Computer Networks 193 (2021) 108102

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

A network traffic forecasting method based on SA optimized ARIMA–BP neural network[☆]

Hanyu Yang [a], Xutao Li [a], Wenhao Qiang [a], Yuhan Zhao [a], Wei Zhang [b,*], Chang Tang [c,*]

[a] School of Cyber Security, Qilu University of Technology (Shandong Academy of Sciences), Jinan 250353, China
[b] Shandong Provincial Key Laboratory of Computer Networks, Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences), Jinan 250000, China
[c] School of Computer Science, China University of Geosciences, Wuhan, 430074, China

ARTICLE INFO

MSC:
00-01
99-00

Keywords:
ARIMA
BP neural network
Hybrid model
Network traffic
Simulated Annealing Algorithm

ABSTRACT

Network traffic forecasting provides key information for network management, resource allocation, traffic attack detection. However, traditional linear and non-linear network traffic forecasting models cannot achieve enough prediction accuracy for future traffic prediction. In order to resolve this problem, a network traffic prediction method based on SA (Simulated Annealing) optimized ARIMA (Autoregressive Integrated Moving Average model)-BPNN (Back Propagation Neural Network) is proposed in this paper, which makes comprehensive use of linear model ARIMA, non-linear model BPNN and optimization algorithm SA. With enhancement of the BPNN global optimization ability, it can fully realize the potential of mining linear and non-linear laws of historical network traffic data, hence improving the prediction accuracy. This paper selects the historical network traffic data of two different sampling points in the WIDE project to predict, and utilizes the MAE(Mean Absolute Error), RMSE(Root Mean Square Error), and the MAPE(Mean Absolute Percentage Error) as the evaluation index of the prediction effect. Experimental results show that our proposed method outperformed traditional network traffic prediction model, with several improvements in network traffic prediction accuracy.

1. Introduction

Living in a society where digital transmission and mobile traffic pervade, we can get tons of information on the Internet through various mobile devices. The Internet has feed people with knowledge of the world around us, which in turn makes the network business developing rapidly [1]. It is inevitable that the network traffic volume has grown dramatically ever since [2]. Literally, network traffic refers to the total traffic of the network link per unit time. Network traffic is often collected within a certain time interval to obtain a time series in the process of network traffic collection [3]. The prediction of network traffic is of great significance nowadays to improve the network efficiency and optimize network resources allocation. After an accurate network traffic prediction, Network operators can adjust the network congestion control mechanism in a timely manner to reduce network delay and packet loss rate, cut the risk of network congestion, and improve user's Quality of Experience (QoE). Meanwhile, network traffic prediction not only provides key information for network management, resource allocation, but also makes a huge difference against malicious network traffic attacks [4], with the potential of uncovering breaches vulnerable to network attacks and intrusions. It is a crucial technology to against numerous network security incidents and of great importance for maintaining the whole network performance [5,6].

Network traffic prediction can be categorized into linear prediction and non-linear prediction [7], where linear prediction mostly includes autoregressive model (AR), moving average model (MA), autoregressive moving average model (ARMA) [8], differential integrated moving average autoregressive model(ARIMA). Non-linear prediction mainly contains the prediction models based on wavelet analysis [9] and neural network model. As a traditional neural network, BP neural network can map any complex non-linear relationships thanks to its non-linear fusion ability, which has been well studied in the field of traffic forecasting [10]. But the BP neural network model training algorithm has inherent characteristics, which causes the neural network model to have the disadvantages of slow convergence rate, and readily to fall into local minimums in practice [11]. The Simulated Annealing Algorithm (SA) is derived from the physical annealing process of solids, it is a stochastic optimization algorithm based on Monte Carlo iterative solution method. As a random search technology, SA has been widely

☆ Fully documented templates are available in the elsarticle package on CTAN.

* Corresponding authors.
E-mail addresses: wzhang@sdas.org (W. Zhang), tangchang@cug.edu.cn (C. Tang).

https://doi.org/10.1016/j.comnet.2021.108102
Received 6 December 2020; Received in revised form 24 February 2021; Accepted 6 April 2021
Available online 15 April 2021
1389-1286/

used in various engineering problems [12]. The algorithm sets a certain initial temperature, then cools down continuously, combines the probabilistic jumping property to continuously search for the global optimal solution in the global solution space. Therefore, the global optimization ability of the simulated annealing algorithm (SA) can just solve the problem, which is the artificial neural networks that are easily to fall into local minimums.

Under the influence of many inherent factors, network traffic possesses the characteristics of sudden change, weak coupling and nonlinearity [10]. Therefore, neither the single linear model nor the nonlinear model can describe the characteristics and change laws of network traffic well. Analyzing linear and non-linear factors in network traffic accurately is the key to improve the network traffic prediction accuracy. Since a single model has limited ability to reveal linear and non-linear relationships in network traffic, these two kind of models can be combined to extract linear and non-linear relationships in network traffic prediction.

A network traffic forecasting method based on SA optimized ARIMA–BPNN (ARIMA–SA–BPNN) is proposed in this paper. For the ARIMA–BP model, ARIMA itself can only capture the linear features in the time series, without the ability to capture non-linear features, as well as the model requires more historical data beforehand. For non-linear features, utilizing learning and data processing ability of the neural network model can mine the non-linear information in the network traffic time series. Meanwhile, the simulation of annealing algorithm is used to improve the global optimization ability of the BPNN. During the process of ARIMA–SA–BP, at first, the ARIMA model is used to extract the linear features of network traffic. With strong processing abilities for time series-based data, ARIMA can still retain the original information of its historical data while extracting linear features.

