Hindawi
International Journal of Distributed Sensor Networks
Volume 2023, Article ID 9970205, 11 pages
https://doi.org/10.1155/2023/9970205

Research Article

Improved Population Intelligence Algorithm and BP Neural Network for Network Security Posture Prediction

Yueying Li and Feng Wu

College of Information Engineering, Xinyang Agriculture and Forestry University, Xinyang 464000, China

Correspondence should be addressed to Feng Wu; wufeng@xyafu.edu.cn

Received 6 January 2023; Revised 2 March 2023; Accepted 14 March 2023; Published 29 March 2023

Academic Editor: Janos Botzheim

To address the problems of low prediction accuracy and slow convergence of the network security posture prediction model, a population intelligence optimization algorithm is proposed to improve the network security posture prediction model of the BP neural network. First, the adaptive adjustment of the two parameters with the increase of iterations is achieved by improving the inertia weights and learning factors in the particle swarm optimization (PSO) algorithm so that the PSO has a large search range and high speed at the initial stage and a strong and stable convergence capability at the later stage. Secondly, to address the problem that PSO is prone to fall into a local optimum, the genetic operator is embedded into the operation process of the particle swarm algorithm, and the excellent global optimization performance of the genetic algorithm is used to open up the spatial vision of the particle population, revive the stagnant particles, accelerate the update amplitude of the algorithm, and achieve the purpose of improving the premature problem. Finally, the improved algorithm is combined with the BP neural network to optimize the BP neural network and applied to the network security posture assessment. The experimental comparison of different optimization algorithms is applied, and the results show that the network security posture prediction method of this model has the smallest error, the highest accuracy, and the fastest convergence, and can effectively predict future changes in network security posture.

1. Introduction

In recent years, with the continuous development of new technologies such as virtualization technology, cloud computing technology, and Internet of Things technology and the popularity and depth of their applications, the scale and complexity of networks have been increasing, and the current network environment has undergone unprecedented changes [1]. The architecture of such networks brings greater convenience to attackers, and the attack methods are more civilianized, collaborative, and can be carried out in stages, leading to a proliferation of network security attacks. Kaspersky Security Bulletin 2020 statistics show that a total of 666,809,967 attacks against online class resources were detected throughout the year, with 33,412,568 unique malicious samples identified and 549,301 pieces of ransomware detected. This means that 10.18% of the world's Internet users have experienced at least one malware-based attack [2]. The CNCERT Internet Security Threat Report released by the National Internet Emergency Response Center (CNCERT) of China in December 2020 showed that the number of terminals infected with Trojan horses or botnet malicious programs in the territory in this month alone was nearly 1.24 million [3–5]. The number of websites tampered with in the territory is 14,988, and the number of websites with backdoors implanted in the territory is 1,577. The number of counterfeit pages against domestic websites was 33,044. The national information security vulnerability sharing platform (CNVD) collected and collated 1,221 information system security vulnerabilities. Among them, there are 469 high-risk vulnerabilities and 872 vulnerabilities that can be exploited to carry out remote attacks. Security issues have become the primary factor restricting the current network development [6].

In order to cope with the increasingly complex and covert network security threats, organizations have deployed

a large number of network security devices and systems [7] and also use corresponding defense reinforcement means for important information systems. These measures and means ensure the safe and reliable operation of network systems to a certain extent, but they still have certain limitations [8]. First, each security tool and means has its own characteristics and purposes and is confined to its own management domain. Second, most of them belong to passive static protection and cannot adapt to the current network complex dynamic changes in security needs. Again, various security tools are fragmented and functionally dispersed, and they lack a unified and effective management and scheduling mechanism, forming a "security silo."

Network security posture prediction technology is one of the more effective active defense technologies to deal with network security threats [9]. Using this technology, network managers can get a comprehensive understanding of the current security risk status and security threats facing networks and information systems. It grasps the dynamic situation of network security as a whole and judges, evaluates, and predicts the network security situation and development trend so that corresponding remedial and preventive measures can be taken in a timely manner [10, 11]. In securing network security, network security posture prediction plays a crucial role and has a very important research value. Therefore, this technology has become a hot spot for research in the industry.

A network security posture prediction method based on the immune optimization principle is proposed, and this method can quickly identify the vulnerabilities of network security [12]. A detection model using a static analysis approach was used specifically to detect vulnerabilities in software [13]. This approach requires specialized tools to collect fingerprint information from the system and is susceptible to the limitations of the collection tools, which causes the system to have a reduced accuracy of vulnerability prediction. Some scholars use the PCA algorithm to identify abnormal traffic, which can effectively identify abnormal conditions [14]. However, due to the high complexity of the algorithm, it cannot meet the requirements of real-time monitoring of the network. An LSTM-based network anomaly detection model was proposed [15]. This model can extract effective feature information in the temporal characteristics and historical quantitative characteristics of network traffic, but the computational efficiency of this approach is relatively low.

Some scholars mapped the posture elements and posture values by combining the ideas of depth-separable convolution and convolutional decomposition techniques [16]. However, this model ignores the important difference between the original data attributes. A BP neural network-based posture prediction model was proposed to optimize the model parameters by introducing a simulated annealing algorithm into the crowd search algorithm [17]. Some scholars have also proposed a differential WGAN-based posture prediction method [18]. The method uses a generating adversarial network (GAN) to simulate the development process of posture, introduces Wasserstein distance as the loss function of GAN, and adds a differential term to improve the prediction accuracy of posture values. Literature [19] provided an overview of prediction and methods in web security, discussing and comparing attack prediction, intent recognition, intrusion prediction, and web security posture prediction. Literature [20] proposed a network security state prediction tool based on Semantic Web, which can be applied in the field where the system configuration is constantly changing (such as computer network). Some scholars have combined attack graphs with hidden Markov models (HMM) for network security assessment [21]. This method uses HMM to assess the network condition by combining the intrusion detection results and the attack graph generated by MulVAL. The method can effectively determine the attack intent and make the results more intuitive and comprehensive. Some scholars have proposed a network anomaly warning method based on generalized network temperature (GNT) and deep learning [22]. The method classifies the network congestion caused by DDoS. Then, Bi-GRU is used to predict the network traffic characteristics and discover the congestion state classes corresponding to each feature set by the stacking model. Some other scholars have proposed a fractional cumulative gray model based on the GA-PSO optimizer. This model determines the optimal fractional order by the fractional gray model and is suitable for cases with insufficient data and irregular sample sizes. However, the model mainly focuses on short-term prediction, and the medium- and long-term prediction performances need to be improved [23].

