

7 (x,y)

RESEARCH ARTICLE

Prediction model for the water jet falling point in fire extinguishing based on a GA-BP neural network

ChaoYi Zhang^{1*}, Ruirui Zhang², ZhiHui Dai¹, BingYang He¹, Yan Yao³

1 School of Technology, Beijing Forestry University, Beijing, China, **2** National Agricultural Intelligent Equipment Technology Research Center, Beijing, China, **3** Automation College, Beijing University of Posts and Telecommunications, Beijing, China

* zhangchaoyi@bjfu.edu.cn



OPEN ACCESS

Citation: Zhang C, Zhang R, Dai Z, He B, Yao Y (2019) Prediction model for the water jet falling point in fire extinguishing based on a GA-BP neural network. PLoS ONE 14(9): e0221729. <https://doi.org/10.1371/journal.pone.0221729>

Editor: Jie Zhang, Newcastle University, UNITED KINGDOM

Received: April 17, 2019

Accepted: August 13, 2019

Published: September 4, 2019

Copyright: © 2019 Zhang et al. This is an open access article distributed under the terms of the [Creative Commons Attribution License](https://creativecommons.org/licenses/by/4.0/), which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.

Data Availability Statement: All relevant data are within the manuscript and its Supporting Information files.

Funding: This work was supported by: Beijing Natural Science Foundation (6194041); Beijing Excellent Talent Training Project (2017-2019); Fundamental Research Funds for the Central Universities (NO.2015ZCQ-GX-04); and by National Key Laboratory Opening task (KF2N2014W01-002).

Competing interests: The authors have declared that no competing interests exist.

Abstract

Past research on the process of extinguishing a fire typically used a traditional linear water jet falling point model and the results ignored external factors, such as environmental conditions and the status of the fire engine, even though the water jet falling point location prediction was often associated with these parameters and showed a nonlinear relationship. This paper constructed a BP (Back Propagation) neural network model. The fire gun nozzle characteristics were included as model inputs, and the water discharge point coordinates were the model outputs; thus, the model could precisely predict the water discharge point with small error and high precision to determine an accurate firing position and allow for the timely adjustment of the spray gun. To improve the slow convergence and local optimality problems of the BP neural network (BPNN), this paper further used a genetic algorithm to optimize the BPNN (GA-BPNN). The BPNN can be used to optimize the weights in the network to train them for global optimization. A genetic algorithm was introduced into the neural network approach, and the water jet landing prediction model was further improved. The simulation results showed that the prediction accuracy of the GA-BP model was better than that of the BPNN alone. The established model can accurately predict the location of the water jet, making the prediction results more useful for firefighters.

Introduction

An analysis of the trajectory and state of the water jet is a key step in analysing the accuracy of fire extinguishing; the determination of the firing point location provides the basis for the adjustment of the relevant equipment parameters [1–3]. The location of the ignition point is the basis of water jet control, so intelligent fire-fighting equipment is the synergistic combination of ignition source location detection and jet flow control [4]. Most of the water jet control methods are based on the relationship between the ignition point and the angle of the water gun, but this approach cannot determine whether the water jet accurately falls at the ignition point [5]. The ultimate goal of firefighting water gun control is to accurately hit the ignition source point, in which case fire detection and the calculation of the ignition point space

coordinates are adequate for a water sprinkler fire in the early stage of development [6]; the basic water jet state and its control situation can affect fire extinguishing, so research on the water jet is a key focus for fixed-point fire extinguishing [7].

At present, research methods on water jets in fixed-point fire extinguishing are mainly divided into two kinds: research methods that use image processing and those based on the jet trajectory equation. The research statuses of these two methods are described below.

(1) Water jet recognition and control based on image processing. Chen Jing of the Nanjing University of Aeronautics and Astronautics [8] proposed an image processing method to obtain the position of the whole water jet trajectory; her method can predict the water jet trajectory of the shelter by fitting a curve in the case of an occlusion to obtain the position of the water landing. The water cannon position can be determined by the relationship between the water shooting point and the ignition point, and then the angle of the water cannon is adjusted and revised to coincide with the fire source [9]. Sun Weilu and Zhao [10] divided water supply systems into jet and non-jet classes and they collected the pixel images by using colour clustering analysis methods [10]. The colour information was used to find the minimum RGB chromatic aberration of the current detection point in the jet direction to set the starting point for the next detection. After determining the jet direction, the ignition point was searched for in this direction. Finally, the point closest to the last image endpoint was located as the current endpoint in a suspected endpoint that satisfied the set condition. Su Hao [11] used the idea of target tracking by taking single image frames from a continuous dynamic video sequence captured by a camera as a research object to identify the water jet trajectory in each image frame and determine the water shooting location. The particle swarm algorithm [11] was used to identify the water jet trajectory against a mobile background to find and detect the target location in situations where prior information was lacking [11].

(2) Water jet research based on the trajectory equation. The fixed-point fire water flow trajectory equation is based on a classical physics equation that uses the fire water cannon parameters, for example, the working pressure of the water gun injection, work flow, sprinkler installation height and muzzle pitching angle, and external environmental factors (such as the wind speed and wind direction) to formulate the trajectory equation of the water jet. The trajectory equation is then used to predict the water flow point to achieve fixed-point fire extinguishing. Because the external factors are complex in actual situations, establishing a practical water jet trajectory equation is a critical and difficult problem. Wan Feng and other people at the Shanghai University [12–13], according to the theory of particle kinematics and external ballistics, deduced the jet motion differential equations of a water jet in air, assuming that the jet micro-element is a particle, and obtained the fitting equation by modifying the exponent on the basis of the actual jet trajectory. Liao Xiaodong and others [14] used the jet velocity as a micro-element and used it as a research object for stress analysis, water jet velocity is ejected from nozzles or orifices and diffused through the air. Through the analysis of forces in the air, the water jet trajectory equation, under the action of many influential factors, was established, and a simulation analysis of the jet trajectory of a fire gun was carried out by using MATLAB software, this method [14] overcame the problem proposed by Wan feng [12] by not constraining the working pressure and the flow parameter. Hu Guoliang and others [15] pointed out that due to a series of external factors, even if the water is injected into the space at the centre of the fire, the final flow point may exceed the scope of the margin of error for effective extinguishing. Therefore, considering gravity and the air resistance of a water flow, a force analysis was carried out. According to particle kinematics, ballistics and hydrodynamic theories, the equation of the water jet trajectory was established to model the relationships among the coordinates of each point, the height of the water cannon, muzzle pitching angle, outlet jet velocity and drag coefficient of air.

Existing methods used in water jet research have attained some achievements, but these methods have particular issues; the image processing methods require special cameras, and the environmental lighting conditions affect the results; for methods based on the jet trajectory equation, some equations need to be established under ideal conditions, and the influence of external factors (such as wind) was considered less [16–17].

With the further development and extensive application of artificial intelligence technology, the field of artificial intelligence has provided new ideas and research methods for many fields. Some studies have used artificial intelligence to study fire-fighting equipment, such as Liu Xuefei [18], who used a fire detector signal as an input to a fuzzy neural network to analyse a fire. Wang Fei [19] used the adaptive weighting algorithm to complete the data fusion of information acquired by a sensor and ignition source information.

