electronics

MDPI

Review

Artificial Neural Networks Based Optimization Techniques: A Review

Maher G. M. Abdolrasol [1,*], S. M. Suhail Hussain [2], Taha Selim Ustun [2,*], Mahidur R. Sarker [3], Mahammad A. Hannan [4], Ramizi Mohamed [1], Jamal Abd Ali [5], Saad Mekhilef [6] and Abdalrhman Milad [7]

[1] Department of Electric, Electronics and System Engineering, Faculty of Engineering and Built Environment, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; ramizi@ukm.edu.my

[2] Fukushima Renewable Energy Institute, AIST (FREA), National Institute of Advanced Industrial Science and Technology (AIST), Koriyama 963-0298, Japan; s.suhail.md@gmail.com

[3] Institute of IR 4.0, Universiti Kebangsaan Malaysia, Bangi 43600, Selangor, Malaysia; mahidursarker@ukm.edu.my

[4] Department of Electrical Power Engineering, Universiti Tenaga Nasional, Kajang 43000, Selangor, Malaysia; hannan@uniten.edu.my

[5] General Company of Electricity Production Middle Region, Ministry of Electricity, Baghdad 10001, Iraq; eng_jhy@yahoo.com

[6] School of Science, Computing and Engineering Technologies, Swinburne University of Technology, Melbourne, VIC 3122, Australia; smekhilef@swin.edu.au

[7] Department of Civil and Environmental Engineering, College of Engineering and Architecture, University of Nizwa, Birkat-al-Mouz, Nizwa 616, Oman; a.milad@unizwa.edu.om

* Correspondence: maher.abdolrasol@gmail.com (M.G.M.A.); selim.ustun@aist.go.jp (T.S.U.)

Citation: Abdolrasol, M.G.M.; Hussain, S.M.S.; Ustun, T.S.; Sarker, M.R.; Hannan, M.A.; Mohamed, R.; Ali, J.A.; Mekhilef, S.; Milad, A. Artificial Neural Networks Based Optimization Techniques: A Review. *Electronics* **2021**, *10*, 2689. https://doi.org/10.3390/electronics10212689

Academic Editors: Ahmad Taher Azar and Amir Mosavi

Received: 24 August 2021
Accepted: 29 October 2021
Published: 3 November 2021

Abstract: In the last few years, intensive research has been done to enhance artificial intelligence (AI) using optimization techniques. In this paper, we present an extensive review of artificial neural networks (ANNs) based optimization algorithm techniques with some of the famous optimization techniques, e.g., genetic algorithm (GA), particle swarm optimization (PSO), artificial bee colony (ABC), and backtracking search algorithm (BSA) and some modern developed techniques, e.g., the lightning search algorithm (LSA) and whale optimization algorithm (WOA), and many more. The entire set of such techniques is classified as algorithms based on a population where the initial population is randomly created. Input parameters are initialized within the specified range, and they can provide optimal solutions. This paper emphasizes enhancing the neural network via optimization algorithms by manipulating its tuned parameters or training parameters to obtain the best structure network pattern to dissolve the problems in the best way. This paper includes some results for improving the ANN performance by PSO, GA, ABC, and BSA optimization techniques, respectively, to search for optimal parameters, e.g., the number of neurons in the hidden layers and learning rate. The obtained neural net is used for solving energy management problems in the virtual power plant system.

Keywords: artificial neural networks; optimization algorithms; machine learning; ANN enhancement; PSO; BSA; ABC; GA

1. Introduction

Artificial intelligence (AI) helps computers or inanimate objects based on computers to think or act as humans do. AI research focuses on how the human brain thinks, learns, decides, and works to solve problems. AI is a vast field that aims to create intelligent machines [1]. Machine learning (ML) is a branch of AI that recognizes and learns different data set patterns [2]. As a definition, ML is an AI application that allows systems to learn automatically and improve by the experience and is devoid of being programmed implicitly [3,4]. The common algorithms used in the ML are neural networks, support vector machines, decision trees, random forest, logistic regression, and many more. Also, some others are subsections of the neural network, such as generative adversarial network (GAN) by Goodfellow in [5,6]. The deep learning (DL) approach utilizes a hierarchy of concepts in a field that assists a computer in building knowledge from experience [7]. This approach has found its use especially in visual object or speech recognition as well as genomics and medicines [8,9]. The neural networks are a family of deep learning (DL) and ML methods based on artificial neural networks (ANNs) with multi-hiding layers [10–14]. Neural networks are applied in many different implementations with slight variations in their structures, such as recurrent neural networks (RNN), Artificial neural network (ANN), and convolutional neural networks (CNN) [15,16]. Due to their feature engineering and decision boundaries, the novel neural network approaches are preferred over machined learning in some fields like self-driving vehicles, unmanned drones, and complex deep learning problems [17]. The decision boundary is used to classify any data point as one of the two classes, positive or negative. For this reason, if the data is not separable for any reason, the neural networks in deep learning will not be a good choice [18,19].

Artificial neural networks are computational algorithms that are utilized to model data. Their design is based on the biological nervous system, hence the name [20,21]. An ANNs contain a set of processing elements called neurons that are interrelated components. These neuron structures act as a harmonious rhythm to solve certain complex problems. ANN can be used in scenarios when it is difficult to extract trends or detect patterns. ANNs have recently gained popularity after almost 50 years of existence. Through their rapid increases and importance, the underlying logic behind ANNs has existed; however, due to the pervasive and ubiquitous adoption of powerful computational tools in our contemporary society, ANNs have had a sort of renaissance, much to the benefit of experts, engineers, and consumers.

The current cutting-edge in deep-learning and ANNs focuses highly on their ability to model and interpret complex data and their ability to scale due through optimization and parallelization [22]. The current framework for designing ANNs is widely available, with a myriad of tools facilitating their development. Python, C++, Google's Tensorflow, Theano, Matlab, and Spark contain a robust set of mathematical operations that necessitate ANNs. Due to the algorithm behind ANNs, the models are inherently liable for extracting meaning from imprecise or intricate problems. Speaking reductively, ANNs are data modeling tools that are trained on a given dataset.

Optimization problems often require good optimization methods to minimize or maximize objective functions. These functions can often not solve problems accurately, for example, when they are not linear or polynomial and must be approximated. Full or partial derivatives are used in some algorithms to linearize these functions at specific points [23], whereas evolutionary algorithms (EA) may be employed for approximation. The objective function approximation in optimization problems makes it possible to apply other artificial intelligent techniques through a non-linear regression to resolve an optimization problem. The objective function's derivate should be polynomial to calculate the optimization problem's solution. Algorithms are normally used to optimize, e.g., weights, optimize network architecture, optimize learning rules, neurons, activation function, and bias. Another way to optimize and enhance the ANN is by using an optimizer to replace the neural network's original algorithms with optimization algorithms, replacing the backpropagation with any optimization techniques to solve certain associated issues. However, using an optimization algorithm in place of back-propagation, like using the Liebenberg Marquardt neural network with any optimization techniques for fast or accurate achievement in the neural network training. This research review highlights improving the neural network by optimizing algorithms by handling neural network parameters or training parameters to find the finest structure network pattern to solve the problems with high accuracy and faster. This review included testing results for improving the ANN performance using four optimization algorithms to search for ANN's optimal parameters, such as the number of neurons in the hidden layers and learning rate. The obtained neural net is used for solving energy management problems in the virtual power plant system.

Supporting AI, ML, DL with optimization techniques has gained importance in the last few years. There is a lot of ongoing research using optimization to enhance or boost performance by finding the optimal parameters values to help architecture design. In [24], a fuzzy logic controller design improvement for PV Inverters utilizes differential search optimization to find the optimal membership function patterns, which improve the fuzzy controller to a higher level of accuracy. In [25] the ML, this approach optimizes the Support Vector Machine model parameters and simultaneously locates the best features subset. Allowing the optimization technique to do the job is the smartest way to improve almost any AI or ML performance [26]. It is essential to process pre-setting to guarantee optimal results for almost any application. In the DL and particularly deep neural networks (DNNs) and ANN, the more hidden layers, number of neurons, and complex activation functions, the better the outcomes but will cost more time and more complexity of the network [27,28]. So, to use the optimum numbers of parameters by trial and error is a time-consuming and impossible way to follow. From another point of view, the ANN with human estimation parameters setup could bring outcomes, but how to confirm this is the best outcome of the ANN? For these reasons, the optimization algorithm can solve these issues, and this review delivers a detailed analysis of various examples of ANN-based optimization techniques. For instance, In [29–31], optimization techniques optimize ANN parameters to solve different electricity and communications fields by finding optimal parameters for the optimum ANN structure.

The rest of this paper is organized as follows: Section 2 presents the materials and methods used. Section 3 addresses the challenges and motivations for ANN-based optimization, while Section 4 presents a review of optimization algorithms, Section 5 addresses neural network structure types. Section 6 is the complete overview of neural networks enhanced by optimization algorithms, Section 7 is an application on artificial neural network-based optimization algorithms, Section 8 covers artificial neural network training-based optimized parameters and finally, Section 9 presents the conclusions and future work.

2. Materials and Methods

A literature survey for material contents was done to present, identify, analysis, classify and review the distinguished ANN-based optimization techniques for various applications and controller enhancement. In this comprehensive review, the survey has gone through voluminous publisher databases. For example, IEEE Xplore library, Web of Science, Elsevier Scopus, and MDPI open access for putting into practice the search queries to ensure all selected articles meet the essential quality measures, novelty, originality, high impact, and high h-index. Following the guidelines [32–34], to present an in-depth review and understanding of utilized various keywords to find significant journals within the scope of the research, including multiple types of neural networks, such as ANN, RNN, CNN, GNN, and many more and different kinds of optimization techniques such as PSO, EA, CSA, and many more. In this section, the enhancement of the architecture of the neural network is further evaluated. Different neural network structures are presented, selecting and validating the superiority of using various optimization techniques to search for optimal ANN parameters and comparing the performance [35] to designate the best parameters results in the comparative performance output of the ANN controllers.

Most of the included articles were about the focus of this research review, that is how to boost the performance of neural networks using optimization algorithms by modifying the neural network structure. The screening stage comprises three stages. Firstly, matching articles were excluded, bringing in about 433 best articles which were examined in the next step, where the significant papers were reviewed by looking at their title, keywords and abstract. This step resulted in 306 documents for additional investigation. The third stage is the eligibility step, in which the full texts of papers were studied, in which 219 were counted as eligible for review of references. In this review, only the meaningful and suitable literature has been considered by evaluating the article's relevant content and the critical topic of attention of the review. Accordingly, the related papers were designated based on the number of citations and research interest. This review methodology process comprises several stages, and the Prisma guidelines according to [36,37] were followed. Figure 1 shows how the methodology for utilizing optimization to find optimal parameters of neural networks. A schematic diagram of the review section process, evaluation, and quality control of the database using the Prisma guidelines is shown in Figure 2.

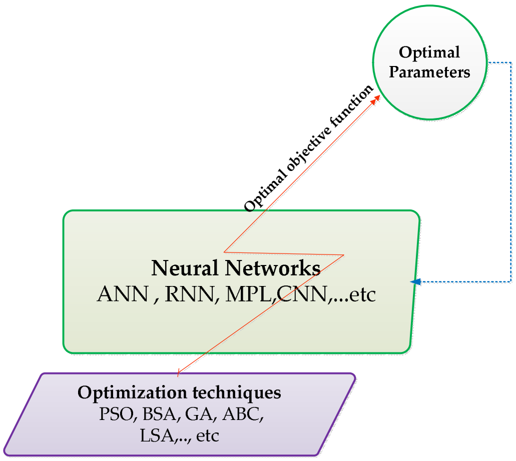


Figure 1. Methodology of utilizing optimization to find optimal parameters of neural networks.

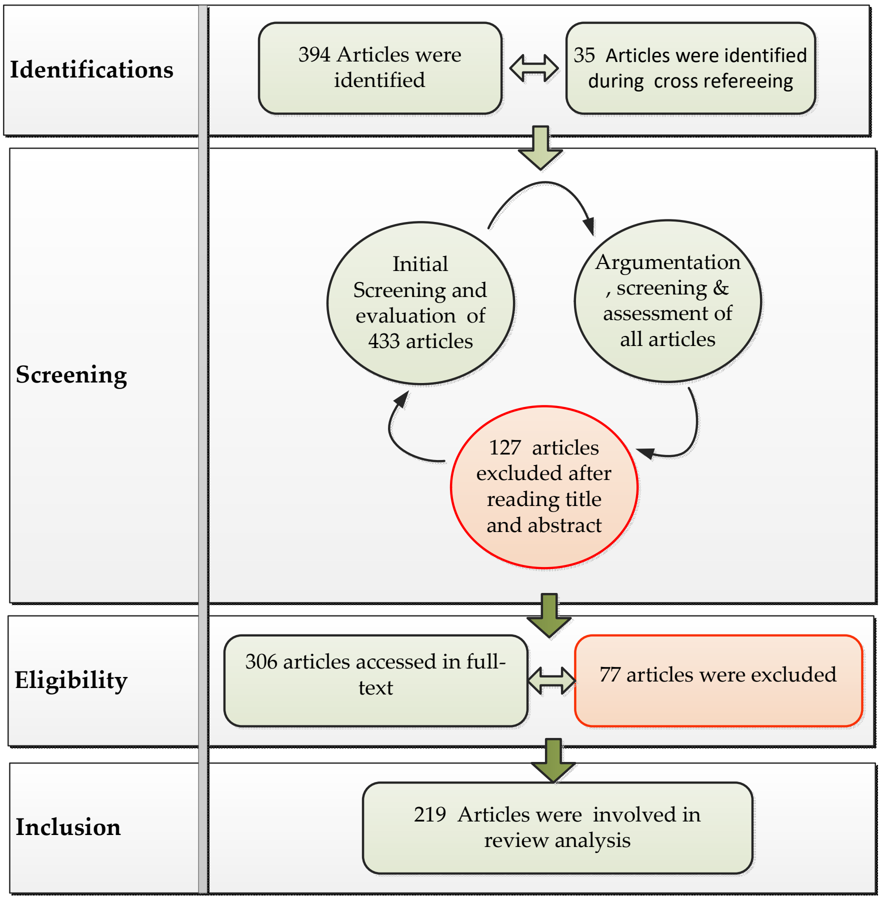

Identifications
394 Articles were identified
35 Articles were identified during cross refereeing
Screening
Initial Screening and evaluation of 433 articles
Argumentation , screening & assessment of all articles
127 articles excluded after reading title and abstract
Eligibility
306 articles accessed in full-text
77 articles were excluded
Inclusion
219 Articles were involved in review analysis

Figure 2. Schematic diagram of the literature selection, evaluation and quality control process of the database using the Prisma guidelines.

3. Challenges and Motivations for ANN-Based Optimization

Neural networks can study large volumes of data with complex features and extract different patterns in a relevantly short amount of time. Therefore, they are useful for many industrial applications, such as predicting certain behaviors, detecting anomalies or errors in data, detecting certain images, sounds or pictures. They could use self-learning to produce the best output, with unlimited provided inputs [36]. The neural networks modeling approach is very flexible and quick to solve problems, and it does not rely on physics-founded algorithms to build models. They are easy to modify and deal with based on operator experiences and merge with the ANN structure model. Neural networks are good for solving complex non-linear relationships as their inputs are saved in their particular networks as an alternative to a database line. For that reason, the loss of data will not disturb the NN operation process. The following points list the main motivations associated with the use of ANNs:

- Ability to accept unlimited inputs and outputs; this unique advantage makes ANNs more important and popular than other AI methods, making them suitable for small or huge dataset analysis.
- Skills to learn and model non-linear and complex relationships, the ANNs can handle various real-life applications in different fields that are complicated and non-linear; this is a very significant advantage.
- Skills of training without complete information and the data may produce output, and the performance depends on the importance of the missing data.
- Distinct from the other deep learning prediction techniques, ANNs do not need any enforcement restrictions on the input variables, such as how the data need to be distributed.
- Skills to create ML: ANNs can learn events and sort wise decisions via commenting to reach similar events better.
- Multi-processing capability, the ANNs can assure numerical efficiency with their power of performing several duties simultaneously.
- Ability to tolerate faults, whereby ANNs can produce output results even if some cells are corrupted, and this advantage allows ANNs to tolerate faults.
- Ability to generalize; as soon as the ANNs learn from the initial input relations, they can conjecture unknown relationships in anonymous data, thus making the model generalized and allowing it to predict unknown data.
- In using distributed memory during the ANN learning, an essential process is adjusting samples and indoctrinating the network according to the desired output by viewing these samples to the network. This process allows the network to achieve and select the instance straight proportionally and by failing to show the event to the network in its full features, and the network may yield false outputs.

However, artificial neural networks, although considered one of the best general algorithms solving problems, they are very much a stochastic problem, where model weights are used and every iteration is reorganized with the backpropagation of the error algorithm signal. ANNs' performance is good, yet several disadvantages and challenges face the ANN to assure the proper network structure, duration, best tuning parameters, trial and error, and more explanations that must rely on an expert user. The next points express the main challenges for ANNs as follows:

- Mysterious network behavior, After the ANN produces an analytical result, it is unexplained why or how selecting these outputs and rejecting the others may make it untrusted in the network.
- Appropriate network architecture design. ANNs have no exact law to determine the best structure design or a proper network structure must be achieved by experience and trial and error.

- Obscure duration time for the network; the optimum results may be produced during the training phase as expected because the network minimizes to a certain level the error on the sample to allow the training completion.
- Depending on the hardware, ANNs need powerful dual processors and ANN structures. This drawback is called the realization that the whole approach is equipment-dependent.
- Gradual corruption slows down the process over time and it suffers relative degradation, and the network problems do not immediately degrade directly.
- Difficulty recognizing network problems if they exist since ANNs are based on numerical data that explain the difficulties in numerical values before being introduced to the ANNs. This could depend on the researcher's ability to display the mechanism and influence the network's performance.

Artificial neural network applications that have increased dramatically in the world in the middle of the last century are developing very fast. At present, besides the computer capabilities, the advantages of ANNs have been examined, and the problems users have encountered. However, it is very important not to neglect the ANN network's disadvantages, which are a developing science branch, and should excluded one after another, and the advantages of the ANNs are growing progressively. That is, the avenue of using ANNs will be an increasingly important indispensable part of our lives. The enhancement of ANNs by using optimization methods could eliminate some of their disadvantages in picking the best network structure using the proper optimization techniques. The challenge is finding a system coding that enables appropriate tuning of neural structures in professional networks, including the best number of neurons, hidden layers, weights, bias, and self-shaping architecture and multi-stage objective functions.

It is very important to select and adjust the best suitable neural network parameters for any given application, as there are many possibilities. However, not every neural network can could act perfectly in all applications. Some types are more practical in particular applications; for example, CNN is good for images and videos, while RNN is good for text and classification problems, so the networks need to be studied and adjusted, and the problems need to be compared and contrasted. Somehow, to enhance the neural networks with optimization it is important to select the neural network parameter optimizer to obtain the best outputs.

Like other AI algorithms, neural networks can deal with non-linear and complicated problems with a high volume of data. The superiority of neural networks over other different AI algorithms lies in that they are very effective for many inputs and outputs. While it is true that fuzzy or adaptive neuro-fuzzy inference systems (ANFIS) techniques have drawbacks, they can accept many inputs, although they are limited in the number of outputs they can support. Neural networks do not have this limitation which makes them work better for classification and regression studies.

4. Review of Optimization Algorithms

An optimization algorithm is an essential tool for selecting the best solution from a set of all possible solutions for analyzing, classifying, or improving existing systems or data. Above all optimization problems get at least one objective function or more objective functions. However, the target is to determine the optimal key that fulfills the complementarity conditions. Optimization problems are found in numerous scientific areas, such as medicine, engineering, business, and many more. Optimization algorithms are classified into various types: deterministic optimization and global optimization [23,37], continuous optimization [21,38], multi-objective optimization [39–41], etc. Overall, each optimization method is designed to serve specific targets. For instance, the local algorithms solve some optimization problems, such as discrete variables or integers, whereas it is easy for global algorithms. The global optimization algorithms can be categorized as either evolutionary algorithms or deterministic algorithms.

There are hundreds of common optimization techniques for many relevant scientific code archives. The challenge is knowing the best techniques to suit a particular optimization problem because some techniques use derivatives while others do not. The conventional methods normally use first order derivatives, and others use the second derivative for their objective function. The search type is either direct or stochastic search for targeted objective functions that result in its function's maximum or minimum output.

Normally, the most popular kinds of optimization problem facing neural networks involve continuous function optimization. Their input pretext for their function estimates numeric values for either the input function or the output function. However, the more available information for the target function, the more accurate the achieved optimization will be. In contrast, the differentiable function can determine any sample in the input search. Optimization algorithms are, in general, categorized into two groups: deterministic and heuristic algorithms. Deterministic techniques exploit their analytical capabilities, while in contrast, heuristic techniques are more flexible and efficient than deterministic techniques through fast-to-obtained solutions, decreasing the number of global solutions. Global optimization algorithms are used to find the global minimum or maximum in complex problems. This is harder than local optimization with bound constraints and does not require derivatives. Both the local and global optimizations are a matching set in solving linear, non-linear, quadratic, and least squares constrained or unconstrained, dense or sparse, forward or reverse communication, continuous, mixed-integer, integer problems [42]. The optimization techniques are classified according to the underlying principle of a biological and physical-based algorithm. The first category is a biology-based algorithm such as genetic algorithm (GA), harmony search algorithm (HSA), particle swarm optimization (PSO), bacteria foraging optimization (BFO), cuckoo search algorithm (CSA), bee colony algorithm (BCA), ant colony optimization (ACO), firefly algorithm (FA) [43], backtracking search algorithm (BSA), lightning search algorithm (LSA), etc. The second category is physics-based algorithms such as simulated annealing (SA), gravitational search algorithm (GSA), chaotic optimization algorithm (COA), etc. [44,45]. In this review, some of the most popular optimization algorithms are explained.