In recent years, neural networks have played an important role in situational assessment with their powerful nonlinear mapping capability, and some intelligent algorithms have been successively used in neural networks to improve the accuracy and efficiency of situational assessment [24–26]. In view of this, this paper proposes a network security posture prediction model with an improved population intelligence algorithm and BP neural network by using the improved population intelligence algorithm to optimize the parameters of the BP neural network. The method effectively solves the problems of low accuracy and efficiency of existing network security posture prediction, which is of great practical significance to current network development.

The innovation points of this paper are as follows:

(1) The adaptive adjustment of two parameters with increasing iterations is achieved by improving the inertia weights and learning factors in the particle swarm optimization (PSO) algorithm

(2) Embedding the genetic operator into the operation process of the particle swarm algorithm and using the excellent global optimization performance of the genetic algorithm to open up the spatial horizon of the particle population and accelerate the update amplitude of the algorithm

(3) The improved algorithm is combined with the BP neural network to optimize the BP neural network, thus significantly improving the model prediction accuracy and convergence speed

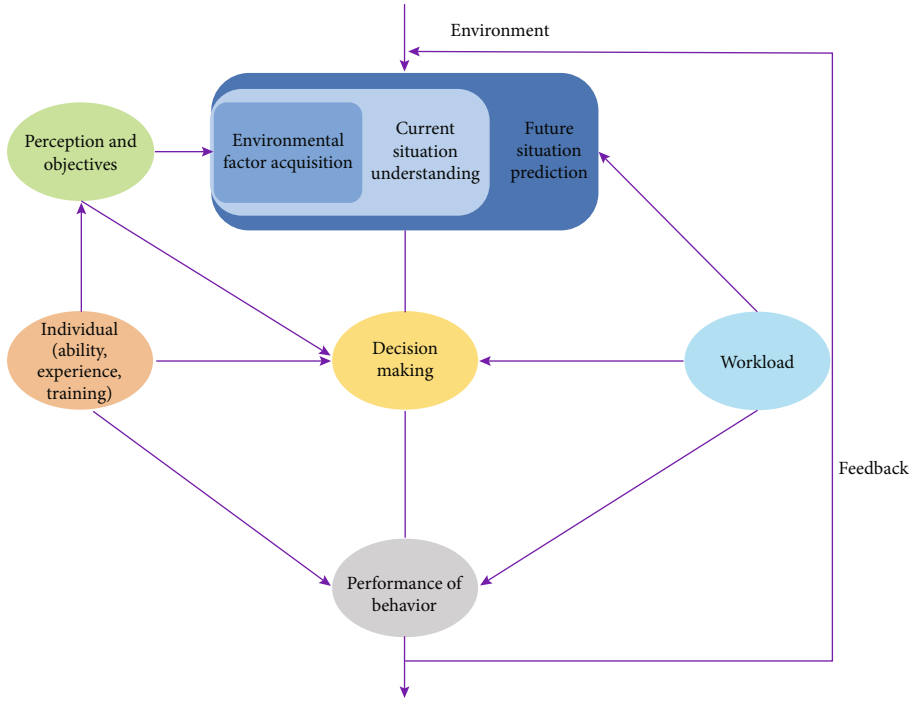


Figure 1: The situational awareness model.

2. State of the Art

2.1. The Concept of Cyber Security Situational Awareness. The aerospace field pioneered the concept of situational awareness, and the situational awareness model was divided into three levels [27, 28]. The information from the situational awareness model is shown in Figure 1.

With the in-depth study of the posture, the understanding of the posture has become more profound. As shown in Figure 2, raw data refers to the unprocessed monitoring data obtained through various data source tools, which reflects that raw data is the direct result of monitoring. Information refers to the preprocessing of raw data, mainly filtering out dirty data and other data that are repeated and unnecessary. Knowledge is the search for patterns based on the information and the use of scientific means to further dig out the deeper valuable activity content. Understanding is the analysis of the knowledge data to obtain certain characteristics and the tendency of their intentions. State prediction is the evaluation of current and prediction of future data characteristics.

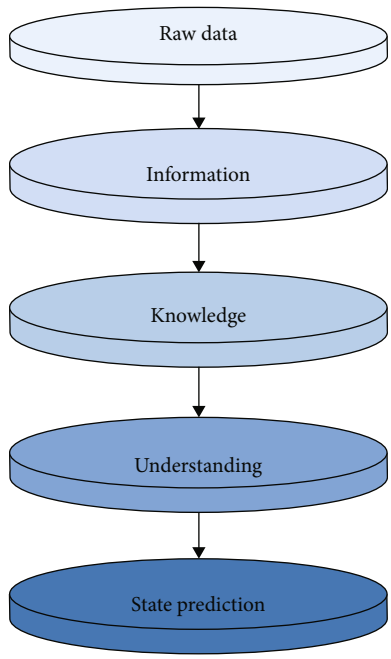


Figure 2: Cognitive mapping of situational awareness.

2.2. Principles and Methods of Network Security Posture Prediction. Common methods for network security posture prediction include gray theory, neural networks, time series, and hidden Markov models [29–32]. The gray theory is an applied mathematical discipline that studies the phenomenon that information is partly clear and partly unclear and with uncertainty. Neural networks are more widely used, not only in prediction but also as an essential theoretical basis in fields such as image recognition. Time series are more common indicators of the characteristics of data analysis models and often have more stringent requirements for the time sequence.

The prediction model of time series is an important research hotspot in the field of time series and usually obtains a series of consecutive data on equally interval time points as time series. There are two common implementation methods, which are the direct prediction method and the iterative prediction method.

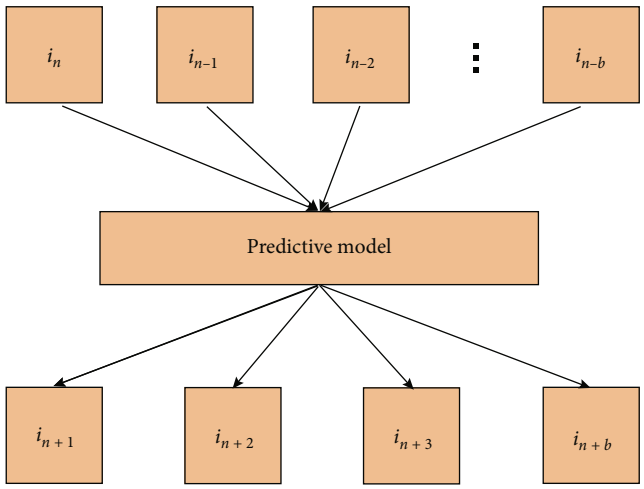


Figure 3: Multistep time series direct prediction method.

