sensors

Article

Intrusion Detection of UAVs Based on the Deep Belief Network Optimized by PSO

Xiaopeng Tan, Shaojing Su, Zhen Zuo *, Xiaojun Guo and Xiaoyong Sun

College of Artificial Intelligence, National University of Defense Technology, Changsha 410073, China; tanxiaopeng14@nudt.edu.cn (X.T.); susj-5@163.com (S.S.); jeanakin@nudt.edu.cn (X.G.); sxy15084973192@163.com (X.S.)

* Correspondence: z.zuo@nudt.edu.cn; Tel.: +86-130-7733-7333

Received: 16 October 2019; Accepted: 12 December 2019; Published: 14 December 2019

Abstract: With the rapid development of information technology, the problem of the network security of unmanned aerial vehicles (UAVs) has become increasingly prominent. In order to solve the intrusion detection problem of massive, high-dimensional, and nonlinear data, this paper proposes an intrusion detection method based on the deep belief network (DBN) optimized by particle swarm optimization (PSO). First, a classification model based on the DBN is constructed, and the PSO algorithm is then used to optimize the number of hidden layer nodes of the DBN, to obtain the optimal DBN structure. The simulations are conducted on a benchmark intrusion dataset, and the results show that the accuracy of the DBN-PSO algorithm reaches 92.44%, which is higher than those of the support vector machine (SVM), artificial neural network (ANN), deep neural network (DNN), and Adaboost. It can be seen from comparative experiments that the optimization effect of PSO is better than those of the genetic algorithm, simulated annealing algorithm, and Bayesian optimization algorithm. The method of PSO-DBN provides an effective solution to the problem of intrusion detection of UAV networks.

Keywords: deep belief network; PSO; parameter optimization; intrusion detection

1. Introduction

In recent years, with the rapid development of cloud computing and artificial intelligence technology, the Internet of Things technology has also ushered in vigorous development. Various intelligent devices can receive a large amount of information through data exchange and interconnection. The popularity of the Internet of Things technology and the intelligence of devices have brought great convenience to people, but the use of new technologies and smart devices has also brought new security and privacy risks [1–3]. As the Internet of Things nodes collect and store large amounts of user privacy data, Internet of Things systems have become important targets for cyber attackers. In this case, protecting personal privacy and data security is very important [4–6].

With the progress of technology and the continuous reduction in manufacturing costs, the Internet of Things system composed of unmanned aerial vehicles (UAVs) has entered industrial production and people's daily life from the military field. Nowadays, UAVs have been widely used in film and television shooting, agricultural monitoring, meteorological monitoring, forest fire detection, emergency rescue, and other fields. However, while UAVs bring various conveniences to our production and life, the network security problems they face have been gradually exposed [7,8].

When multiple UAVs cooperate to perform tasks, it is necessary to build information connection channels between them to form a mobile self-organizing network of UAVs. The UAVs in the network realize the real-time sharing of information through this mobile network, which no longer needs to be forwarded by a ground station, and this effectively improves the survivability and combat ability of the UAV group. As a UAV network is a subclass of the mobile ad hoc network, the common attacks in the mobile ad hoc network will also threaten the UAV network.

Due to the diversity of network access methods and the openness of networks, UAV networks are facing inevitable security threats [9–12]. The defense function of traditional network security technology is mostly passive, and it is difficult to resist network attacks with changeable technology. As an active defensive network security technology, intrusion detection technology makes up for the shortcomings of traditional security technology [13–16].

While intrusion detection systems have attracted much attention from users, there are still some problems to be improved in their practical application. Traditional intrusion detection systems generally suffer from an insufficient performance and inefficiency, especially in modern computer networks with high bandwidth and large traffic. In the face of attacks, which are becoming more and more complex, automated, and distributed, traditional intrusion detection systems cannot meet the needs of current network security. In order to improve the detection efficiency and reduce the false alarm rate of intrusion detection systems, more and more researchers have introduced machine learning algorithms into the field of intrusion detection and have made good progress [17–21].

Shah et al. [22] investigated the performance of two open-source intrusion detection systems, Snort and Surcata. The results show that using an optimized support vector machine (SVM) and firefly algorithm can achieve the best detection effect. Kabir et al. [23] proposed a new method based on a least-squares support vector machine (LS-SVM) for an intrusion detection system. Wang et al. [24] proposed an intrusion detection framework based on the SVM, with feature augmentation. By transforming the logarithmic marginal density ratio to form original features, new and better transform features can be obtained, which greatly improves the detection ability of the model. Ahmed et al. [25] proposed a learning algorithm for an intrusion detection system based on a neural network (NN), which has a good performance in terms of its convergence speed and learning time. Hu et al. [26] proposed a distributed intrusion detection framework, in which a local parameterized detection model is constructed in each node using the online Adaboost algorithm. Ma et al. [27] proposed a novel approach called SCDNN, which combines spectral clustering (SC) and deep neural network (DNN) algorithms. The experimental results indicate that the SCDNN classifier performs better than the back-propagation neural network and support vector machine.

However, with the deepening of research, deep learning has gained wider application and a more outstanding performance in massive data analysis, which can be used to solve intrusion detection problems of massive, high-dimensional, and nonlinear data. By constructing a nonlinear network structure with multiple hidden layers, low-dimensional features, which are easier to classify in the data, can be obtained, and the accuracy of intrusion detection is improved [28–32]. Hinton et al. [33] proposed a deep learning method, called a deep belief network, which has attracted wide attention in academic circles. The deep belief network can transform high-dimensional and nonlinear data features into abstract features, which are more suitable for pattern classification, through layer-by-layer feature extraction. Qu et al. [34] proposed an intrusion detection model based on a deep belief network, which effectively improves the detection of abnormal data. Liang et al. [35] proposed an intrusion detection method based on a deep belief network and extreme learning machine, which improves the recognition rate of intrusion detection and the efficiency of the algorithm operation.

The number of nodes in the hidden layer of a deep belief network is not easy to determine. In this paper, the particle swarm optimization (PSO) algorithm is used to find the optimal number of hidden layer nodes. Common intelligent search algorithms include the genetic algorithm [36,37], ant colony algorithm [38,39], simulated annealing algorithm [40], and particle swarm optimization. The genetic algorithm cannot effectively converge in a limited time. The ant colony algorithm is slow in terms of its solving time and is prone to prematurity. The actual effect of the simulated annealing algorithm is greatly affected by the parameters, including the global optimization and calculation efficiency. The Bayesian optimization algorithm [41] is also commonly used for hyperparameter optimization. Its advantage is that it has fewer iterations, but it is easy to fall into local optimization.