The particle swarm optimization algorithm is one of the most popular evolutionary optimization algorithms [46]. The PSO algorithm principle depends on the velocity and position of particles [47]. The authors described that the PSO algorithm is utilized to automatically design an ANN method to improve the synaptic mass, architecture, and transfer functions for each neuron [48–52]. Nevertheless, PSO has some drawbacks: it is vulnerable to becoming stuck in local minima and selecting control parameters incorrectly, resulting in a bad solution. In [48], an ANN-based PSO method was used to predict the thermal properties of molecular structure.

Another popular algorithm is the gravitational search algorithm, a physics-based optimization algorithm inspired by Newton's motion and gravity laws [49]. The GSA optimization method has been used in some applications to find the best solution for a short-term training feedforward approach to ANN problems and to improve the performance [53–57]. In [50], the authors addressed an ANN-based GSA optimization approach to enhance kidney image quality classification for a bio-medical application. The study in [51] presented a GSA optimization-based ANN to solve geotechnical engineering issues for improving geogrid-reinforced soil structures.

One optimization algorithm is the neural network algorithm (NNA), which is inspired by the functioning of biological nervous systems and artificial neural networks [52]. NNA has recently been used in machine learning, as an intelligent controller, in biodiversity assessment, intelligent feature recognition, and for uncertain data streams to provide a way of learning features, predicting highly nonlinear functions, discovering useful hidden representations of the input because it does not require mathematical models and it achieves good prediction for ANN [58–61]. However, NNA controllers require massive data and long-time training and learning. In [53], an artificial bee colony (ABC) and an NNA intelligent feature recognition for STEP-NC-compliant manufacturing can adjust geometric and topological information. The study in [54] addressed estimating biodiversity assessment based on AI and NNA.

Another powerful optimizer is the BSA which generates a trial population and then takes partial advantage of its experiences from previous generations. Crossover is developed in the trial population. The initial trial populations are taken from mutations. The described benefits of BSA are in their search exploration process, which has the advantage of using the mutation and crossover strategies. Though it has some limitations, such as time-consuming computation because of the dual population algorithm, one parameter is only used to control the amplitude of the search direction matrix in the mutation phase, and crossover is complex [55–57]. In [29,30], BSA is applied for a fuzzy logic speed controller optimization approach for induction motor drives. The deterministic global optimization in numerical optimization helps to search for global solutions for optimization problems [42].

The lightning search algorithm was first proposed by Shareef and his colleagues [58]. Afterwards Ali upgraded it with quantum mechanics theories to generate a quantum-inspired LSA (QLSA) [59]. The LSA optimization approach has been utilized in numerous applications [30,60,61]. The study in [30] described an LSA-based ANN method home energy management scheduling controller for residential demand response strategies. The study in [62] proposed a neural networks-based LSA to find the optimized feedforward learning process to solve datasets. In [63], the author addressed finding the optimal Kp and Ki value of the LSA-based PI voltage controller and implementing it into the dSPACE controller. Table 1 lists the advantages and disadvantages of the most popular nature-inspired optimization techniques. However, not all optimization algorithms and their variants provide superior solutions to specific problems. Also, even though some of the optimization techniques are efficient, they still need further improvement to enhance their performance. Besides, how to speed up the convergence of an algorithm is still a very challenging question, so new Nature-inspired optimization techniques must be continuously developed to advance the field of computational intelligence or heuristic optimization [60,61,64–73].

Table 1. Advantages and disadvantages of the most popular nature-inspired optimization techniques.

Technique	Advantages	Disadvantages
PSO [62]	- Fast convergence. - The capability of solving complex problems in a different application domain.	- Easily get trapped in local minima. - Improper selection of control parameters leads to a poor solution.
GA [63]	- It does not require derivative information - Suitable for a large number of variables,	- No guarantee of finding the global minimum, - Long time for convergence, - Hard to fine-tune all the parameters, like mutation rate, crossover parameters, etc., this is often done by just trial and error.
NNA [52]	- Easy to learn and implement. - It obtains good results when dealing with lower-dimensional optimization problems.	- Abrupt switching to the exploitation stage by quickly varying wavelength and pulse emission rate. - Difficult to solve high-dimensional optimization problems.
ABC [64]	- Strong robustness - Fast convergence and flexibility	- Premature convergence in the later search period. - Accuracy problems that in some cases cannot meet the optimal solution.
LSA [58]	- Suitable for the search exploration process. - Has the advantage of using the mutation and crossover strategies.	- Time-consuming in computation because of the use of the dual population algorithm. - One parameter only controls the amplitude of the search direction matrix in the mutation phase. - Crossover is complex.

Table 1. *Cont.*

Technique	Advantages	Disadvantages
EA [65]	- Robust concerning noisy evaluation functions. - Easily to adjust to the problem	- Usually provide reasonably good performance - Premature convergence to a local global minimum.
BSA [73]	- Suitable for the search exploration process. - Has good mutation and crossover strategies.	- Time-consuming in computation because of the dual population algorithm.
GSA [67]	- Faster solution convergence	- Easily gest trapped in local minima, and weakness in its strategy to diversify the algorithm's population
FA [68]	- Easy to implement - Capable of automatic subdivision and dealing with multimodality.	- Gets trapped in several local minimal. - Performs local searches - Does not memorize the history of the better situation, and may end up missing situations

5. Neural Networks in Deep Learning

Deep learning (DL) is a subset of ML which is based on learning data representations, unlike task-specific algorithms. It is inspired by the function and structure of the brain, known as artificial neural networks. The approach utilizes a hierarchy of concepts in a field that assists a computer in building knowledge from experience. This technique does not require computer knowledge to be provided through human input as it is automatically gathered. The hierarchy of concepts facilitates breaking complex concepts into simpler ones with several layers [8]. DL techniques use several layers of abstraction to learn when there is more than one processing layer. This approach has found its use especially in visual object or speech recognition as well as genomics and medicines. DL implements a back-propagation approach to detect patterns in complex datasets. It does this by considering how the internal parameters should be altered to move from one representation layer to the next. Deep convolutional and recurrent nets have facilitated breakthroughs in image and audio processing as well as text and speech detection, respectively [16,69].

Neural networks have different implementations with slight variations, including RNN, ANN, and CNN [15,16]. Due to their feature engineering and decision boundaries, such novel NN approaches are preferred over machine learning by the people active in the study of self-driving vehicles, unmanned drones, or complex deep learning problems [17]. The decision boundary is a technique used to classify any data point as belonging to one of two classes, positive or negative. For this reason, if the data is not separable for any reason, neural networks will not be a good choice in deep learning. On the other hand, feature engineering is composed of two steps: feature selection and extraction. These two components make up the model building. The multi-layer ANNs are also neurons placed similarly to the human brain. Each neuron is connected to other neurons with certain coefficients. During training, information is distributed to these connection points to learn the network structure and functioning [18].

6. Neural Networks Structure Types

In deep learning, many neural network types use different principles to determine rules for various applications and formulate the foundation for most pre-trained models. The most well-known neural networks are ANN [70], CNN [71], and RNN [72]. On the other hand, many neural networks are developed with unique structures to serve different software. For example, radial basis function neural networks, modular neural networks, multilayer perceptron neural networks, and sequence-to-sequence models neural networks use their unique strengths to serve and fit some applications well compared to other networks. DNN [73] and another deep learning NN are so-called graph neural networks (GNNs), designed for graphic data classification problems [74,75]. LSTM recurrent neural network models are excellent for text classification problems. An ANN is based on the use of simultaneous optimization techniques was used to model theophylline tablet formulations in [76]. A generative neural network has been used in adjoint electromagnetic simulations [23].

6.1. Artificial Neural Networks

An ANN is a cluster of multiple perceptrons or neurons at each layer; when the input data is sorted in the forward direction, this is called the feed-forward neural network [15,77]. The basic structure of an ANN consists of three layers: the input layer, hidden layers, and output layer. The input layer receives the input data; the hidden layers compute the input data, and the output layer provides outcomes. Each layer's duty in the neural networks attempts to learn specific decimal weights to be set at the end of the learning process. The ANN approach is good for solving image data, text data, and tabular data problems. The advantage of ANN is its skill of dealing with nonlinear functions and learning weights that help map any input to the output for any data. The activation functions provide nonlinear properties to their ANN, which can benefit the net to learn any complex relation associated with input data and output data, known as a universal approximation. Many researchers adopt ANNs to solve complex relations, for example, the coexistence of cellular and WiFi networks in an unlicensed spectrum [78].

Another example is feed-forward neural network probabilistic neural network (PNN) in [79] and knowledge-based neural network described in [80,81]. In [82] this approach was used for modeling a solar field in direct steam generation parabolic trough. ANN is used as an optimizer in many research projects to solve bundling problems; for example, in [83] it was used to optimize a flight trajectory for rockets. An ANN optimized the design and optimization of microwave circuits in [84]. Model-aided wireless AI embedding expert knowledge in DNN to solve wireless system optimization to find the best architecture of an ANN [22]. ANN is also used to optimize and control thin film growth processes [85]. A sampling method for the ANN model's optimal design [86]. A feedforward neural network optimization is applied to synthesize fault-tolerance [87]. ANNs, together with the Xinanjiang model to employed to explore nonlinear transformations [88]. Some optimized artificial neural network models for predicting chlorophyll dynamics were done to decrease the cost of aquatic environmental in-situ monitoring and increase bloom forecasting accuracy [89]. A problem of crude oil distillation systems was solved using ANN by optimizing heat-integration [90]. ANN solved the optimization problem and extraction of anthocyanins in black rice using orthogonal arrays [91]. ANN solves the optimization problems in traffic lights timing traffic light controller [92]. Also, ANN is used as an optimizer and applied to waves energy converters (WEC) to predict overtopping rates as part of a sustainable optimization of coastal or harbor defense structures and their conversion for constructing a predictive model [1]. The architecture of the artificial neural network is shown in Figure 3. Each neuron output includes an activation function of a sum of all inputs weights, while the neuron input is a sum of all weights included in the bias, as shown in Figure 4. The bias is a constant used to adjust the output and the weighted sum of the inputs to the neuron, while the activation functions are a powerhouse for neural networks [93–99]. The neural network weights updates in the back-propagation process are done to get the gradients as a neural network using many hidden layers. The gradient may vanish and explode during the backward propagation [100,101].

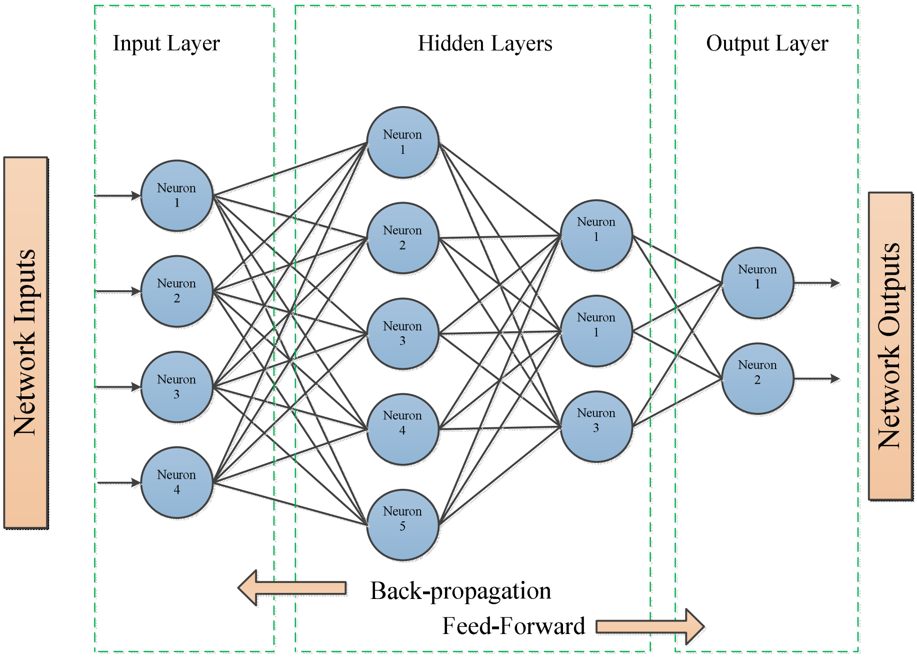


Figure 3. Artificial neural network architectures with feed-forward and backpropagation algorithms.

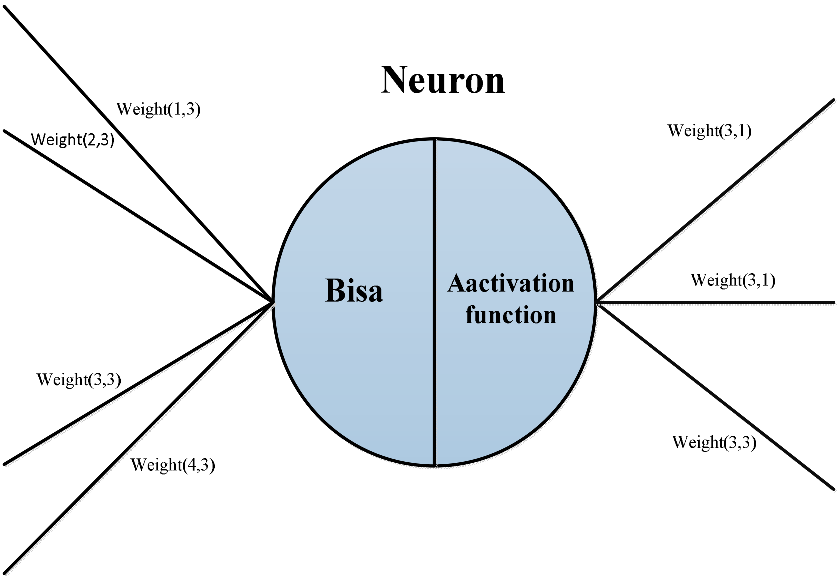


Figure 4. Details of a single perceptron neuron for an artificial neural network.

6.2. Recurrent Neural Network

Recurrent neural network architecture is from a neural network family, though, is displays differences compared to ANN, in which the looping constraint on the hidden layer turns back to RNN [15]. The feedback constraint is back-propagated to ensure that the subsequent data is looped into the input data from the last step in each neuron's first step, as shown in Figure 5.

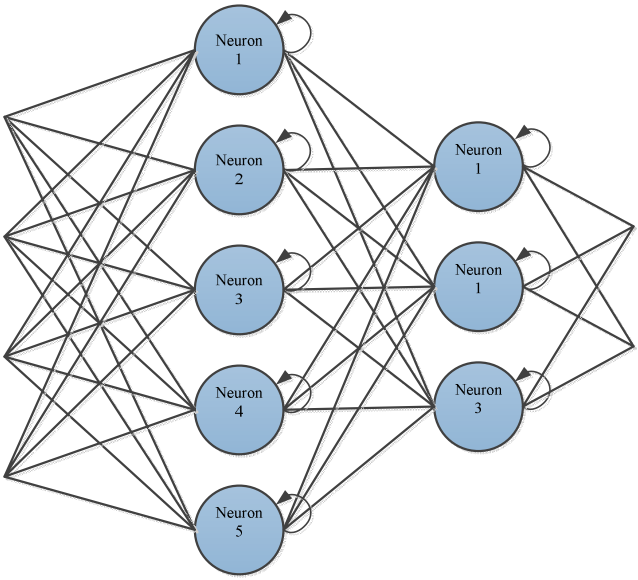


Figure 5. Details of the recurrent neural network architectures network.

RNN is normally used to solve problems associated with text data, time-series data, and audio data. Because the parameters go through different time steps, these steps are called parameter sharing, ending with fewer parameters to be trained [102]. This action could save computational time because the gradient computes only at the last step and vanishes in every neuron in the RNN. The error is back-propagated from the previous time step to the first step. The error at each time step is calculated, allowing us to update the weights. The Elman neural network (ENN) has similar concept properties to RNNa, and it has standard back-propagation known as the Elman backpropagation algorithm (EBP). RNN is used in many applications for real-world problem-solving, as in [38].

6.3. Convolution Neural Network

A convolutional neural network is a neural network family compared to a multilayer perceptron (MLP) [93]. The CNN has hidden layers called convolutional layers. Also, CNN has other non-convolutional layers [15,103]. The basic concept of CNN structure is the convolutional layers that pull through the input weight and transform the neurons' input on the activation function.

To the next convolutional layer, is the convolutional operation, as shown in Figure 6. Each convolutional layer specifies the number of filters used to detect the patterns of shapes in specific object shapes, for example (circle, squire, corner, eyes, feathers, etc.). These filters help extract the right and relevant features from the input data. CNN is the most widely used type of neural network for analyzing images. However, image analysis is but one use of CNN and it can be used for other data analysis problems such as classification problems. Most generally, CNN is a critical neural network specializing in picking out patterns and making sense of them. This pattern detection is what makes CNN so useful for image analysis. These CNN models are used across different applications and domains, especially in image and video processing projects. CNN is related to solving images as multiple-image-based depth estimation or estimates depths in the edges; basically, they are used to classify the edges in backgrounds or reflections [5]. CNN was used to detect wildfire smoke images [104] and forest fire smoke recognition [105].

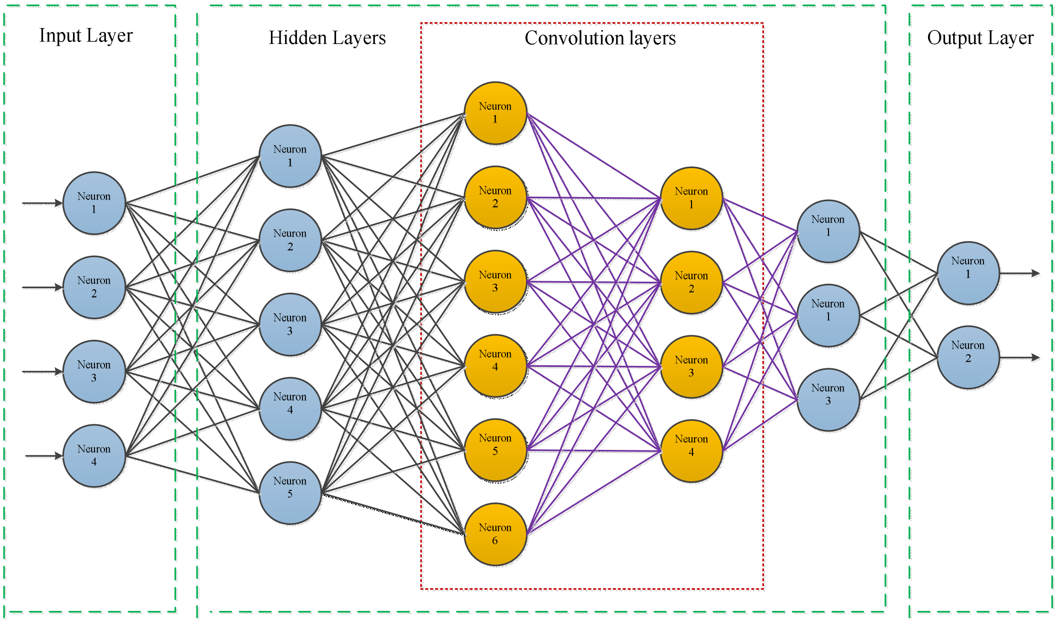


Figure 6. Details of the convolution neural network architectures network with convolution layers.

7. Overview of Neural Networks Enhanced by Optimization Algorithms

The optimization technique aims to improve the applications by finding the minimal error, minimal cost, maximum performance and efficiency. It can be categorized into two principal ideologies: physical and biological-based [44]. for example, chaotic optimization algorithm (COA), simulated annealing (SA), gravitational search algorithm, etc. or a biology-based algorithm such as genetic algorithm (GA), practical swarm optimization, bacterial foraging optimization, harmony search algorithm, cuckoo search algorithm, ant colony optimization, dolphin swarm algorithm (DSA) bee colony algorithm, firefly algorithm, LSA, backtracking search algorithm, etc. [46]. In this review, some common optimization algorithms that enhance the performance of neural networks are discussed in detail in the following subsections.

7.1. Artificial Neural Networks Based Particle Swarm Optimization

The PSO method was first discovered by Eberhart and Kennedy inspired by the movement of organisms such as bird flocking and fish schooling in 1995 [106]. PSO uses a velocity vector to update each particle's current position in the swarm [107]. The PSO-based neural network is used extensively compared to the other algorithms and applied by many researchers in different applications. For example, it is used to solve mathematical problems of predicting the uniaxial compressive strength of rock samples from other states in Malaysia [108]. Also, this combination of ANN-based PSO is used for detecting trip purposes from smartphone-based travel surveys of GPS data [109]. ANN-based PSO is used smartly to improve the prediction performance model for Wi-Fi indoor localization strategies by reducing the maximum location error with astonishing results [110]. In [111], PSO optimization swas used to design a dynamic modular neural network based on adaptive PSO to solve the problem related to a subnetwork output. Optimization enhances many algorithms and applications to solve complex linear and nonlinear problems; for example, an efficient PSO-based ANN was utilized for the nonlinear mathematical model of Troesch's problem. PSO was used to obtain a unique numerical solution by weights optimization for the final network [112]. Network weight optimization is very popular for optimizing the initial weights or entire network weights. For example, in [72], optimization is used to find the best weights for self-adaptive parameters and strategy-based PSO (SPS-PSO) algorithm to optimize feedforward NN (FNN) design. Again, weights optimization using ANN-based PSO solves a non-linear channel equalization problem as in [113], and in [49], PSO's weight optimization automatically designs an ANN methodology. Using PSO for search for hyperparameters is also widely discussed and tested in many kinds of studies and the outcome improves many applications. For example, CNN-based PSO optimized the hyperparameter linearly to decrease CNN weights in the final network [114]. Also, PSO optimization boosts neural networks by searching for the optimal hyperparameters for network architecture design in [115]. PSO-based deep NN was used to optimize the number of hidden layer nodes for digital modulation recognition applications. Another study [116] discusses optimizing the number of hidden layer nodes used for global solar irradiance prediction in extremely short-time intervals with hybrid backpropagation neural networks based on PSO optimization. Table 2 presents some examples of PSO research for neural network architectures focusing on weights and neurons in hidden layer optimization problems.