(1) Direct prediction method: the direct prediction method is a very simple way of thinking in which the model is built using multiple input and multiple output forms. Multiple training values are input at one time, and multiple prediction values are also output at one time as the final result of the prediction. The structure of the prediction model is shown in Figure 3

(2) Iterative forecasting method: the model of the iterative forecasting method predicts only a single time series node at a time. Multiple time series are input to the prediction model, and the prediction result for one future time step is output. The iterative forecasting method treats the prediction result as a known value and adds it to the input time series of the current forecasting model to predict the next time step. And so on, all the prediction results are obtained after b iterations, and the structure of the prediction model is shown in Figure 4

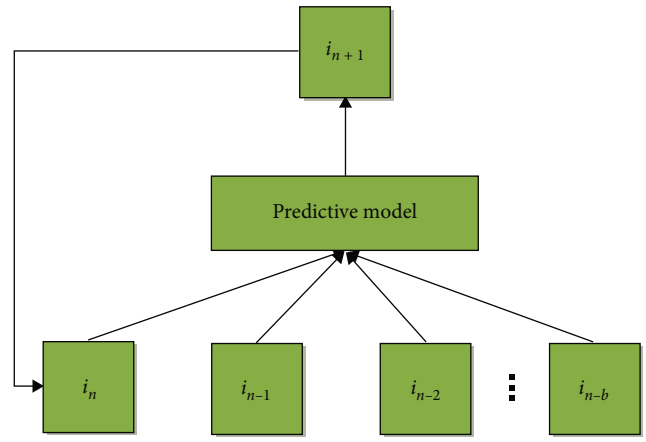


Figure 4: Multistep time series iterative prediction method.

3. Methodology

3.1. BP Neural Network. A BP (back propagation) neural network is a model system that implements a mapping of nonlinear relationships between multidimensional data pairs and trains, adjusts, and corrects weights and thresholds according to an error feedback algorithm. It consists of an input, an implicit (intermediate), and an output (see Figure 5).

It contains n input-side neurons, p hidden layer neurons, and m output-side neurons, with a total of $t \times u + p \times w + u + w$ weights and thresholds. The neural nodes at the middle end receive the external input data from the neurons at the input layer, and the input data are processed at the middle end, and then the output becomes the input data at the output end, which becomes the final result after the processing at the output layer.

3.2. Particle Swarm Algorithm. The particle swarm optimization (PSO) algorithm [33] belongs to the population-intelligent evolutionary method which has both global search performance and local fast convergence characteristics. It maps the problem of solving the objective to the migration behavior of a flock of birds.

The particles continuously update their velocity and position vectors by perceiving and comparing their own and surrounding particles' position and velocity parameters with the fitness measure until they find the global optimal position in the whole population that meets the set fitness requirement. The evolutionary iterations of velocity and position vectors of the population particles are formulated as follows:

$$\begin{aligned} q_{\mathrm{xd}}(n+1) &= \mathrm{mq}_{\mathrm{xd}}(n) + c_1 \times \mathrm{rand}\,() \times (u_{\mathrm{best}} - i_{\mathrm{xd}}(n)) \\ &\quad + c_2 \times \mathrm{rand}\,() \times (A_{\mathrm{best}} - i_{\mathrm{xd}}(n)), \\ i_{\mathrm{xd}}(n+1) &= i_{\mathrm{xd}}(n) + q_{\mathrm{xd}}(n), \end{aligned} \tag{1}$$

where $q_{\mathrm{xd}}(n)$ is the velocity vector of the particle; c_1 and c_2 are the learning factor coefficients; rand () is a random number between 0 and l; $i_{\mathrm{xd}}(n)$ is the current position vector of the particle; $u_{\mathrm{best}}(n)$ represents the current best position of the particle; $A_{\mathrm{best}}(n)$ represents the current global optimum position of the population. m is called the self-adjusting inertia weighting factor coefficient, and its expression is as follows:

$$\begin{aligned} m &= (m_{\mathrm{ini}} - m_{\mathrm{end}})(n_x/n_{\max})^2 \\ &\quad + (m_{\mathrm{ini}} - m_{\mathrm{end}})(2n_x/n_{\max}) + m_{\mathrm{ini}}. \end{aligned} \tag{2}$$

The maximum number of iterative generations $n_{\max}$ for the population, the current number of iterative generations n_x generations, m_{ini}, and m_{end} represent the initial maximum inertia weights and the initial minimum weights, respectively. In order to balance the global search ability and local optimization efficiency, the evolution of m is nonlinearly decreasing with time.

3.3. Improved Population Intelligence Algorithm. Intelligent algorithms, as an emerging stochastic optimization algorithm, have better global search performance, a broader search space, and faster algorithm operation than

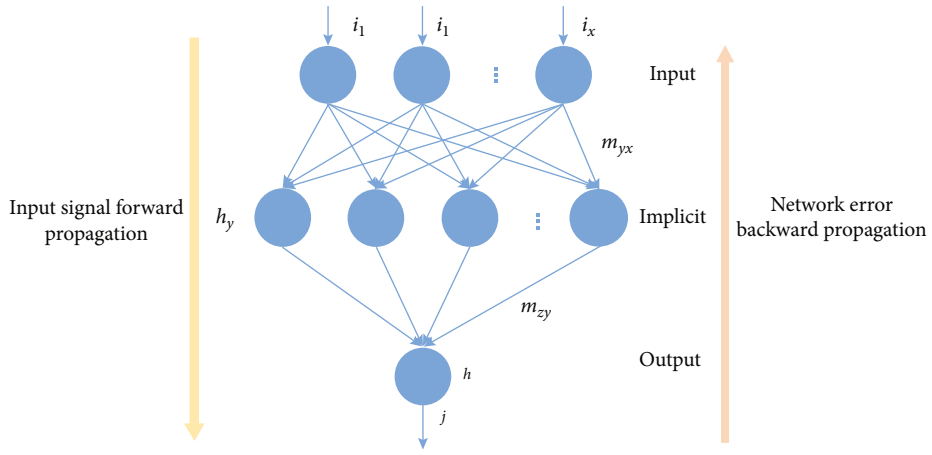


Figure 5: Neural network topology diagram.

traditional, direct optimization algorithms that solve for the optimal value of the objective function, such as error feedback algorithms. Therefore, in this paper, the basic particle swarm algorithm is improved from the need for situation prediction, and a more efficient search algorithm is designed to obtain the ideal parameter combination model.