It is very important to predict the water shooting point for effective fire extinguishing. The traditional water jet point model is a linear model, and the results ignore external factors, such as the environmental conditions and the fire engine status, which must be included considering that the prediction of the water shooting location has a complicated non-linear relationship with the firefighting equipment parameters, environmental factors and water shooting dynamics [20–21]. The BPNN has a unique advantage in dealing with nonlinear relationships and has been a widely used prediction model [22]. Therefore, this study used a BPNN to establish a model. The fire gun nozzle characteristics were included as model inputs, and the water discharge point coordinates were the model outputs [23]. The model accurately located the fire source, and allowed for the timely adjustment of the spray gun.

Data preprocessing

According to reference [24], we selected the input variables of the water jet prediction model: the height (h) of the gun, the horizontal angle (α), the pitch angle (β), the pressure (p), the flow rate (q), the wind influence index ($wind_x$) and ($wind_y$). The output variable was the water jet landing point (x,y). Then, the parameters were trained, and some preprocessing was performed; these parameters needed to be normalized before training.

(1) Data preprocessing

Because of the index system, the measurement units of each data index were different, and the range difference was large. To have comparability among all indexes, the data were preprocessed for normalization before the weight calculation so that the data range of each index was between [0,1]. The calculation formula was as follows:

$$G_{ij} = \begin{cases} \frac{y_{ij} - \min y_{ij}}{\max y_{ij} - \min y_{ij}}, y_{ij} \text{ was positive} \\ \frac{\max y_{ij} - y_{ij}}{\max y_{ij} - \min y_{ij}}, y_{ij} \text{ was negative} \end{cases} \quad (1)$$

(2) Determine the weights

① The weighted mean variance method

In this paper, the mean-variance weighting method was used to assign weights to each index, and the detailed calculation steps were as follows:

The mean value of each index:

$$E(G_j) = \frac{1}{n} \sum_{i=1}^n y_{ij} \quad (2)$$

The average variance of each index:

$$\sigma(G_j) = \sqrt{\frac{1}{n} \sum_{i=1}^n [y_{ij} - E(G_j)]^2} \quad (3)$$

The weight coefficient of the index set:

$$W(G_j) = \frac{\sigma(G_j)}{\sum_{i=1}^m \sigma(G_j)} \quad (4)$$

The result of the composite indicators was calculated as follows:

$$F(G_j) = \sum_{j=1}^m W(G_j) \cdot G_j \quad (5)$$

In addition, according to the performance description of neural network, the determinant coefficient R^2 is calculated for the linear fitness index of the prediction system.

$$SST = \sum_{i=1}^n (y_i - \bar{y})^2 \quad (6)$$

$$SSR = \sum_{i=1}^n (\hat{y}_i - \bar{y})^2 \quad (7)$$

$$SSE = \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (8)$$

Where, SST is the sum of total squares, SSR is the sum of regression squares and SSE is the sum of residual squares. According to the relationship between these formulas, there is $SST = SSR + SSE$. Therefore, the determinant coefficient is defined as:

$$R^2 = \frac{SSR}{SST} = \frac{\sum_{i=1}^n (\hat{y}_i - \bar{y})^2}{\sum_{i=1}^n (y_i - \bar{y})^2} = 1 - \frac{SSE}{SST} \quad (9)$$

In the simulation, the "Mapminmax" function in MATLAB can realize the above calculation process to perform the normalization operation. The function maps the original data to the range of the sample set, [0,1]. The result was a structured and normalized matrix that contained the mapping relationship. In this way, when new data were obtained, it could be normalized directly without the normalization of all of the original data, and the training output results could be calculated according to the mapping information. In MATLAB, The functions of error and determinant coefficients of neural networks are defined as follows:

error = abs(T_sim—T_test)./T_test;

R2 = (N * sum(T_sim .* T_test)—sum(T_sim) * sum(T_test))^2 / ((N * sum((T_sim).^2)—(sum(T_sim))^2) * (N * sum((T_test).^2)—(sum(T_test))^2));

Materials and methods

Prediction model of the water jet falling point based on the BPNN

Construction of the prediction model based on a BPNN. The fire water jet model designed in this paper was based on the BPNN. First, according to the requirements of the BPNN, the hierarchical structure of the network was determined by specifying the number of nodes in the input, hidden, and output layers and selecting the transfer functions. In terms of the number of network layers, the traditional three-layer BPNN structure, which includes an implied layer, was chosen, which is shown in (Fig 1).

In the input layer, the number of nodes was set to 7 to represent the 7 input variables of the model. The output layer had two outputs: the data type and the data value. The results of the BPNN algorithm were expressed in binary form and were used for classification and identification; the number of categories determined the number of output nodes. In this paper, the output was the coordinate value of the drop point, that is, the coordinate value of (x, y) , so the number of output nodes was set to 2.

Transfer functions and training processes. In the past, there have been many studies on the transmission function of the BPNN and their deformation [25–26]. According to the previous literature and the characteristics of the water droplet drop point, the implicit layer used the hyperbolic tangent S function, which is shown in Eq (10). A linear function was selected as the transfer function of the output layer, which is shown in Eq (11). These two equations had corresponding functions in the MATLAB Toolbox library.

$$f(x) = \frac{2}{1 + e^{-xx}} - 1 \quad (-1 < f(x) < 1) \quad (10)$$

$$f(x) = kx + c \quad (11)$$

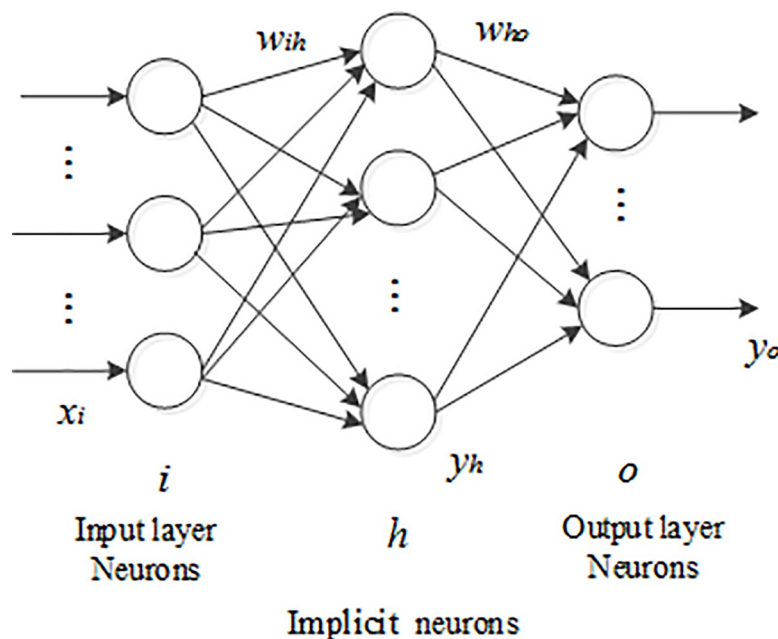


Fig 1. Structure of the BPNN. It is a traditional three-layer BPNN structure. By specifying the number of nodes in the input layer, hidden layer and output layer and selecting the transfer function, the hierarchical structure of the network is determined.