Secondly, the network traffic residual predicted by the ARIMA model serves as an input in the BPNN, which is optimized by the SA algorithm. Non-linear factors in the residual data pool can be fully captured via its ability to perform non-linear fusion and global optimization. Finally, the final network traffic prediction value is to add the ARIMA model prediction results and the SA-BPNN residual prediction value. This module improves the accuracy of network traffic prediction in two ways. Adaptation of this module solves the problem of fully extracting linear and non-linear features in network traffic. Moreover, this module uses simulated annealing algorithm to make up for the shortcomings of BPNN, and enhance the BPNN global optimization ability. The main contributions of this paper can be summarized as follows:

- We proposed a network traffic prediction method based on SA optimized ARIMA–BPNN. This method can extract linear and non-linear features from traffic data, and can predict network traffic with more accuracy.
- Our work make advances in adjusting the network congestion control mechanism, optimizing the network infrastructure architecture with any other fields that require more accurate network traffic prediction results.
- Experimental results based on real-world network traffic data sets, which validated that our method outperforms current existing network traffic prediction models, like single prediction model and hybrid prediction model, such as LSTM, WNN, ARIMA–BPNN.

The rest of this article is organized as follows: The second part introduces existing studies done by researchers in the field of network traffic prediction. In the third part, we made some elaboration of our principle and analyzes the basis of the modules that comprise the method in this paper. Then we move on to discuss steps of our proposed approach. The fifth part uses real network traffic data sets to verify the effect of the model. Our paper closes with summary of the researches methods in the article and shed some light on future research directions.

2. Related work

We focus on the related work of other scholars in network traffic prediction. These works can be roughly divided into two categories: The first category is the research on the optimization and improvement of the model based on the defects of the existing network traffic prediction model. The second category is to study the characteristics of network traffic on the basis of building a combination forecasting model of network traffic.

Haitao Li [13] proposed a network traffic prediction method based on the firefly swarm optimization algorithm to optimize the BP neural network (GSOBPNN) to improve the convergence speed and learning ability of BPNN and reduce the network traffic prediction error. Yue Hou [14] and others proposed an improved differential evolution BP algorithm to optimize the fuzzy neural network (IDEBPFNN) network traffic prediction method, by optimizing the FNN network output parameters to improve the traffic prediction accuracy. Hongjun Yang [15] proposed a network traffic prediction method based on dynamic adaptive search step (DASSS) improved seeker algorithm (SOA) optimized wavelet neural network (WNN). Ramakrishnan [16] et al. used recurrent neural network (RNN) and its variants long short-term memory (LSTM) network and gated recurrent unit (GRU)) structure to analyze and predict network traffic and proved that it can capture complex network traffic The non-linear relationship and long-term dependence. Shihao Wang [17] et al. studied a long and short-term memory (LSTM) neural network model to predict network traffic with non-linear characteristics. Yusuke Tokuyama [18] et al. proposed a one-hot coding method to improve the traditional RNN-VTD model and predict network traffic. Chih-Wei Huang [19] and others constructed a convolutional neural network (CNN) and recurrent neural network (RNN) combined model and applied it to the research of Mobile Internet traffic prediction. Ming Li [20] and others combined Attention mechanism, Residual Neural Network (ResNet) and Recurrent Neural Network (RNN) to predict wireless network traffic.

In the above research, the main body of the prediction model used is neural network, which has the advantage of fully mining the non-linear factors in network traffic, so as to obtain better prediction results. But its common defect is that it ignores the discovery of linear factors in network traffic. Therefore, the above method is not suitable for processing network traffic data that contains both linear and non-linear factors. Therefore, this paper proposes a network traffic forecasting method based on the ARIMA–SA–BPNN combined model. Both the ARIMA model and the SA-BPNN model are the main body of traffic forecasting. Not only can they fully extract the linear and non-linear factors in the network traffic, but the model has a simple structure, the realization cost is small and has high prediction accuracy.

3. Fundamental analysis

3.1. Simulated annealing algorithm

The original idea of the simulated annealing (SA) algorithm was proposed by scholars such as N. Metropolis in 1953. In 1983, scholars such as S. Kirkpatrick introduced the annealing idea into the field of combinatorial optimization successfully [21]. The Simulated Annealing algorithm is derived from the principle of physical solids annealing in practical, that is, heating the solid, then cooling it down slowly after the temperature rises high enough. In the process of heating the solid to raise its temperature, as the temperature rises up, the internal energy of the particles increases gradually, from the state of orderly arrangement to the state of disordered arrangement. When the solid reaches enough temperature, the arrangement of the particles is sufficiently disordered, and then the heating process stopped and the solid is gradually cooled. During the cooling process, the internal energy of the particles gradually decreases and changes from the disordered state to orderly state. When the temperature decreases to the lowest

Table 1
The meaning of ARIMA (p, d, q) model parameter.

Name of parameter	Meaning
p	Autoregressive coefficient
d	Differential times
q	Number of moving average terms

temperature, the total internal energy of the particles inside the solid is the smallest. At this time, a global optimal solution is displayed. And when the solid temperature decrease slowly to the lowest temperature, the heat balance taken place inside the solid can be referred as the local optimal solution [22].

The simulated annealing algorithm is a greedy algorithm, but the algorithm introduces random factors in the searching process. That is, in the process of searching for the global optimal solution, the SA algorithm can accept a solution that is worse than the current solution with a certain probability. Therefore, it is possible to jump out of the current local optimal solution and find the global optimal solution. The main steps of the simulated annealing algorithm are as follows:

Step 1 Initialize the related parameters. The initial temperature is T, the ending temperature is T_{end}, the initial solution state is S, and the number of iterations for each T value is m.

Step 2 For each new state generated, a new solution S_{new} can be got.

Step 3 Calculate the increment ΔS between the new solution S_{new} and the initial solution S.

$$\Delta S = C\left(S_{new}\right) - C(S) \tag{1}$$

In fact, $C(S)$ is the evaluation function. The evaluation function is the objective function, which needs to be set according to the specific objectives of all optimizations. For example, this article optimizes the BP neural network, the evaluation function is about the error function value corresponding to the current connection weight of the BP neural network. In this paper, the SA algorithm is used to deal with the minimization problem.
If you deal with the maximization problem, you only need to add a negative sign to the evaluation function to convert the processing minimization problem to the processing maximization problem.