Genetic algorithm (GA) [34] is a stochastic global search method that evolves from the genetic idea of simulating the natural evolution of superiority and gradual evolution. The genetic algorithm has good global optimization performance and is often introduced into the PSO algorithm to expand the particle search space. The following is the specific operation process of the three genetic operation operators embedded in this paper:

(1) Selection operation implementation: this operation takes the original set fitness as the evolutionary direction and filters the particles based on the fitness corresponding to the population particle positions in the PSO algorithm. The particles with low fitness and dynamic motion are kept and continue to evolve for the best. The particles with high fitness and stagnation are introduced to other genetic operators for operation, and the filtered particles effectively preserve the diversity of the population

(2) Implementation of the hybridization operation: the hybridization operation randomly selects those particles whose position and velocity are smaller than the specified position and velocity factors in the late evolutionary stage of the PSO algorithm to enter the pairing period. These stagnant particles are hybridized with probability u_c, and the positions and velocities of the particles are randomly changed with each other to obtain the new combination of individuals. Two new individuals are exchanged with the following positions

$$\begin{aligned} i_g(n)' &= \text{rand}\,() \times i_g(n) + (1 - \text{rand}\,()) \times \text{i}_h(n), \\ i_h(n)' &= \text{rand}\,() \times i_h(n) + (1 - \text{rand}\,()) \times i_g(n), \end{aligned} \tag{3}$$

where $i_g(n)'$ and $i_h(n)'$ represent the new positions of the 2 particles, respectively. And $i_g(n)$ and $i_h(n)$ represent the positions of the original 2 particles. The velocities of the later generations after the exchange are as follows:

$$\begin{aligned} q_g(n)' &= \frac{q_g(n) + q_h(n)}{\left|q_g(n)\right| + \left|q_h(n)\right|} q_g(n), \\ q_h(n)' &= \frac{q_g(n) + q_h(n)}{\left|q_g(n)\right| + \left|q_h(n)\right|} q_g(n). \end{aligned} \tag{4}$$

(3) Mutation operation implementation: the mutation operation also selects those particles that satisfy the position, velocity less than the specified position, and velocity factor into the mutation period during the iteration of the above PSO algorithm. The real numbers corresponding to the velocity and position vectors of these selected particles are reinitialized and assigned with probability pm, and the mutated particles replace the previous generation

3.4. Algorithm of This Paper. This paper also adopts a three-layer structure. The number of neuron nodes at the input and output ends is determined according to the nature of the mapping relationship of the original target problem. The number of nodes at the implicit end is determined by the complexity of the neural network, recognition ability, operation efficiency, and whether it is easy to fall into local traps. The following equation is used as reference information for the determination of the number of nodes at the implied end:

$$\begin{aligned} B &< \sum_{x=0}^{u} c_u^x, \\ u &= \sqrt{t \times w}, \\ u &= \sqrt{t + w} + \varepsilon. \end{aligned} \tag{5}$$

The number of neurons at the implied end, u, is related to the number of neurons at each end, the number of samples B, and is a constant between 1 : 10.

The selection of the training samples is a critical step. The number of target samples is often very large, and we select only a portion of the training samples. In order to guarantee the accuracy of the prediction results and reduce the dimensionality of the input vector, the learning samples are selected from a large sample set by applying the principle of correspondence analysis in statistical analysis. The samples have all the characteristics of the initial target problem as much as possible.

Selection of initial weights and thresholds: when randomly initializing the weights and thresholds of the network, the parameters are set as small as possible. Such a setting is beneficial to speed up the learning speed of the algorithm. Therefore, the selection of the number of iterations starts by considering the efficiency of training and reducing the error.

The backpropagation (BP) neural network employs a supervised learning approach, utilizing error feedback and the gradient descent algorithm to fine-tune the weights and thresholds of the neural network. This iterative process ensures the optimization of the network's performance and minimizes the error between its predicted outputs and the desired target values. However, it has the problem of slow convergence of training speed and easy to fall into local minima. The intelligent algorithm runs fast, has strong global search performance, does not require high analytical properties of the target object, and is suitable for large-scale parallel-running, complex nonlinear model systems. In this paper, the improved PSO algorithm improves the BP algorithm to optimize the learning parameters in order to improve the learning speed and parameter optimization accuracy.

The algorithm design idea: the weight and threshold optimization problem of the neural network is converted into a search problem for the optimal position of the particle population, and the inverse of the mean square error EMSE is used as the fitness to guide the direction of the search. The steps of the training algorithm are as follows:

(1) Set the range of values for each parameter of the particle population, and then randomly generate the initialized population

(2) Objective problem: the weights and threshold parameters are optimally mapped to the population particle position parameter search

(3) Determine the number of training samples and input

(4) Calculate the velocity and position of the particles based on the particle swarm evolution formula update

(5) Find the global optimal position A_{best}.

(6) Calculate the weights, thresholds, and fitness of the global optimal particle position mapping

(7) Determine whether the highest fitness of the population satisfies the set value, if so, output the weights, thresholds, and fitness of the global optimal position A_{best} and terminate the iteration; otherwise, continue to the next step

(8) Compare the current position of the particle with the historical best position; if the current position is better, update the current position as the best position of the particle. Then, compare the best position of each individual with the global optimal position. If the current distance $D < \alpha$ and the velocity $q < \beta$, the genetic operator is introduced to crossover the particles, and the mutation operation generates a new generation of dynamic particles. The particles that do not satisfy the conditions are kept to continue the evolutionary iteration

(9) Loop the number of iterations $n = n + 1$, if $n > N_{\max}$, exit; otherwise, go to the fourth step to continue the update

Figure 6 shows the flow chart of the training algorithm.

4. Result Analysis and Discussion

In order to verify the effectiveness of the parameter optimization method of the algorithm in this paper, this chapter compares it with the parameter optimization methods original GA, original PSO, and the state-of-the-art metaheuristic algorithms (GWO (gray wolf optimization) and ACO (ant colony optimization)). The experimental results are shown in Table 1.