<https://doi.org/10.1371/journal.pone.0221729.g001>

Many reference algorithms have been studied for the training of a BPNN, and different training methods have some influence on the accuracy of the prediction. From the point view of the flow impact prediction model, the LM algorithm was used to train the network, and the method for BPNN weight correction already appeared in many studies.

In this paper, the data were randomly divided into 3 sample datasets: training, confirmation and testing. The confirmation dataset was dynamically changing, and the error of each training observation decreased continuously. The error value from each training iteration was continuously monitored, and when the value was less than the threshold, the training was considered complete. After several iterations of training, the BPNN confirmation error value tended to remain steady, and the network performance was stable; if the training continued, the network's prediction results would not be much improved, so to make the error not decline further, the maximum number of BPNN training iterations was limited to avoid the "overfitting" phenomenon. The main parameters of the BPNN were set, as shown in Table 1:

In practical applications, the number of hidden layer nodes needs to be set. If the number of nodes is too little, the network learning ability is weakened, and increasing the number of hidden layer nodes can effectively improve the learning performance. However, if the number of nodes is very large, the time complexity of network training increases, and the network training time becomes longer. Because there was no fixed method for determining the number of nodes in the hidden layer, the test method was generally used, that is, an empirical formula was used to calculate a range for the number of hidden layer nodes, and the number of hidden layer nodes was set based on the results of different experiments. In this paper, the number of hidden layer nodes was obtained using the empirical formula shown in Eq (12).

$$n = \sqrt{p + q} + a \quad (12)$$

where n is the number of hidden layer nodes, p is the number of input layer nodes, q is the number of output layer nodes, and a is an integer between 1 and 10. In this paper, the number of input layer neurons was set to 7 according to the inputs water gun height (h), horizontal angle (α), pitching angle (β), pressure (p), flow (q), wind influence index ($wind_x$) and ($wind_y$). The number of output layer neurons was 2, that is, the coordinate value of (x,y). According to Eq (12), the range for the number of nodes in the hidden layer was calculated as 4~13. The exact number was determined by an experimental method.

The parameters of the BPNN were set, as shown in Table 2. The number of hidden layer nodes was used to construct the network, and it is determined by the validation data, training and test datasets were used to train and test, the sample error was calculated. The mean square error of the validation data was obtained to predict the neural network performance according to the number of hidden layers nodes. The hidden layer configuration with the lowest network error was observed, and 10 neurons were selected for the hidden layer.

Experimental results

In order to verify the accuracy of the method in this paper, we build a water jet experiment platform, which includes a set of automatic control water gun system, and can freely control the position of the water gun, as shown in Fig 2.

In order to change the height of water gun, the platform was fixed and placed on the elevator. The elevator can move upward and downward by controlling the positive and reverse

Table 1. Neural network training parameters.

Maximum training epochs	Target error	μ	Reduction factor	Increment factor	Maximum failure limit steps
500	0.006	1	0.5	1	20

<https://doi.org/10.1371/journal.pone.0221729.t001>

Table 2. Mean square error of the validation data with different numbers of neurons in the hidden layer.

Hidden layers	3	4	5	6	7	8	9	10	11	12
Mean square error	0.0223	0.0178	0.0127	0.0147	0.0150	0.0180	0.0127	0.0171	0.0185	0.0147
Determination Coefficient	0.18	0.14	0.16	0.15	0.13	0.16	0.14	0.15	0.13	0.16

<https://doi.org/10.1371/journal.pone.0221729.t002>

rotation with DC motor. When it works, it can rise or fall at a uniform speed. The maximum load is about 200 kg, which satisfy the experiment requirements.

To verify the model accuracy, it was necessary to explain the evaluation index. This paper compared the absolute and relative error results, and the experimental results of the test set are shown in Fig 3 (Note: All scientific data in this paper were obtained from the author's experiments. They are the data of the relationship between the water jet landing point coordinates and the spray gun, which are primary data and have been uploaded to the server.)



Fig 2. Physical chart of water jet gun experiment platform. A water jet experimental platform including automatic control water gun system was established.

<https://doi.org/10.1371/journal.pone.0221729.g002>

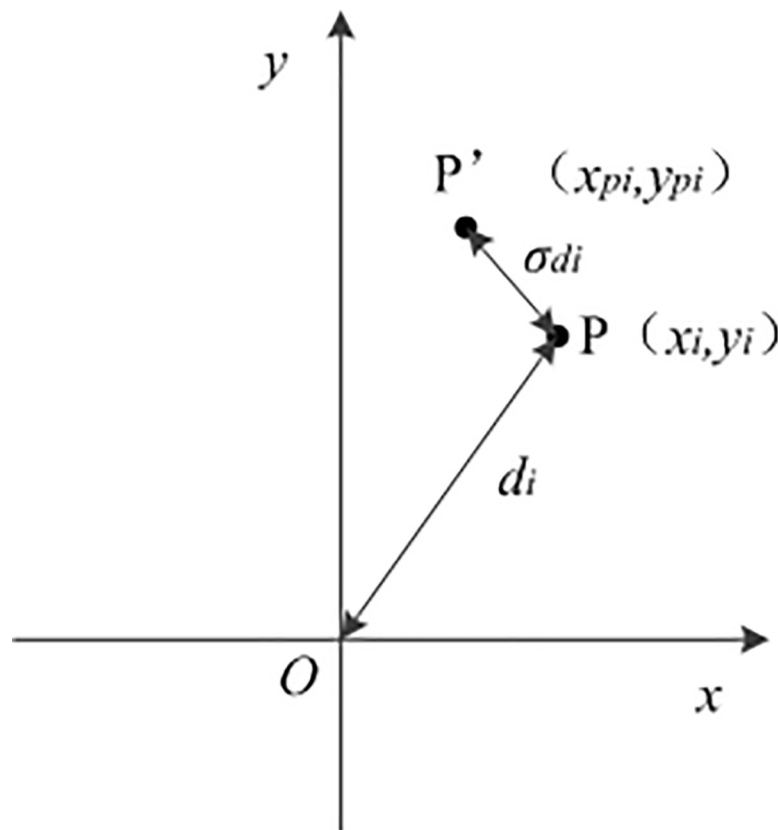


Fig 3. Schematic diagram of the error between the predicted and actual placements. The error schematic diagram of the experimental results and the predicted position. X is the abscissa of the water falling point, Y is the longitudinal coordinate of the water falling point. P is the landing point of water jet and P' is the deviation position.

<https://doi.org/10.1371/journal.pone.0221729.g003>

(1) Average distance error

The mean error was the average distance error between the predicted and actual water jet falling point, which was used (σ_d) to express Eqs (13) and (14) as follows:

$$\sigma_{di} = \sqrt{(x_{pi} - x_i)^2 + (y_{pi} - y_i)^2} \quad (13)$$

$$\sigma_d = \frac{1}{m} \sum_{i=1}^m \sigma_{di} = \frac{1}{m} \sum_{i=1}^m \sqrt{(x_{pi} - x_i)^2 + (y_{pi} - y_i)^2} \quad (14)$$

where σ_{di} is group i 's average linear distance error between the predicted and actual water jet falling point; x_{pi} and y_{pi} are the coordinates of the position of the water flow point (p') outputted from the BPNN for group i ; x_i and y_i are the coordinates of the position of the corresponding actual water flow point (p) for group i ; m is the test set sample size, which was set as 500; and σ_d is the average linear distance error obtained for the test dataset.