Step 4 If $\Delta S < 0$, designate S_{new} as the new current solution. If $\Delta S > 0$, it means that the state of the new solution is not as good as the state of the old solution. Then accept the new solution with a certain probability P, otherwise keep the old solution.

$$P = \exp(-\Delta S/T) \tag{2}$$

Step 5 Repeat **Steps 2** to **4**, if the termination condition is met, output the current solution as the optimal solution.

3.2. ARIMA model

The ARIMA model is a time series forecasting method proposed by scholars Box and Jenkins in the 1970s. Its full name is called Autoregressive Integrated Moving Average model [23]. The main advantage of the ARIMA model is that it only needs relevant time series data and has good short-term forecasting capabilities [24].

The ARIMA model is composed of three parts: "AR", "I", and "MA", where "AR" means autoregressive, "I" means difference, and "MA" equals to Moving average. In the ARIMA (p, d, q) model, the specific meanings of its parameters are shown in Table 1.

The specific mathematical form of the model is shown below:

$$y_t' = \alpha_0 + \sum_{i=1}^{p} \alpha_i y_{t-i}' + \varepsilon_t + \sum_{i=1}^{q} \beta_i \varepsilon_{t-i} \tag{3}$$

In fact, $y_t' = (1-L)^d y_t$, $L^i y_t = y_{t-i}$.

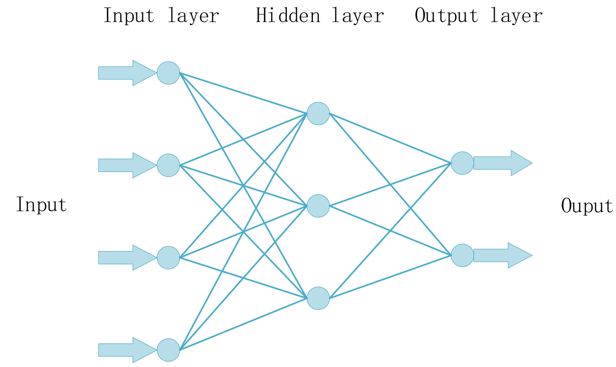


Fig. 1. BP neural network structure.

The formula can be transformed into:

$$\left(1 - \sum_{i=1}^{p} \alpha_i L^i\right)(1-L)^d y_t = \alpha_0 + \left(1 + \sum_{i=1}^{q} \beta_i L^i\right)\varepsilon_t \tag{4}$$

In Eqs. (3) and (4), l is the lag operator, α_i is the coefficient of autoregressive lag term, β_i is the coefficient of moving average lag term, and ε_t is the residual.

The main steps of the ARIMA model are as follows:

Step 1 Obtain time series data and perform data preprocessing, use ADF test to determine whether the series is stable, and then determine the parameter d of the ARIMA model. If the series is not stable, perform a difference operation until the series is stable.
The first-order difference formula is as follows:

$$\Delta y_t = y_t - y_{t-1} = (1-L)y_t \tag{5}$$

Step 2 For the stationary time series obtained through Step 1, calculate the autocorrelation coefficient ACF and partial autocorrelation coefficient PACF respectively. Estimate the sequence parameters p, q through ACF and PACF, and then determine the model parameters p, q through the AIC rule.

Step 3 Check the model and predict the data.

3.3. BP neural network

Artificial Neural Networks (ANNs) are derived from the human nervous system, often used as an alternative mathematical tool for classification, pattern recognition, prediction, and other tasks performed on automatically correlated data. BP (Back Propagation) neural network is a multi-layer feedforward neural network trained by error back propagation algorithm [25,26]. Its main structure is composed of input layer and output layers and several hidden layers, as shown in Fig. 1. BP Neural Network adopts the idea of gradient descent algorithm, and adjusts the weight and threshold of the neural network continuously through the back propagation algorithm to achieve the goal of minimizing the error between the network output value and the actual value.

In the BP Neural Network, the relationship between input and output can be expressed by the following formula:

$$y_T = \omega_0 + \sum_{j=1}^{n} \omega_j g\left(\omega_{0j} + \sum_{i=1}^{m} \omega_{ij} y_{T-i}\right) + \varepsilon_T \tag{6}$$

Among them, y_T represents output; y_{T-i} means input ($i = 1,2,3,\ldots$); m is the number of nodes in the input layer of the BPNN; n is the number of hidden layer nodes; ω_{ij} (i = 1,2, 3,…,m; j = 1,2,3,…,n) are the parameters of the model; $g(x)$ are the transfer functions.

$$g(x) = \frac{1}{1+e^{-x}} \tag{7}$$

Transform formula (6) into:

$$y_T = f\left(y_{T-1}, y_{T-2}, \dots, y_{T-p}, \omega\right) + \varepsilon_T \tag{8}$$

Formula (8) reflects the non-linear function mapping relationship, is a parameter vector, and the function is determined by the network structure and weight parameters.

The number of neurons in the hidden layer directly affects the mapping ability of the BPNN to deal with complex problems. The number of neurons in the hidden layer is set as follows according to the empirical formula in the literature [27]:

$$n = \sqrt{p+m} + b \tag{9}$$

In the above formula, m is the number of neurons in the input layer, p is the number of neurons in the output layer. b is the adjustment constant, usually 1–10 (not limited to this range), but its final value needs to be Analyze according to specific experiments.

If the expected output of the neuron in the output layer is O_T, then the quadratic form error rule function E_T each sample is:

$$E_T = \frac{1}{2}\left(O_T - y_T\right)^2 \tag{10}$$

According to the above formula, the output error of the neurons in the output layer propagates back to the previous layers, and the weights and offset parameters between the layers are updated and corrected.