The results in Table 1 show that the evaluation accuracy of the proposed algorithm is the highest after optimization, which proves that the parameters obtained by the proposed algorithm are optimal. This is due to the fact that the algorithm in this paper uses chromosomes for information sharing, which makes the movement of the population to the optimal region relatively uniform and covers a large area, which facilitates global meritocracy. In the other four methods, the whole search and update process follows the current optimal solution, and the information flow is unidirectional. According to the time complexity analysis, it can be seen that the time complexity of the algorithm in this paper is the same as that of the other four methods. Therefore, compared with the tiny time advantage of the other four methods, the advantage of the parameter optimization method of this paper's algorithm in terms of accuracy is more obvious.

In order to evaluate the goodness of model fit, in this experiment, the original GA, original PSO, GWO, ACO, and proposed algorithm are modeled separately using the ten-fold cross-validation method. The samples were first randomly disrupted, and 75% of the samples in each category were used for the training set and 25% for the testing set. The prediction results for the testing set are shown in Table 2.

The results show that the model built by the data preprocessed by the algorithm in this paper is much better than the results preprocessed by the other four algorithms. It is proved that the model built by the algorithm in this paper

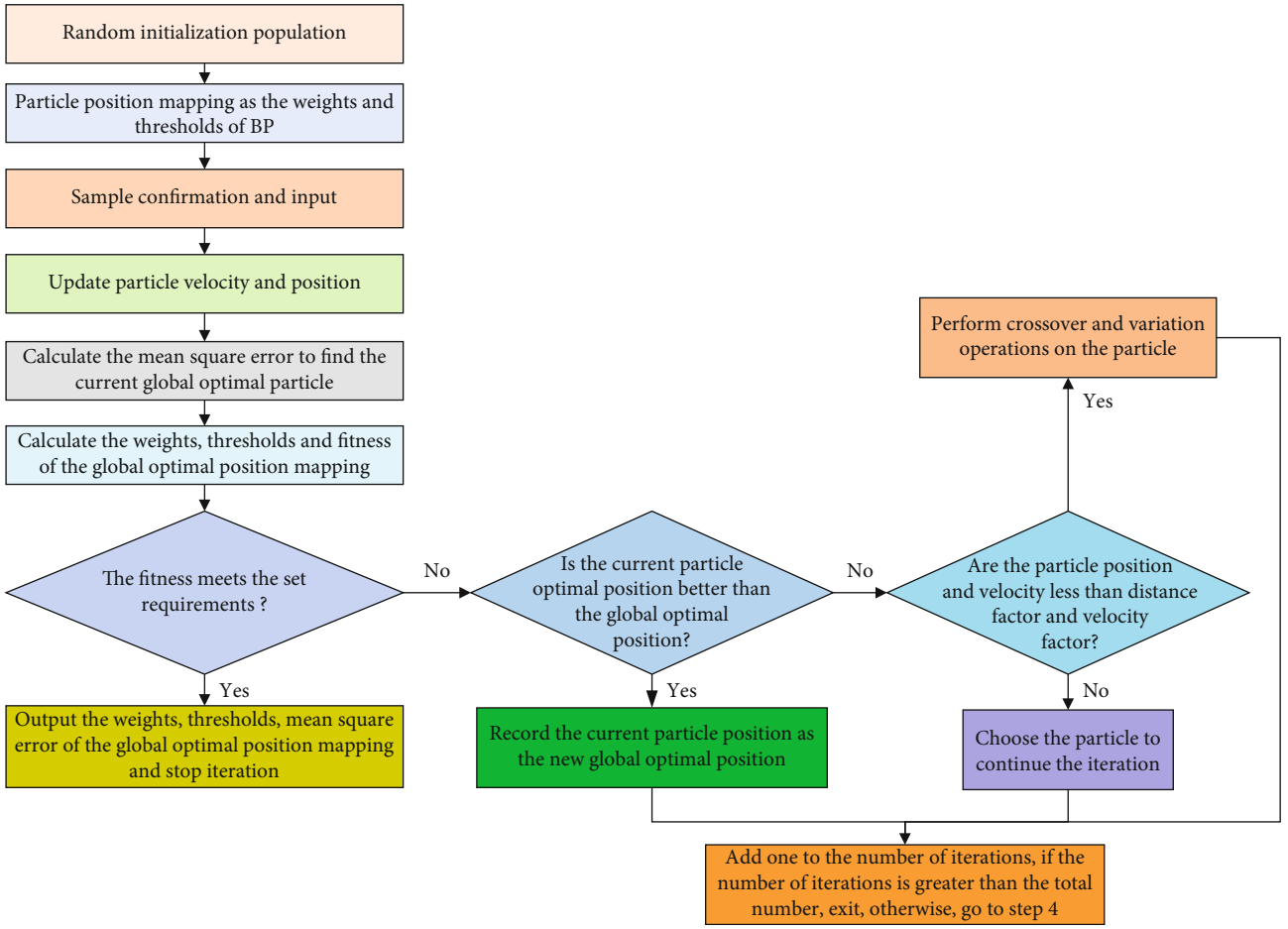


Figure 6: Improved algorithm training BP neural network process.

Table 1: Experimental results of different parameter optimization methods.

	Original GA	Original PSO	ACO	GWO	Proposed algorithm
Accuracy	96.65%	93.46%	95.74%	96.86%	98.78%
Time	499.28 s	483.77 s	490.13 s	485.36 s	470.25 s

is optimal, and the accuracy of the training set reaches 95.83%, and the accuracy of the test set reaches 97.55%. Meanwhile, the goodness-of-fit R^2 is 0.8467 in the training set and 0.8816 in the testing set, which indicates that the model of this paper has a good fitting effect.

In this experiment, we choose MSE as the loss function for the experiments, and the value of MSE is the expected value of the square of the difference between the estimated value and the true value. Its calculation formula is as follows:

$$\mathrm{MSE} = \frac{1}{T}\sum_{n=1}^{T}\left(\mathrm{Observed}_n - \mathrm{Predicted}_n\right)^2 \quad (6)$$

where T represents the initial number of parallel BP neural networks, $Observed_n$ denotes the estimated value of the assessment result, and $Predicted_n$ denotes the true value of the assessment. Equation (6) illustrates that the value of MSE is inversely proportional to the accuracy of security posture assessment, that is, the smaller the value of MSE, the more accurate the assessment is.