(2) Average distance relative error

The relative error = (absolute value of the measurement error) \div (the actual measured value). This value represented the magnitude of the error value relative to the actual measurement. The coordinates used in this paper for the water flow point are in the form (x, y) . When the horizontal direction of the water flow was 0, the transverse coordinate of the position was close to 0, then the transverse relative error was large, and the prediction result was inaccurate.

Here, the ratio of the water flow point distance error to the actual distance between the water shooting location and the origin was calculated as the relative error; the relative error of the water ejection point of the sample set was evaluated as the criterion, and the calculation was as follows:

$$\delta_d = \frac{1}{m} \sum_{i=1}^m \frac{\sigma_{d_i}}{d_i} \times 100\% \quad (15)$$

where d_i is the distance between the actual water jet point and the origin point for group i and $d_i = \sqrt{x_i^2 + y_i^2}$.

MATLAB was used to simulate the experiments for the prediction model for the water jet displacement points of a sprinkler gun. The simulation parameters were set as follows (Table 3):

From the data acquisition and preprocessing of the above sections, 500 data samples were obtained, and a group of 450 samples was randomly selected from the 500 samples for the neural network training sample dataset. These 450 samples are selected according to the time of water jet. Since the experiment of testing water drop point is a continuous test, the selected training and test set are based on the time of water jet gun. When we select the data of drop point, according to the continuous injection of water flow, randomly select the data in a certain period of time. 90% is the training set, that is, 90% of the time period is selected as the training sample, so that the data in the remaining 10% of the time period is used as the test set, which is more accurate. The BPNN was trained, and the remaining 30 data samples were grouped for the test sample dataset to verify the output accuracy.

Based on the abovementioned characteristics, a prediction model of the water jet falling point based on the BPNN was established. First, the results were tested 5 times with the same training set and test set, as shown in Table 4.

From Table 4, it can be seen that the BPNN had large prediction deviations for the same training and test datasets. In the prediction results, the difference between the maximum and minimum errors was 0.055, and the difference between the maximum relative error and the minimum relative error was 1.9%. This was because for each training session, the initial values of the network parameters were not the same, resulting in a different starting point of the BPNN. At the same time, the convergence of the network was not the same at the end of the training session, so the network output had a large deviation.

To illustrate the prediction effect of the BPNN on the same sample datasets, the experimental results of groups 1 and 4 were compared with respect to the predicted and measured values, as shown in Fig 4. Groups 1 and 4 were picked out from the test dataset.

- a. Comparison of coordinate values in the X direction
- b. Comparison of coordinate values in the Y direction

It can be seen from Fig 4 that most of the predicted and actual landings were relatively close, but there were still a few instances where the water jet landings had large deviations. This also reflected the previous setting of the BPNN model weights and the initial data value settings, which influence the forecast results. It can also be seen from Fig 4 that each time the

Table 3. Parameters of the BPNN prediction model.

Network structure	Training function	Implicit layer transfer function	Output-layer transfer function	Maximum training epochs	Target Error	μ	Reduction factor	Increment factor	Maximum failure limit steps
7-10-2	trainlm	tansig	purelin	500	0.006	1	0.5	1	20

<https://doi.org/10.1371/journal.pone.0221729.t003>

Table 4. Prediction results of the BPNN with the same test dataset.

Group	Average error distance/m	Average relative distance error	Average determinant coefficient
1	0.457	7.36%	0.23
2	0.474	6.71%	0.23
3	0.419	6.27%	0.21
4	0.438	6.82%	0.21
5	0.474	8.17%	0.23

<https://doi.org/10.1371/journal.pone.0221729.t004>

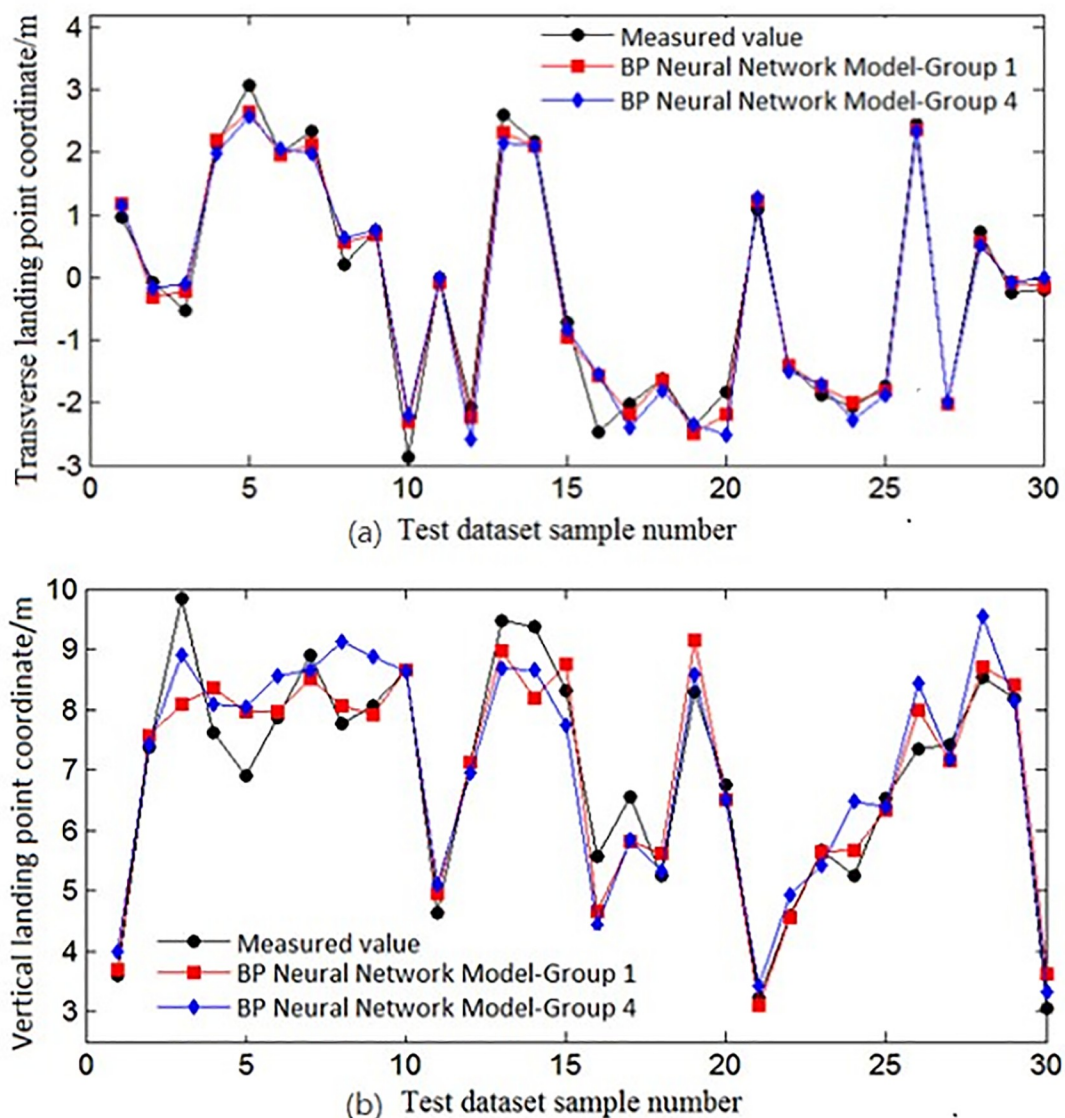


Fig 4. Comparison of two prediction results of the BPNN with the same data samples. The experimental results of group 1 and group 4 were compared with the predicted and measured values. (a) A comparison of coordinate values in the X direction. (b) Comparisons of coordinate values in Y direction.