Based on the above analysis, the prediction scheme of feature information extraction step by step for network traffic can be adopted. Firstly, the advantages of ARIMA model in processing linear features in data are utilized to extract linear features in network traffic and separate them from original traffic data. Then, the SA-BPNN model is constructed to train and predict the network traffic data that only contains non-linear characteristics after separation. Finally, the linear feature data predicted by ARIMA model and the non-linear feature data predicted by SA-BPNN model are added to get the final prediction result.

4. Network traffic prediction analysis based on SA optimized ARIMA-BPNN

4.1. Construction of ARIMA-BPNN hybrid model

In the practice of network traffic forecasting, the network traffic data can be affected by many factors. Both linear and non-linear trends existed in historical data. It is possible to lead to excessive predict errors if a single ARIMA model or BPNN model are adopted to predict network traffic. In fact the ARIMA model is good at predicting the linear series data, while the BPNN is good at predicting the non-linear series data [28]. Therefore, the ARIMA model could be used to predict the network traffic at first, followed by the prediction result of the ARIMA model which contains the linear law information of the network traffic. The prediction residual of the ARIMA model contains the non-linear law information of the network traffic. Afterwards, the network traffic residuals predicted by the ARIMA model are used as the input of the BPNN, that is, the BPNN is used to predict the residuals, the non-linear law of network traffic are included in the predicted results. Finally, add the prediction results of the ARIMA model and the BPNN to obtain the predicted value of the ARIMA–BPNN hybrid model. The flowchart is shown in Fig. 2.

4.2. BPNN weights are optimized by SA

Traditional BPNN algorithm is much more sensitive to the initial weight, and can fall into the local minimum of the error easily. It is also difficult to converge to the global optimal solution, which affects the accuracy of network training [29,30], lowering the accuracy in potential performance of BPNN to predict traffic residual. Because SA algorithm has a probabilistic jumping property, that is, it can jump out of the current local optimal solution with a certain probability, and finally reach the global optimal solution. Therefore, we combined traditional BPNN with the SA algorithm, with the BPNN used as the main frame. The SA algorithm optimized the weights of the BPNN, which can improve the shortcomings of the BPNN that is easily trapped in local minimums, thereby improving the accuracy of network training. The flowchart of SA optimization BPNN is shown in Fig. 3.

In Fig. 3, $f(w)$ is the objective function.

$$f = \frac{1}{2}\sum_{k=1}^{p} \Delta {y_k}^2 \tag{11}$$

Among them, Δy_k is the error of the output value of the output layer neuron k to the corresponding expected output value.

The method set in this article to determine that the current solution is the minimum (termination condition): If it is satisfied that a value does not change for multiple consecutive iterations, it is considered to be the minimum (for example, if a value does not change for 50 consecutive iterations, it is considered to be the minimum).

The steps of calling the simulated annealing algorithm to optimize the weights of the BPNN are as follows:

Step 1 Initialize the parameters of the SA algorithm and BPNN.
SA algorithm initialization: The initial temperature is T_0, the termination temperature is a T_{min}, and the cooling rate is α, the upper limit of the number of each temperature disturbance is N.
BPNN initialization: The connection weight of each layer of the neural network is ω, error threshold is ε, the number of hidden layer neurons is n.

Step 2 The selected sample executes the following steps until the termination condition is met and then exits.

Step 3 The sample C_i uses BPNN to perform the forward propagation step, and obtains the error value $E_T(\omega)$ according to formula (10). If $E_T(\omega) < \varepsilon$, the algorithm ends, otherwise continue to perform the following steps.

Step 4 Calculate the new network connection weight according to the following formula.

$$\omega_{nev} = \omega + rand\left(1 - \frac{T_{now}}{f}\right)^K sgn(rand - 0.5)\left(\omega_{\max} - \omega_{new}\right) \tag{12}$$

Above the formula, rand is a random number in the interval (0,1), K is Boltzmann's constant, f is the maximum number of iterations, T_{now} is the current temperature, $\omega_{\max}, \omega_{\min}$ are the maximum and minimum values of BPNN connection weight respectively.

Step 5 Calculate ΔE_T by the following formula.

$$\Delta E_T = E_T\left(\omega_{new}\right) - E_T(\omega) \tag{13}$$

Step 6 Use Metropolis acceptance criteria to judge the new weight generated in step 4, and check whether ω_{new} can be accepted. If $\Delta E_T < 0$, then accept ω_{new}, otherwise, accept ω_{new} as the current solution with a certain probability p.

$$p = e^{-\frac{\Delta E_T}{KT_{now}}} \tag{14}$$

If ω_{new} is accepted, then $\omega = \omega_{new}$, and judge whether $E_T(\omega) < \varepsilon$, if not, continue to perform the following Steps, if yes, terminate the algorithm.

Step 7 For each temperature, repeat Steps 4, 5, and 6 until the equilibrium state or the upper limit N of disturbances is reached.