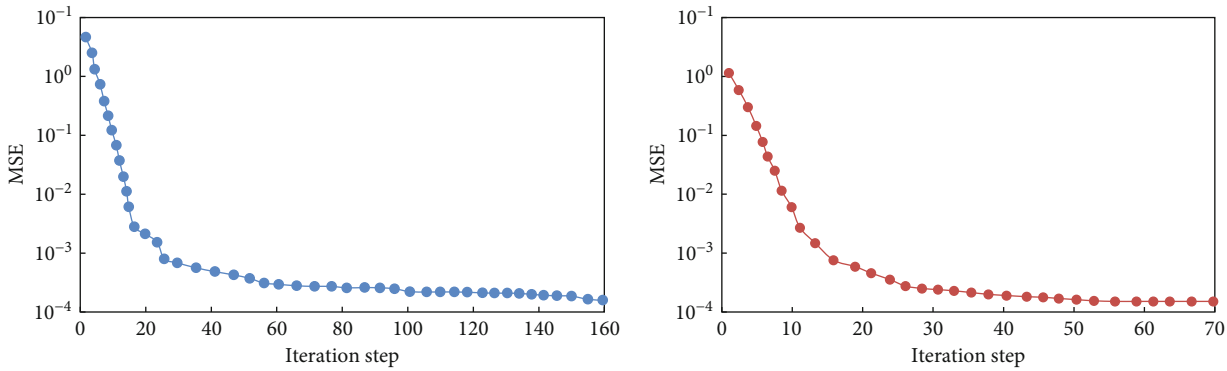
In order to verify the advantages of the improved model over the original BP neural network prediction model in terms of convergence speed, a comparison experiment was conducted to verify the low prediction accuracy and slow convergence speed of the conventional BP neural network prediction model. The training error curves of the proposed model and the original BP neural network training model are shown in Figure 7.

It can be seen from Figure 7 that the BP neural network prediction model takes 160 iterative training steps to basically achieve the set training target error requirement. In this paper, the improved BP neural network prediction model has basically achieved a stable training effect after 80 steps of iterative training, and the set training error requirement is achieved. It can be seen that the convergence speed of the improved prediction model in this paper is significantly accelerated.

The accuracy of the model is verified by comparing the original GA, original PSO, and other two classical machine learning algorithms support vector machine regression

Table 2: Recognition results of five models.

Model	Training set			Testing set		
	Accuracy%	RMSE	R^2	Accuracy%	RMSE	R^2
Original GA	72.17	0.3832	0.4934	78.35	0.6442	0.5013
Original PSO	73.83	0.3331	0.5825	82.65	0.6124	0.5127
ACO	86.42	0.2753	0.6326	91.75	0.4739	0.6538
GWO	83.26	0.2148	0.6733	94.45	0.3928	0.7024
Proposed	95.83	0.2019	0.8467	97.55	0.2646	0.8816



(a) Original BP-NN model training error curve

(b) The proposed model training error curve

Figure 7: Convergence rate of the prediction model before and after improvement.

(SVR) and random forest regression (RFR) by the method in this paper. The results of the comparison experiments are shown in Figure 8.

From Figure 8, it can be seen that the SVR model and the RFR model have better adaptability than the original GA and the original PSO, and the accuracy of the prediction model is higher. However, the core problem of the SVR model is the determination of the kernel function and the selection of relevant parameters, and the constructed prediction model is limited to a certain extent. The RFR model leads to large fluctuations in the prediction results due to the randomness of the random forest itself. The loss value of this paper is lower than that of other methods, and the convergence process is smoother and faster than that of other methods, which has higher accuracy and stability.

The network domain posture evaluation method proposed in this paper is compared with the other two comparison algorithms as well as the original GA and PSO methods to verify the performance advantages of the proposed method. The evaluation performance metrics use the common accuracy, precision, recall, and F1 combined metrics, and in addition, the sampling ratio of 0.9 is used for the method in this paper. The results of the comparison experiments are shown in Figure 9.

As can be seen from Figure 9, the prediction accuracy of the proposed algorithm is significantly ahead of the other two compared methods, as well as the original GA and original PSO methods. The proposed method achieves an accuracy of 97%, compared to 95% for the RFR, 93% for the SVR, and 88% and 89% for the original GA and PSO methods, respectively. In addition, the performance of this method is better than that of other algorithms in the precision, recall, and F1 combined indexes. In particular, the precision of this method reaches 100%. In other words, the false positive rate is 0, and all positive samples are correctly classified.

In order to evaluate the impact of the model on the web attack classification results, a set of five network models were designed in a simulated network environment to do a set of comparison experiments, as shown in Table 3.

Table 3 shows the performance of the five models on the web attack dataset with four evaluation metrics. The results show that the models in this paper have the best performance, with a high-performance improvement compared to the original GA and the original PSO. The experiments demonstrate that the models in this paper are basically unaffected by web attacks and can guarantee the security posture evaluation work in the web domain regardless of whether they are subject to web attacks or not.

In order to further verify the algorithm's situational assessment capability, the evaluation results are fitted with the actual situational trends in this paper, and the obtained results are shown in Figure 10.

As shown in Figure 10, the situational level "safe" is marked as 1, the situational level "low risk" is marked as 2, the situational level "medium risk" is marked as 3, the situational level "high risk" is marked as 4, and the situational level "emergency" is marked as 5. At sample numbers 5, 7, 17, 23, 31, and 33, the network's posture level fluctuates significantly, indicating that the network is subject to a high level of cyber threat at these times. At sample number 7, there is a "medium risk" warning indicating that the network is being attacked by a higher level of threat and

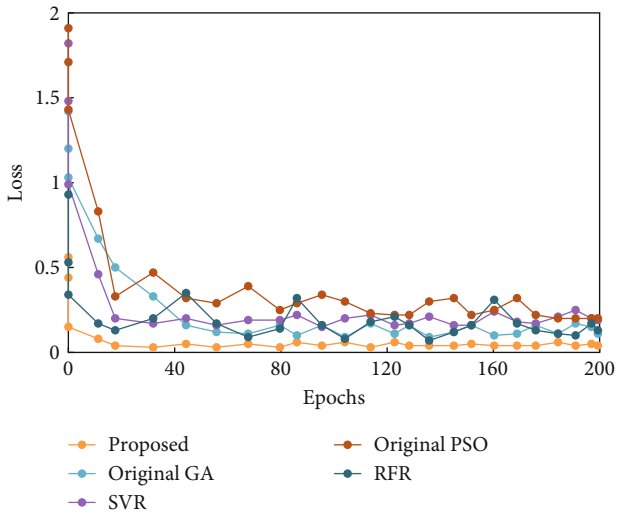


Figure 8: Comparison of loss values of different algorithms.