<https://doi.org/10.1371/journal.pone.0221729.g004>

prediction result had a certain volatility, that is, the prediction model was unstable, an inherent problem was caused by the convergence of the BPNN.

Water jet prediction model based on a GA-BPNN

Genetic algorithm optimization of the BPNN. Literature on the optimization of BPNNs based on genetic algorithms [27–28] is extensive. The GA algorithm can improve the BPNN's slow convergence, which easily falls into the local optima and other issues [25][29]. The optimization goals were the optimization of network weights, the network structure, and learning rules [26][30]. In this paper, the weights and thresholds were optimized by a genetic algorithm. The genetic algorithm mainly included several coding design steps for the individual fitness evaluation, genetic operators, and main operating parameters. If these parameters were determined, the BPNN can use the algorithm to match and optimize the weights in the network so that it can be trained to obtain the global optimization state [27–28][31–32].

Construction of the GA-BP neural network prediction model. In the previous section, the principle of the genetic algorithm optimization of a BPNN was briefly introduced. Next, the construction of the BPNN water jet prediction model based on genetic algorithm optimization (hereafter referred to as the GA-BPNN prediction model) will be specifically described.

There have been many previous studies on using genetic algorithms for BPNNs. We used this model to predict the water jet landing point model. The workflow is shown in (Fig 5):

The algorithm for optimizing the weights of a BPNN using a genetic algorithm is as follows:

The first step was the individual coding and population initialization. In the BPNN, the weights of each neuron were arranged according to certain rules, and the individuals were generated. An encoding process was used to obtain the individuals in a random manner. This process was actually the process used to generate the initial population of a genetic algorithm. This article adopted a real number coding method to encode the individuals, assuming that the chromosome length was S , that is, the encoding length, as seen in Eq (16).

$$S = p \times n + n \times q + n + q \quad (16)$$

where p , n and q were introduced earlier and the initial values were given. After the calculation, $S = 116$. When solving for the global optimal solution of the neural network, the larger the population, the easier it was to find an optimal solution, but the computational complexity would also increase. Therefore, the number of populations was set to 80 in this paper.

The second step consists of the adaptive degree calculation. The fitness function is shown in Eq (17).

$$F = \frac{1}{E} \quad (17)$$

where E is the mean square error of the output prediction result and the actual expectation. The value reflects the probability of individual survival. The higher the value, the more accurate the output prediction result is.

The third step consists of the selection of individuals. In the genetic algorithm, the principle of individual selection is to select the best individuals for producing the next generation from the existing population and address the convergence problem of an optimal individual, i.e., a local optimal solution was found. In this paper, the individuals were selected by using the roulette selection method. The probability of selection is shown in Eq (18):

$$P_i = \frac{f_i}{\sum_{k=1}^s f_k} \quad (i = 1, 2, \dots, s) \quad (18)$$

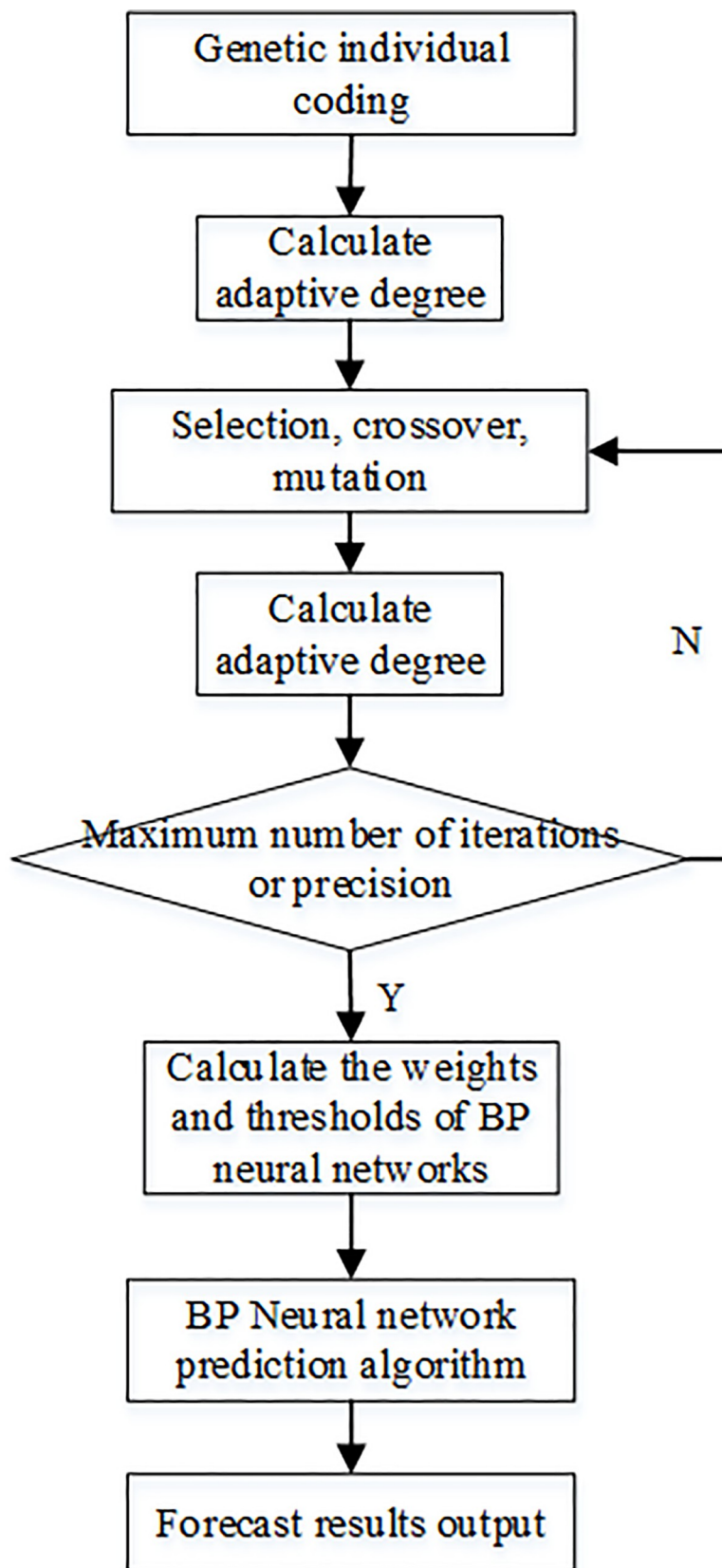


Fig 5. Flow chart of the GA-BP neural network prediction algorithm. The work chart of BPNN water jet prediction model based on genetic algorithm optimization was constructed.