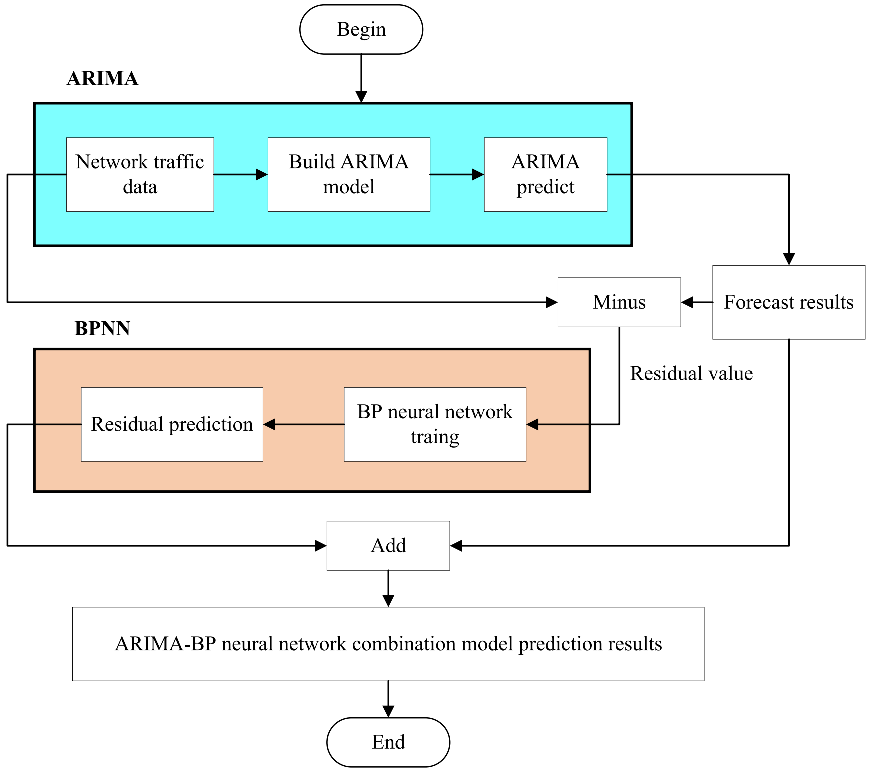


Fig. 2. Flow chart of ARIMA–BPNN hybrid model.

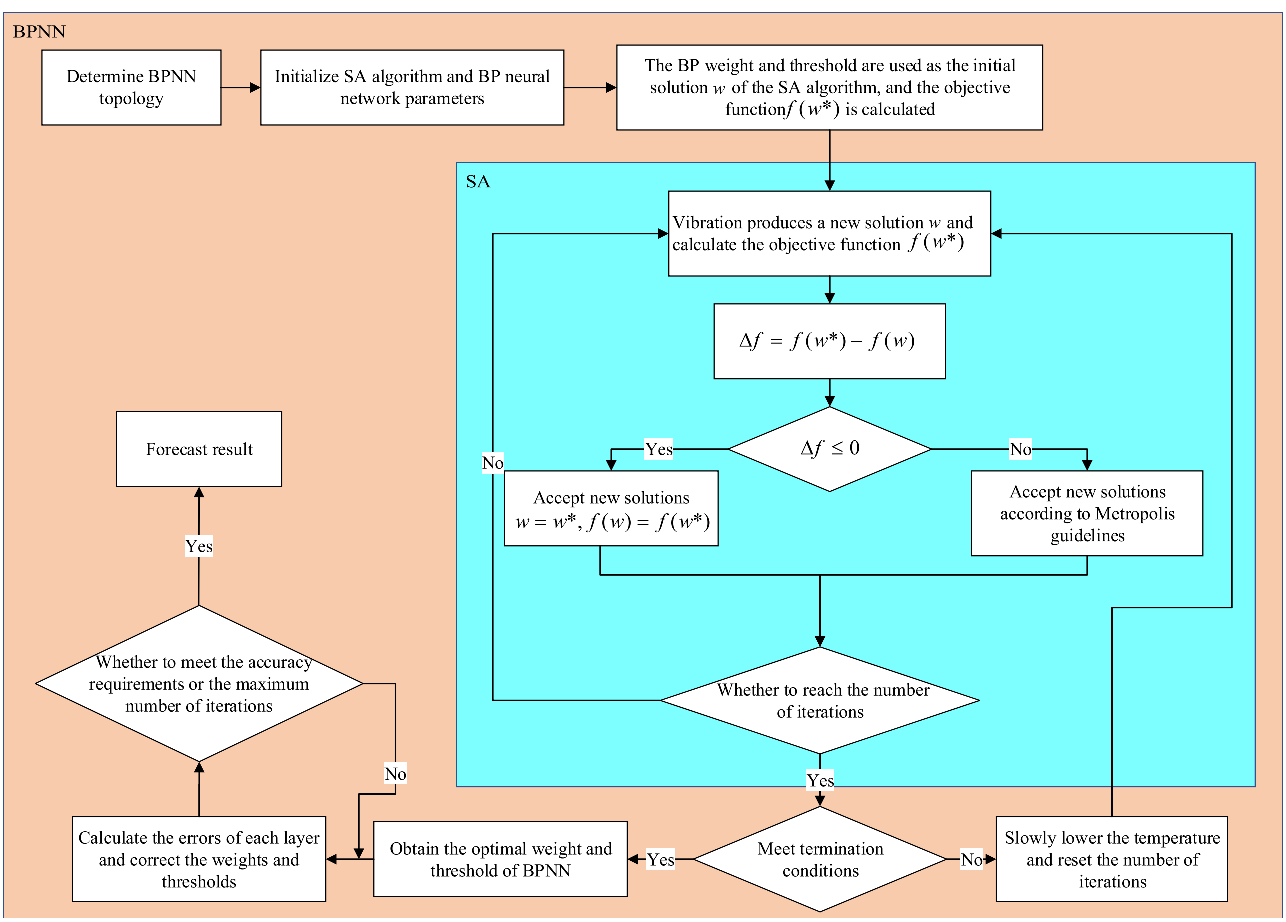


Fig. 3. Flow chart of SA optimized BPNN.

Step 8 The temperature is cooled down according to formula (14). If the temperature reaches to the termination temperature T_{min}, skip to Step 2, otherwise skip to Step 4.

$$T_{i+1} = \alpha T_i \tag{15}$$

α can take a positive number less than 1 directly, that is, decreased exponentially. At the same time, there are two other decreasing methods:

$$\begin{cases} T_{i+1} = \frac{T_i}{\log(1+M)} \\ T_{i+1} = \frac{T_i}{1+M} \end{cases} \tag{16}$$

M is the number of current iteration.