Table 2. Studies involving PSO for neural network design based on weights and neuron number optimization.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
dynamic MNN [111]	Adaptive PSO	To calculate the weights	Design of dynamic modular neural network
DNN [115]	PSO	To optimize the number of hidden layer nodes	Digital modulation recognition
ANN [117]	Simulation annealled PSO	initial weights and biases of the neural network are optimized	Endpoint sulfur content in Kambara reactor desulfurization
BiLSTM NN [118]	ADPSO	To optimize the hyperparameters of BiLSTM neural network	Ship motion attitude prediction
IT2FNNs [119]	PSO & BBBC	For parameter optimization for Takagi-Sugeno-Kang TSK type IT2FNNs	Design interval type-2 fuzzy neural networks IT2FNNs
FNN [72]	SPS-PSO	Weight optimization problem parameters	For parameter and self-adaptive mechanism strategies.
ANN [49]	PSO	Train a set of synaptic weights	To evaluate the fitness of each solution and find the best ANN design
ANN [113]	PSO	Find the optimal weights of the network	Non-linear channel equalization
ANN [116]	PSO	For optimizing the number of hidden layers and neurons used and the learning rate	Global solar irradiance prediction at extremely short-time-intervals
CNN [114]	PSO	For hyperparameter optimization with linearly decreasing weights	CNN architecture design
ANN [120]	PSO	For an optimal number of hidden layers and learning rate	Microgrid scheduling and management

A combination of PSO optimization and neural networks is the most common combination between optimization algorithms and AI and is used in many application software and controllers. There is much ongoing research on this combinationa; for example, a PSO-based ANN was used to enhance forecasting software reliability [121], while in [122], one was used for data-based fault-tolerant control. PSO assists different types of neural networks in different ways. For example, a PSO-based BP neural network used to solve big-data mining approach problems associated with financial risk management with the Internet of Things (IoT) constructs a nonlinear parallel optimization model [3]. There are some applications done on a giant scale, for example, The Kambara reactor desulfurization used a combining ANN-based optimization techniques and a simulated annealing algorithm with PSO (SAPSO) for determining optimal parameter structures such as a number of hidden layers, neurons, and activation functions training to solve desulfurization model performance problems [117]. In [118], an issue of ship motion attitude prediction was solved by using the adaptive dynamic PSO (ADPSO) algorithm and bidirectional long short-term memory (LSTM). That is done by searching for the hyperparameters of bidirectional (BiLSTM) neural networks. In [119] interval type-2 fuzzy neural networks (IT2FNNs)-based PSO and a big bang big crunch (BBBC) functional for parameter optimization were used for Takagi-Sugeno-Kang type problem. Sadik and his co-workers have successfully used a hybrid PSO-ANN algorithm for indoor and outdoor track cycling wireless sensor localization. and the algorithm was used for improving the distance estimation accuracy of mobile nodes [29].

The PSO optimization with AI saves lives in many biomedical applications that help many smart applications in hospitals, clinics, and therapists by assisting smart diagnoses or smart robots. Some applications in this area can be highlighted; for example, in [123], a hybrid ANN-PSO is used for predicting airblast-overpressure by estimating quarry blasting and influential parameters in four granite quarry sites in Malaysia. Also, in [124] ANN-PSO is used to manage groundwater resources to solve the groundwater management problems of groundwater in France's Dore river basin [124], whereas in western Australia, short-term traffic flow predictors for forecasting traffic flow conditions on a section of freeway using Intelligent Swarm PSO-based ANNs were used [125]. In [126], a functional-link-based neural fuzzy network (FLNFN)-based hybrid cooperative PSO and cultural algorithm were proposed for solving problems related to orthogonal polynomials and linearly independent functions in a functional expansion of the functional link neural networks. in [127], PSO was enhanced with a periodic mutation strategy (PMS) and neural networks with mutation application strategy and diversity variety for solving problems of an airfoil in transonic flow. A photovoltaic thermal nanofluid-based collector system used ANN and PSO to solve a complex non-linear relationship between input and output parameters [128]. Some researchers have used a neural network to improve the PSO search performance oppositely [129–131]. Improved PSOs revolve around feed-forward ANNs, as in [31], to present a unique evolutionary ANN algorithm called IPSONe. In [132], a neural network with a fuzzy algorithm and PSO is used for a brain-computer interface classifier for wheelchair commands, whereas PSO is used to optimize with a cross-mutated-based ANN (FPSOCM-ANN). A PSO combined with ANN for data classification with an opposition-based PSO neural network (OPSONN) algorithm was used for the NN training to solve data classification problems [133]. Taguchi PSO solves high-dimensional global numerical optimization problems for ANN design concerning tensile strength for steel bars [131]. A nonlinear neural network predictive control strategy based on tent-map chaotic PSO (TCPSO) was used for achieving a nonlinear optimization for advanced convergence and high accuracy [129]. ANN is the most common neural network and the PSO is the most common optimization method; for that reason, they have been used and compared in some cases with other AI or optimization techniques. For example, training ANNs over a hybrid PSO and cuckoo search (PSO-SC) algorithms that have been done by adopting feedforward neural networks (FNNs) to solve algorithm performance problems [130]. Table 3 presents studies involving PSO for neural network design and application enhancement.

Table 3. Studies involving PSO for neural networks design and application enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [123]	PSO	To predict airblast-overpressure (AOp) in quarry blasting	Airblast-overpressure induced influential parameters in four granite quarry sites in Malaysia
ANN [124]	PSO	To minimize pumping cost and solve ground management issues	Management of groundwater of the Dore river basin in France
ANN [125]	PSO	To solve traffic flow predictors problems	Forecast traffic flow conditions on a freeway in Australia
Neural fuzzy Network [126]	CCPSO	To increase the global search capacity using the belief space	Several predictive applications
ANN [127]	PSO & PMS	A periodic mutation application strategy with diversity variety for six benchmark test functions	Airfoil in transonic flow

Table 3. *Cont.*

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [128]	PSO	To identify a complex non-linear relationship between input and output parameters	Photovoltaic thermal nanofluid
ANN [129]	tent-map chaotic PSO (TCPSO)	To perform the nonlinear optimization to enhance the convergence and accuracy	Numerical simulations of two benchmark functions
ANN [130]	Hybrid PSO-CS Algorithm	To investigate the algorithm performance with two benchmark problems	Benchmark classification for ANN structures.
ANN [121]	neighborhood fuzzy PSO	To enhance forecasting of software reliability	Forecasting of software reliability
Critic NN [122]	PSO	Solve the Hamilton-Jacobi-Bellman equation more efficiently.	Data-based fault-tolerant control
ANN [29]	PSO	To improve the distance estimation accuracy of mobile nodes	Wireless sensor localization technique
ANN [131]	Taguchi PSO (TPSO)	To solve high-dimensional global numerical optimization problems.	Optimize the chemical composition of a steel bar
ANN [112]	PSO	To obtain the numerical solution of Troesch's problem	Non-linear Troesch's problem
ANN [110]	Affinity Propagation (AP) & PSO	To reduce the maximum location error and enhance the prediction performance	Wi-Fi-based indoor localization system
ANN [108]	PSO	To predict unconfined compressive strength (UCS) of rocks	Predicting UCS rocks from different states in Malaysia
ANN [31]	PSO	To evolve the structure and weights of ANNs	Evaluated on several benchmarks
ANN [132]	Fuzzy PSO	Classification of a three-class mental task-based brain-computer interface	Brain-computer interface for wheelchair commands
ANN [133]	PSO	For training on opposition based PSO neural network (OPSONN) algorithm	Data classification

7.2. Artificial Neural Networks-Based Genetic Algorithms

Holland firstly introduced the genetic algorithm concept in 1975. It is a stochastic global adaptive search optimization technique based on the mechanisms of natural selection [134]. The GA algorithm solves optimization problems by applying a series of crossover, mutation, and fitness evaluations to multiple chromosomes. This algorithm is initialized to a population containing several chromosomes, in which each one represents the optimal solution of the problem that is evaluated by an objective function [87]. Many researchers use the GA for different applications. Some such research was about renewable energies applications, such as the maximum power point tracking for PV and wind systems, to improve distribution systems' reliability and power quality [135]. Ongoing studies focus more on GA for enhancing the ANN than other neural networks in comparison. For example, the GA is used for outline capturing using rational functions and ANN to solve energy management applications such as scheduling and economic dispatches [136]. It is also used for solving reliability problems of structural laminated composite materials [137]. In [138], it solves bankruptcy prediction problems, while that combination is used to solve circular tubes with functionally graded thickness problems with multiple objective crashworthiness optimizations [139].

ANN-based GA is applied in many ways; some are related to optimizing the ANN structures design. For example, in [140], an ANN used GA to optimize parameters to determine the number of hidden neurons, bias values, and the connection weights between nodes to solve time series forecasting problems. Also, it is used for weights optimization of ANN on a pre-specified neural network applied on a mobile ad-hoc network [141]. GA-based ANN is used for solving many issues, such as producing spectra for prediction, parameter fitting, inverse design, and performance to design network architectures and select optimal hyperparameters [142]. In the same way, it is used to compute a heat transfer study in [143] to determine suitable parameters for maximum weight reduction. GA is also used to select optimal network parameters for a deep-NN model architecture to model prospective university students' admission [144].

Many types of research use these two smart concepts by merging them for many applications and classification problems. In [145], an ANN hybridized with GA was used to optimize lipase production from *Penicillium roqueforti* ATCC 10110 in solid-state fermentation in. A multi-layer ANN united with GA was employed to solve problems of pectinase-assisted extraction of cashew apple juice [146]. Some studies discuss parametric study problems of the transcritical power cycle and regenerator by selecting objective functions for parametric optimization [147]. In [148], a nanofluid flow in flat tubes using computational fluid dynamics problems was solved using multi-objective ANN optimization and non-dominated sorting GA (NSGA). Also, decouples capacitor placement on a power delivery network, while another example used for analog circuit design space exploration for automated sizing of integrated circuits [149].

In some cases, the neural network works with more than one optimization for either comparison or combination reasons. For example, the GA and PSO work together on ANN to find the best values of the rational functions' parameters for optimizing surface roughness [12]. in [150,151] the GA is used with Adadelta DNN (GA-ADNN) to predict catenary comprehensive pantograph and catenary monitor status models. Table 4 presents different studied involving GA for neural network design and application enhancement. In [152], a hybrid PSO with GA for ANN training for short-term load forecasting and GA optimization was used to solve power grid investment risk problems by optimizing the weight and threshold of the BP neural network, while in A GA was also applied on three neural networks (MLP), radial basis functions neural network (RBFNN), and a GA-derived generalized regression neural network (GRNN) for discovering the optimal weights to solve the problem of predicting groundwater salinity [153]. Also,

Table 4. Studies involving GA for neural network design and application enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [152]	PSO &GA	To overcome the training issue of local minima traps	Short-term load forecasting
ANN [137]	GA	To overcome high computational cost by using multilayer perceptron NN	Design of anisotropic laminated composite structures
ANN [143]	GA	To determine suitable parameters for maximum weight reduction	Heat transfer analysis in perforated plate fins
ANN [138]	GA	To solve the data imbalance problem caused by simultaneous ANN optimization.	Corporate bankruptcy prediction
DNN [144]	GA	To select optimal network parameters of the Deep-NN	Binary classification for university student admissions
ANN [139]	Crashworthiness optimization and GA	To design parameter alternatives and determine optimal combinations.	Circular tubes having a functionally graded thickness
ANN [140]	GA	To find the number of hidden neurons, bias values of hidden neurons, and the connection weights between nodes.	Time-series forecasting for real-life data
ANN [145]	GA	To optimize lipase production through the ANN model	Lipase production from *Penicillium roqueforti* ATCC 10110 in solid-state fermentation
ANN [146]	GA	Optimum extraction parameters	Low-temperature extraction of cashew apple juice
ANN [147]	GA	To optimize the thermal efficiency, exergy efficiency, and specific network.	Transcritical power cycle with regenerator
ANN [148]	Non-dominated Sorting GA	To numerically solve problems in various flat tubes for nanofluid flow analysis and regime	Nanofluid flow in flat tubes

Table 4. *Cont.*

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [17]	GA	To minimize the number of decoupling capacitors for reducing the differences between the input impedance	PCB decoupling
ANN [149]	GA	For analog circuit optimization system automated sizing of integrated circuits	Analog design space exploration
ADNN [150]	GA	To prevent prediction models from falling into local optimum and a comprehensive catenary model	Pantograph and catenary
ANN [151]	GA	To optimize the weight and threshold of a BP neural network	Power grid investment risk problems
ANN [141]	GA	For weight optimization in a pre-specified neural network	Applied on a mobile ad-hoc network
ANN [142]	GA	To design the network architecture and select the hyperparameters for ANNs	Plasmonic waveguide systems
MLP, RBFNN & GRNN [153]	GA	Search for optimal weights	Predicting groundwater salinity

7.3. *Artificial Neural Networks-Based Artificial Bee Colony*

Many optimizations are used for optimizing neural networks to find the values of linkage weights either alone or associated with biases and neurons in hidden layers. Many researchers have considered the ABC for boosting neural network performance either by optimizing the hyperparameters or somehow merging to enhance the neural network or applications. An example of improving the ANN is an efficient model based on the ABC optimization algorithm with neural networks [154]. The ABC algorithm uses an alternative learning scheme to optimize neuron connection weights for the design of ANN structures used for electric load forecasting to obtain an optimized set of neuron connection weights [155]. In [156], intrusion detection for cloud computing using ANN neural networks and an ABC and fuzzy logic for identified normal and abnormal network traffic packets by optimizing the values of linkage weights and biases [156]. Deep neural networks are good for classification problems, and some studies use the ABC algorithm with DNN. For example, in the ABC algorithm search for hybridization parameters of DNN structure, this study included autoencoder layers cascaded to a softmax classification layer [157].

Also, a modular neural network presents a modular NN model based on the ABC algorithm for electric load forecasting with synaptic weights optimization [158]. On the other hand, some research is merging the neural networks with ABC to solve specific problems. For example, a study using a swarm-inspired algorithm with ANN to protect against dual attacks using the concept of ANN as a deep learning algorithm and the swarm-based ABC optimization technique [8]. Table 5 lists studies involving ABC for neural network design and application enhancement.

Table 5. Studies involving ABC for neural networks design and application enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [155]	ABC	To optimized set of neuron connection weights	Electric load forecasting
Modular NN [158]	ABC	For synaptic weights optimization	Classifier designed for NN
MLP network [156]	ABC & Fuzzy clustering algorithms	To optimize linkage weights and biases	Intrusion detection for cloud computing
DNN [8]	swarm-based ABC	To optimize DNN parameter protection against dual attacks	Mobile ad hoc network for mitigation of black and gray holes attacks
DNN [157]	ABC & BFGS	For hybridization parameters of deep neural networks	Data classification of dimensions and sizes

7.4. Artificial Neural Networks Based Evolutionary Algorithm

Most of the research discussing neural networks-based evolutionary algorithms improves the neural networks' design to either reduce the training time or solve problems encountered by ANNs [159]. This combination is used in many applications to solve different problems, for example, an adaptive co-optimization of ANNs using EA for global radiation forecasting using hybrid ANN models. It predicted monthly radiation by typical weather and geographic data-adaptive the EAs utilized to improve prediction performance was developed to train the neural networks [160]. At the same time, another study used multiverse optimization for new natural EA together with ANN to develop advanced detection approaches for intrusion detection systems [161]. The combined effort between ANN and EA is reported in many research studies, yet only some significant research has been considered in this review. For example, in a correlation analysis of the training process, self-organizing combined with genetic EA, is applied to boost built structures of neural network's performance and efficiency [162], while another research study evaluated a model-based optimization process for high voltage alternating current systems [163]. Though some studies use EA optimization for ANN weights optimization, this unique combination is used in mobile communications to solve weights optimization problems in ANN optimal modeling by applying a framework for predicting received signal strength [164]. Also, in the chemistry field, the EA introduces chemical reaction optimization (CRO), used as a global optimization technique to replace BP in training neural networks [165], for better performance and saving more time for the training process. An optimization technique, EA based on pieces of training, approximates the solution of fractional differential equations [166]. Table 6 presents research involving EA for neural network structure design and application enhancement.

Table 6. Studies involving EA for neural networks design and application enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [160]	EA	To co-optimize the ANN properties	Global radiation forecasting
ANN [161]	Multiverse optimizer (MVO)/EA	To allow ENN to solve problems encountered by ANNs	Intrusion detection systems using multiverse optimization via a benchmark dataset
ANN [162]	Self-organized genetic EA	To improve the performance efficiency and structural efficiency of the built ANN	Structure of neural network and its implementation
ANN [163]	EA	For optimization and ANN for modeling	High voltage AC systems
ANN [164]	EAs	For self-adaptive control parameters and dynamically adjust the population size for ANN weight optimization	Unmanned aerial vehicle measurements for mobile communications
ANN [166]	EA	To adjust the weights to satisfy the differential equations	Differential equations of fractional order
ANN [165]	EA/CRO	To replace backpropagation in training neural networks	ANN architecture design

7.5. Artificial Neural Networks-Based Backtracking Search Algorithm

BSA optimization technique is an evolutionary computation technique for producing a trial population that includes two new crossovers and mutation operators proposed by [62]. BSA dominates searching for the best value of the populations and searches in the space boundary to get the exploitation capabilities and very robust exploration [62]. BSA dominates the search's value for the best populations and the boundary of the space to provide very sturdy exploration and exploitation capabilities. Thus, considerable research has proven it as one of the most powerful optimization techniques [62]. Numerous researchers widely use BSA in modern applications, such as solving the state of charge of lithium-ion batteries by improving a backpropagation neural network (BPNN) by optimizing hidden layer neurons' optimal value learning rate [167]. The BSA improved neural network with random weights by combining BSA and a neural network with random weights (NNRWs) to optimize the hidden layer parameters of the single-layer feed-forward network (SLFN), and NNRWs is used to derive the output layer weights [168]. In [169] a modified BSA (MBSA) has been improve by learning and niching together with ANN training and in [170] an ANN prediction method based on adaptive BSA was used for optimizing the connection weights matrix of the echo state network reservoir [170]. These studies involving BSA for neural network design and applications enhancement for the best neural network structure boost the performance level and reduce the time-consuming network setup. Table 7 presents different research projects involving BSA for design and application enhancement.

Table 7. Studies involving BSA for neural network design and application enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
Back-propagation NN [167]	BSA	To find the optimal values of hidden layer neurons and learning rate	Estimating state of charge of lithium-ion batteries
SLFN [168]	BSA	To optimize the neural network with random weights, and derive the output layer weights.	Improve neural network design
ANN [169]	Modified BSA	For learning and niching strategies such as learning strategy, a niching strategy, and a mutation strategy	Chaotic time series prediction and benchmark functions
Echo State Network/RNN [170]	Adaptive BSA	To optimize the connection weights matrix of the echo state network reservoir	Echo state network architecture design
ANN [171]	Binary BSA	To optimize the number of nodes in hidden layers and learning rate	Energy management to reduce the cost

7.6. Artificial Neural Networks Based Other Optimization Search Algorithms

Neural network-based optimization algorithms are a hot topic in the research field. Many algorithms have been studied in the past ten years; their combination has become very attractive because of the incredible outcomes from that merging or enhancement. As a result, many studies have been conducted in different applications in life; in this section, some significant research has been investigated to highlight the importance of enhancing neural networks with optimizations. A short wind speed forecasting-based prediction problem has been solved by ANN hybrid with crisscross optimization in [172]. ANN also solves the reliability-based design problem of double-loop reliability-based optimization approaches [173]. Deterministic global optimization and ANN to solve the convex and concave envelopes of the nonlinear activation function in [4]. A graph neural network called RouteNet solves complex relationships between topology, routing, and input traffic to produce accurate estimates solutions in [81].

This section focuses on mixing different types of neural networks with other optimization algorithm techniques. For example, FNN training employs a symbiotic organisms search (SOS) algorithm to solve the UCI machine learning repository problems [84]. An ANN model using the teaching–learning-based optimization algorithm (TLBO) solves energy consumption estimates in Turkey [174]. Also, the ANNs with ant colony optimization (ACO) assess residential buildings' performance by training the NN based on ACO instead of the BP algorithm [175]. In [176], a social spider optimization was used to improve the training phase of ANN with multilayer perceptrons for the context of Parkinson's disease recognition. A dynamic optimization problem (DOPs) used with neural network (NN)-based information transfer method (NNIT) used for solving issues associated with environmental changes in [177]. Also, an automated optimization-oriented strategy for designing high power amplifiers using DNNs with a deep learning regression network and electromagnetic-based Thompson sampling efficient multi-objective optimization (TSEMO) [178]. Another continuous optimization based on deep RNNs uses metaheuristic algorithms to solve the difficulties of optimization problems for noise to signal ratio [40]. A neural network in numerous learning problems and backpropagation (BP) methods as correntropy-based conjugate gradient BP (CCG-BP) in [179]. A DNN based on a secure precoding scheme is a deep AN scheme for solving artificial noise scheme problems in multiple-input single-output (MISO) wiretap channels [180]. A deep CNN (DCNN) structure modeling for reconstruction enhancement and decreasing online prediction in ANN is used for anthropomorphic manipulators in [181]. In [182], three DNNs called deep multilayer perceptron (DMLP), long-short memory (LSTM) neural network, and CNN, were used to build prediction-based portfolio optimization models in the Chinese stock market. This combination has come to an optimal prediction without optimization in comparison to the other studies. A hybrid method for electricity price forecasting by ANN and artificial cooperative search algorithm (ACS) for the combination of mutual information and neural network (NN) in [183]. Table 8 presents research involving various optimization techniques based on neural network design and application enhancement.