The PSO algorithm is easy to understand, easy to implement, fast in convergence, and can obtain the global optimal solution. Therefore, the PSO algorithm is selected as the optimization algorithm. The PSO algorithm is a kind of evolution algorithm based on a population. Through individual cooperation and group sharing, particles find the optimal solution of individuals and the optimal solution of the whole community to complete the optimization. Aburomman et al. [42] proposed a novel ensemble construction method that uses PSO-generated weights to create an ensemble of classifiers, which has a better accuracy in intrusion detection. Bamakan et al. [43] proposed an attack detection method based on multi-criteria linear programming and the PSO algorithm to improve the accuracy of attack detection.

In this paper, the deep learning method of the deep belief network (DBN) and the parameter optimization method of the PSO are introduced into the field of intrusion detection, and an intrusion detection model based on the PSO-DBN is proposed. The model uses the PSO algorithm to optimize the number of nodes in the DBN hidden layer to obtain the optimal network structure. Then, each restricted Boltzmann machine (RBM) network is trained from the bottom to the top, and the low-dimensional representation of the original data is obtained in the unsupervised learning process, which significantly reduces the dimensionality of the data, retains key features of the data, and removes the redundancy features. Finally, the back-propagation (BP) algorithm is used to classify the low-dimensional representation and fine-tune the RBM network at the same time. The PSO-DBN method, proposed in this paper, is compared to the artificial neural network (ANN), SVM, Adaboost, and DNN methods using the KDD Cup 99 dataset [44]. The experimental results show that the optimization effect of PSO is better than those of the genetic algorithm, simulated annealing algorithm, and Bayesian optimization algorithm, and the PSO-DBN model is superior to other machine learning methods, which effectively improves the accuracy of intrusion detection.

The remaining sections of this paper are organized as follows. Section 2 describes the principle of the DBN. Section 3 describes the parameter optimization based on the PSO algorithm. Section 4 describes the intrusion detection based on the PSO-DBN. Section 5 describes the experimental results and discussion, including the dataset, evaluation indicators, results, and comparison. Finally, Section 6 summarizes the paper.

2. Principle of the DBN

The detection model based on the DBN method is shown in Figure 1. The input layer includes five types of network data, including the Normal, Probing, DoS, U2R, and R2L data. A DBN is a neural network model, composed of multiple RBMs. When applying a DBN network in intrusion detection, the network structure should be trained first, to determine the connection weight and neuron bias of the network. The DBN mainly includes pre-training and reverse fine-tuning in the process of training the model. First, each layer of the RBM network is trained independently and unsupervised in the pre-training process to ensure that as much feature information as possible is retained when the feature vectors are mapped to different feature spaces. Then, the BP network is set-up in the last layer of the DBN, and the output eigenvector of the RBM is received as its input eigenvector. Then, supervised training is conducted for the entity relationship classifier. Moreover, each layer of the RBM network can only ensure that the weights in its own layer are optimal for the feature vector mapping of that layer, not for the whole DBN. Therefore, it is necessary for the BP network to spread error information, from top to bottom, in each layer of the RBM and fine-tune the DBN network. The process of training the model of the RBM network can be regarded as the initialization of the weights of a deep BP network, which makes the DBN overcome the shortcomings of the BP network, which easily falls into the local optimum due to the random initialization of weights.

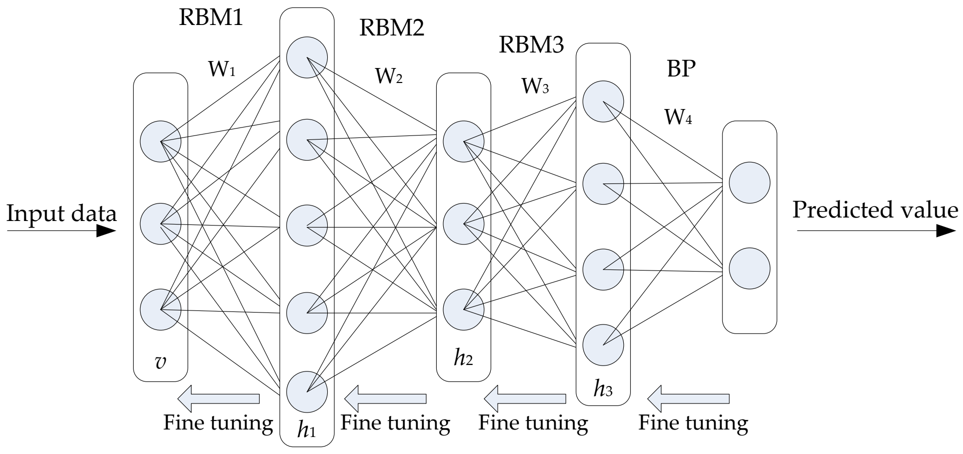


Figure 1. Detection model based on the deep belief network (DBN).

A single RBM is a neural network model consisting of a visible layer and a hidden layer [45]. Figure 1 shows a network structure consisting of 3 layers of the RBM, where v is the visible layer connecting the intrusion detection data, h is the hidden layer, which is used to extract the effective features of the input data, and W is the connection weight of the visible layer and the hidden layer. The neurons of the same layer in the network structure are not connected to each other, and the neurons of the adjacent layers are connected to each other by weights. The inactivated and activated states are represented by a binary, 0 and 1, for neurons in the network.

The RBM is an energy-based model [46], where v_i is used to represent the state of neuron i in the visible layer, with corresponding bias a_i, h_j is used to represent the state of neuron j in the hidden layer, with corresponding bias b_j, and the connection weight of neuron i and j is w_{ij}. The energy of the RBM can be expressed as

$$E(\boldsymbol{v},\boldsymbol{h}|\theta) = -\sum_{i=1}^{n} a_i v_i - \sum_{j=1}^{m} b_j h_j - \sum_{i=1}^{n}\sum_{j=1}^{m} v_i w_{ij} h_j. \tag{1}$$

In the equation, $\boldsymbol{\theta} = \left(w_{ij}, a_i, b_j\right)$ is the RBM parameter, and n and m are the number of neurons in the visible layer and hidden layer, respectively.