<https://doi.org/10.1371/journal.pone.0221729.g005>

where f_i is the individual i 's fitness value in the population and S is the number of individuals in the population. To achieve the best fitness for individuals in the subsequent crossover and mutation operations, the best-preserved individual strategy was used, i.e., the individuals with the best fitness were replaced by the individuals with the worst fitness in this iteration.

The fourth step consists of the crossover operation. The crossover operation in the genetic algorithm is related to the coding method. It is a step of acquiring a new individual through the exchange of the parental genomic chromosomes, which is determined by the crossover probability. This article used the real-coded methods, as shown by Eqs (19) and (20):

$$X_A^{i'} = \alpha X_B^i + (1 - \alpha) X_A^i \quad (19)$$

$$X_B^{i'} = \alpha X_A^i + (1 - \alpha) X_B^i \quad (20)$$

where α is a random number between [0–1]; X_A and X_B are a pair of chromosomes chosen at random for the crossover operation; X_A^i and X_B^i are the corresponding gene positions for X_A and X_B , respectively; $i \in [1, S]$ (S is the length of the chromosome), and $X_A^{i'}$ and $X_B^{i'}$ are the new genes that produced by the crossover operation [33].

The fifth step involves the mutation operation. In a GA, individual variation is the cause of population diversity. By using the mutation operation, the GA can avoid the premature convergence of the neural network. This article used Eq (21) for the mutation operation. Through the non-uniform mutation operator, the iteration step size can be adjusted according to different generations so that the convergence range was further reduced compared with that of the BPNN.

$$\begin{cases} X_A^{i'} = X_A^i + (U_{\max} - X^i) \times \gamma & \beta \geq 0.5 \\ X_A^{i'} = X_A^i - (X^i - U_{\min}) \times \gamma & \beta < 0.5 \end{cases} \quad (21)$$

where $\gamma = [r \times (1 - \frac{t}{t_{\max}})]^2$, r is a random number from 0 to 1, t is the evolutionary algebra, t_{\max} is the set maximum number of generations, $\beta < 1$, X^i is the gene at the mutation position of the mutation chromosome, $X^i \in [U_{\min}, U_{\max}]$, and $X^{i'}$ is the new gene produced by the mutation operation.

The sixth step is to cycle and terminate. Determining the termination of the genetic algorithm mainly depends on whether an iteration has reached the requirements. If none of the termination conditions are satisfied, the cycle is continued for steps 2~5.

Finally, after terminating the genetic algorithm, the optimal individual results obtained were used in the BPNN; that is, the weights on each layer were re-evaluated, and the subsequent steps were performed, such as training. In this way, the predictions were made, and the model outputs were closer to the actual water jet drop point locations.

Results and discussion

This section covers the verification of the simulation and the analysis of the abovementioned algorithm. The water jet prediction model based on a GA-BPNN is constructed and is compared to the water jet prediction model using the traditional BPNN. The parameter setting part of the GA-BPNN was consistent with that of the BP network in the previous section. In addition, the GA-BPNN genetic algorithm parameter settings are shown in Table 5.

Table 5. Parameters of the genetic algorithm.

Parameters	Value	Instructions
Population size	80	Total number of individuals per generation
Crossover probability	0.25	The probability of the two paired chromosome gene exchanges in the parent
Variation probability	0.01	The probability of a chromosomal gene mutation in the father
The largest evolutionary generation	10	The maximum number of iterations of the genetic algorithm; one of the evolutionary stop conditions
Maximum continuous evolutionary algebra	5	One of the evolutionary stop conditions

<https://doi.org/10.1371/journal.pone.0221729.t005>

The same data were used as with the BPNN simulation experiments in the previous section, this article used the same experiment to predict the falling point of water flow. Here, the GA-BPNN algorithm was first used to preprocess the data, and the same training and sample datasets were processed and operated upon. The experiment was performed 4 times, as shown in Table 6.

From Table 6, we can see that for the same dataset, using the GA-BPNN prediction results, the differences were very small for the prediction accuracies, the average relative error of the maximum and minimum difference was 0.34%, and the error range was 0.14 m. The results indicate the performance relative to that using the BPNNs alone, especially in terms of the ability of local convergence in GA, so that each iteration will choose the best individuals to pass on from one generation to the next. Through the optimization of the weights and thresholds between the neural network layers, the prediction results were achieved more accurately, and the prediction differences were further reduced.

To further compare the results of the GA-BPNN and BPNN for the prediction of the water jet falling point, the following four sets of training and test datasets were selected for experimentation. Ten predictions were performed separately, and the average of the 10 errors was taken as the prediction result. The data results were recorded, as shown in Table 7.

An output result error always exists. This was because the prediction process of the neural network itself had errors, and the inaccuracies in the actual measurement and the settings of the experimental simulation environment will also bring about certain errors. To further eliminate the influence of these accidental errors, this experiment focused on different data and performed multiple sets of measurements and simulations on the two predictive models. From Table 7, we can see that the predictions from the model based on the GA-BPNN were better than those of the BPNN model in terms of distance error and relative error.

The above experimental schematic diagram of the graphical results is given below, as shown in Fig 6. For the different groups, the same sample set and training set used by the GA-BP and BPNNs were used for predictive modelling. A total of 20 experiments were performed. The best experiment was used as a comparison to observe the two networks' predicted and measured values.

Table 6. Prediction results of the GA-BPNN with the test dataset.

Group	Average error distance	Average relative error	Average determinant coefficient
1	0.301	4.34%	0.15
2	0.310	4.68%	0.15
3	0.396	4.51%	0.18
4	0.304	4.47%	0.15

<https://doi.org/10.1371/journal.pone.0221729.t006>

Table 7. Comparison of prediction results for the BP and GA-BP models.

Group	BPNN prediction model		GA-BPNN prediction model	
	Average error distance	Average relative error	Average error distance	Average relative error
1	0.471	7.59%	0.317	4.59%
2	0.485	7.25%	0.337	4.96%
3	0.507	7.62%	0.343	4.74%
4	0.491	7.36%	0.345	4.66%