Table 8. Overview of a variety of optimization techniques based on neural network design and application enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
FNNs [84]	SOS	For training of FNNs	UCI machine learning repository
ANN [174]	TLBO	To replace the BP with TLBO	Estimates of energy consumption in Turkey
ANN [176]	Social-spider Optimization	To improve the training phase of ANN with multilayer perceptrons	Parkinson's disease identification
DMLP, LSTM, CNN [182]	DNNs	DNNs to predict each stock's future return also DNNs are applied to measure the risk of each stock	Portfolio optimization models utilizing the stocks market of China
ANN [177]	NNIT & EA	To solve dynamic optimization problems	Moving peaks benchmark
DNN [178]	TSEMO & DNN	To get the number of passive components in the input and output matching networks	Designing high power amplifier circuit topologies
RNN [40]	Metaheuristic Algorithms	For the objective analytic function of a continuous optimization problem	Estimate tree structures
ANN [179]	CCG-BP	Optimizing common correntropy-based BP algorithms based on MSE	Improving training in NNs for enhancing the signal-to-noise ratios
DNN [180]	Deep AN	For optimal precoding scheme	Artificial noise scheme wiretap channels
ANN [181]	DCNN	For reconstruction enhancement and reducing online prediction time	Anthropomorphic manipulators
ANN [183]	ACS	To select the input variables subsets for forecasting of electricity price	Forecasts of short-term electricity prices in a deregulated market

The following examples enhance neural networks by optimizing their weights connections, for example, a prediction of time series to adjust the weights in the ANNs model with parameter-free simplified swarm optimization (SSO) [184]. ANN-based biogeography-based optimization (BBO) also solved electrical energy forecasting problems for long-term forecasting of India's sector-wise electrical energy demand [185]. Again, an enhanced ANN with a shuffled complex evolutionary global optimization algorithm with principal component analysis—University of California Irvine (SP-UCI) for the weight training for feedforward ANN [186]. Another example of weights linkages optimization is done in a metaheuristic, bird mating optimizer (BMO), which was used to train feedforward ANNs in [21]. Also, a quantum-based algorithm was used to design an ANN with few connections and high classification performance by simultaneously optimizing the network structure and the connection weights [187]. Neural network training with a weighting mechanism-based optimization algorithm was used to resolve some algorithms' undesirable convergence behavior and improve Adam and AMSGrad [188]. A unified automated model generation algorithm uses optimization to automatically determine the type and topology of the mapping structure in a knowledge-based neural network model to force some weights of the mapping neural networks to zeros while leaving other weights nonzeros optimized in [88]. An Elman neural network was used to train the connection weights between the layers based on a whale optimization algorithm (WOA) to solve the problem of falling into local best solutions [189]. Another optimization of connection weights in neural networks using the WOA for training ANN and verified by comparisons with BP algorithm other evolutionary techniques was described in [190]. An evolutionary nonlinear adaptive filter approach via cat swarm functional link ANN (CS-FLANN) was employed for solving unwanted noise problems by picking the optimum weights of NN filters in [191]. Cat swarm optimization (CSO) was also used to train the ANN for structure design by simultaneously optimizing the connection weights [192]. A calibration method was done to improve the robot positional accuracy of industrial manipulators using a teaching-learning-based optimization (TLBO) method to optimize the weights and bias in ANN in [193]. ANNs based sparse optimization simultaneously estimates the weights and model structure of an ANN in [194]. Table 9 lists optimization-based neural network weights optimization enhancements.

Table 9. Overview of a variety of optimization-based neural network weights optimization enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [185]	BBO	To obtain the best global weight parameters	Long-term sector-wise power forecasting
ANN [186]	(SP-UCI)	To the weight-training process of a three-layer feed-forward ANN	Gradient-based optimization schemes
ANN [184]	SSO	To adjust the weights in ANNs	For ANN modeling
ANN [21]	BMO	For weight training of ANNs	Solving three real-world classification problems
ANN [187]	Quantum-based algorithm	Few connections and high classification performance using connection weights.	ANN design and structure
ANN [194]	sparse optimization	To simultaneously estimate the weights and model structure of an ANN	Model structure of ANN
ANN [188]	NWM-Adam	To resolve the undesirable convergence behavior by weighting mechanism-based first-order gradient descent optimization	For effective neural network training
Knowledge-based NN [88]	i_1Optimizer	To force some weights of the NNs to zeros while leaving other weights as non-zeros.	For unified automated parametric modeling algorithm
Elman Neural Network [189]	WOA	To train the connection weights between the layers	Network soft-sensor model of conversion velocity in a polymerization process
ANN [190]	WOA	Optimizing connection weights of ANN controlling parameters weights and biases	ANN structure design
FLANN [191]	CSO	For the selection of an optimum weight of the neural network filter	Gaussian noise removal from tomography Images
ANN [192]	CSO & OBD	For optimization of the connection weights	ANN structure design
ANN [193]	TLBO	To optimize weights and bias of the NN	Robot manipulator

The following studies overview some examples of optimization enhancement of various optimization-based neural network parameters (hidden layers, learning rate, neurons, and wights). For example, a hybrid lightning search algorithm (LSA)-based ANN can predict the optimal ON/OFF status for home appliances for home energy management by tuning the learning rate value and the number of nodes in the hidden layers in [30]. Also, in [195], FNNs are based on artificial fish swarm optimization (AFSA) to replace the BP process in ANN. A DNN based on multi-objective was used for solving the connecting structure DNNs, particularly the layerwise structure learning method, in [80]. An optimization method for CNNs based on the difference between the present and the immediate past gradient diffGrad optimization technique to solve the problem with basic stochastic gradient descent (SGD) in [103]. A global optimal known as Bayesian optimization (BayesOpt) is a machine learning-based global optimization technique to solve a simple objective function problem in CNN [196]. A systematic quantitative and qualitative analysis and guidelines using CNN-based Ben's spiker algorithm [197]. In [198], the microcanonical optimization algorithm (MOA) is used to select the best hyperparameter architecture for CNN for a variant of simulated annealing. DNNs with stochastic optimization acceleration update the network parameters to solve PID controller problems in [199]. A hybrid neuro-fuzzy network-based differential biogeography-based optimization (DBBO) for on-line population classification in an earthquakes optimizer searches for the best parameter for the main network and the subnetwork [200]. An adaptive memetic algorithm with a rank-based mutation (AMARM) is used to design ANN architectures by a simultaneously fine-tune number of hidden neurons as well as connection weights in [201]. ANN-based path loss prediction for wireless communication network multilayer perceptron (MLP) neural network generates low dimensional environmental features and eliminates redundant information among similar environmental types [202]. Table 10 overviews various optimization-based neural network parameters (hidden layers, learning rate, neurons) optimization enhancement.

Table 10. Overview of various optimization-based neural network parameters (hidden layers, learning rate, neurons, and weights) optimization enhancement.

Neural Networks	Optimizer	Optimizer Problem	Application Improved
ANN [195]	AFSA	To obtain hidden layers trained by the back-propagation algorithm	Design of neural networks
ANN [202]	PL	To eliminate redundant information by impacts of the number of neurons in the hidden layer, number of hidden layers, number of training samples	Wireless communication network
DNN [80]	Multi-objective Optimization	To find the optimal structure with high representation ability and better generalization for each layer.	Structure of DNN model
CNN [103]	diffGrad	To adjust each parameter for faster gradient changing parameters	Image categorization experiments
ANN [30]	LSA	Using suitable learning rate value and number of nodes in the hidden layers	Home energy management scheduling
CNN [196]	BayesOpt	To utilize both a simple objective function and a proper optimization	low-rank decomposition
CNN [197]	Ben's Spiker algorithm	For parameter optimization	Signal-to-noise ratio
CNN [198]	MOA	For hyper-parameter optimization and architecture selection for CNN	Using six widely-used image recognition datasets
DNN [199]	SGD-M	Use past and present gradients for DNN parameter updates	PID Controller
Hybrid neuro-fuzzy [200]	DBBO	For parameter optimization of both the main network and the subnetwork	Online population classification in earthquakes
ANN [201]	AMARM	To simultaneously fine-tune the number of hidden neurons and connection weights	Design ANN architectures.

7.7. Optimization Search Algorithm-Based Artificial Neural Networks

In this subsection, neural networks work as an optimizer for optimization techniques to optimize algorithm parameters. For example, as in a study, the fitness function value in a pressurized water reactor core was optimized by pattern optimization using a grey wolf algorithm (GWO)-based ANN to solve the best configuration for fuel assemblies [203], while in [204] parameter prediction, using an ANN as a tool in finding the parameter optimization of resistance spot welding optimization (RSW) to solve the sensitivity of exact measurement for aluminum alloy was decribed. A topology optimization accelerated-based deep learning study discussed learning a cross-sectional image of an interior permanent magnet motor represented in RGB and trained a CNN to infer the torque properties to decrease computational cost for the optimization topology (TO) [11]. Table 11 presents studies involving neural networks for improving optimization techniques design and application enhancement.

Table 11. Studies involving neural networks for improving optimization techniques design and application enhancement.

Neural Networks Optimizer	Optimization Algorithm Enhanced	Optimizer Problem	Application Improved
ANN [203]	GWO	ANN is applied to estimate the fitness function value of GWO	Pressurized water reactor
ANN [204]	RSW	ANN as a tool in finding the parameter optimization of RSW	A sensitive to exact measurement of aluminum alloy
CNN [11]	TO	The trained CNN approximately evaluates individuals	Cross-sectional image of an interior permanent magnet motor

8. Application on Artificial Neural Networks Based Optimization Algorithms

Previous research on optimal scheduling controllers was developed for energy management, reliable power generation, cost minimization, and carbon emission calculation [64]. A binary BSA algorithm and binary PSO are utilized to search for optimal binary schedules [120,205]. These algorithm techniques have powerful optimization skills, search exploration process, fast convergence for the solution, and other conventional optimization techniques and overcome local minima traps. Besides, developing an enhanced ANN-based BBSA and ANN-based BPSO schedule controller ensures the best performance across different load conditions. [206,207]. An ANN is on track as a prediction technique to find the best weight values for neural nets designed for efficient system operation. In this paper, these nets operate in optimum ON/OFF status by training on input and output data patterns obtained from scheduling controllers [206,208]. This section presents ANN-based optimization algorithms implementation for ANN-PSO, ANN-GA, ANN-ABC, and ANN-BSA, respectively, to search for the optimal values of the number of nodes in hidden nodes layer1 and layer2 as well as the best value of the learning rate. The algorithms apply limitations of, e.g., (max and min) number of nodes in each hidden layer and the learning rate. The output data is relayed on a binary schedule (25x24) obtained from the scheduling controller. The input data includes six inputs, including solar irradiances, wind speed, energy price, battery status, gird status, and diesel fuel status refer to [206,208]. In all the ANN-based algorithms, the iteration of ANN is set to 100 iterations, and the population size is 20 populations. Table 10 presents a brief list describing the data and the limitations of the aforementioned algorithm techniques. The mean absolute error (MAE) is an objective function that enhances the ANN performance by decreasing the error function expressed in the general flow chart of optimization of ANN optimal parameters, as shown in Figure 7. All the inputs and outputs of the ANN-based optimization algorithm training for the virtual power plants system in [208] can be expressed by the following Equations (1) and (2):

$$Input = \begin{bmatrix} solar\ irradiances\ (R) \\ wind\ Speed\ (W) \\ energy\ price\ (E) \\ battey\ status\ (B) \\ grid\ status\ (G) \\ diesel\ fuel\ status\ (D) \end{bmatrix} \tag{1}$$

$$Output = \begin{bmatrix} DG_{(1,1)} & \cdots & DG_{(1,25)} \\ \vdots & \ddots & \vdots \\ DG_{(24,1)} & \cdots & DG_{(24,25)} \end{bmatrix} \tag{2}$$

ANN deep neural networks using feed-forward structures have been adopted in this study. The use of trainlm as a network training function that updates weight and bias values according to Levenberg-Marquardt optimization is considered the fastest back-propagation algorithm in the Matlab toolbox. However, it does require more memory than other algorithms. Hidden layers are chosen to be two layers using the sigmoid activation function and the optimization is adapted to search for the number of the nodes in both hidden layers; the optimization algorithms are also set to search for the optimal value for learning rate. This optimization process is done using random trail values in ANN training based on the aforementioned inputs and outputs data. The optimal trail is the minimalist mean absolute error (MAE). Those trials from each optimization algorithm have been done separately, and each optimization takes days to come up with the best parameters. All these algorithms have addressed the limitations of the search for a set of trails, as presented in Table 12. The algorithms use random ANN trial parameters as the initial step pre iteration process includes ANN training for 10000 epochs to evaluate the minimum objective function. Figure 8 shows that the numbers of inputs and outputs layers are known based on the data [209]. The duration time of each ANN training is unexpected could take a long or short time depending on the trial training points of the ANN training. The training can show good or bad performance from the very early stages of the training, but this is not sure because it sometimes behaves differently and improves or remains in the middle or end of the training.

In this study, the BSA objective of enhancing the ANN structure toward optimal parameters was the best among the other techniques, which minimize the MAE to reaches a value of 0.0062 [171]. The GA objective was 0.0080, which is not very far from the BSA objective. Simultaneously, the MAE of the PSO and the ABC was greater at 0.0144 and 0.0172, respectively, compared to the other two techniques, as shown in Figure 9. The BSA's main principles have been done through its crossover consists of two parts. The first part generates the binary matrix, and the second part compares population X(i,j) and the trial population. Crossover is used to obtain an updated map(i,j). Also, this part works on the control mechanism of boundaries for a trial population. As presented, enhancing the neural network could help the system enormously and the enhanced ANN is proven to be overwhelmingly impressive, or at least competitive, by training and testing is as important as the optimal design of the ANN structure. This study also introduces a novel way of solving optimization tasks by the neural network.

Table 12. Optimization algorithms data and limitations.

Symbol	Description
P	Controller input data
t	Controller output data
$iteration_{ANN} = 100$	Maximum iterations for ANN
$Population\ size = 20$	Size of the population
$Lower_{LR} = 0$	Min value of LR
$Upper_{LR} = 1$	Max value of LR
$Lower_{N1} = 6$	Min value of nodes in hidden layer1
$Upper_{N1} = 30$	Max value of nodes in hidden layer1
$Lower_{N2} = 6$	Min value of nodes in hidden layer2
$Upper_{N2} = 30$	Max value of nodes in hidden layer2

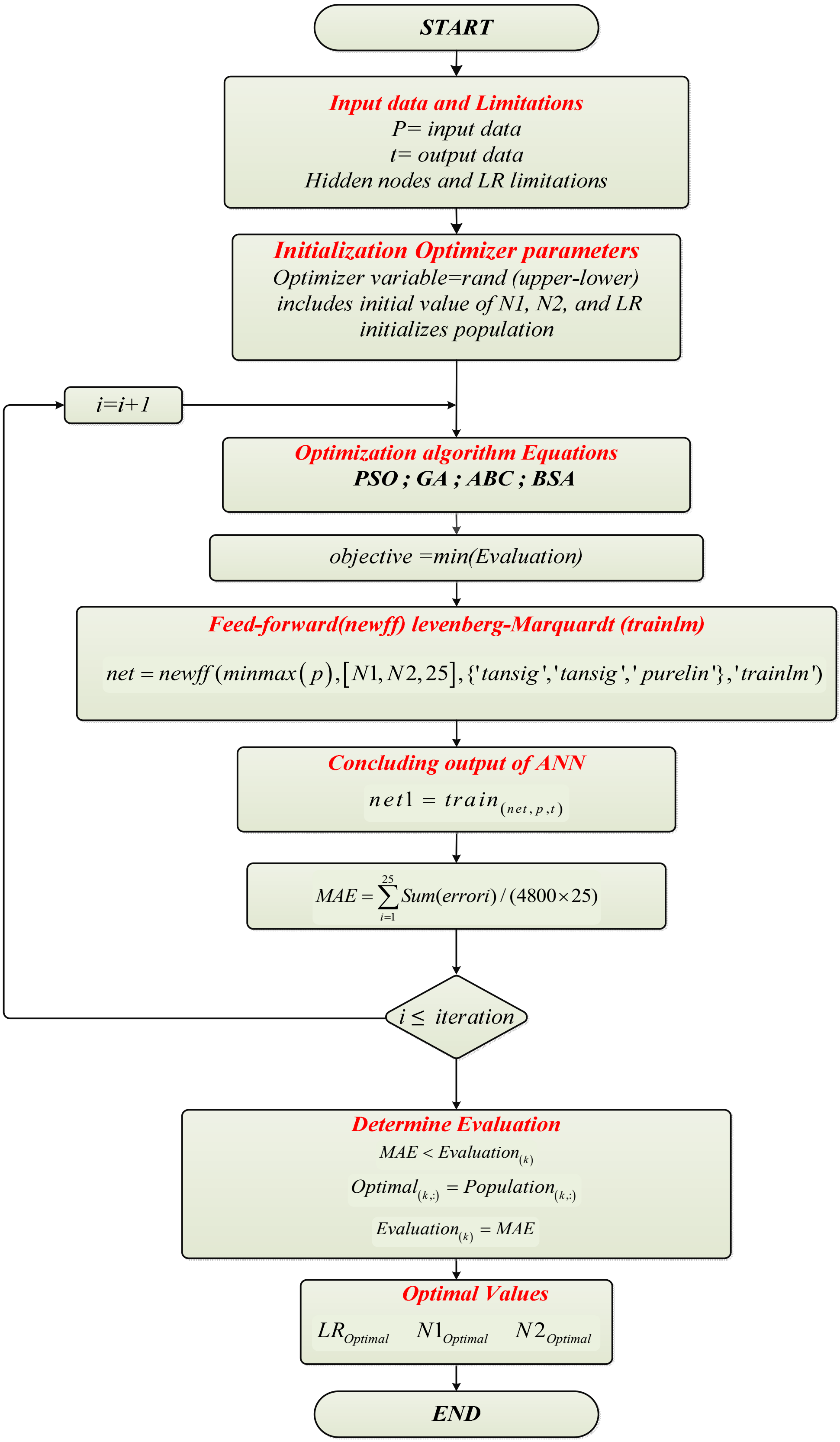

START
Input data and Limitations
P= input data
t= output data
Hidden nodes and LR limitations
Initialization Optimizer parameters
Optimizer variable=rand (upper-lower)
includes initial value of N1, N2, and LR
initializes population
i=i+1
Optimization algorithm Equations
PSO ; GA ; ABC ; BSA
objective =min(Evaluation)
Feed-forward(newff) levenberg-Marquardt (trainlm)
net = newff (minmax(p),[N1, N2, 25],{'tansig','tansig','purelin'},'trainlm')
Concluding output of ANN
net1 = train(net,p,t)
MAE = Σ_{i=1}^{25} Sum(errori) / (4800×25)
i ≤ iteration
Determine Evaluation
MAE < Evaluation(k)
Optimal(k,:) = Population(k,:)
Evaluation(k) = MAE
Optimal Values
LR_Optimal N1_Optimal N2_Optimal
END

Figure 7. General flow chart of optimization algorithms for ANN-PSO, ANN-GA, ANN-ABC, and ANN-BSA for 100 iteration.

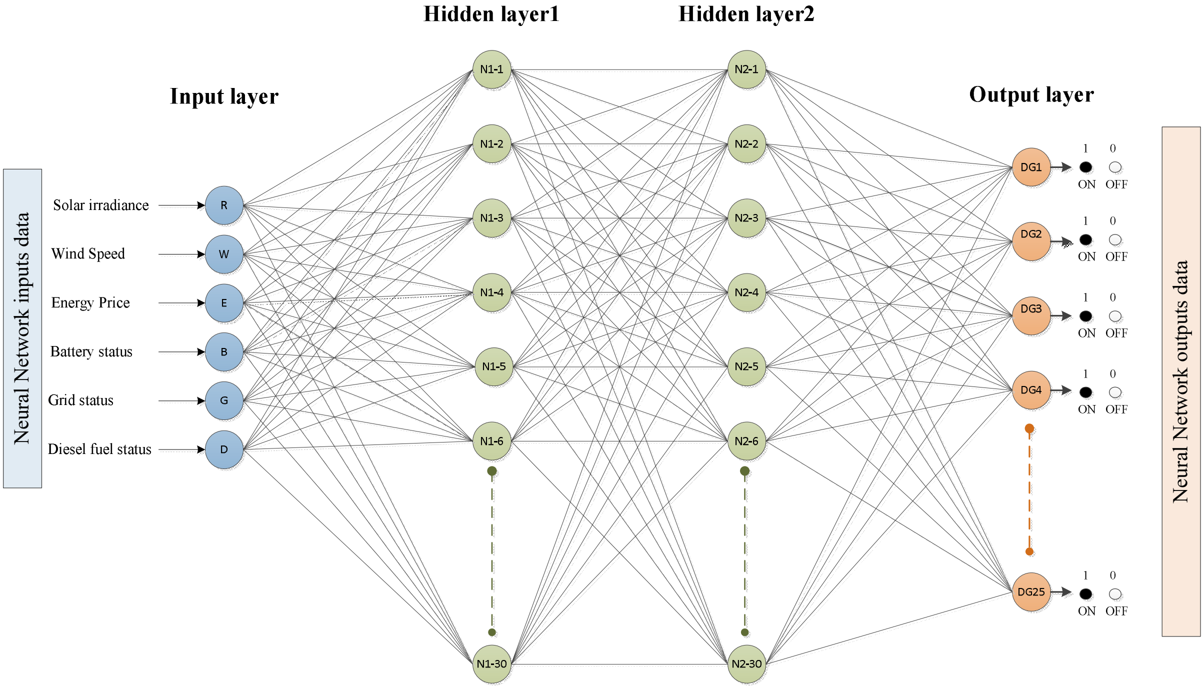


Figure 8. ANN Architecture Based on PSO, GA, ABC, and BSA algorithms, using input and output data obtained from [208].