4.3. Prediction of network traffic

By constructing the ARIMA–BPNN hybrid model, the linear and non-linear factors in the historical network traffic data can be fully explored. On this basis, the SA algorithm is used to improve the network connection weight of the BPNN in the ARIMA–BPNN hybrid model. Except for the phenomenon where BPNN is trapped in the local optimal solution instead of the global optimal solution in the search process, which improves its global optimization ability, and makes the predicted result more accurate. The overall forecasting steps of the SA-optimized ARIMA–BPNN for network traffic are introduced as follows:

Step 1 Select a suitable network traffic data set, and normalize the network traffic data to converge the value of the network traffic to the interval [0,1].

$$x^* = \frac{x - x_{\min}}{x_{\max} - x_{\min}} \tag{17}$$

x^* represents the network traffic data after normalization, x_{max} represents the maximum value in the network traffic data, and x_{min} represents the minimum value in the network traffic data.

Step 2 Determine the parameters p, d and q of the ARIMA model based on the data characteristics of the network traffic, and use the ARIMA (p, d, q) model to predict the network traffic, and obtain the preliminary prediction value of the network traffic, the pseudo code is shown as below:

Algorithm 1 ARIMA

```
Require: x
Ensure: y
for i = 0; i < 6; i++ do
    if adf(x) = ture then
        x ← Diff
        break
    else
        x ← Difference(x)
        continue
p, q ← AIC(x)
y ← ARIMA(x, p, d, q)
```

Step 3 Use the real value in the network traffic data set in Step 1 to subtract the preliminary network traffic prediction value in Step 2 to obtain the residual value of the network traffic prediction.

Step 4 Determine the initial temperature, annealing rate and other parameters of the SA algorithm, initialize the weights of the BPNN and network training parameters.

Step 5 Use the SA algorithm to optimize the network weight of the BPNN to improve the global optimization ability, the pseudo code is shown as below:

Algorithm 2 SA optimization

```
Require: Rescur,Resbestcur,T0,Tend,itrnum,Mark
Ensure: Resbest
for t → T0  to  Tend do
    itrnum++
    t ← decaysacle * t
    for i → 1  to  Mark do
        p ← 0
        while p = 0 do
            Resnew ← Rescur + StepFactor * range * (rand(size(lb)) − 0.5)
            if sum(Resnew > ub) + sum(Resnew < lb) == 0 then
                p = 1
        if objfunBP(Rescur) > objfunBP(Resnew) then
            Resbestcur ← Resbest
            Resbest = Resnew
        if objfunBP(Rescur) > objfunBP(Resnew) then
            Rescur ← Resnew
            AcceptP++
        else
            changer = −(objfunBP(Resnew) + objfunBP(Rescur))/Bolt * T0
            p1 = exp(changer)
            if p1>rand then
                Rescur ← Resnew
                AcceptP++
```

Step 6 Use the SA optimized BPNN to predict the residual value of the network traffic prediction obtained in Step 3, and obtain the result of the residual prediction.

Step 7 Add the preliminary network traffic prediction value of the ARIMA (p, d, q) model in Step 2 and the residual prediction value in Step 6, the result which is the final network traffic prediction value, the pseudo code of BPNN for model training is as follows:

Algorithm 3 BPNN

```
Require: y,net
Ensure: result
net.train(net, inputn, outputn)
inputntest ← mapminmax(inputtest)
BPsim ← sim(net, inputntest)
SABPout ← mapminmax(reverse, BPsim)
result ← y + SABPout
```

Fig. 4 is the flow chart of ARIMA–SA–BPNN model for network traffic prediction.

5. Instance simulation and results analyzation

5.1. Factors involved in network traffic prediction and data set selection

In order to make the prediction of network traffic more accurate, following points discussed below should be considered before selecting the network traffic data set and predicting it.

- Consider the unique characteristics of network traffic. In general, the more historical data, the easier to obtain the change law of the data, and the more accurate the prediction result, but in fact, the network traffic sequence has different changes in different time periods, that is, periodicity [31]. Therefore, when selecting a network traffic data set, attention should be paid to the time scale and scale of the selected historical data, so as to avoid redundant information from affecting the prediction results. Considering that the upgrade speed of network technology is relatively fast and the number of network users is increasing year by year, network traffic is not suitable for long-term forecasting, and short-term forecasting should be made for it.

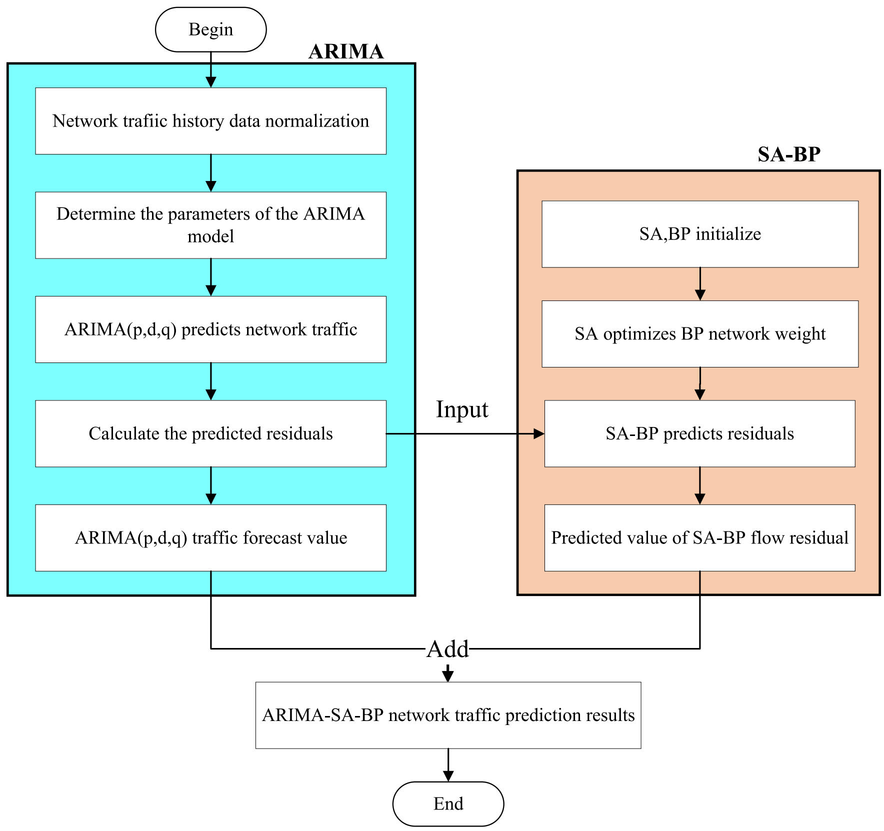


Fig. 4. The flow chart of ARIMA–SA–BPNN model prediction.