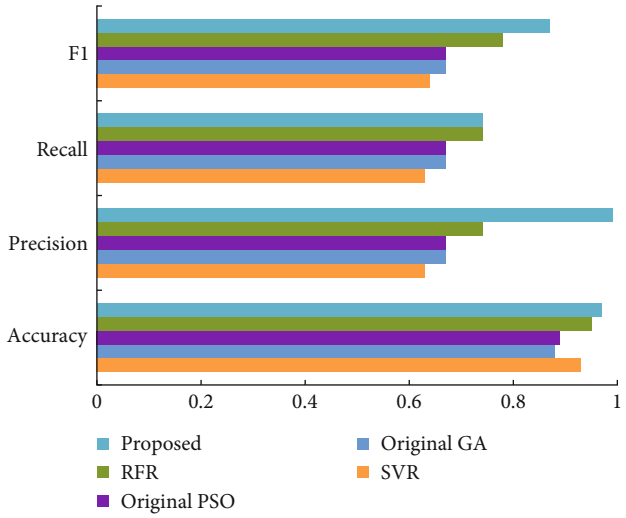


Figure 9: Performance comparison of different algorithms evaluation.

Table 3: Performance of the five models on the web attack dataset.

Model	F1-score	Recall	Precision	Accuracy
Original PSO	0.585	0.496	0.549	0.878
Original GA	0.596	0.517	0.595	0.892
SVR	0.797	0.697	0.696	0.928
RFR	0.826	0.714	0.767	0.949
Proposed	0.865	0.738	0.998	0.979

requiring countermeasures for security defense. The "high risk" level warnings appear in sample 33, indicating that the network is facing a high-security threat and needs to be protected or rescued in time. Based on the two fitted curves comparing the real and assessed values, it can be seen that the proposed method yields a situational assessment that fits perfectly with the real security situation. In contrast, the method in RFR is wrong twice, in samples 7 and 33,

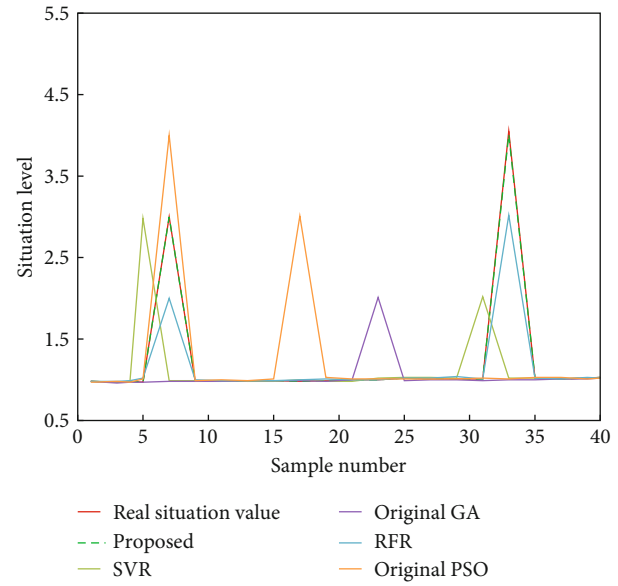


Figure 10: Comparison of the fitting effects of different methodologies for the assessment of legal posture.

respectively. The method of SVR and the methods based on the original GA and PSO have three misjudgments, respectively. The above experimental results fully prove that the method in this paper can adapt to the network domain security posture assessment work in the current network environment and can accurately fit the real security posture change of the network.

5. Conclusion

Based on the powerful nonlinear mapping capability of the BP neural network and also for the problems of low efficiency and insignificant parameter optimization in the training process of the BP algorithm, this paper proposes an improved population intelligence algorithm and BP neural network for a security posture assessment model. The BP neural network is optimized by improving the PSO adaptive adjustment of the global and local optimization-seeking capabilities to make the prediction results more accurate. The weights and pigeon values of the BP network are optimized, and the prediction model is built to predict the network security posture using the inherent hidden parallelism and good global merit-seeking ability of the improved population intelligence algorithm. The algorithm is able to improve the stability of the algorithm while maintaining fast convergence, which improves the shortcomings of traditional BP networks in prediction applications. At the same time, the comparative analysis with other prediction algorithms further verifies the accuracy of this model.

Data Availability

The labeled dataset used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare no competing interests.

Acknowledgments

This study is sponsored by the Xinyang Agriculture and Forestry University.

References

[1] M. D. Renzo, M. Debbah, D. T. Phan-Huy et al., "Smart radio environments empowered by reconfigurable AI meta-surfaces: an idea whose time has come," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Article ID 129, 2019.

[2] M. Talal, A. A. Zaidan, B. B. Zaidan et al., "Comprehensive review and analysis of anti-malware apps for smartphones," *Telecommunication Systems*, vol. 72, no. 2, pp. 285–337, 2019.

[3] W. M. Pardridge, "Drug and gene targeting to the brain with molecular Trojan horses," *Nature Reviews Drug Discovery*, vol. 1, no. 2, pp. 131–139, 2002.

[4] R. A. Badierah, V. N. Uversky, and E. M. Redwan, "Dancing with Trojan horses: an interplay between the extracellular vesicles and viruses," *Journal of Biomolecular Structure and Dynamics*, vol. 39, no. 8, pp. 3034–3060, 2021.

[5] R. U. Khan, X. Zhang, R. Kumar, A. Sharif, N. A. Golilarz, and M. Alazab, "An adaptive multi-layer botnet detection technique using machine learning classifiers," *Applied Sciences*, vol. 9, no. 11, p. 2375, 2019.

[6] S. Sengupta, A. Chowdhary, A. Sabur, A. Alshamrani, D. Huang, and S. Kambhampati, "A survey of moving target defenses for network security," *IEEE Communication Surveys and Tutorials*, vol. 22, no. 3, pp. 1909–1941, 2020.