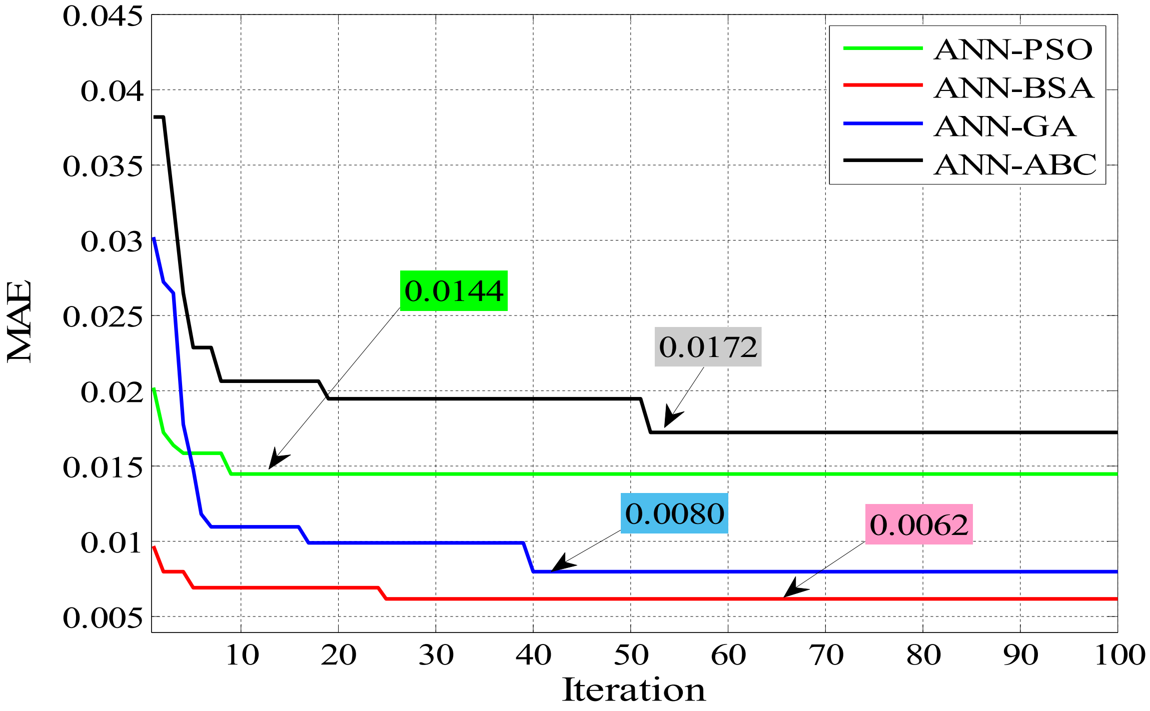


Figure 9. Objectives of optimization algorithms for ANN-PSO, ANN-GA, ANN-ABC, and ANN-BSA for 100 iterations.

9. Artificial Neural Network Training-Based Optimized Parameters

When applying the optimized parameters in ANN training using the input and output data for each optimization technique separately, this training process will result in a net for each optimization algorithm. The obtained net is the masterpiece, and it is the intelligent controller that can replace the ordinary controller to predict unexpected non-linear input to result in a wise decision. The enhanced ANN saves training time and ANN parameters chosen wisely by the optimization algorithms. The results are better than human decisions no matter what type of optimization is used [171,208,209]. The pseudocode of ANN training based on the optimal parameters, and the outcome is a Net for ANN-PSO, ANN-GA, ANN-ABC, and ANN-BSA [210–212]. Since the net output is 0 or 1 hourly pattern, we can call this net an intelligent binary controller [120,171,206,208]. The following is the Algorithm 1.

Algorithm 1. Pseudocode of ANN training based on optimized parameters obtained from optimization algorithms.

```
Input: (solar irradiances, wind speed, energy price, battery status, gird status, and diesel fuel status)
Output: ANN-Net of the binary matrix of (24 × 25)
N1 = optimal value obtained
N2 = optimal value obtained we can call this net an intelligent binary controller
LR = optimal value obtained
// ANN
Applying Feed-Forward neural network ($newff$) and Levenberg-Marquardt ($trainlm$)
$net = newff(minmax_{(p)}, [N1, N2, 25], \{'tansig', 'tansig', 'purelin'\}, 'trainlm')$
$net{\cdot}trainParam{\cdot}epochs = 10,000$
$net{\cdot}trainParam{\cdot}lr = LR$
$net{\cdot}trainParam{\cdot}goal = 0$
$net1 = train_{(net,p,t)}$
$gen\sin(Net1, -1)$
Output is an ANN-Net with input data and 25 outputs
```

The optimal enhanced net of ANN-BSA in a Matlab Simulink block is shown in Figure 10, involving six inputs and twenty-five binary outputs on an hourly basis to manage distributed generators throughout the virtual power plant system. The net block is generated after the training completer by using Equation (3). Table 13 presents the ANN training-based PSO, GA, ABC, and BSA using the optimized parameters. The generated ANN Net module is an AI controller; it is considered a masterpiece and smart controller. This Net could be implemented in cheap microchips and used as a smart device to control hug systems to serve in a very effective smart way cheaply.

$$gen\sin(Net1, -1) \tag{3}$$

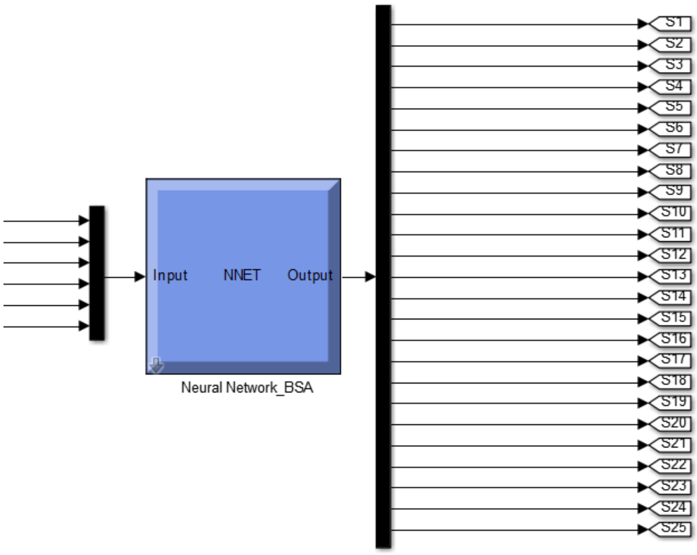
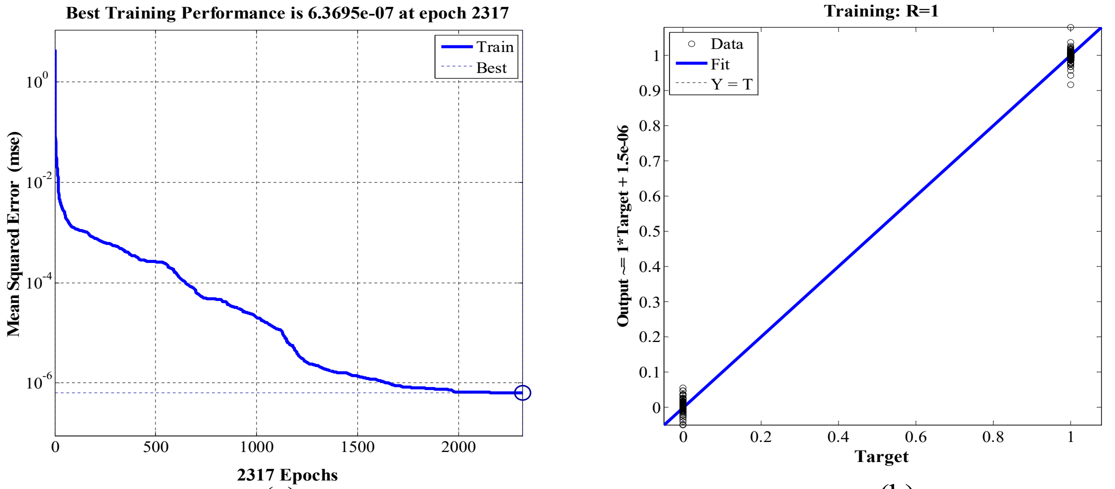


Figure 10. MATLAB Simulink block of Neural Network Net of ANN-BSA.

Table 13. Artificial neural network training-based PSO, GA, ABC, and BSA using the optimized parameters obtained.

Optimization Algorithm	No. of Nodes in Hidden Layer1 (N1)	No. of Nodes in Hidden Layer2 (N2)	Learning Rate Value (LR)	Training Time	Training Performance (MSE)
PSO	18	30	0.7	20:00:48	3.99×10^{-6}
GA	23	28	0.6	20:31:36	5.46×10^{-6}
ABC	26	29	0.45	30:32:29	2.52×10^{-5}
BSA	22	27	0.6	4:30:29	6.37×10^{-7}

The following figures represent the training performance and regressions for ANN deep neural network after using the optimization algorithms' optimal parameters. This study shows a fair compression based on each optimization technique to find the best parameters to serve the system in the best way. These hybrid techniques could save huge trial and error time during training and find the required best parameters, using smaller nets to save valuable time during the training and testing. Any of the optimization algorithms used could give better results than manual parameters tuning. Yet, some techniques could find the best fitness faster and more efficiently than others, as ANN-BSA in Figure 11, which shows the best training performance of 6.3695×10^{-7} at 2317 epochs and regression (R) reach to best of 1 regression training which the best results it may obtain by training.

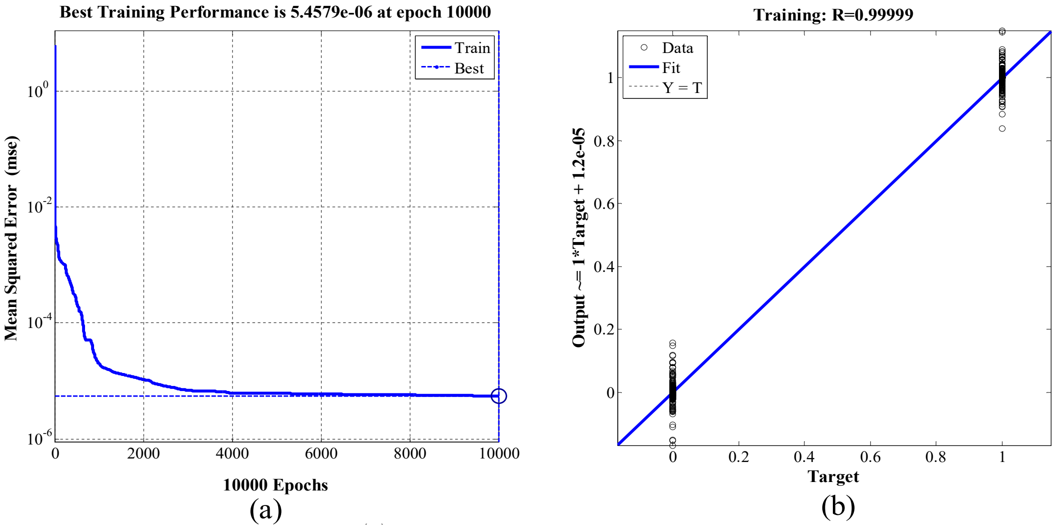


(a) (b)

Figure 11. (**a**) Performance and (**b**) regression of ANN training after applying optimal parameters of ANN-BSA.

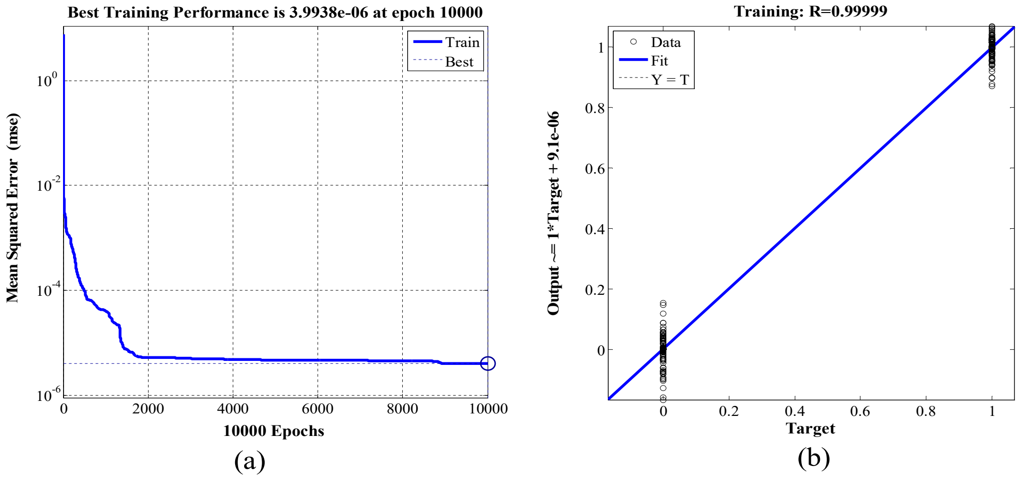
However, other optimization techniques trained for 10,000 epochs on their optimal parameter have good results somewhat near in results. However, in Figure 12, ANN-GA shows the best training performance of 5.4579×10^{-6}, and regression (R) reach 0.99999 it is very close to unity. Figures 13 and 14 show the best training performance of 3.9938×10^{-6} and 2.5178×10^{-5} and regression of 0.99999 and 0.99995 for ANN-PSO and ANN-ABC [171].

Fair comparison results of Bus1 of 14-bus IEEE test system for virtual power plants utilize the optimized ANN net based on half-hour binary patterns for managing each distributed generation (DG) unit in the system. The binary (ANN-BPSO), binary (ANN-BABC), binary ANN-BGA, and binary (ANN-BBSA) is a controller with binary output 0 or 1 to switch each DG ON or Off based on the inputs. Figure 15 shows that the entire algorithm saved a huge amount of power. Yet, all the saved power was considered with sharing new distributed resources to inject power to the loads instead of supplying power from the utility grid [210]. However, most of the optimized Nets have done an excellent job. However, some Nets are better than the others based on their objectives as can be seen that the total power for the 24 h of the ANN-BBSA Net was 1182.5 MW in comparison to other optimized Nets 1211.3 MW, 1184.3 MW, and 1252.9 MW for, ANN-BGA, ANN-BPSO, ANN-BABC, respectively.



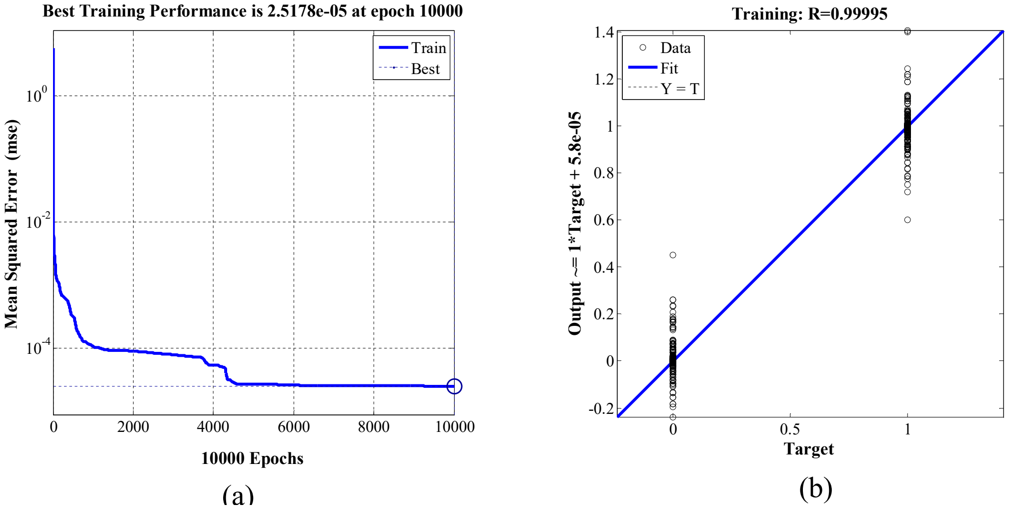
(a) (b)

Figure 12. (**a**) performance and (**b**) regression of ANN training after applying optimal parameters of ANN-GA.



(a) (b)

Figure 13. (**a**) Performance and (**b**) regression of ANN training after applying optimal parameters of ANN-PSO.



(a) (b)

Figure 14. (**a**) Performance and (**b**) regression of ANN training after applying optimal parameters of ANN-ABC.

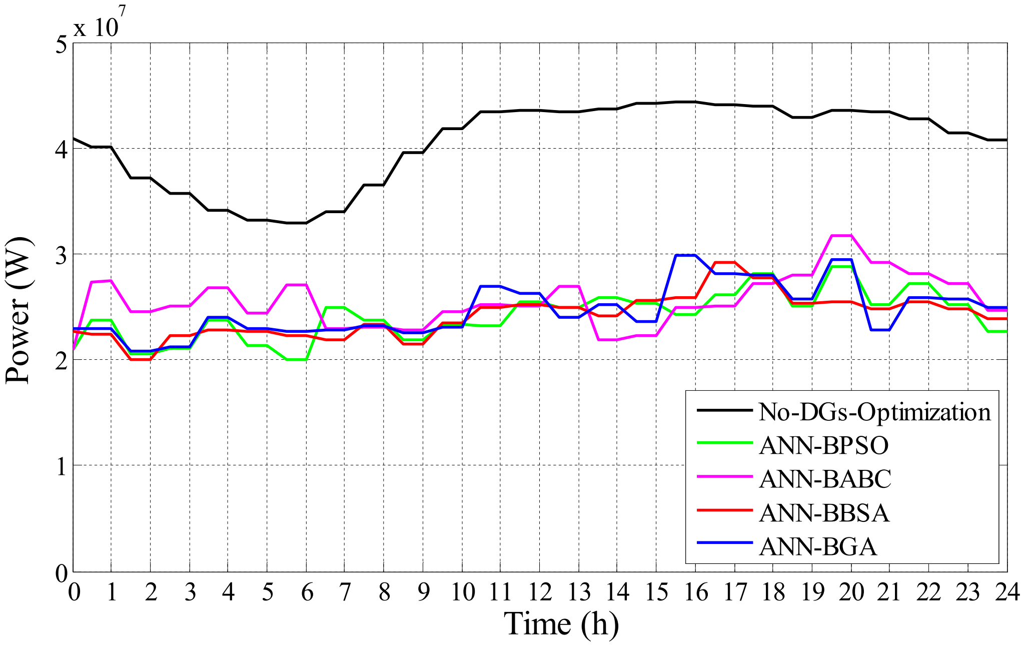


Figure 15. Original Bus1 of 14-bus IEEE test system compared to ANN-based binary optimization algorithms ANN-BPSO, ANN-BABC, ANN-BGA, and ANN-BBSA.

Much research is conducted to address the ANN enhancement to present extraordinary results compared to using the same ANN. The difference between the first and the second results is because the involvement of optimization techniques to find the best parameters. This method has been evaluated compared to other trends research discussing similar issues in evaluating the applied approach compared to other methods, Table 14 presents a comparison of the proposed technique with other enhancing neural networks by finding the optimal parameters of no. of nodes in hidden layers and learning rate. Table 15 shows an overview of neural networks-based optimization techniques for the optimal number of nodes in hidden layers and learning rate. The table states that ANN-based optimization techniques have gained a momentum trend in the last five years. This enhancement becomes essential in most AI applications used for ANFIS and fuzzy to optimize the best membership function shapes. Also, the optimization techniques are used to enhance the PI controllers to select the best parameters. Also, in many ML to improve the classification or regression are utilized.

Table 14. Comparison of the proposed technique with other techniques of enhancing neural networks by finding the optimal parameters of No. of nodes in hidden layers and learning rate.

Optimization Algorithm	Objective Function (MAE)	No. of Nodes in Hidden Layer 1 (N1)	No. of Nodes in Hidden Layer 2 (N2)	Learning Rate Value (LR)	No. of Input and Output	Regression (R)	Training Performance (MSE)
Hybrid LSA-ANN [30]	9.128×10^{-9}	6	4	0.6175	5 and 4	1	9.128×10^{-9}
PSO-DNN [115]	-	20	60	0.1	12	-	-
Hybrid ANN-PSO [29]	0.1742	18	16	0.071	3 and 1	0.99991	-
BPNN-PSO [116]	0.1911×10^{-2}, 0.2032×10^{-2}	14 and 9	9 and 11	0.7373, 0.6481	7 and 1	0.99993, 0.99999	4.3×10^{-5}
ANN-PSO tested	0.0144	18	30	0.7	6 and 25	0.99999	3.99×10^{-6}
ANN-GA tested	0.0080	23	28	0.6	6 and 25	0.99999	5.46×10^{-6}
ANN-ABC tested	0.0172	26	29	0.45	6 and 25	0.99995	2.52×10^{-5}
ANN-BSA tested	0.0062	22	27	0.6	6 and 25	1	6.37×10^{-7}

Table 15. Overview of significant studies on NNs based optimization using node numbers in hidden layers and learning rate.