From the energy function, the joint probability distribution of $(\boldsymbol{v},\boldsymbol{h})$ can be obtained as follows:

$$p(\boldsymbol{v},\boldsymbol{h}|\theta) = \frac{1}{Z(\theta)} \exp(-E(\boldsymbol{v},\boldsymbol{h}|\theta)), \tag{2}$$

where $Z(\theta) = \sum\limits_{\boldsymbol{v}}\sum\limits_{\boldsymbol{h}} \exp(-E(\boldsymbol{v},\boldsymbol{h}|\theta))$ is the normalization factor.

For the training sample with the number of N, parameter θ is obtained by learning the maximum logarithmic likelihood function of the sample, which is

$$\theta^* = \underset{\theta}{\operatorname{argmax}} L(\theta) = \underset{\theta}{\operatorname{argmax}} \sum_{n=1}^{N} \log p(v^n|\theta), \tag{3}$$

where $p(\boldsymbol{v}|\theta) = \frac{1}{Z(\theta)} \sum\limits_{\boldsymbol{h}} \exp(-E(\boldsymbol{v},\boldsymbol{h}|\theta))$ is the likelihood function of v.

In the process of training, due to the complexity of calculating the normalization factor $Z(\theta)$, Gibbs and other sampling methods are generally used to approximate it [47]. Hinton proposed a fast learning algorithm using contrast divergence (CD) to train the network parameters, which improves the training efficiency and promotes the development of the RBM. The CD algorithm calculates the state of the neurons in the hidden layer by the vector value of the neurons in the visible layer, and then reconstructs the state of the neurons in the visible layer using the neurons in the hidden layer and calculates the state of the neurons in the hidden layer again using the reconstructed neurons in the visible layer, so that a new state of the neurons in the hidden layer can be obtained.

As the activation states of each neuron in the same layer of the RBM are independent of each other, the jth neuron in the hidden layer is calculated according to the state of the neurons in the visible layer, and the activation probability is as follows:

$$p\Big(h_j = 1\Big|\boldsymbol{v}, \boldsymbol{\theta}\Big) = \sigma\left(b_j + \sum_i v_i w_{ij}\right), \tag{4}$$

where $\sigma = \frac{1}{1+\exp(-x)}$ is the sigmoid activation function.

The ith neuron in the visible layer is reconstructed by the hidden layer, and the activation probability is as follows:

$$p(v_i = 1|\boldsymbol{h}, \boldsymbol{\theta}) = \sigma\left(a_i + \sum_j w_{ij} h_j\right). \tag{5}$$

Further, the updated equations of the RBM weights and bias parameters can be obtained as follows:

$$\begin{cases} w_{ij}^{k+1} = w_{ij}^{k} + \varepsilon\Big(\big\langle v_i h_j\big\rangle_{data} - \big\langle v_i h_j\big\rangle_{recon}\Big) \\ a_i^{k+1} = a_i^{k} + \varepsilon(\langle v_i\rangle_{data} - \langle v_i\rangle_{recon}) \\ b_j^{k+1} = b_j^{k} + \varepsilon\Big(\big\langle h_j\big\rangle_{data} - \big\langle h_j\big\rangle_{recon}\Big) \end{cases}. \tag{6}$$

Among them, $\langle\cdot\rangle_{data}$ is the distribution, defined by the model of the original intrusion detection data, $\langle\cdot\rangle_{recon}$ is the distribution defined by the reconstructed model, ε is the learning rate, k is the number of iterations of the CD algorithm, w_{ij}^{k+1} is the updated weight matrix, and a_i^{k+1} and b_j^{k+1} are the bias vectors, after the visible layer and the hidden layer have been updated.

3. Parameter Optimization Based on the PSO Algorithm

The PSO algorithm is inspired by the behavioral characteristics of bird predation and is used to solve the optimization problem. Each particle in the algorithm represents a potential solution to the problem, and each particle corresponds to a fitness value, which is determined by the fitness function. The velocity of the particle determines the direction and distance of the particle movement. The velocity is dynamically adjusted to the movement experience of the particle itself and other particles, thus realizing the optimization of the individual in the solvable space [48].

The PSO algorithm first initializes a group of particles in the solvable space, and in each iteration, the particles update themselves by tracking two extreme values. One is the optimal solution found by the particle itself, which is generally called the individual extreme value; the other is the current optimal solution, found by the whole population, which is generally called the global extreme value. The individual extreme value and global extreme value are updated continuously in the iteration process, and the final output global extreme value is the optimal solution, obtained by the algorithm [49].

It is supposed that in a D-dimensional search space, the population consisting of n particles is $X = (X_1, X_2, \cdots, X_n)$, where the ith particle represents a D-dimensional vector, $X_i = (x_{i1}, x_{i2}, \cdots, x_{it})^T$, which represents the position of the ith particle in the D-dimensional search space and also a potential solution to the problem. According to the fitness function, the fitness value corresponding to the position of each particle can be calculated. The fitness function defined in this paper is as follows:

$$F_{fitness} = 1 - correct/sum, \tag{7}$$

where correct represents the number of data that are correctly classified, and sum represents the total number of data.

Assuming that the velocity of the ith particle is $V_i = (V_{i1}, V_{i2}, \cdots, V_{iD})^T$, its individual extreme value is $P_i = (P_{i1}, P_{i2}, \cdots, P_{iD})^T$, and the global extreme value of the population is

$P_g = \left(P_{g1}, P_{g2}, \cdots, P_{gD}\right)^T$. In each iteration, the particle updates its velocity and position through the individual and global extreme value. The updating equation is as follows:

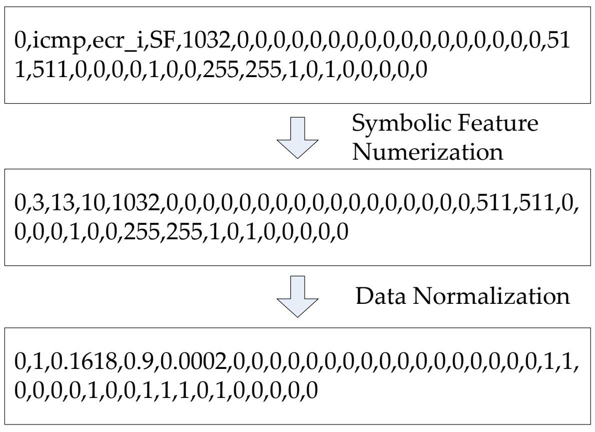
$$V_{id}^{k+1} = \omega V_{id}^{k} + c_1 r_1 \left(P_{id}^{k} - X_{id}^{k}\right) + c_2 r_2 \left(P_{gd}^{k} - X_{id}^{k}\right), \tag{8}$$

$$X_{id}^{k+1} = X_{id}^{k} + V_{id}^{k+1}, \tag{9}$$

where d represents the dth dimension of the variable, $d = 1, 2, \cdots, D$; i represents the ith particle, $i = 1, 2, \cdots, n$; k is the current number of iterations; V_{id} is the velocity of the dth dimension of the kth iteration of particle i; P_{id}^{k} is the coordinates of the individual optimal value, found by particle i in the dth dimension of the kth iteration; P_{gd}^{k} is the position of the global optimal solution, found by the entire particle swarm in the dth dimension of the kth iteration; c_1 and c_2 are learning factors, which are used to adjust the maximum step size for the optimal position of the individual and the optimal position of the group; r_1 and r_2 are random numbers distributed between [0, 1], called inertia factors, and the larger the value, the larger the range of the search; and ω is the inertia weight, which is a parameter introduced to balance the global search ability and local search ability. In order to prevent a blind search of particles, it is generally recommended to limit their position and velocity to a certain interval: $[-X_{\max}, X_{\max}]$ and $[-V_{\max}, V_{\max}]$.

4. Intrusion Detection Based on the PSO-DBN

UAV mobile ad hoc network intrusion detection can be regarded as a classification problem. First, the intrusion detection dataset is preprocessed. The preprocessing process is shown in Figure 2. Each connection record in the KDD Cup 99 dataset consists of 41 attribute features, including 3 symbolic features and 38 numeric features. In this paper, the attribute mapping method is used to transform symbolic features into numeric features. For example, there are three values for the attribute feature, 'protocol type,' in column 2: tcp, udp, and icmp, which can be processed according to tcp = 1, udp = 2, and icmp = 3. Similarly, the 70 symbol values of the attribute feature, 'service,' and the 11 symbol values of the 'flag' can establish the mapping relationship between the symbol value and the corresponding numerical value.



```
0,icmp,ecr_i,SF,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,51
1,511,0,0,0,0,1,0,0,255,255,1,0,1,0,0,0,0,0
```

```
0,3,13,10,1032,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,511,511,0,
0,0,0,1,0,0,255,255,1,0,1,0,0,0,0,0
```

```
0,1,0.1618,0.9,0.0002,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,
0,0,0,0,1,0,0,1,1,1,0,1,0,0,0,0,0
```

Figure 2. Preprocessing process of the connection record.

Then, the obtained data are normalized, and the data are normalized within a range of [0, 1], according to Equation (10), to ensure that the attributes are within the same order of magnitude.

$$\widetilde{x}(i) = \frac{x(i) - x_{\min}}{x_{\max} - x_{\min}}. \tag{10}$$

In the equation, $\widetilde{x}(i)$ is the normalized value of the input variable; $x(i)$ is the original value of the input variable; and $x_{\max}$ and $x_{\min}$ are the maximum and minimum values of the original data, respectively.

After preprocessing the intrusion detection data, the DBN network structure is initialized, and then the PSO algorithm is used to optimize the number of nodes in each layer of the DBN hidden layer to obtain an optimal network structure. Common hyperparameters in the DBN include the learning rate, the number of network layers, and the number of nodes in each layer. For the learning rate, it mainly controls the learning progress of the model. The larger the learning rate, the faster the learning speed. Generally speaking, users can intuitively set the optimal value of the learning rate by using experience values or other types of learning materials. For the number of network layers, the larger the number of layers, the more complicated the calculation. Compared to image processing, the dimension of the dataset used in this paper is not very high, and the selected network layers can meet the requirements of intrusion detection. In DBN, the selection of the number of nodes in each layer is very important. It not only has a great impact on the performance of the established DBN network model, but can also easily lead to "over fitting" in training if it is not properly selected. At present, the calculation formulas for determining the number of nodes in each layer proposed in most literatures are for the case of very large training samples, and the obtained results are not necessarily optimal. In fact, the number of nodes in each layer obtained by various calculation formulas greatly varies. In order to avoid "over fitting" during training as much as possible, and to ensure a high enough network performance and generalization ability, it is necessary to optimize the number of nodes in each layer.

The intrusion detection model based on the PSO-DBN is shown in Figure 3. The process of building the DBN includes pre-training and reverse fine-tuning. First, the forward propagation of the DBN is established through the training of the RBM model, and better initial model parameters are obtained. Then, the output error information of the training samples is calculated by the BP algorithm and propagated, from top to bottom, in each layer of the RBM, and the parameters of the DBN model are finely adjusted.

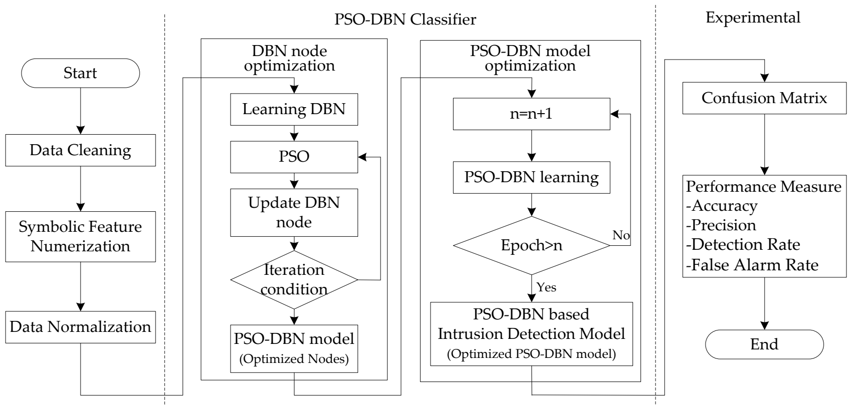


Figure 3. Intrusion detection model based on the particle swarm optimization (PSO)-DBN.

In the process of optimizing the number of nodes in the hidden layer, the prediction error of the classifier is selected as the fitness function of the model. Through the iteration condition of the PSO, the number of nodes in the hidden layer of the DBN is constantly updated, and the optimized PSO-DBN model is obtained. When the PSO-DBN model is completed, supervised learning is performed using BP to obtain an improved performance in updating the values of the weights of the nodes. Therefore, learning is performed after assigning a suitable number of epochs of the BP.