[7] S. Ndichu, S. McOyowo, H. Okoyo, and C. Wekesa, "A remote access security model based on vulnerability management," *International Journal of Information Technology and Computer Science*, vol. 12, no. 5, pp. 38–51, 2020.

[8] F. Wei, S. Roy, X. Ou, and Robby, "Amandroid: A precise and general inter-component data flow analysis framework for security vetting of android apps," *ACM Transactions on Privacy and Security*, vol. 21, no. 3, pp. 1–32, 2018.

[9] M. Banerjee, J. Lee, and K. K. R. Choo, "A blockchain future for internet of things security: a position paper," *Digital Communications and Networks*, vol. 4, no. 3, pp. 149–160, 2018.

[10] Y. Li, G. Huang, C. Wang, and Y. C. Li, "Analysis framework of network security situational awareness and comparison of implementation methods," *EURASIP Journal on Wireless Communications and Networking*, vol. 2019, no. 1, Article ID 205, 2019.

[11] Z. Fan, Y. Xiao, A. Nayak, and C. Tan, "An improved network security situation assessment approach in software defined networks," *Peer-to-Peer Networking and Applications*, vol. 12, no. 2, pp. 295–309, 2019.

[12] B. Yang and M. Yang, "Data-driven network layer security detection model and simulation for the internet of things based on an artificial immune system," *Neural Computing and Applications*, vol. 33, no. 2, pp. 655–666, 2021.

[13] G. Lin, S. Wen, Q. L. Han, J. Zhang, and Y. Xiang, "Software vulnerability detection using deep neural networks: a survey," *Proceedings of the IEEE*, vol. 108, no. 10, pp. 1825–1848, 2020.

[14] Z. Wang, D. Han, M. Li, H. Liu, and M. Cui, "The abnormal traffic detection scheme based on PCA and SSH," *Connection Science*, vol. 34, no. 1, pp. 1201–1220, 2022.

[15] Z. Zhao, C. Xu, and B. Li, "A LSTM-based anomaly detection model for log analysis," *Journal of Signal Processing Systems*, vol. 93, no. 7, pp. 745–751, 2021.

[16] S. Zhang, L. Zhou, X. Chen, L. Zhang, L. Li, and M. Li, "Network-wide traffic speed forecasting: 3D convolutional neural network with ensemble empirical mode decomposition," *Computer-Aided Civil and Infrastructure Engineering*, vol. 35, no. 10, pp. 1132–1147, 2020.

[17] X. Liu, Z. Liu, Z. Liang, S. P. Zhu, J. A. F. O. Correia, and A. M. P. de Jesus, "PSO-BP neural network-based strain prediction of wind turbine blades," *Materials*, vol. 12, no. 12, p. 1889, 2019.

[18] Z. Shen, X. Zhao, C. Pang, and L. Zhang, "GAN-FDSR: GAN-based fault detection and system reconfiguration method," *Sensors*, vol. 22, no. 14, p. 5313, 2022.

[19] E. Doynikova, E. Novikova, and I. Kotenko, "Attacker behaviour forecasting using methods of intelligent data analysis: a comparative review and prospects," *Information*, vol. 11, no. 3, p. 168, 2020.

[20] M. Noura, A. Gyrard, S. Heil, and M. Gaedke, "Automatic knowledge extraction to build semantic web of things applications," *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 8447–8454, 2019.

[21] J. Yang, Y. Yang, L. Zheng, R. Cheng, and S. Lin, "Network security situation assessment based on attack graph techniques," *Journal of Physics: Conference Series*, vol. 2310, no. 1, article 012071, 2022.

[22] Y. Feng and C. Wang, "Network anomaly early warning through generalized network temperature and deep learning," *Journal of Network and Systems Management*, vol. 31, no. 2, pp. 1–34, 2023.

[23] R. Huang, X. Fu, and Y. Pu, "A novel fractional accumulative grey model with GA-PSO optimizer and its application," *Sensors*, vol. 23, no. 2, p. 636, 2023.

[24] S. Liaqat, K. Dashtipour, K. Arshad, K. Assaleh, and N. Ramzan, "A hybrid posture detection framework: integrating machine learning and deep neural networks," *IEEE Sensors Journal*, vol. 21, no. 7, pp. 9515–9522, 2021.

[25] K. Ansari, "Cooperative position prediction: beyond vehicle-to-vehicle relative positioning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 21, no. 3, pp. 1121–1130, 2020.

[26] B. Hu, Z. H. Guan, N. Xiong, and H. C. Chao, "Intelligent impulsive synchronization of nonlinear interconnected neural networks for image protection," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 8, pp. 3775–3787, 2018.

[27] P. A. Di Tore, "Situation awareness and complexity: the role of wearable technologies in sports science," *Journal of Human Sport and Exercise*, vol. 10, no. 1proc, pp. 500–506, 2015.

[28] R. Gutzwiller, J. Dykstra, and B. Payne, "Gaps and opportunities in situational awareness for cybersecurity," *Digital Threats: Research and Practice*, vol. 1, no. 3, pp. 1–6, 2020.

[29] U. Brose, P. Archambault, A. D. Barnes et al., "Predator traits determine food-web architecture across ecosystems," *Nature Ecology & Evolution*, vol. 3, no. 6, pp. 919–927, 2019.

[30] X. Li, H. Chen, and B. Ariann, "Computer network security evaluation model based on neural network," *Journal of Intelligent Fuzzy Systems*, vol. 37, no. 1, pp. 71–78, 2019.

[31] Y. Liu, S. Garg, J. Nie et al., "Deep anomaly detection for time-series data in industrial IoT: a communication-efficient on-device federated learning approach," *IEEE Internet of Things Journal*, vol. 8, no. 8, pp. 6348–6358, 2021.

[32] M. K. Mustafa, T. Allen, and K. Appiah, "A comparative review of dynamic neural networks and hidden Markov model methods for mobile on-device speech recognition," *Neural Computing and Applications*, vol. 31, no. S2, pp. 891–899, 2019.

[33] N. K. Jain, U. Nangia, and J. Jain, "A review of particle swarm optimization," *Journal of The Institution of Engineers (India): Series B*, vol. 99, no. 4, pp. 407–411, 2018.

[34] S. Katoch, S. S. Chauhan, and V. Kumar, "A review on genetic algorithm: past, present, and future," *Multimedia Tools and Applications*, vol. 80, no. 5, pp. 8091–8126, 2021.