- Due to the fact that data set is divided during model training, there are two common ways to divide the data set:
  Method 1: Divide the data set into 2 independent parts. One part is used as a training set for model training, while the other one is used as a verification set to verify the fitting degree between the model prediction result and the actual data.
  Method 2: Perform training and verification on the complete data set, but it requires a different time span. The time span used for verification is one time interval ahead of the time span used for training.
- Different network traffic data sets contain different information. Hence, we should try our best to choose different network traffic data sets for network traffic prediction to make sure that the experimental results are more convincing, and the test prediction method is more feasible.
  Based on the above analysis, the text selected two different data sets in the WIDE Project (http://mawi.wide.ad.jp/mawi/) to predict network traffic. It collects network traffic from a certain backbone network from Japan to the U.S.A according to different time periods (such as a certain minute every day, every month).
  Data set 1: The sampling point is the daily tracking (samplepoint-F) of the WIDE transmission link in the upstream ISP since July 1, 2006. This article collects the historical network traffic data of the sampling point from March 1, 2020 to August 8, 2020, and uses the first 150 days of data as the training set to establish the network traffic prediction model, and the next 11 days of data as the validation set to carry out the prediction effect test.
  Data set 2: The sampling point is the weekly trace (samplepoint-G) from the main IX link of WIDE to DIX-IE. This paper collects the weekly network traffic historical data of the sampling point from January 2, 2019 to December 25, 2019, and uses the data of the first 41 weeks as the training set to establish the network traffic prediction model, and the data for the next 10 weeks is used as the validation set to verify the prediction effect.

5.2. Set the simulation parameters

The process of select the model parameters and experimental platform used in this article is as follows:

ARIMA model

Step 1 Draw a sequence diagram Fig. 5 according to the historical data of data set 1 network traffic.

Step 2 Judge the stationarity of the sequence preliminarily by observing the data traffic sequence diagram, and then use the ADF test for stationarity test. The ADF test results are shown in Table 2. The image is relatively stable, the t value of the ADF test is −4.9862715, which is less than 1% of the corresponding statistic. At the same time, p is 0.00002364, which can confirm that the time series of the data traffic is a stationary series.

Step 3 Use the autocorrelation coefficient and partial correlation coefficient to estimate the model order. The results of the data correlation detection are shown in Table 3.
Because ACF is truncated at $q = 4$ and PACF is truncated at $p = 6$, determine the model order $p = 6$, $q = 4$, and use the AIC criterion [32] to determine the order, and finally use the AIC function in python and parameters are determined to be $p = 6$, $q = 4$. So the parameters p, d, and q of the ARIMA model in this paper are 6, 0, and 4 respectively, which is the ARIMA(6,0,4) model.

$$\begin{aligned} Y_t = 465.9876 - 0.7943Y_{t-1} - 0.8010Y_{t-2} - 0.5685Y_{t-3} \\ -0.2394Y_{t-4} - 0.4710Y_{t-5} - 0.3279Y_{t-6} + \varepsilon_t \\ +0.9764\varepsilon_{t-1} + 0.9349\varepsilon_{t-2} + 0.6622\varepsilon_{t-3} + 0.0081\varepsilon_{t-4} \end{aligned} \tag{18}$$

Y_t represents the predicted value of the ARIMA(6,0,4) model for network traffic.

Step 4 Perform ARIMA model checking, and use residuals to test the quality of the model. Quantile–Quantile Plot is shown in Fig. 6.

Table 2
ADF test.

		t-Statistic	P
ADF statistical test		−4.9862715	0.00002364
Significance level	1%	−3.4744158	
	5%	−2.8808783	
	10%	−2.5770812	

It can be seen from Fig. 6 that the residuals basically satisfy the normal distribution.
By observing the ACF diagram of the residuals in Fig. 7, we can see that the sequence residuals are basically white noise. Through the two methods above, the ARIMA model is tested to determine that the model meets the requirements and can be applied to network traffic prediction.
According to the parameter selection process above, the parameters p, d, and q of the ARIMA model in this paper are $6, 0, 4$ respectively, which is the ARIMA(6,0,4) model used in data set 1 to predict the network traffic. Repeating the above steps 1 to 4, the parameters of the ARIMA model used in data set 2 are respectively 1,1,0, which is the ARIMA(1,1,0) model.

BPNN and SA algorithm

Data set 1: The network structure of the BPNN uses a 10-9-1 three-layer BP network, that is, there are 10 nodes in the input layer (10-dimensional network traffic history data), 9 nodes in the hidden layer, and 1 node in the output layer(network traffic prediction data), the weight learning rate is 0.05, and the threshold learning rate is 0.01.

Data set 2: The network structure of the BPNN uses a 4-6-1 three-layer BP network, that is, the input layer has 4 nodes, the hidden layer has 6 nodes, and the output layer has 1 node, the weight learning rate is 0.05, and the threshold learning rate is 0.01.
For the above two data sets, the initial temperature T of the SA algorithm is set to 10, the termination temperature is set to 3, and the annealing rate is set to 0.85.

Experiment platform The prediction of network traffic by ARIMA model is implemented by Python programming on the PyCharm, and the BPNN and SA algorithm are both implemented on Matlab.