Enhancement Method	Reference	Year	Application Enhanced
ANN-based GA	[140]	2021	Time-series forecasting for real-life data
ANN-based binary BSA	[171]	2021	Virtual power plant
ANN-based PSO	[120]	2021	Microgrid energy management
ANN-based LSA	[30]	2016	Home energy management
DNN-based PSO	[115]	2019	Digital modulation recognition
ANN-based AMARM	[201]	2017	Design ANN architectures.
BPNN-based PSO	[116]	2021	Global solar irradiance prediction
ANN-based BSA	[167]	2018	Estimating state of charge of lithium-Ion battery
ANN-based PSO	[211]	2016	Wireless sensor localization for cycling tracking
ANN-based PL	[202]	2020	Wireless communication network
ANN-based ABC	[155]	2014	Electric load forecasting
ANN-based SAPSO	[117]	2020	Kambara reactor desulfurization

10. Conclusions and Future Work

This review includes extensive research on ANNs' importance, advantages and types of utilization in a series of applications and also neural network enhancement based on optimizations for network architecture design, training and testing. The literature shows that optimization for AI generally and neural networks specifically has been a hot topic during the past ten years and has increased year by year up to 2021. The review has undertaken neural network enhancements by optimizing the parameters, such as weight optimizations, initial weight optimizations, bias and learning rate optimizations, number of hidden layers, number of nodes in hidden layers, and activation functions. On the other hand, the enhancement could be trained by modifying the neural network's regular algorithms, for example, replacing the feed-forward or the back-propagation for tuning network weights based on the error rate per epochs. This review covers a test case study of ANN-based optimization algorithm techniques to provide a quick example of ANN improvement. As presented, the hybrid or mix techniques of ANN-PSO, ANN-GA, ANN-ABC, and ANN-BSA are compared in a fair comparison for their objectives, regressions, training performance, training time, and application of microgrid energy management. The influence of each technique can economize time for selecting parameters and for the training. The enhanced ANN nets are tested on distributed energy resources in the form of an energy management system. The quick results show that virtual power plants save a reasonable amount of supplied power. From this review, the research emphasizes enhancing the neural network by optimization algorithms used to search for ANN best parameters and training parameters to achieve the best structure network have been satisfied based on the comparison tables as well as the testing results for improving the ANN performance by PSO, GA, ABC, and BSA, optimization techniques. Also, this review has proved that neural network is a very hot topic recently and could improve both neural networks or maybe other AI techniques or ML to search for optimal parameters to solve the problems in a short time and efficient way. This review and the case study also include several important and targeted recommendations for the further development of the ANN-based optimization method, such as:

- Generally, ANN intelligent methods are associated with powerful optimization tools, such as PSO, ABC, BSA, and GA techniques, in various engineering applications, such as electromagnetism, signal processing, and pattern recognition and classification, robotics. Nevertheless, they have a problem with constancy and cost. Thus, future research should be conducted on the appropriate optimization method selection, finding the system's optimal value, such as cost-effect components with high accuracy.
- The conventional NN technologies create issues; for example, the human brain is highly complex, non-linear, and sensitive [212]. Therefore, additional investigation is needed on human brain monitoring optimization to obtain high accuracy. The low timing loss under the high-risk, complex situation to achieve high reliability,

modularity, efficiency, and performance; further investigation of the system's proper optimization selection is needed.

- Despite the benefits of optimization algorithms in reducing technical loss, low error, and cost, their use in ANN has been very limited. Only computational intelligence optimization algorithms have made significant progress toward optimizing the controller design and the price. As a result, advanced optimization algorithms will be better choices for ANN design.
- Enhancement of ANN parameters with optimization could result from new algorithms that save more time adjusting the ANN toward optimal architectures by avoiding trial and error or random selection. Like this, the optimal solution is considered as a smaller network, a straightforward calculation method, and less time could be achieved.

The neural networks of intelligence dependence evolutionary algorithms improve neural networks' design by reducing training time or solving problems using the ANN method. For quick tracking, less steady-state errors, and high performance, ANN techniques can be used to monitor robotic sensing and control monitor and achieve bidirectional power management. However, real-time data integrity, reduced operations time, expensive processing equipment, and the need for good parameter selection and manual tuning are all disadvantages. As a result, more research is required to select proper optimization methods for enhancing neural network structure design is important.

DL methods are fast evolving for higher performance. There are adequate review articles about the progressing algorithms in particular application domains. Future work could be carried out considering other DL methods such as denoising autoencoder, deep belief networks, and long short-term memory. Further study and review can enhance or hybridize ML with optimization techniques, random forest, Markov chain Monte Carlo, or support victor machines. Future work can also consider many optimizations to improve AI and ML to boost their performance [213–215]. Future studies can consider DL from another perspective, for example, continuous or online optimization.

Author Contributions: Conceptualization, T.S.U., M.G.M.A. and J.A.A.; methodology, M.G.M.A. and M.R.S.; software, M.G.M.A.; A.M. and J.A.A., validation M.A.H., S.M.; formal analysis, T.S.U.; investigation, S.M.S.H. and T.S.U.; resources, T.S.U. and R.M.; data curation, M.G.M.A. and R.M.; writing—original draft preparation, M.G.M.A., R.M.; writing—review and editing, S.M., S.M.S.H. and T.S.U. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

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

Abbreviations

ABC	Artificial bee colony
ACO	Ant colony optimization
ACS	Artificial cooperative search algorithm
ADNN	Adadelta deep neural networks
ADPSO	Adaptive dynamic particle swarm optimization
AFSA	Artificial fish swarm optimization
AI	Artificial intelligence
AMARM	Adaptive memetic algorithm with a rank-based mutation
AMSG	Adam optimization stochastic gradient descent
ANFIS	Adaptive neuro-fuzzy inference systems
ANN	Artificial neural networks
ANN-ABC	Artificial neural networks-based artificial bee colony
ANN-BSA	Artificial neural networks-based backtracking search algorithm
ANN-BABC	Artificial neural networks-based bainary artificial bee colony

ANN-BBSA	Artificial neural networks-based binary backtracking search algorithm
ANN-BGA	Artificial neural networks-based binary genetic algorithm
ANN-BPSO	Artificial neural networks-based binary particle swarm optimization
ANN-GA	Artificial neural networks-based genetic algorithm
ANN-PSO	Artificial neural networks-based particle swarm optimization
Aop	Airblast-overpressure
AP	Affinity propagation
BABC	Binary artificial bee colony
BBBC	Big bang big crunch
BBO	Biogeography-based optimization
BBSA	Binary backtracking search algorithm
BCA	Bee colony algorithm
BFGS	Limited memory Broyden Fletcher Goldfarb Shannon
BFO	Bacteria foraging optimization
BGA	Binary genetic algorithm
BMO	Bird mating optimizer
BP	Backpropagation
BPNN	Backpropagation neural network
BPNN-PSO	Backpropagation neural network-based particle swarm optimization
BPSO	Binary particle swarm optimization
BSA	Backtracking search algorithm
CCG-BP	Correntropy-based conjugate gradient-backpropagation
CCPSO	Cultural cooperative particle swarm optimization
CNN	Convolutional neural networks
COA	Chaotic optimization algorithm
CRO	Chemical reaction optimization
CS	Cuckoo search
CSA	Cuckoo search algorithm
CSO	Cat swarm optimization
DBBO	Differential biogeography-based optimization
DCNN	Deep convolutional neural networks
DG	Distributed generation
DL	Deep learning
DMLP	Deep multilayer perceptron
DNN	Deep neural networks
DOP	Dynamic optimization problem
DSA	Dolphin swarm algorithm
EA	Evolutionary algorithms
EBP	Elman backpropagation algorithm
EFA	Electromagnetism-based firefly algorithm
ENN	Elman neural network
FA	Firefly algorithm
FLANN	Functional link artificial neural networks
FLNFN	Functional-link-based neural fuzzy network
FNN	Optimize feedforward NN
GA	Genetic algorithm
GAN	Generative adversarial network
GNN	Graph Neural Networks
GRNN	Generalized regression neural network
GSA	Gravitational search algorithm
GWO	Grey wolf algorithm
HSA	Harmony search algorithm
IT2FNN	Interval type-2 fuzzy neural networks
LR	Learning Rate

LSA	Lightning search algorithm
LSA-ANN	Lightning search algorithm-based particle swarm optimization
LSTM	Long short-term memory
MAE	Mean absolute error
MBSA	Modified backtracking search algorithm
MISO	Multiple-input single-output
ML	Machine learning
MLP	Multilayer perceptron
MNN	Modular neural network
MOA	Microcanonical optimization algorithm
MSE	Mean squire error
MVO	Multiverse optimizer
NN	Neural networks
NNA	Neural network algorithm
NNIT	Neural network-based information transfer
NNRW	Neural network with random weights
NSGA	Non-dominated sorting GA
OBD	Optimal brain damage
OPSONN	Opposition-based PSO neural network
PI	Proportional integral
PID	Proportional integral derivative
PL	Path Loss
PMS	Periodic mutation strategy
PNN	Probabilistic neural network
PSO	Particle swarm optimization
PSO-DNN	Particle swarm optimization-based deep neural network
PV	Photovoltaic
QLSA	Quantum-inspired lightning search algorithm
RBF	Radial basis functions
RBFNN	Radial basis functions neural network
RNN	Recurrent neural networks
RSW	Resistance spot welding optimization
SAPSO	Simulation annealing algorithm with particle swarm optimization
SGD	Stochastic gradient descent
SLFN	Single-layer feed-forward network
SOS	Symbiotic organisms search
SPS-PSO	Self-adaptive parameters and strategy-based PSO
SSO	Simplified swarm optimization
TCPSO	Tent-map chaotic particle swarm optimization
TLBO	Teaching–learning-based optimization algorithm
TO	Optimization topology
TPSO	Taguchi particle swarm optimization
TSEMO	Thompson sampling efficient multi-objective optimization
UCI	University of California Irvine
UCS	Unconfined compressive strength
WEC	Wave energy converters
WOA	Whale optimization algorithm

References

1. Oliver, J.M.; Esteban, M.D.; López-Gutiérrez, J.-S.; Negro, V.; Neves, M.G. Optimizing Wave Overtopping Energy Converters by ANN Modelling: Evaluating the Overtopping Rate Forecasting as the First Step. *Sustainability* **2021**, *13*, 1483. [CrossRef]
2. Mosavi, A.; Salimi, M.; Ardabili, S.F.; Rabczuk, T.; Shamshirband, S.; Varkonyi-Koczy, A.R. State of the Art of Machine Learning Models in Energy Systems, a Systematic Review. *Energies* **2019**, *12*, 1301. [CrossRef]
3. Zhou, H.; Sun, G.; Fu, S.; Liu, J.; Zhou, X.; Zhou, J. A big data mining approach of PSO-Based BP neural network for financial risk management with IoT. *IEEE Access* **2019**, *7*, 154035–154043. [CrossRef]

4. Schweidtmann, A.M.; Mitsos, A. Deterministic Global Optimization with Artificial Neural Networks Embedded. *J. Optim. Theory Appl.* **2019**, *180*, 925–948. [CrossRef]
5. Li, T.; Chan, Y.H.; Lun, D.P.K. Improved Multiple-Image-Based Reflection Removal Algorithm Using Deep Neural Networks. *IEEE Trans. Image Process.* **2021**, *30*, 68–79. [CrossRef] [PubMed]
6. Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; Bengio, Y. Generative adversarial networks. *Commun. ACM* **2020**, *63*, 139–144. [CrossRef]
7. Ardabili, S.; Mosavi, A.; Dehghani, M.; Várkonyi-Kóczy, A.R. Deep Learning and Machine Learning in Hydrological Processes Climate Change and Earth Systems a Systematic Review. In *Engineering for Sustainable Future*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 101, pp. 52–62. [CrossRef]
8. Rani, P.; Kavita; Verma, S.; Nguyen, G.N. Mitigation of Black Hole and Gray Hole Attack Using Swarm Inspired Algorithm with Artificial Neural Network. *IEEE Access* **2020**, *8*, 121755–121764. [CrossRef]
9. Milad, A.; Adwan, I.; Majeed, S.A.; Yusoff, N.I.M.; Al-Ansari, N.; Yaseen, Z.M. Emerging Technologies of Deep Learning Models Development for Pavement Temperature Prediction. *IEEE Access* **2021**, *9*, 23840–23849. [CrossRef]
10. Moayedi, H.; Bui, D.T.; Gör, M.; Pradhan, B.; Jaafari, A. The feasibility of three prediction techniques of the artificial neural network, adaptive neuro-fuzzy inference system, and hybrid particle swarm optimization for assessing the safety factor of cohesive slopes. *ISPRS Int. J. Geo-Inform.* **2019**, *8*, 391. [CrossRef]
11. Sasaki, H.; Igarashi, H. Topology optimization accelerated by deep learning. *IEEE Trans. Magn.* **2019**, *55*, 1–5. [CrossRef]
12. Shamshirband, S.; Mosavi, A.; Rabczuk, T.; Nabipour, N.; Chau, K. Prediction of significant wave height; comparison between nested grid numerical model, and machine learning models of artificial neural networks, extreme learning and support vector machines. *Eng. Appl. Comput. Fluid Mech.* **2020**, *14*, 805–817. [CrossRef]
13. Gonçalves, R.; Ribeiro, V.M.; Pereira, F.L.; Rocha, A.P. Deep learning in exchange markets. *Inf. Econ. Policy* **2019**, *47*, 38–51. [CrossRef]
14. Mosavi, A.; Ardabili, S.; Várkonyi-Kóczy, A.R. List of Deep Learning Models. In *Engineering for Sustainable Future*; Springer: Berlin/Heidelberg, Germany, 2019; Volume 101, pp. 202–214. [CrossRef]
15. Kim, K.G. Deep learning book review. *Nature* **2019**, *29*, 1–73. [CrossRef]
16. Lecun, Y.; Bengio, Y.; Hinton, G. Deep learning. *Nature* **2015**, *521*, 436–444. [CrossRef]
17. Cecchetti, R.; de Paulis, F.; Olivieri, C.; Orlandi, A.; Buecker, M. Effective PCB Decoupling Optimization by Combining an Iterative Genetic Algorithm and Machine Learning. *Electronics* **2020**, *9*, 1243. [CrossRef]
18. Mijwil, M.M. Artificial Neural Networks Advantages and Disadvantages. *Linkedin* **2018**, 1–2. Available online: https://www.linkedin.com/pulse/artificial-neural-networks-advantages-disadvantages-maad-m-mijwel/ (accessed on 2 April 2021).
19. Nabipour, N.; Dehghani, M.; Mosavi, A.; Shamshirband, S. Short-Term Hydrological Drought Forecasting Based on Different Nature-Inspired Optimization Algorithms Hybridized with Artificial Neural Networks. *IEEE Access* **2020**, *8*, 15210–15222. [CrossRef]
20. Jafarian, F.; Taghipour, M.; Amirabadi, H. Application of artificial neural network and optimization algorithms for optimizing surface roughness, tool life and cutting forces in turning operation. *J. Mech. Sci. Technol.* **2013**, *27*, 1469–1477. [CrossRef]
21. Askarzadeh, A.; Rezazadeh, A. Artificial neural network training using a new efficient optimization algorithm. *Appl. Soft Comput. J.* **2013**, *13*, 1206–1213. [CrossRef]
22. Zappone, A.; Di Renzo, M.; Debbah, M.; Lam, T.T.; Qian, X. Model-Aided Wireless Artificial Intelligence: Embedding Expert Knowledge in Deep Neural Networks for Wireless System Optimization. *IEEE Veh. Technol. Mag.* **2019**, *14*, 60–69. [CrossRef]
23. Jiang, J.; Fan, J.A. Simulator-based training of generative neural networks for the inverse design of metasurfaces. *Nanophotonics* **2019**, *9*, 1059–1069. [CrossRef]
24. Mutlag, A.H.; Shareef, H.; Mohamed, A.; Hannan, M.A.; Abd Ali, J. An improved fuzzy logic controller design for PV inverters utilizing differential search optimization. *Int. J. Photoenergy* **2014**, *2014*. [CrossRef]
25. Aljarah, I.; Al-Zoubi, A.M.; Faris, H.; Hassonah, M.A.; Mirjalili, S.; Saadeh, H. Simultaneous Feature Selection and Support Vector Machine Optimization Using the Grasshopper Optimization Algorithm. *Cogn. Comput.* **2018**, *10*, 478–495. [CrossRef]
26. Ghazvinei, P.T.; Darvishi, H.H.; Mosavi, A.; bin Wan Yusof, K.; Alizamir, M.; Shamshirband, S.; Chau, K. Sugarcane growth prediction based on meteorological parameters using extreme learning machine and artificial neural network. *Eng. Appl. Comput. Fluid Mech.* **2018**, *12*, 738–749. [CrossRef]
27. Ardabili, S.; Mosavi, A.; Várkonyi-Kóczy, A.R. Systematic Review of Deep Learning and Machine Learning Models in Biofuels Research. *Engineering for Sustainable Future* **2019**, *Volume 101*, 19–32. [CrossRef]
28. Taghizadeh-Mehrjardi, R.; Emadi, M.; Cherati, A.; Heung, B.; Mosavi, A.; Scholten, T. Bio-Inspired Hybridization of Artificial Neural Networks: An Application for Mapping the Spatial Distribution of Soil Texture Fractions. *Remote Sens.* **2021**, *13*, 1025. [CrossRef]
29. Gharghan, S.K.; Nordin, R.; Ismail, M.; Ali, J.A. Accurate Wireless Sensor Localization Technique Based on Hybrid PSO-ANN Algorithm for Indoor and Outdoor Track Cycling. *IEEE Sens. J.* **2016**, *16*, 529–541. [CrossRef]
30. Ahmed, M.; Mohamed, A.; Homod, R.; Shareef, H. Hybrid LSA-ANN Based Home Energy Management Scheduling Controller for Residential Demand Response Strategy. *Energies* **2016**, *9*, 716. [CrossRef]
31. Yu, J.; Xi, L.; Wang, S. An improved particle swarm optimization for evolving feedforward artificial neural networks. *Neural Process. Lett.* **2007**, *26*, 217–231. [CrossRef]

32. Dineva, A.; Mosavi, A.; Ardabili, S.F.; Vajda, I.; Shamshirband, S.; Rabczuk, T.; Chau, K.-W. Review of Soft Computing Models in Design and Control of Rotating Electrical Machines. *Energies* **2019**, *12*, 1049. [CrossRef]
33. Ayub, S.; Guan, B.H.; Ahmad, F.; Oluwatobi, Y.A.; Nisa, Z.U.; Javed, M.F.; Mosavi, A. Graphene and Iron Reinforced Polymer Composite Electromagnetic Shielding Applications: A Review. *Polymers* **2021**, *13*, 2580. [CrossRef] [PubMed]
34. Ayub, S.; Guan, B.H.; Ahmad, F.; Javed, M.F.; Mosavi, A.; Felde, I. Preparation Methods for Graphene Metal and Polymer Based Composites for EMI Shielding Materials: State of the Art Review of the Conventional and Machine Learning Methods. *Metals* **2021**, *11*, 1164. [CrossRef]
35. Moayedi, H.; Mosavi, A. An Innovative Metaheuristic Strategy for Solar Energy Management through a Neural Networks Framework. *Energies* **2021**, *14*, 1196. [CrossRef]
36. Nosratabadi, S.; Mosavi, A.; Duan, P.; Ghamisi, P.; Filip, F.; Band, S.S.; Reuter, U.; Gama, J.; Gandomi, A.H. Data Science in Economics: Comprehensive Review of Advanced Machine Learning and Deep Learning Methods. *Mathematics* **2020**, *8*, 1799. [CrossRef]
37. Mosavi, A.; Faghan, Y.; Ghamisi, P.; Duan, P.; Ardabili, S.F.; Salwana, E.; Band, S.S. Comprehensive Review of Deep Reinforcement Learning Methods and Applications in Economics. *Mathematics* **2020**, *8*, 1640. [CrossRef]
38. Chen, H.; Heidari, A.A.; Chen, H.; Wang, M.; Pan, Z.; Gandomi, A.H. Multi-population differential evolution-assisted Harris hawks optimization: Framework and case studies. *Futur. Gener. Comput. Syst.* **2020**, *111*, 175–198. [CrossRef]
39. Wang, Z.; Chen, B.; Wang, J.; Chen, C. Networked microgrids for self-healing power systems. *IEEE Trans. Smart Grid* **2016**, *7*, 310–319. [CrossRef]
40. Tian, Y.; Peng, S.; Zhang, X.; Rodemann, T.; Tan, K.C.; Jin, Y. A Recommender System for Metaheuristic Algorithms for Continuous Optimization Based on Deep Recurrent Neural Networks. *IEEE Trans. Artif. Intell.* **2020**, *1*, 5–18. [CrossRef]
41. Balachennaiah, P.; Suryakalavathi, M.; Nagendra, P. Optimizing real power loss and voltage stability limit of a large transmission network using firefly algorithm. *Eng. Sci. Technol. Int. J.* **2016**, *19*, 800–810. [CrossRef]
42. Wong, L.A.; Shareef, H.; Mohamed, A.; Ibrahim, A.A. Novel quantum-inspired firefly algorithm for optimal power quality monitor placement. *Front. Energy* **2014**, *8*, 254–260. [CrossRef]
43. Ramli, L.; Sam, Y.M.; Mohamed, Z.; Khairi Aripin, M.; Fahezal Ismail, M.; Ramli, L. Composite nonlinear feedback control with multi-objective particle swarm optimization for active front steering system. *J. Teknol.* **2015**, *72*, 13–20. [CrossRef]
44. Lin, M.H.; Tsai, J.F.; Yu, C.S. A review of deterministic optimization methods in engineering and management. *Math. Probl. Eng.* **2012**, *2012*, 756023. [CrossRef]
45. Bui, D.K.; Nguyen, T.N.; Ngo, T.D.; Nguyen-Xuan, H. An artificial neural network (ANN) expert system enhanced with the electromagnetism-based firefly algorithm (EFA) for predicting the energy consumption in buildings. *Energy* **2020**, *190*, 116370. [CrossRef]
46. Hannan, M.A.; Ali, J.A.; Hossain Lipu, M.S.; Mohamed, A.; Ker, P.J.; Indra Mahlia, T.M.; Mansor, M.; Hussain, A.; Muttaqi, K.M.; Dong, Z.Y. Role of optimization algorithms based fuzzy controller in achieving induction motor performance enhancement. *Nat. Commun.* **2020**, *11*, 3792. [CrossRef]
47. Hannan, M.A.; Ali, J.A.; Mohamed, A.; Hussain, A. Optimization techniques to enhance the performance of induction motor drives: A review. *Renew. Sustain. Energy Rev.* **2018**, *81*, 1611–1626. [CrossRef]
48. Miao, K.; Feng, Q.; Kuang, W. Particle Swarm Optimization Combined with Inertia-Free Velocity and Direction Search. *Electronics* **2021**, *10*, 597. [CrossRef]
49. Garro, B.A.; Vázquez, R.A. Designing Artificial Neural Networks Using Particle Swarm Optimization Algorithms. *Comput. Intell. Neurosci.* **2015**, *2015*. [CrossRef] [PubMed]
50. Conforth, M.; Meng, Y. Toward evolving Neural networks using Bio-inspired algorithms. In Proceedings of the Artificial Intelligence and Soft Computing—ICAISC 2008, Zakopane, Poland, 22–26 June 2008; pp. 413–419.
51. Garro, B.A.; Sossa, H.; Vázquez, R.A. Back-Propagation vs Particle Swarm Optimization Algorithm: Which Algorithm is better to adjust the Synaptic Weights of a Feed-Forward ANN? *Int. J. Artif. Intell.* **2011**, *7*, 208–218.
52. Rosli, A.D.; Adenan, N.S.; Hashim, H.; Abdullah, N.E.; Sulaiman, S.; Baharudin, R. Application of Particle Swarm Optimization Algorithm for Optimizing ANN Model in Recognizing Ripeness of Citrus. *IOP Conf. Ser. Mater. Sci. Eng.* **2018**, *340*. [CrossRef]
53. Lazzús, J.A. Neural network-particle swarm modeling to predict thermal properties. *Math. Comput. Model.* **2013**, *57*, 2408–2418. [CrossRef]
54. Rashedi, E.; Nezamabadi-pour, H.; Saryazdi, S. GSA: A Gravitational Search Algorithm. *Inf. Sci.* **2009**, *179*, 2232–2248. [CrossRef]
55. Do, Q.H. A hybrid Gravitational Search Algorithm and back-propagation for training feedforward neural networks. In Proceedings of the Advances in Intelligent Systems and Computing; Springer: Berlin/Heidelberg, Germany, 2015; Volume 326, pp. 381–392.
56. Chaitanya, S.M.K.; Rajesh Kumar, P. Oppositional Gravitational Search Algorithm and Artificial Neural Network-based Classification of Kidney Images. *J. Intell. Syst.* **2020**, *29*, 485–496. [CrossRef]
57. Momeni, E.; Yarivand, A.; Dowlatshahi, M.B.; Armaghani, D.J. An efficient optimal neural network based on gravitational search algorithm in predicting the deformation of geogrid-reinforced soil structures. *Transp. Geotech.* **2021**, *26*, 100446. [CrossRef]
58. Zhang, Y.; Jin, Z.; Chen, Y. Hybrid teaching–learning-based optimization and neural network algorithm for engineering design optimization problems. *Knowl.-Based Syst.* **2020**, *187*. [CrossRef]