5. Experimental Results and Discussion

5.1. Dataset and Evaluation Indicators

This paper uses the KDD Cup 99 dataset as the training and testing set. The dataset is derived from the intrusion detection assessment program of the US Department of Defense Advanced Research Projects Agency (DARPA). It is hosted by the MIT Lincoln Laboratory. It is the benchmark dataset of network intrusion detection. It provides labeled training data and test data for researchers and is widely used for testing various intrusion detection methods. In this paper, 10% of the data was randomly selected from the "10% KDD Cup 99 training set," as the training data, and 10% of the data was randomly selected from the "KDD Cup 99 corrected labeled test data set," as the test data. The specific data distribution is shown in Table 1.

Table 1. Distribution of various types of data in the dataset.

Types\Distribution	Training Data	Testing Data	Label
Normal	9626	6047	1
Probing	452	412	2
DoS	39216	23037	3
U2R	9	18	4
R2L	97	1586	5

In the intrusion detection system, the accuracy (ACC), precision (PRE), detection rate (DR), and false alarm rate (FAR) are usually used as indicators to evaluate the system. ACC is the proportion of correctly classified samples and is defined as follows:

$$ACC = \frac{TP + TN}{TP + TN + FN + FP}, \tag{11}$$

where TP refers to the number of positive instances detected as positive instances, TN refers to the number of negative instances detected as negative instances, FP refers to the number of negative instances detected as positive instances, and FN refers to the number of positive instances detected as negative instances.

PRE is the proportion of samples that have an actual intrusion behavior in the samples detected as intrusion behavior, and it is defined as

$$PRE = \frac{TP}{TP + FP}. \tag{12}$$

DR is the proportion of the number of detected intrusion samples in the total number of intrusion samples, and it is defined as

$$DR = \frac{TP}{TP + FN}. \tag{13}$$

FAR is the proportion of the number of normal samples that are falsely reported as intrusions in the total number of normal samples, and it is defined as

$$FAR = \frac{FP}{TN + FP}. \tag{14}$$

The average reconstruction error (ARE) between the reconstructed data and the original data in each RBM network is also used as the criterion for performance evaluation, and it is calculated as follows:

$$ARE = \frac{\sum_{k=1}^{n} \sum_{i=1}^{41} \left(v_{ki} - v'_{ki}\right)^2}{n}, \tag{15}$$

where k is the sample number; v_{ki} is the original data of the kth sample; v'_{ki} is the kth sample data after reconstruction; and n is the number of samples.

5.2. Results and Comparison

The experimental environment of this paper is based on MATLAB R2013a and the data mining software, Weka. Compared to image processing, the dimension of the KDD Cup 99 dataset is not very high, so a DBN structure with four hidden layers can satisfy the experimental requirements. In order to verify the superiority of the PSO-DBN algorithm, proposed in this paper, a comparative experiment is carried out using the ANN, SVM, DNN, and Adaboost algorithms. The parameters of the PSO algorithm are shown in Table 2.

Table 2. Parameters of the PSO algorithm.

Parameter Name	Value	Description
inertia weight	1	the recommended global and local convergence speed
acceleration factor	1, 1	c1 + c2 $\leq$ 4
iterations	20	if more than 20, the experimental results have no significant changes
population	20	the number of particles involved in the search at the same time
particle dimension	4	The number of hidden layers is 4

Figure 4 shows the results of the PSO algorithm for optimizing the number of hidden layer nodes under different iterations. In this paper, the classification error is used as the fitness function. In the process of PSO optimization, when the classification error is at its minimum, the optimal result can be obtained. The experimental results show that the numbers of hidden layer nodes optimized by the PSO algorithm are 39, 29, 14, and 7, and the minimum error is 0.0923.

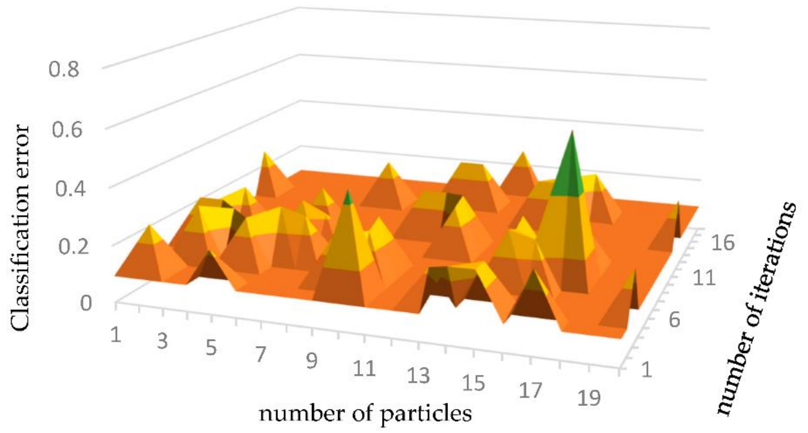


Figure 4. Optimization results of the PSO algorithm.

The average reconstruction error of the RBM is shown in Figure 5. As can be seen from the figure, the higher the number of iterations of the RBM, the smaller the average reconstruction error. When the number of iterations is more than 4, the average reconstruction error tends to be flat.

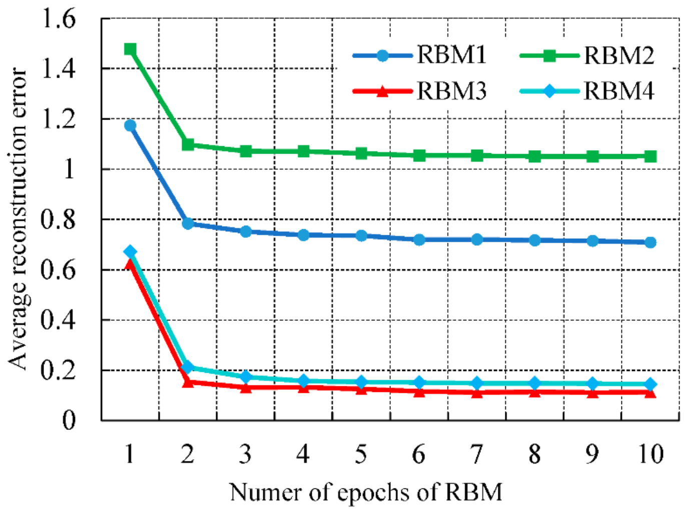


Figure 5. Average reconstruction error of the restricted Boltzmann machine (RBM).