<https://doi.org/10.1371/journal.pone.0221729.t007>

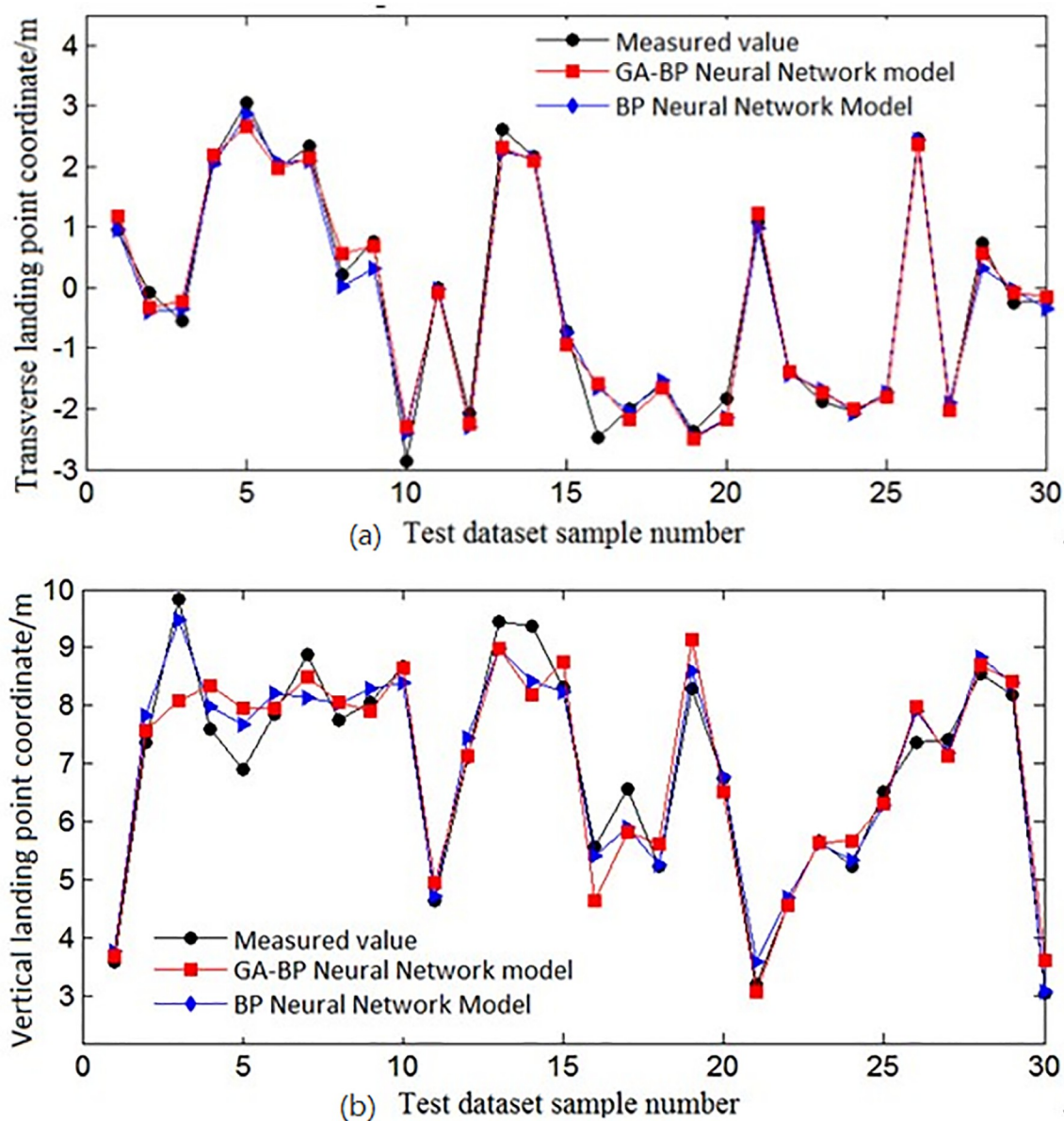


Fig 6. Comparison of the predicted values and measured values of the two predictions. The best experimental data are used as comparison, and the same sample set and training set used by the GA-BP and BPNNs were used for prediction modeling. (a) A comparison of coordinate values in the X direction. (b) A comparison of coordinate values in the Y direction.

<https://doi.org/10.1371/journal.pone.0221729.g006>

- a. Comparison of the coordinate values in the X direction
- b. Comparison of the coordinate values in the Y direction

It can be seen from Fig 6 that due to multiple sets of experiments and taking the data from the experiment with the best results as the comparison, the actual error in these two algorithms was not large. However, it still can be seen that for the predictions based on the GA-BP, the predicted water jet landing point was basically consistent with the actual landing point. Compared with the BPNN prediction, the error appeared to be relatively large, and the difference between these two algorithms is reflected in Table 7.

From the comparison between Tables 6 and 7, it can be seen that the BPNN was inferior to the GA-BPNN in terms of local convergence, and there was some deviation in the final prediction. This deviation will be judged by firefighters in an actual fire extinguishing process.

Conclusion

This paper used the classical BPNN to model and predict the water jet falling point of a fire gun and designed a network structure suitable for the fire water jet landing point model, including the data collection, data preprocessing, network weight setting, transfer function selection and other parameter settings. Then, experiments and predictions were carried out. As a result, it was found that the prediction error of the water jet landing point based on the BPNN was relatively large. Then, a water jet prediction model based on the GA-BPNN was proposed. The article adopted the optimization structure of the genetic algorithm to analyse and optimize the neural network weights and thresholds, and, to a large extent, address the problems of the BPNN training process in terms of local minima and other issues. The simulation results showed that the accuracy of the GA-BP model prediction was better than that of the BPNN. The established model can accurately predict the location of the water jet, making the prediction result more useful for firefighters.

Supporting information

S1 File. Structure of experiment system.
(PDF)

S2 File. Water jet coordinate data set.
(RAR)

S3 File. Platform experiment data of water jet system.
(PDF)

Acknowledgments

We would like to thank Feng Chen and Jianhui Lin for the discussion and advice. We would also like to cordially thank Fengjun Chen for advice and fruitful brainstorming sessions. We would also like to acknowledge Yili Zheng's suggestions for revision papers.

This work was supported by Beijing Natural Science Foundation(6194041), Beijing Excellent Talent Training Project (2017–2019), National Key Laboratory Opening task (KF2N2014W01-002).

Author Contributions

Conceptualization: ChaoYi Zhang, Ruirui Zhang.

Data curation: ZhiHui Dai.

Formal analysis: ChaoYi Zhang, BingYang He, Yan Yao.

Investigation: ZhiHui Dai.

Methodology: ChaoYi Zhang, Ruirui Zhang.

Project administration: ChaoYi Zhang.

Supervision: Yan Yao.

Writing – original draft: ChaoYi Zhang.

Writing – review & editing: ChaoYi Zhang, ZhiHui Dai, BingYang He.