59. Sun, Y.; Cao, M.; Sun, Y.; Gao, H.; Lou, F.; Liu, S.; Xia, Q. Uncertain data stream algorithm based on clustering RBF neural network. *Microprocess. Microsyst.* **2021**, *81*, 103731. [CrossRef]
60. Faris, H.; Aljarah, I.; Al-Madi, N.; Mirjalili, S. Optimizing the Learning Process of Feedforward Neural Networks Using Lightning Search Algorithm. *Int. J. Artif. Intell. Tools* **2016**, *25*, 1650033. [CrossRef]
61. Sarker, M.R.; Mohamed, R.; Saad, M.H.M.; Mohamed, A. DSPACE Controller-based enhanced piezoelectric energy harvesting system using PI-lightning search algorithm. *IEEE Access* **2019**, *7*, 3610–3626. [CrossRef]
62. Zhang, Y.; Zhang, Y.; He, K.; Li, D.; Xu, X.; Gong, Y. Intelligent feature recognition for STEP-NC-compliant manufacturing based on artificial bee colony algorithm and back propagation neural network. *J. Manuf. Syst.* **2021**. [CrossRef]
63. Li, C. Biodiversity assessment based on artificial intelligence and neural network algorithms. *Microprocess. Microsyst.* **2020**, *79*, 103321. [CrossRef]
64. Civicioglu, P. Backtracking Search Optimization Algorithm for numerical optimization problems. *Appl. Math. Comput.* **2013**, *219*, 8121–8144. [CrossRef]
65. Guha, D.; Roy, P.K.; Banerjee, S. Application of backtracking search algorithm in load frequency control of multi-area interconnected power system. *Ain Shams Eng. J.* **2018**, *9*, 257–276. [CrossRef]
66. Abdolrasol, M.G.M.; Mohamed, A.; Hannan, M.A. Virtual power plant and microgrids controller for energy management based on optimization techniques. *J. Electr. Syst.* **2017**, *13*, 285–294.
67. Shareef, H.; Ibrahim, A.A.; Mutlag, A.H. Lightning search algorithm. *Appl. Soft Comput. J.* **2015**, *36*, 315–333. [CrossRef]
68. Abd Ali, J.; Hannan, M.; Mohamed, A. A Novel Quantum-Behaved Lightning Search Algorithm Approach to Improve the Fuzzy Logic Speed Controller for an Induction Motor Drive. *Energies* **2015**, *8*, 13112–13136. [CrossRef]
69. Liu, L.; Liu, W.; Cartes, D.A. Particle swarm optimization-based parameter identification applied to permanent magnet synchronous motors. *Eng. Appl. Artif. Intell.* **2008**, *21*, 1092–1100. [CrossRef]
70. Tabassum, M.; Mathew, K. A Genetic Algorithm Analysis towards Optimization solutions. *Int. J. Digit. Inf. Wirel. Commun.* **2014**, *4*, 124–142. [CrossRef]
71. Chao, K.-H.; Hsieh, C.-C. Photovoltaic Module Array Global Maximum Power Tracking Combined with Artificial Bee Colony and Particle Swarm Optimization Algorithm. *Electronics* **2019**, *8*, 603. [CrossRef]
72. Xue, Y.; Tang, T.; Liu, A.X. Large-scale feedforward neural network optimization by a self-adaptive strategy and parameter based particle swarm optimization. *IEEE Access* **2019**, *7*, 52473–52483. [CrossRef]
73. Chen, D.; Zou, F.; Lu, R.; Li, S. Backtracking search optimization algorithm based on knowledge learning. *Inf. Sci.* **2019**, *473*, 202–226. [CrossRef]
74. Sahu, R.K.; Panda, S.; Padhan, S. A novel hybrid gravitational search and pattern search algorithm for load frequency control of nonlinear power system. *Appl. Soft Comput. J.* **2015**, *29*, 310–327. [CrossRef]
75. Yang, X.S.; He, X. Firefly algorithm: Recent advances and applications. *Int. J. Swarm Intell.* **2013**, *1*, 36. [CrossRef]
76. Hassan, L.; Abdel-Nasser, M.; Saleh, A.; Omer, O.A.; Puig, D. Efficient Stain-Aware Nuclei Segmentation Deep Learning Framework for Multi-Center Histopathological Images. *Electronics* **2021**, *10*, 954. [CrossRef]
77. Arora, V.; Mahla, S.K.; Leekha, R.S.; Dhir, A.; Lee, K.; Ko, H. Intervention of Artificial Neural Network with an Improved Activation Function to Predict the Performance and Emission Characteristics of a Biogas Powered Dual Fuel Engine. *Electronics* **2021**, *10*, 584. [CrossRef]
78. Ketkar, N. Convolutional Neural Networks. In *Deep Learning with Python*; Apress: Berkeley, CA, USA, 2017; pp. 63–78.
79. Medsker, L.R.; Jain, L.C. Recurrent Neural Networks Design and Applications. *J. Chem. Inf. Model.* **2013**, *53*, 1689–1699.
80. Liu, J.; Gong, M.; Miao, Q.; Wang, X.; Li, H. Structure Learning for Deep Neural Networks Based on Multiobjective Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 2450–2463. [CrossRef]
81. Rusek, K.; Suarez-Varela, J.; Almasan, P.; Barlet-Ros, P.; Cabellos-Aparicio, A. RouteNet: Leveraging Graph Neural Networks for Network Modeling and Optimization in SDN. *IEEE J. Sel. Areas Commun.* **2020**, *38*, 2260–2270. [CrossRef]
82. Scarselli, F.; Gori, M.; Tsoi, A.C.; Hagenbuchner, M.; Monfardini, G. The graph neural network model. *IEEE Trans. Neural Netw.* **2009**, *20*, 61–80. [CrossRef]
83. Takayama, K.; Morva, A.; Fujikawa, M.; Hattori, Y.; Obata, Y.; Nagai, T. Formula optimization of theophylline controlled-release tablet based on artificial neural networks. *J. Control. Release* **2000**, *68*, 175–186. [CrossRef]
84. Wu, H.; Zhou, Y.; Luo, Q.; Basset, M.A. Training feedforward neural networks using symbiotic organisms search algorithm. *Comput. Intell. Neurosci.* **2016**, *2016*. [CrossRef]
85. Alsenwi, M.; Yaqoob, I.; Pandey, S.R.; Tun, Y.K.; Bairagi, A.K.; Kim, L.W.; Hong, C.S. Towards coexistence of cellular and WiFi networks in unlicensed spectrum: A neural networks based approach. *IEEE Access* **2019**, *7*, 110023–110034. [CrossRef]
86. Kusy, M.; Zajdel, R. Application of Reinforcement Learning Algorithms for the Adaptive Computation of the Smoothing Parameter for Probabilistic Neural Network. *IEEE Trans. Neural Netw. Learn. Syst.* **2015**, *26*, 2163–2175. [CrossRef]
87. Suganthi, L.; Iniyan, S.; Samuel, A.A. Applications of fuzzy logic in renewable energy systems—A review. *Renew. Sustain. Energy Rev.* **2015**, *48*, 585–607. [CrossRef]
88. Na, W.; Feng, F.; Zhang, C.; Zhang, Q.J. A Unified Automated Parametric Modeling Algorithm Using Knowledge-Based Neural Network and l1 Optimization. *IEEE Trans. Microw. Theory Tech.* **2017**, *65*, 729–745. [CrossRef]
89. Guo, S.; Pei, H.; Wu, F.; He, Y.; Liu, D. Modeling of solar field in direct steam generation parabolic trough based on heat transfer mechanism and artificial neural network. *IEEE Access* **2020**, *8*, 78565–78575. [CrossRef]

90. Do Nascimento, E.O.; De Oliveira, L.N. Numerical Optimization of Flight Trajectory for Rockets via Artificial Neural Networks. *IEEE Lat. Am. Trans.* **2017**, *15*, 1556–1565. [CrossRef]
91. Rayas-Sánchez, J.E. EM-based optimization of microwave circuits using artificial neural networks: The state-of-the-art. *IEEE Trans. Microw. Theory Tech.* **2004**, *52*, 420–435. [CrossRef]
92. Chaffart, D.; Ricardez-Sandoval, L.A. Optimization and control of a thin film growth process: A hybrid first principles/artificial neural network based multiscale modelling approach. *Comput. Chem. Eng.* **2018**, *119*, 465–479. [CrossRef]
93. Zhang, Z.; Cheng, Q.S.; Chen, H.; Jiang, F. An Efficient Hybrid Sampling Method for Neural Network-Based Microwave Component Modeling and Optimization. *IEEE Microw. Wirel. Components Lett.* **2020**, *30*, 625–628. [CrossRef]
94. Deodhare, D.; Vidyasagar, M.; Sathiya Keerthi, S. Synthesis of fault-tolerant feedforward neural networks using minimax optimization. *IEEE Trans. Neural Netw.* **1998**, *9*, 891–900. [CrossRef]
95. Song, X.; Kong, F.; Zhan, C.; Han, J. Hybrid Optimization Rainfall-Runoff Simulation Based on Xinanjiang Model and Artificial Neural Network. *J. Hydrol. Eng.* **2012**, *17*, 1033–1041. [CrossRef]
96. Tian, W.; Liao, Z.; Zhang, J. An optimization of artificial neural network model for predicting chlorophyll dynamics. *Ecol. Modell.* **2017**, *364*, 42–52. [CrossRef]
97. Ochoa-Estopier, L.M.; Jobson, M.; Smith, R. Operational optimization of crude oil distillation systems using artificial neural networks. *Comput. Chem. Eng.* **2013**, *59*, 178–185. [CrossRef]
98. Cui, S. Artificial neural network-based optimization of extraction of anthocyanins in black rice. *Food Sci. Technol.* **2012**, 1. Available online: https://en.cnki.com.cn/Article_en/CJFDTotal-SSPJ201201057.htm (accessed on 18 March 2021).
99. De Oliveira, M.B.W.; De Almeida Neto, A. Optimization of traffic lights timing based on Artificial Neural Networks. In Proceedings of the 2014 17th IEEE International Conference on Intelligent Transportation Systems, ITSC 2014, Qingdao, China, 8–11 October 2014; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2014; pp. 1921–1922.
100. Mukherjee, A.; Jain, D.K.; Goswami, P.; Xin, Q.; Yang, L.; Rodrigues, J.J.P.C. Back Propagation Neural Network Based Cluster Head Identification in MIMO Sensor Networks for Intelligent Transportation Systems. *IEEE Access* **2020**, *8*, 28524–28532. [CrossRef]
101. Rusydi, M.I.; Anandika, A.; Rahmadya, B.; Fahmy, K.; Rusydi, A. Implementation of Grading Method for Gambier Leaves Based on Combination of Area, Perimeter, and Image Intensity Using Backpropagation Artificial Neural Network. *Electronics* **2019**, *8*, 1308. [CrossRef]
102. Yang, F.; Moayedi, H.; Mosavi, A. Predicting the Degree of Dissolved Oxygen Using Three Types of Multi-Layer Perceptron-Based Artificial Neural Networks. *Sustainability* **2021**, *13*, 9898. [CrossRef]
103. Dubey, S.R.; Chakraborty, S.; Roy, S.K.; Mukherjee, S.; Singh, S.K.; Chaudhuri, B.B. DiffGrad: An Optimization Method for Convolutional Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 4500–4511. [CrossRef]
104. Zhao, L.; Hu, Z. Detection of Wildfire Smoke Images Based on a Densely Dilated Convolutional Network. *Electronics* **2019**, *8*, 1131. [CrossRef]
105. Zhao, E.; Liu, Y.; Zhang, J.; Tian, Y. Forest Fire Smoke Recognition Based on Anchor Box Adaptive Generation Method. *Electronics* **2021**, *10*, 566. [CrossRef]
106. Marini, F.; Walczak, B. Particle swarm optimization (PSO). A tutorial. *Chemom. Intell. Lab. Syst.* **2015**, *149*, 153–165. [CrossRef]
107. Kerdphol, T.; Fuji, K.; Mitani, Y.; Watanabe, M.; Qudaih, Y. Optimization of a battery energy storage system using particle swarm optimization for stand-alone microgrids. *Int. J. Electr. Power Energy Syst.* **2016**, *81*, 32–39. [CrossRef]
108. Momeni, E.; Jahed Armaghani, D.; Hajihassani, M.; Mohd Amin, M.F. Prediction of uniaxial compressive strength of rock samples using hybrid particle swarm optimization-based artificial neural networks. *Meas. J. Int. Meas. Confed.* **2015**, *60*, 50–63. [CrossRef]
109. Xiao, G.; Juan, Z.; Zhang, C. Detecting trip purposes from smartphone-based travel surveys with artificial neural networks and particle swarm optimization. *Transp. Res. Part C Emerg. Technol.* **2016**, *71*, 447–463. [CrossRef]
110. Li, N.; Chen, J.; Yuan, Y.; Tian, X.; Han, Y.; Xia, M. A Wi-Fi Indoor Localization Strategy Using Particle Swarm Optimization Based Artificial Neural Networks. *Int. J. Distrib. Sens. Netw.* **2016**, *12*, 4583147. [CrossRef]
111. Qiao, J.F.; Lu, C.; Li, W.J. Design of dynamic modular neural network based on adaptive particle swarm optimization algorithm. *IEEE Access* **2018**, *6*, 10850–10857. [CrossRef]
112. Yadav, N.; Yadav, A.; Kumar, M.; Kim, J.H. An efficient algorithm based on artificial neural networks and particle swarm optimization for solution of nonlinear Troesch's problem. *Neural Comput. Appl.* **2017**, *28*, 171–178. [CrossRef]
113. Das, G.; Pattnaik, P.K.; Padhy, S.K. Artificial Neural Network trained by Particle Swarm Optimization for non-linear channel equalization. *Expert Syst. Appl.* **2014**, *41*, 3491–3496. [CrossRef]
114. Serizawa, T.; Fujita, H. Optimization of convolutional neural network using the linearly decreasing weight particle swarm optimization. *arXiv* **2020**, arXiv:2001.05670.
115. Shi, W.; Liu, D.; Cheng, X.; Li, Y.; Zhao, Y. Particle Swarm Optimization-Based Deep Neural Network for Digital Modulation Recognition. *IEEE Access* **2019**, *7*, 104591–104600. [CrossRef]
116. Aljanad, A.; Tan, N.M.L.; Agelidis, V.G.; Shareef, H. Neural Network Approach for Global Solar Irradiance Prediction at Extremely Short-Time-Intervals Using Particle Swarm Optimization Algorithm. *Energies* **2021**, *14*, 1213. [CrossRef]
117. Wu, S.; Yang, J.; Zhang, R.; Ono, H. Prediction of Endpoint Sulfur Content in KR Desulfurization Based on the Hybrid Algorithm Combining Artificial Neural Network with SAPSO. *IEEE Access* **2020**, *8*, 33778–33791. [CrossRef]
118. Zhang, G.; Tan, F.; Wu, Y. Ship Motion Attitude Prediction Based on an Adaptive Dynamic Particle Swarm Optimization Algorithm and Bidirectional LSTM Neural Network. *IEEE Access* **2020**, *8*, 90087–90098. [CrossRef]

119. Wang, J.; Kumbasar, T. Parameter optimization of interval Type-2 fuzzy neural networks based on PSO and BBBC methods. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 247–257. [CrossRef]
120. Abdolrasol, M.G.M.; Mohamed, R.; Hannan, M.A.; Al-Shetwi, A.Q.; Mansor, M.; Blaabjerg, F.G. Artificial Neural Network Based Particle Swarm Optimization for Microgrid Optimal Energy Scheduling. *IEEE Trans. Power Electron.* **2021**. [CrossRef]
121. Roy, P.; Mahapatra, G.S.; Dey, K.N. Forecasting of software reliability using neighborhood fuzzy particle swarm optimization based novel neural network. *IEEE/CAA J. Autom. Sin.* **2019**, *6*, 1365–1383. [CrossRef]
122. Lin, H.; Zhao, B.; Liu, D.; Alippi, C. Data-based fault tolerant control for affine nonlinear systems through particle swarm optimized neural networks. *IEEE/CAA J. Autom. Sin.* **2020**, *7*, 954–964. [CrossRef]
123. Hajihassani, M.; Jahed Armaghani, D.; Sohaei, H.; Tonnizam Mohamad, E.; Marto, A. Prediction of airblast-overpressure induced by blasting using a hybrid artificial neural network and particle swarm optimization. *Appl. Acoust.* **2014**, *80*, 57–67. [CrossRef]
124. Gaur, S.; Ch, S.; Graillot, D.; Chahar, B.R.; Kumar, D.N. Application of Artificial Neural Networks and Particle Swarm Optimization for the Management of Groundwater Resources. *Water Resour. Manag.* **2013**, *27*, 927–941. [CrossRef]
125. Chan, K.Y.; Dillon, T.; Chang, E.; Singh, J. Prediction of short-term traffic variables using intelligent swarm-based neural networks. *IEEE Trans. Control Syst. Technol.* **2013**, *21*, 263–274. [CrossRef]
126. Lin, C.J.; Chen, C.H.; Lin, C.T. A hybrid of cooperative particle swarm optimization and cultural algorithm for neural fuzzy networks and its prediction applications. *IEEE Trans. Syst. Man Cybern. Part C Appl. Rev.* **2009**, *39*, 55–68. [CrossRef]
127. Volkan Pehlivanoglu, Y. A new particle swarm optimization method enhanced with a periodic mutation strategy and neural networks. *IEEE Trans. Evol. Comput.* **2013**, *17*, 436–452. [CrossRef]
128. Kalani, H.; Sardarabadi, M.; Passandideh-Fard, M. Using artificial neural network models and particle swarm optimization for manner prediction of a photovoltaic thermal nanofluid based collector. *Appl. Therm. Eng.* **2017**, *113*, 1170–1177. [CrossRef]
129. Song, Y.; Chen, Z.; Yuan, Z. New chaotic PSO-based neural network predictive control for nonlinear process. *IEEE Trans. Neural Netw.* **2007**, *18*, 595–600. [CrossRef]
130. Chen, J.F.; Do, Q.H.; Hsieh, H.N. Training artificial neural networks by a hybrid PSO-CS Algorithm. *Algorithms* **2015**, *8*, 292–308. [CrossRef]
131. Chou, P.Y.; Tsai, J.T.; Chou, J.H. Modeling and Optimizing Tensile Strength and Yield Point on a Steel Bar Using an Artificial Neural Network with Taguchi Particle Swarm Optimizer. *IEEE Access* **2016**, *4*, 585–593. [CrossRef]
132. Chai, R.; Ling, S.H.; Hunter, G.P.; Tran, Y.; Nguyen, H.T. Brain-Computer Interface Classifier for Wheelchair Commands Using Neural Network with Fuzzy Particle Swarm Optimization. *IEEE J. Biomed. Health Inform.* **2014**, *18*, 1614–1624. [CrossRef]
133. Bangyal, W.H.; Ahmad, J.; Rauf, H.T.; Shakir, R. Evolving artificial neural networks using opposition based particle swarm optimization neural network for data classification. In *Proceedings of the 2018 International Conference on Innovation and Intelligence for Informatics, Computing, and Technologies, 3ICT 2018, Zallaq, Bahrain, 18–19 November 2018*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2018.
134. Darrel, W. A Genetic Algorithm Tutorial. *Stat. Comput.* **1994**, *4*, 65–85. [CrossRef]
135. Daraban, S.; Petreus, D.; Morel, C. A novel MPPT (maximum power point tracking) algorithm based on a modified genetic algorithm specialized on tracking the global maximum power point in photovoltaic systems affected by partial shading. *Energy* **2014**, *74*, 374–388. [CrossRef]
136. Irshad, M.; Khalid, S.; Hussain, M.Z.; Sarfraz, M. Outline capturing using rational functions with the help of genetic algorithm. *Appl. Math. Comput.* **2016**, *274*, 661–678. [CrossRef]
137. Gomes, H.M.; Awruch, A.M.; Lopes, P.A.M. Reliability based optimization of laminated composite structures using genetic algorithms and Artificial Neural Networks. *Struct. Saf.* **2011**, *33*, 186–195. [CrossRef]
138. Kim, H.J.; Jo, N.O.; Shin, K.S. Optimization of cluster-based evolutionary undersampling for the artificial neural networks in corporate bankruptcy prediction. *Expert Syst. Appl.* **2016**, *59*, 226–234. [CrossRef]
139. Baykasoğlu, A.; Baykasoğlu, C. Multiple objective crashworthiness optimization of circular tubes with functionally graded thickness via artificial neural networks and genetic algorithms. *Proc. Inst. Mech. Eng. Part C J. Mech. Eng. Sci.* **2017**, *231*, 2005–2016. [CrossRef]
140. Erzurum Cicek, Z.I.; Kamisli Ozturk, Z. Optimizing the artificial neural network parameters using a biased random key genetic algorithm for time series forecasting. *Appl. Soft Comput.* **2021**, *102*, 107091. [CrossRef]
141. Dharmistha, M.; Vishwakarma, D. Genetic Algorithm based Weights Optimization of Artificial Neural Network. *Int. J. Adv. Res. Electr. Electron. Instrum. Energy* **2013**, *1*, 206–211.
142. Zhang, T.; Wang, J.; Liu, Q.; Zhou, J.; Dai, J.; Han, X.; Zhou, Y.; Xu, K. Efficient spectrum prediction and inverse design for plasmonic waveguide systems based on artificial neural networks. *Photonics Res.* **2019**, *7*, 368–380. [CrossRef]
143. Chidambaram, B.; Ravichandran, M.; Seshadri, A.; Muniyandi, V. Computational Heat Transfer Analysis and Genetic Algorithm-Artificial Neural Network-Genetic Algorithm-Based Multiobjective Optimization of Rectangular Perforated Plate Fins. *IEEE Trans. Components Packag. Manuf. Technol.* **2017**, *7*, 208–216. [CrossRef]
144. Efosa, C.I.; Kingsley, C.U. Architecture Optimization Model for the Deep Neural Network for Binary Classification Problems. *i-Manags. J. Softw. Eng.* **2019**, *14*, 18. [CrossRef]