The accuracy under different BP epochs in the PSO-DBN model is shown in Figure 6. It can be seen, from the figure, that when the number of epochs is small, the accuracy of the proposed model increases with the increase in the number of BP iterations. When the number of epochs is 37, the accuracy of the model reaches a maximum of 92.44%. After that, the accuracy of the model decreases with the increase in the number of epochs and tends to be flat. Therefore, when the number of BP epochs is 693, the effect of the model is optimal.

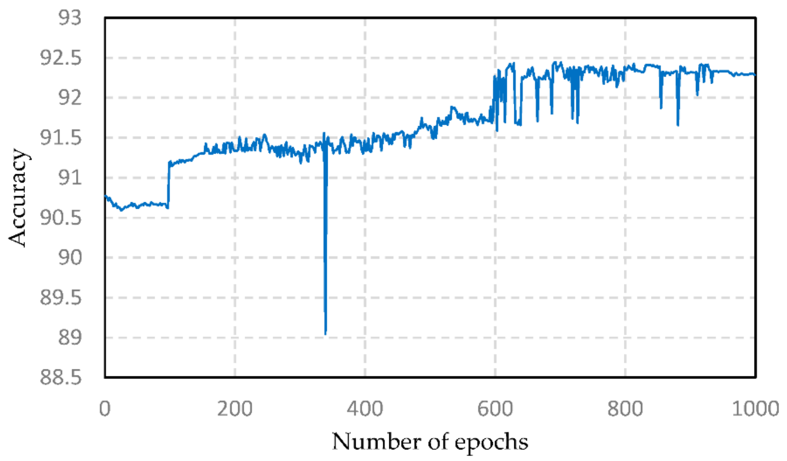


Figure 6. Accuracy under different BP epochs.

The PSO-DBN model, proposed in this paper, is compared to the ANN, SVM, Adaboost, and DNN classification methods. The ANN algorithm uses a three-layer structure, including an input layer, an intermediate layer, and an output layer. The other parameters are similar to those of the DBN. The type of SVM algorithm is set to C-support vector classification (C-SVC), and the kernel function is the radial basis function. The weight threshold of the Adaboost algorithm is set to 100, and the number of iterations is set to 10. The DNN algorithm uses an eight-layer structure, including an input layer, six intermediate layers, and an output layer. The performance comparison of ANN, SVM, Adaboost, DNN, and PSO-DBN is shown in Table 3. It can be seen from the table that in dealing with the classification problem of intrusion detection, PSO-DBN has the lowest false alarm rate, the highest accuracy rate, detection rate, and precision, and the best classification effect.

Table 3. Performance comparison of ANN, support vector machine (SVM), Adaboost, deep neural network (DNN), and PSO-DBN.

Result\Methods	ANN	SVM	Adaboost	DNN	PSO-DBN
ACC	90.79%	83.97%	90.00%	91.36%	92.44%
DR	89.68%	80.63%	89.00%	89.71%	91.20%
PRE	99.61%	99.54%	99.35%	99.60%	99.82%
FAR	1.46%	1.54%	2.40%	1.49%	0.68%

For the optimization problem in this paper, comparative experiments of the genetic algorithm (GA), simulated annealing algorithm (SA), and Bayesian optimization algorithm (BOA) are carried out. The genetic algorithm is a kind of computing model that simulates natural selection and the genetic mechanism. It is an algorithm that finds the optimal solution by simulating the natural evolution process. The simulated annealing algorithm imitates the behavior of the burning object during the annealing process to find the optimal solution. The actual effect of the algorithm is greatly affected by the parameters. The Bayesian optimization algorithm finds an acceptable extreme value by guessing what the black box function (objective function) looks like without knowing what the black box function looks like. The advantage of the Bayesian optimization algorithm is that it has fewer iterations, but it is easy to fall into the local optimum. The parameters of the GA and SA are shown in Tables 4 and 5, and the description of the BOA is shown in Table 6.

Table 4. Parameters of the GA.

Number of Generation	Population Size	Mutation Rate	Crossover Rate
20	10	0.05	0.8

Table 5. Parameters of the SA.

Decay Scale	Step Factor	Start Temperature	Final Temperature
0.85	0.2	8	3

Table 6. Description of the Bayesian optimization algorithm (BOA).

Prior Function	Acquisition Function	Number of Iteration	Objective Function
Gaussian process Regression	EI Function	30	Error in Classification

Table 7 shows the optimized number of nodes in each hidden layer. According to the optimized number of nodes in each hidden layer, the classification effect of the DBN network can be obtained. Table 8 shows the results of intrusion detection under various optimization algorithms. As can be seen from the table, the DBN network optimized by PSO has the lowest false alarm rate and the highest accuracy, detection rate, and precision. Therefore, the optimization effect of PSO is the best among the four optimization algorithms.

Table 7. Number of nodes in each hidden layer after optimization.

Hidden Layer Nodes\Methods	GA	SA	BOA	PSO
Layer 1	36	35	35	39
Layer 2	21	33	29	29
Layer 3	16	21	16	14
Layer 4	11	10	6	7

Table 8. Performance comparison of GA-DBN, SA-DBN, BOA-DBN, and PSO-DBN.

Result\Methods	GA-DBN	SA-DBN	BOA-DBN	PSO-DBN
ACC	91.66%	91.41%	91.30%	92.44%
DR	90.42%	90.22%	90.70%	91.20%
PRE	99.50%	99.50%	99.76%	99.82%
FAR	1.87%	1.89%	0.89%	0.68%

6. Conclusions

Intrusion detection for UAV networks is an important subject in the field of the security of UAV networks. The deep belief network optimized by PSO is a very effective method. Through the unsupervised learning of the RBM and the supervised learning of the BP, the DBN can effectively solve the intrusion detection problem of massive, high-dimensional, and nonlinear data. The DBN not only has a strong feature extraction ability for high-dimensional feature vectors, but it also has an efficient classification ability. Based on the DBN method, the PSO algorithm is used to optimize the number of hidden layer nodes of the DBN, to optimize its network structure. The experimental results show that the accuracy of the PSO-DBN algorithm, proposed in this paper, reaches 92.44%, which is higher than those achieved by the methods of ANN, SVM, Adaboost, and DNN. In addition, the optimization effect of PSO is better than those of GA, SA, and BOA. The PSO-DBN algorithm is very suitable for the tasks of information extraction in high-dimensional space, improves the intrusion recognition ability, and provides an effective solution to the problem of intrusion detection.