References

1. Jiang L, Ruolin W, Hui Z, Peng M, Qiang Z, Jiangyang Z, et al. (2015). Middle or low water pressure direct spiral double helix converging nozzle structure optimization and flow field analysis. *High Technology Letters*, 21(3), 301–308., et al. Middle or low water pressure direct spiral double helix converging nozzle structure optimization and flow field analysis. *High Technology Letters*. 2015; 21(3): 261–268. <https://doi.org/10.3772/j.issn.1006-6748.2015.03.003>
2. Zou L, Zhu GX, Chen Z, Pei YG, Zong Z Numerical Investigation on the Water Entry of Convex Objects Using a Multiphase Smoothed Particle Hydrodynamics Model. *International Journal of Computational Methods*. 2018; 15(2): 1–12. <https://doi.org/10.1142/S0219876218500081>
3. Enrile F, Besio G, Stocchino A. Shear and shearless Lagrangian structures in compound channels. *Advances in Water Resources*. 2018; 113: 141–154. <https://doi.org/10.1016/j.advwatres.2018.01.006>
4. Rappsilber T, Krüger Simone. Design fires with mixed-material burning cribs to determine the extinguishing effects of compressed air foams. *Fire Safety Journal*. 2018; 98: 3–14. <https://doi.org/10.1016/j.firesaf.2018.03.004>
5. Shu-Man W, Yi-Qing MA, Shao-Zhen YU. Effect of rocket engine jet flow with water injection on air-flow exhaust of gas-flow-guided channel. *Binggong Xuebao/Acta Armamentarii*. 2017; 38(1): 97–105.
6. Diez G, Luis E, Martín SD. Vertical atmosphere and fire. *Forest Fires: Detection, Suppression and Prevention*, 2009; (1): 1–32
7. Zhang XW, Yu YG, Mang SS. Numerical Analysis of Influence of Charge Parameters on Flow Field around Sealed Muzzle of Underwater Machine Gun. *Binggong Xuebao/Acta Armamentarii*. 2018; 39(1): 18–27.
8. Jing C, Min Z. Image Segmentation and Recognition Algorithm for the Fire Cannon's Jet Trajectory. *Electronic Sci. & Tech*. 2010; 23(3): 43–45.
9. Kuo-Hsien H, Shao-Fan L, Juhng-Perng S. PTZ camera based auto-targeting control of water cannons on firefighting boats. *ICIC Express Letters*. 2012; 6(3): 811–817.
10. Weilu S, Min Z. The Extraction Algorithm for the Fire Cannon's Jet Trajectory in Colorful Pictures. *Electronic Sci. & Tech*. 2011; 24(3): 27–30.
11. Jie Y, Min Z, Hao S, Liang L. Jet trajectory recognition method based on the fusion with vector search method and particle swarm optimization. *JOURNAL OF ELECTRONIC MEASUREMENT AND INSTRUMENTATION*. 2016; 30(5): 803–809.
12. Feng W, XiaoYang C, YongLin M, WenMin W. Fitting equation of the fire monitor's jet track. *Fire Science and Technology*, 2007; 26(6): 656–658.
13. Brooks H, Haines GE, Lin MC, Sanderson SL. Physical modeling of vortical cross-step flow in the American paddlefish, *Polyodon spathula*. *PLoS ONE*. 2018; 13(3): 1–25. <https://doi.org/10.1371/journal.pone.0193874> PMID: 29561890
14. Xiaodong L, Pingan L, Wenyu C. The jet trajectory study of fire monitor based on MATLAB. *Fire Science and Technology*. 2014; 10: 1169–1172.
15. Long M, Hu GL, Li Z. Analysis of water jet trajectory of auto-targeting fire sprinkler system in interior large space. *Manufacturing Automation*. 2012; 34(15): 57–60. <https://doi.org/10.4028/www.scientific.net/AMR.490-495.171>
16. Jian G, Guanghua Y, Pengfei H, Jinlu G, Lijun C. An improved back propagation neural network prediction model for subsurface drip irrigation system. *Computers and Electrical Engineering*. 2017; 60(5): 58–65. <https://doi.org/10.1016/j.compeleceng.2017.02.016>

17. Yunhao D, Jinxian Q, Shaoyuan F. Application of BP Neural Network Based on Improved GA for Forecast of Groundwater Level Depth. *Water Saving Irrigation*. 2017; (9): 81–84.
18. Xuefei L, Qin J. Simulation of Fire Detection Based on Fuzzy Nerve Network in Atrium Space. *Computer Simulation*. 2012; 29(8): 159–162.
19. Fei W, Yongliang Q, Dongjian H, Diyi C. Design of fire-fighting robot Based on Multi-Sensor information Fusion. *Microcomputer Information*. 2010; 26(29): 156–158.
20. Jiwei W, Chen C, Urso C, Xiaojun Y. Experimental research on the performances of water jet devices and proposing the parameters of borehole hydraulic mining for oil shale. *PLOS ONE*. 2018; 13(6): 1–32. <https://doi.org/10.1371/journal.pone.0199027>
21. Cui K, Qin X. Virtual reality research of the dynamic characteristics of soft soil under metro vibration loads based on BP neural networks. *Neural Computing and Applications*. 2018; 29(5): 1233–1242.
22. JiRen T., YiYu LU, HuiJuan S. Study on visual BP neural network cutting model for abrasive water jet. *Sichuan Daxue Xuebao (Gongcheng Kexue Ban)/Journal of Sichuan University (Engineering Science Edition)*. 2013; 45(3): 164–170.
23. Heskestad G. Scaling the interaction of water sprays and flames. *Fire Safety Journal*, 2002, 37(6): 535–548.
24. Wenfui H, Jiangming K. Analysis of Influencing Factors of Fire-fighting Lance Water Jet Landing Positions. *FOREST ENGINEERING*. 2017; 33(5): 50–55.
25. Yingyi C, Qianqian C, Yanjun C, Huihui Y, Chao Z. Short-term Prediction System of Water Temperature in Pond Aquaculture Based on GA-BP Neural Network. *Transactions of the Chinese Society for Agricultural Machinery*. 2017; 48(8): 172–178.
26. Ronghua J, Shulei Z, Lihua Z, Qiuxia L. Prediction of soil moisture based on multilayer neural network with multi-valued neurons. *Transactions of the Chinese Society of Agricultural Engineering (Transactions of the CSAE)*. 2017; 33(Supp.1): 126–131. <https://doi.org/10.11975/j.issn.1002-6819.2017.z1.019>
27. Ando H, Ambe Y, Ishii A, Konyo M, Tadakuma K, Maruyama S, et al. Aerial Hose Type Robot by Water Jet for Fire Fighting. *IEEE Robotics and Automation Letters*. 2018; 3(2): 1128–1135. <https://doi.org/10.1109/LRA.2018.2792701>
28. Liu WT, Xie X. Space Observation of Carbon Dioxide Partial Pressure at Ocean Surface. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*. 2017; 10(12): 5472–5484. <https://doi.org/10.1109/jstars.2017.2766138>
29. Xuxiang T, Yaoguang W. Research on a dissolved oxygen prediction method for recirculating aquaculture systems based on a convolution neural network. *Computers and Electronics in Agriculture*. 2018; 145: 302–310. <https://doi.org/10.1016/j.compag.2017.12.037>
30. Wei L, Lijuan C, Yaqiong Z, Zhangjie C, Manyin Z, Weigang X, et al. Using a backpropagation artificial neural network to predict nutrient removal in tidal flow constructed wetlands. *Water (Switzerland)*. 2018; 10(1): 83–99. <https://doi.org/10.3390/w10010083>
31. Jha SK, Ahmad Z. Soil microbial dynamics prediction using machine learning regression methods. *Computers and Electronics in Agriculture*. 2018; 147: 158–165. <https://doi.org/10.1016/j.compag.2018.02.024>
32. Yaseen ZM, Ramal MM, Diop L, Jaafar O, Demir V, Kisi O, et al. Hybrid Adaptive Neuro-Fuzzy Models for Water Quality Index Estimation. *Water Resources Management*. 2018; 32(7): 2227–2245. <https://doi.org/10.1007/s11269-018-1915-7>
33. Rodrigues do Amaral Laurence, Rafael Hruschka Estevam. Transgenic: An evolutionary algorithm operator. *Neurocomputing*, 2014, 127(1): 104–113. <https://doi.org/10.1016/j.neucom.2013.08.037>