145. Sales de Menezes, L.H.; Carneiro, L.L.; Maria de Carvalho Tavares, I.; Santos, P.H.; Pereira das Chagas, T.; Mendes, A.A.; Paranhos da Silva, E.G.; Franco, M.; Rangel de Oliveira, J. Artificial neural network hybridized with a genetic algorithm for optimization of lipase production from Penicillium roqueforti ATCC 10110 in solid-state fermentation. *Biocatal. Agric. Biotechnol.* **2021**, *31*, 101885. [CrossRef]
146. Abdullah, S.; Pradhan, R.C.; Pradhan, D.; Mishra, S. Modeling and optimization of pectinase-assisted low-temperature extraction of cashew apple juice using artificial neural network coupled with genetic algorithm. *Food Chem.* **2021**, *339*, 127862. [CrossRef] [PubMed]
147. Rashidi, M.M.; Bég, O.A.; Parsa, A.B.; Nazari, F. Analysis and optimization of a transcritical power cycle with regenerator using artificial neural networks and genetic algorithms. *Proc. Inst. Mech. Eng. Part A J. Power Energy* **2011**, *225*, 701–717. [CrossRef]
148. Safikhani, H.; Abbassi, A.; Khalkhali, A.; Kalteh, M. Multi-objective optimization of nanofluid flow in flat tubes using CFD, Artificial Neural Networks and genetic algorithms. *Adv. Powder Technol.* **2014**, *25*, 1608–1617. [CrossRef]
149. Li, Y.; Wang, Y.; Li, Y.; Zhou, R.; Lin, Z. An Artificial Neural Network Assisted Optimization System for Analog Design Space Exploration. *IEEE Trans. Comput. Des. Integr. Circuits Syst.* **2020**, *39*, 2640–2653. [CrossRef]
150. Qu, Z.; Yuan, S.; Chi, R.; Chang, L.; Zhao, L. Genetic optimization method of pantograph and catenary comprehensive monitor status prediction model based on adadelta deep neural network. *IEEE Access* **2019**, *7*, 23210–23221. [CrossRef]
151. Jiang, Q.; Huang, R.; Huang, Y.; Chen, S.; He, Y.; Lan, L.; Liu, C. Application of BP neural network based on genetic algorithm optimization in evaluation of power grid investment risk. *IEEE Access* **2019**, *7*, 154827–154835. [CrossRef]
152. Abeyrathna, K.D.; Jeenanunta, C. Hybrid particle swarm optimization with genetic algorithm to train artificial neural networks for short-term load forecasting. *Int. J. Swarm Intell. Res.* **2019**, *10*, 1–14. [CrossRef]
153. Barzegar, R.; Asghari Moghaddam, A. Combining the advantages of neural networks using the concept of committee machine in the groundwater salinity prediction. *Model. Earth Syst. Environ.* **2016**, *2*, 26. [CrossRef]
154. Kayabasi, A. An Application of ANN Trained by ABC Algorithm for Classification of Wheat Grains. *Int. J. Intell. Syst. Appl. Eng.* **2018**, *6*, 85–91. [CrossRef]
155. Awan, S.M.; Aslam, M.; Khan, Z.A.; Saeed, H. An efficient model based on artificial bee colony optimization algorithm with Neural Networks for electric load forecasting. *Neural Comput. Appl.* **2014**, *25*, 1967–1978. [CrossRef]
156. Hajimirzaei, B.; Navimipour, N.J. Intrusion detection for cloud computing using neural networks and artificial bee colony optimization algorithm. *ICT Express* **2019**, *5*, 56–59. [CrossRef]
157. Badem, H.; Basturk, A.; Caliskan, A.; Yuksel, M.E. A new efficient training strategy for deep neural networks by hybridization of artificial bee colony and limited–memory BFGS optimization algorithms. *Neurocomputing* **2017**, *266*, 506–526. [CrossRef]
158. Zhuo-Ming, C.; Yun-Xia, W.; Wei-Xin, L.; Zhen, X.; Han-Lin-Wei, X. *Artificial Bee Colony Algorithm for Modular Neural Network*; Springer: Berlin/Heidelberg, Germany, 2013; pp. 350–356.
159. Ding, S.; Li, H.; Su, C.; Yu, J.; Jin, F. Evolutionary artificial neural networks: A review. *Artif. Intell. Rev.* **2013**, *39*, 251–260. [CrossRef]
160. Kılıç, F.; Yılmaz, İ.H.; Kaya, Ö. Adaptive Co-Optimization of Artificial Neural Networks using Evolutionary Algorithm for Global Radiation Forecasting. *Renew. Energy* **2021**. [CrossRef]
161. Benmessahel, I.; Xie, K.; Chellal, M. A new evolutionary neural networks based on intrusion detection systems using multiverse optimization. *Appl. Intell.* **2018**, *48*, 2315–2327. [CrossRef]
162. Chai, Z.; Yang, X.; Liu, Z.; Lei, Y.; Zheng, W.; Ji, M.; Zhao, J. Correlation Analysis-Based Neural Network Self-Organizing Genetic Evolutionary Algorithm. *IEEE Access* **2019**, *7*, 135099–135117. [CrossRef]
163. Nassif, N. Modeling and optimization of HVAC systems using artificial neural network and genetic algorithm. *Build. Simul.* **2014**, *7*, 237–245. [CrossRef]
164. Goudos, S.K.; Tsoulos, G.V.; Athanasiadou, G.; Batistatos, M.C.; Zarbouti, D.; Psannis, K.E. Artificial Neural Network Optimal Modeling and Optimization of UAV Measurements for Mobile Communications Using the L-SHADE Algorithm. *IEEE Trans. Antennas Propag.* **2019**, *67*, 4022–4031. [CrossRef]
165. Yu, J.J.Q.; Lam, A.Y.S.; Li, V.O.K. Evolutionary artificial neural network based on Chemical Reaction Optimization. In Proceedings of the 2011 IEEE Congress of Evolutionary Computation, CEC 2011, New Orleans, LA, USA, 5–8 June 2011; pp. 2083–2090.
166. Pakdaman, M.; Ahmadian, A.; Effati, S.; Salahshour, S.; Baleanu, D. Solving differential equations of fractional order using an optimization technique based on training artificial neural network. *Appl. Math. Comput.* **2017**, *293*, 81–95. [CrossRef]
167. Hannan, M.A.; Lipu, M.S.H.; Hussain, A.; Saad, M.H.; Ayob, A. Neural network approach for estimating state of charge of lithium-ion battery using backtracking search algorithm. *IEEE Access* **2018**, *6*, 10069–10079. [CrossRef]
168. Wang, B.; Wang, L.; Yin, Y.; Xu, Y.; Zhao, W.; Tang, Y. An Improved Neural Network with Random Weights Using Backtracking Search Algorithm. *Neural Process. Lett.* **2016**, *44*, 37–52. [CrossRef]
169. Chen, D.; Lu, R.; Zou, F.; Li, S.; Wang, P. A learning and niching based backtracking search optimisation algorithm and its applications in global optimisation and ANN training. *Neurocomputing* **2017**, *266*, 579–594. [CrossRef]
170. Wu, S.; Wang, Z.; Ling, D. Echo state network prediction based on backtracking search optimization algorithm. In *Proceedings of the 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference, ITNEC 2019, Chengdu, China, 15–17 March 2019*; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2019; pp. 661–664.

171. Hannan, M.A.; Mohamed, R.; Abdolrasol, M.G.M.; Al-Shetwi, A.Q.; Ker, P.J.; Begum, R.A.; Muttaqi, K.M. ANN based binary backtracking search algorithm for virtual power plant scheduling and cost-effective evaluation. In Proceedings of the 2021 IEEE Texas Power and Energy Conference, College Station, TX, USA, 2–5 February 2021.
172. Meng, A.; Ge, J.; Yin, H.; Chen, S. Wind speed forecasting based on wavelet packet decomposition and artificial neural networks trained by crisscross optimization algorithm. *Energy Convers. Manag.* **2016**, *114*, 75–88. [CrossRef]
173. Lehký, D.; Slowik, O.; Novák, D. Reliability-based design: Artificial neural networks and double-loop reliability-based optimization approaches. *Adv. Eng. Softw.* **2018**, *117*, 123–135. [CrossRef]
174. Uzlu, E.; Kankal, M.; Akpinar, A.; Dede, T. Estimates of energy consumption in Turkey using neural networks with the teaching-learning-based optimization algorithm. *Energy* **2014**, *75*, 295–303. [CrossRef]
175. Shi, H.; Li, W. Artificial neural networks with ant colony optimization for assessing performance of residential buildings. In Proceedings of the FBIE 2009—2009 International Conference on Future BioMedical Information Engineering, Sanya, China, 13–14 December 2009; pp. 379–382.
176. Pereira, L.A.M.; Rodrigues, D.; Ribeiro, P.B.; Papa, J.P.; Weber, S.A.T. Social-spider optimization-based artificial neural networks training and its applications for Parkinson's Disease identification. In Proceedings of the IEEE Symposium on Computer-Based Medical Systems, New York, NY, USA, 27–29 May 2014; Institute of Electrical and Electronics Engineers Inc.: Piscataway, NJ, USA, 2014; pp. 14–17.
177. Liu, X.F.; Zhan, Z.H.; Gu, T.L.; Kwong, S.; Lu, Z.; Duh, H.B.L.; Zhang, J. Neural Network-Based Information Transfer for Dynamic Optimization. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 1557–1570. [CrossRef] [PubMed]
178. Kouhalvandi, L.; Ceylan, O.; Ozoguz, S. Automated Deep Neural Learning-Based Optimization for High Performance High Power Amplifier Designs. *IEEE Trans. Circuits Syst. I Regul. Pap.* **2020**, *67*, 4420–4433. [CrossRef]
179. Heravi, A.R.; Abed Hodtani, G. A new correntropy-based conjugate gradient backpropagation algorithm for improving training in neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2018**, *29*, 6252–6263. [CrossRef] [PubMed]
180. Yun, S.; Kang, J.M.; Kim, I.M.; Ha, J. Deep Artificial Noise: Deep Learning-Based Precoding Optimization for Artificial Noise Scheme. *IEEE Trans. Veh. Technol.* **2020**, *69*, 3465–3469. [CrossRef]
181. Su, H.; Qi, W.; Yang, C.; Aliverti, A.; Ferrigno, G.; De Momi, E. Deep neural network approach in human-like redundancy optimization for anthropomorphic manipulators. *IEEE Access* **2019**, *7*, 124207–124216. [CrossRef]
182. Ma, Y.; Han, R.; Wang, W. Prediction-Based Portfolio Optimization Models Using Deep Neural Networks. *IEEE Access* **2020**, *8*, 115393–115405. [CrossRef]
183. Pourdaryaei, A.; Mokhlis, H.; Illias, H.A.; Kaboli, S.H.A.; Ahmad, S.; Ang, S.P. Hybrid ANN and artificial cooperative search algorithm to forecast short-term electricity price in de-regulated electricity market. *IEEE Access* **2019**, *7*, 125369–125386. [CrossRef]
184. Yeh, W.C. New parameter-free simplified swarm optimization for artificial neural network training and its application in the prediction of time series. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 661–665. [CrossRef]
185. Kumaran, J.; Ravi, G. Long-term sector-wise electrical energy forecasting using artificial neural network and biogeography-based optimization. *Electr. Power Components Syst.* **2015**, *43*, 1225–1235. [CrossRef]
186. Yang, T.; Asanjan, A.A.; Faridzad, M.; Hayatbini, N.; Gao, X.; Sorooshian, S. An enhanced artificial neural network with a shuffled complex evolutionary global optimization with principal component analysis. *Inf. Sci.* **2017**, *418–419*, 302–316. [CrossRef]
187. Lu, T.C.; Yu, G.R.; Juang, J.C. Quantum-based algorithm for optimizing artificial neural networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2013**, *24*, 1266–1278. [CrossRef]
188. Yu, Y.; Liu, F. Effective Neural Network Training with a New Weighting Mechanism-Based Optimization Algorithm. *IEEE Access* **2019**, *7*, 72403–72410. [CrossRef]
189. Sun, W.Z.; Wang, J.S. Elman Neural Network Soft-Sensor Model of Conversion Velocity in Polymerization Process Optimized by Chaos Whale Optimization Algorithm. *IEEE Access* **2017**, *5*, 13062–13076. [CrossRef]
190. Aljarah, I.; Faris, H.; Mirjalili, S. Optimizing connection weights in neural networks using the whale optimization algorithm. *Soft Comput.* **2018**, *22*. [CrossRef]
191. Kumar, M.; Mishra, S.K.; Sahu, S.S. Cat Swarm Optimization Based Functional Link Artificial Neural Network Filter for Gaussian Noise Removal from Computed Tomography Images. *Appl. Comput. Intell. Soft Comput.* **2016**, *2016*, 1–6. [CrossRef]
192. Yusiong, J.P.T. Optimizing Artificial Neural Networks using Cat Swarm Optimization Algorithm. *Int. J. Intell. Syst. Appl.* **2012**, *5*, 69–80. [CrossRef]
193. Le, P.N.; Kang, H.J. Robot Manipulator Calibration Using a Model Based Identification Technique and a Neural Network with the Teaching Learning-Based Optimization. *IEEE Access* **2020**, *8*, 105447–105454. [CrossRef]
194. Manngård, M.; Kronqvist, J.; Böling, J.M. Structural learning in artificial neural networks using sparse optimization. *Neurocomputing* **2018**, *272*, 660–667. [CrossRef]
195. Ma, J.-w. *Optimization of Feed-Forward Neural Networks based on Artificial Fish-Swarm Algorithm*. 2004. Available online: https://www.semanticscholar.org/paper/Optimization-of-feed-forward-neural-networks-based-Jian-wei/663ea00fe44a17a89da2838c774ebe665dedcabf (accessed on 15 March 2021).
196. Kim, T.; Lee, J.; Choe, Y. Bayesian optimization-based global optimal rank selection for compression of convolutional neural networks. *IEEE Access* **2020**, *8*, 17605–17618. [CrossRef]
197. Petro, B.; Kasabov, N.; Kiss, R.M. Selection and Optimization of Temporal Spike Encoding Methods for Spiking Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 358–370. [CrossRef] [PubMed]

198. Gulcu, A.; Kus, Z. Hyper-Parameter Selection in Convolutional Neural Networks Using Microcanonical Optimization Algorithm. *IEEE Access* **2020**, *8*, 52528–52540. [CrossRef]
199. Wang, H.; Luo, Y.; An, W.; Sun, Q.; Xu, J.; Zhang, L. PID Controller-Based Stochastic Optimization Acceleration for Deep Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2020**, *31*, 5079–5091. [CrossRef] [PubMed]
200. Zheng, Y.J.; Ling, H.F.; Chen, S.Y.; Xue, J.Y. A Hybrid Neuro-Fuzzy Network Based on Differential Biogeography-Based Optimization for Online Population Classification in Earthquakes. *IEEE Trans. Fuzzy Syst.* **2015**, *23*, 1070–1083. [CrossRef]
201. Sheng, W.; Shan, P.; Mao, J.; Zheng, Y.; Chen, S.; Wang, Z. An Adaptive Memetic Algorithm with Rank-Based Mutation for Artificial Neural Network Architecture Optimization. *IEEE Access* **2017**, *5*, 18895–18908. [CrossRef]
202. Wu, L.; He, D.; Ai, B.; Wang, J.; Qi, H.; Guan, K.; Zhong, Z. Artificial Neural Network Based Path Loss Prediction for Wireless Communication Network. *IEEE Access* **2020**, *8*, 199523–199538. [CrossRef]
203. Naserbegi, A.; Aghaie, M.; Mahmoudi, S.M. PWR core pattern optimization using grey wolf algorithm based on artificial neural network. *Prog. Nucl. Energy* **2020**, *129*, 103505. [CrossRef]
204. Arunchai, T.; Sonthipermpoon, K.; Apichayakul, P.; Tamee, K. Resistance Spot Welding Optimization Based on Artificial Neural Network. *Int. J. Manuf. Eng.* **2014**, *2014*. [CrossRef]
205. Abdolrasol, M.G.M.; Hannan, M.A.; Hussain, S.M.S.; Ustun, T.S.; Sarker, M.R.; Ker, P.J. Energy Management Scheduling for Microgrids in the Virtual Power Plant System Using Artificial Neural Networks. *Energies* **2021**, *14*, 6507. [CrossRef]
206. Hannan, M.A.; Abdolrasol, M.G.M.; Faisal, M.; Ker, P.J.; Begum, R.A.; Hussain, A. Binary Particle Swarm Optimization for Scheduling MG Integrated Virtual Power Plant Toward Energy Saving. *IEEE Access* **2019**, *7*, 107937–107951. [CrossRef]
207. Ahmed, M.S.; Mohamed, A.; Khatib, T.; Shareef, H.; Homod, R.Z.; Ali, J.A. Real time optimal schedule controller for home energy management system using new binary backtracking search algorithm. *Energy Build.* **2017**, *138*, 215–227. [CrossRef]
208. Abdolrasol, M.G.M.; Hannan, M.A.; Mohamed, A.; Amiruldin, U.A.U.; Abidin, I.B.Z.; Uddin, M.N. An Optimal Scheduling Controller for Virtual Power Plant and Microgrid Integration Using the Binary Backtracking Search Algorithm. *IEEE Trans. Ind. Appl.* **2018**, *54*, 2834–2844. [CrossRef]
209. Roslan, M.F.; Hannan, M.A.; Jern Ker, P.; Begum, R.A.; Indra Mahlia, T.M.; Dong, Z.Y. Scheduling controller for microgrids energy management system using optimization algorithm in achieving cost saving and emission reduction. *Appl. Energy* **2021**, *292*, 116883. [CrossRef]
210. Hannan, M.A.; Begum, R.A.; Abdolrasol, M.G.; Hossain Lipu, M.S.; Mohamed, A.; Rashid, M.M. Review of baseline studies on energy policies and indicators in Malaysia for future sustainable energy development. *Renew. Sustain. Energy Rev.* **2018**, *94*, 551–564. [CrossRef]
211. Safari, A.; Babaei, F.; Farrokhifar, M. A load frequency control using a PSO-based ANN for micro-grids in the presence of electric vehicles. *Int. J. Ambient. Energy* **2021**, *42*, 688–700. [CrossRef]
212. Shabbir, J.; Anwer, T. Artificial Intelligence and its Role in Near Future. *arXiv* **2018**, arXiv:1804.01396.
213. Wang, M.; Chen, H. Chaotic multi-swarm whale optimizer boosted support vector machine for medical diagnosis. *Appl. Soft Comput.* **2020**, *88*, 105946. [CrossRef]
214. Shan, W.; Qiao, Z.; Heidari, A.A.; Chen, H.; Turabieh, H.; Teng, Y. Double adaptive weights for stabilization of moth flame optimizer: Balance analysis, engineering cases, and medical diagnosis. *Knowl.-Based Syst.* **2021**, *214*, 106728. [CrossRef]
215. Tu, J.; Chen, H.; Liu, J.; Heidari, A.A.; Zhang, X.; Wang, M.; Ruby, R.; Pham, Q.V. Evolutionary biogeography-based whale optimization methods with communication structure: Towards measuring the balance. *Knowl.-Based Syst.* **2021**, *212*, 106642. [CrossRef]