Author Contributions: Conceptualization, X.T. and S.S.; Formal analysis, S.S.; Investigation, Z.Z.; Methodology, Z.Z. and X.G.; Resources, X.G.; Software, X.T. and X.S.; Supervision, S.S. and Z.Z.; Validation, X.T. and X.S.; Writing—original draft, X.T., Z.Z. and X.G.; Writing—review and editing, S.S.

Funding: This work was supported by the Natural Science Foundation of Hunan Province, under Grant No. 2018JJ3607, and the National Natural Science Foundation of China, under Grant No. 51575517.

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

References

1. Fadi, A.; Zahmatkesh, H.; Shahroze, R. An overview of security and privacy in smart cities' IoT communications. *Trans. Emerg. Telecommun. Technol.* **2019**, e3677. [CrossRef]
2. Lin, J.C.W.; Wu, J.M.T.; Fournier-Viger, P.; Djenouri, Y.; Chen, C.H.; Zhang, Y. A Sanitization Approach to Secure Shared Data in an IoT Environment. *IEEE Access* **2019**, *7*, 25359–25368. [CrossRef]
3. Renuka, K.; Kumar, S.; Kumari, S.; Chen, C. Cryptanalysis and Improvement of a Privacy-Preserving Three-Factor Authentication Protocol for Wireless Sensor Networks. *Sensors* **2019**, *19*, 4625. [CrossRef]
4. Zhang, Y.; Li, P.; Wang, X. Intrusion Detection for IoT Based on Improved Genetic Algorithm and Deep Belief Network. *IEEE Access* **2019**, *7*, 31711–31722. [CrossRef]
5. Javaid, S.; Afzal, H.; Arif, F.; Iltaf, N.; Abbas, H.; Iqbal, W. CATSWoTS: Context Aware Trustworthy Social Web of Things System. *Sensors* **2019**, *19*, 3076. [CrossRef]
6. Ji, J.; Wu, G.; Shuai, J.; Zhang, Z.; Wang, Z.; Ren, Y. Heuristic Approaches for Enhancing the Privacy of the Leader in IoT Networks. *Sensors* **2019**, *19*, 3886. [CrossRef]
7. Sun, X.; Ng, D.; Ding, Z.; Xu, Y.; Zhong, Z. Physical Layer Security in UAV Systems: Challenges and Opportunities. *IEEE Wirel. Commun.* **2019**, *26*, 40–47. [CrossRef]
8. Lei, K.; Zhang, Q.; Lou, J.; Bai, B.; Xu, K. Securing ICN-Based UAV Ad Hoc Networks with Blockchain. *IEEE Commun. Mag.* **2019**, *57*, 26–32. [CrossRef]
9. Arshad, J.; Azad, M.A.; Abdellatif, M.M.; Rehman, M.H.U.; Salah, K. COLIDE: A collaborative intrusion detection framework for Internet of Things. *IET Netw.* **2019**, *8*, 3–14. [CrossRef]
10. Elmasry, W.; Akbulut, A.; Zaim, A.H. Empirical study on multiclass classification-based network intrusion detection. *Comput. Intell.* **2019**, *35*, 919–954. [CrossRef]
11. Qian, Y.-G.; Lu, H.-B.; Ji, S.-L.; Zhou, W.J.; Wu, S.-H.; Lei, J.-S.; Tao, X.-X. A Poisoning Attack on Intrusion Detection System Based on SVM. *Acta Electron. Sin.* **2019**, *47*, 59–65.

12. Vaseer, G.; Ghai, G.; Ghai, D. Novel Intrusion Detection and Prevention for Mobile Ad Hoc Networks: A Single- and Multiattack Case Study. *IEEE Consum. Electron. Mag.* **2019**, *8*, 35–39. [CrossRef]
13. Maza, S.; Touahria, M. Feature selection for intrusion detection using new multi-objective estimation of distribution algorithms. *Appl. Intell.* **2019**, *49*, 4237–4257. [CrossRef]
14. Siddique, K.; Akhtar, Z.; Khan, F.A.; Kim, Y. KDD Cup 99 Data Sets: A Perspective on the Role of Data Sets in Network Intrusion Detection Research. *Computer* **2019**, *52*, 41–51. [CrossRef]
15. Hu, X.; Li, T.; Wu, Z.; Xuan, G.; Wang, Z. Research and application of intelligent intrusion detection system with accuracy analysis methodology. *Infrared Phys. Technol.* **2018**, *88*, 245–253. [CrossRef]
16. Meng, W. Intrusion Detection in the Era of IoT: Building Trust via Traffic Filtering and Sampling. *Computer* **2018**, *51*, 36–43. [CrossRef]
17. Li, L.; Zhang, H.; Peng, H.; Yang, Y. Nearest neighbors based density peaks approach to intrusion detection. *Chaos Solitons Fractals* **2018**, *110*, 33–40. [CrossRef]
18. Belouch, M.; Hadaj, S.E.; Idhammad, M. Performance evaluation of intrusion detection based on machine learning using Apache Spark. *Proc. Comput. Sci.* **2018**, *127*, 1–6. [CrossRef]
19. Nisioti, A.; Mylonas, A.; Yoo, P.D.; Katos, V. From Intrusion Detection to Attacker Attribution: A Comprehensive Survey of Unsupervised Methods. *IEEE Commun. Surv. Tutor.* **2018**, *20*, 3369–3388. [CrossRef]
20. Rustam, Z.; Zahras, D. Comparison between Support Vector Machine and Fuzzy C-Means as Classifier for Intrusion Detection System. *J. Phys. Conf. Ser.* **2018**, *1028*, 012227. [CrossRef]
21. Wang, W.; Sheng, Y.; Wang, J.; Zeng, X.; Ye, X.; Huang, Y.; Zhu, M. HAST-IDS: Learning Hierarchical Spatial-Temporal Features using Deep Neural Networks to Improve Intrusion Detection. *IEEE Access* **2018**, *6*, 1792–1806. [CrossRef]
22. Shah, S.A.R.; Issac, B. Performance Comparison of Intrusion Detection Systems and Application of Machine Learning to Snort System. *Future Gener. Comput. Syst.* **2017**, *80*, 157–170. [CrossRef]
23. Kabir, E.; Hu, J.; Hua, W.; Zhuo, G. A novel statistical technique for intrusion detection systems. *Future Gener. Comput. Syst.* **2017**, *79*, 303–318. [CrossRef]
24. Wang, H.; Gu, J.; Wang, S. An effective intrusion detection framework based on SVM with feature augmentation. *Knowl.-Based Syst.* **2017**, *136*, 130–139. [CrossRef]
25. Ahmed, H.I.; Elfeshawy, N.A.; Elzoghdy, S.F.; El-Sayed, H.S.; Faragallah, O.S. A Neural Network-Based Learning Algorithm for Intrusion Detection Systems. *Wirel. Pers. Commun.* **2017**, *97*, 3097–3112. [CrossRef]
26. Hu, W.; Gao, J.; Wang, Y.; Wu, O.; Maybank, S. Online Adaboost-Based Parameterized Methods for Dynamic Distributed Network Intrusion Detection. *IEEE Trans. Cybern.* **2014**, *44*, 66–82.
27. Ma, T.; Wang, F.; Cheng, J.; Yu, Y.; Chen, X. A Hybrid Spectral Clustering and Deep Neural Network Ensemble Algorithm for Intrusion Detection in Sensor Networks. *Sensors* **2016**, *16*, 1701. [CrossRef]
28. Khan, M.A.; Karim, M.R.; Kim, Y. A Scalable and Hybrid Intrusion Detection System Based on the Convolutional-LSTM Network. *Symmetry* **2019**, *11*, 583. [CrossRef]
29. Yang, Y.; Zheng, K.; Wu, C.; Yang, Y. Improving the Classification Effectiveness of Intrusion Detection by Using Improved Conditional Variational AutoEncoder and Deep Neural Network. *Sensors* **2019**, *19*, 2528. [CrossRef]
30. Asghar, M.Z.; Abbas, M.; Zeeshan, K.; Kotilainen, P.; Hämäläinen, T. Assessment of Deep Learning Methodology for Self-Organizing 5G Networks. *Appl. Sci.* **2019**, *9*, 2975. [CrossRef]
31. Yin, C.L.; Zhu, Y.F.; Fei, J.L.; He, X.Z. A Deep Learning Approach for Intrusion Detection using Recurrent Neural Networks. *IEEE Access* **2017**, *5*, 21954–21961. [CrossRef]
32. Hwang, R.H.; Peng, M.C.; Nguyen, V.L.; Chang, Y.L. An LSTM-Based Deep Learning Approach for Classifying Malicious Traffic at the Packet Level. *Appl. Sci.* **2019**, *9*, 3414. [CrossRef]
33. Hinton, G.E.; Osindero, S.; Teh, Y.-W. A Fast Learning Algorithm for Deep Belief Nets. *Neural Comput.* **2006**, *18*, 1527–1554. [CrossRef]
34. Qu, F.; Zhang, J.; Shao, Z.; Qi, S. An Intrusion Detection Model Based on Deep Belief Network. In Proceedings of the International Conference on Network, Communication and Computing, Kunming, China, 8–10 December 2017; pp. 97–101.
35. Liang, D.; Pan, P. Research on Intrusion Detection System Based on DBN-ELM. In Proceedings of the 2019 International Conference on Communications, Information System and Computer Engineering (CISCE), Haikou, China, 5–7 July 2019; pp. 153–158.

36. Rani, S.; Suri, B.; Goyal, R. On the Effectiveness of Using Elitist Genetic Algorithm in Mutation Testing. *Symmetry* **2019**, *11*, 1145. [CrossRef]
37. Gong, D.W.; Sun, J.; Miao, Z. A Set-based Genetic Algorithm for Interval Many-objective Optimization Problems. *IEEE Trans. Evol. Comput.* **2018**, *22*, 47–60. [CrossRef]
38. Yuan, C.; Sun, X. Server Consolidation Based on Culture Multiple-Ant-Colony Algorithm in Cloud Computing. *Sensors* **2019**, *19*, 2724. [CrossRef]
39. Ke, Y.; Zhang, C.; Ning, J.; Liu, X. Ant-colony algorithm with a strengthened negative-feedback mechanism for constraint-satisfaction problems. *Inf. Sci.* **2017**, *406–407*, 29–41.
40. Assad, A.; Deep, K. A Hybrid Harmony search and Simulated Annealing algorithm for continuous optimization. *Inf. Sci.* **2018**, *450*, 246–266. [CrossRef]
41. Choi, E.; Chae, S.; Kim, J. Machine Learning-Based Fast Banknote Serial Number Recognition Using Knowledge Distillation and Bayesian Optimization. *Sensors* **2019**, *19*, 4218. [CrossRef]
42. Aburomman, A.A.; Reaz, M.B.I. A novel SVM-kNN-PSO ensemble method for intrusion detection system. *Appl. Soft. Comput.* **2016**, *38*, 360–372. [CrossRef]
43. Bamakan, S.M.H.; Amiri, B.; Mirzabagheri, M.; Yong, S. A New Intrusion Detection Approach Using PSO based Multiple Criteria Linear Programming. *Proc. Comput. Sci.* **2015**, *55*, 231–237.
44. KDD Cup 1999 Data. Available online: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (accessed on 20 September 2018).
45. Wang, X.Z.; Zhang, T.; Wang, R. Noniterative Deep Learning: Incorporating Restricted Boltzmann Machine Into Multilayer Random Weight Neural Networks. *IEEE Trans. Syst. Man Cybern.-Syst.* **2017**, *49*, 1299–1308. [CrossRef]
46. Chen, X.J.; Xiang, W.; Yuan, Z.Q.; Xiang, C.; Zhang, Y.W.; Cao, C.X. Spectral characteristics and species identification of rhododendrons using a discriminative restricted Boltzmann machine. *Spectr. Lett.* **2017**, *50*, 65–72.
47. Jiang, H.; Pan, Z.; Nan, L.; You, X.; Deng, T. Gibbs Sampling Based CRE Bias Optimization Algorithm for Ultra-Dense Networks. *IEEE Trans. Veh. Technol.* **2017**, *66*, 1334–1350. [CrossRef]
48. Ishaque, K.; Salam, Z.; Amjad, M.; Mekhilef, S. An Improved Particle Swarm Optimization (PSO)–Based MPPT for PV With Reduced Steady-State Oscillation. *IEEE Trans. Power Electron.* **2012**, *27*, 3627–3638. [CrossRef]
49. Luo, Y.; Yuan, X.; Liu, Y. An improved PSO algorithm for solving non-convex NLP/MINLP problems with equality constraints. *Comput. Chem. Eng.* **2007**, *31*, 153–162.