5.3. Model evaluation index

In order to evaluate the accuracy of each model in the experiment for network traffic prediction, 3 indices were adopted by this paper, which were Mean Absolute Error(MAE), Root Mean Square Error(RMSE), Mean Absolute Percentage Error(MAPE). The calculation formula corresponding to each indices are as follows:

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{19}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{20}$$

$$MAPE = \frac{100\%}{n}\sum_{i=1}^{n}\left|\frac{(y_i - \hat{y}_i)}{y_i}\right| \tag{21}$$

y_i represents the true value, and $\hat{y}_i$ represents the predicted value. For the above three evaluation indices, the smaller the value, the higher the accuracy of the prediction result.

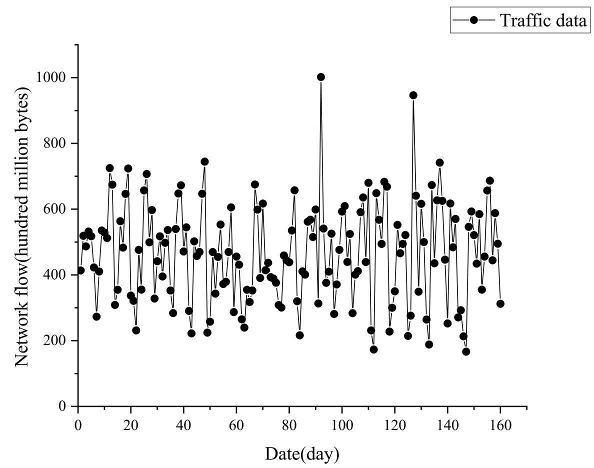


Fig. 5. Sequence diagram of historical network traffic data.

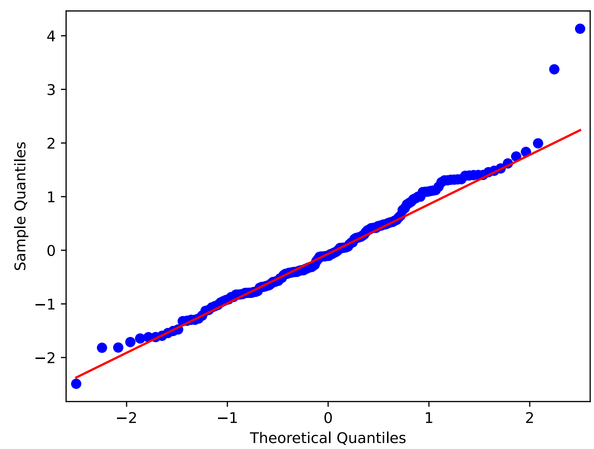


Fig. 6. The QQ diagram of the residual.

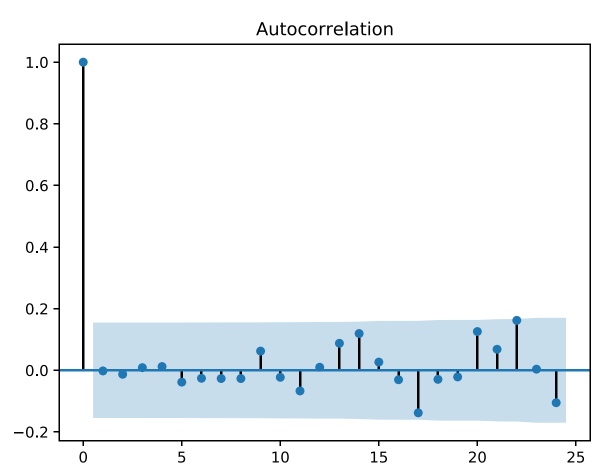


Fig. 7. The ACF diagram of the residual.

5.4. Network traffic prediction and analysis of prediction accuracy

On the basis of ARIMA, BPNN, ARIMA–BPNN and other models introduced previously, this paper selects the Wavelet Neural Network (WNN) and Recurrent Neural Network(RNN)'s variants Long Short-Term Memory(LSTM), to compare with the proposed method in this

Table 3
Data correlation detectiont.

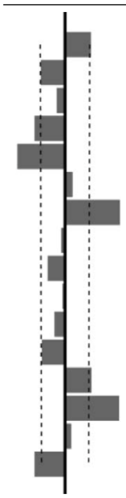
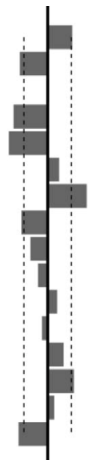
Autocorrelation	Partial correlation		AC	PAC	Q-Stat	Prob
		1	0.164	0.164	4.380	0.036
		2	−0.154	−0.185	8.252	0.016
		3	−0.053	0.008	8.716	0.033
		4	−0.194	−0.227	14.960	0.005
		5	−0.302	−0.259	30.242	0.000
		6	0.048	0.076	30.636	0.000
		7	0.348	0.261	51.143	0.000
		8	−0.024	−0.175	51.242	0.000
		9	−0.110	−0.115	53.324	0.000
		10	−0.016	−0.065	53.366	0.000
		11	−0.067	0.063	54.156	0.000
		12	−0.148	−0.038	57.967	0.000
		13	0.167	0.107	62.872	0.000
		14	0.343	0.175	83.732	0.000
		15	0.040	0.044	84.014	0.000
		16	−0.193	−0.195	90.746	0.000

Table 4
Evaluation index values of each model.

Data sets	Model	RMSE	MAE	MAPE (%)
Data set 1	ARIMA	93.4058	77.3371	14.8792
	BPNN	86.0350	69.7519	14.8408
	ARIMA–BPNN	77.4411	51.8752	10.4091
	LSTM	94.0596	76.9289	15.1091
	WNN	80.3702	59.2483	11.2013
	ARIMA–SA–BPNN	**76.9290**	**54.9551**	**9.9614**
Data set 2	ARIMA	116.6656	103.3013	2.0414
	BPNN	422.4974	346.8287	6.9984
	ARIMA–BPNN	132.7887	105.2373	1.9841
	LSTM	434.6872	319.0046	6.6365
	WNN	310.4659	265.1942	5.0780
	ARIMA–SA–BPNN	**112.7565**	**96.7524**	**1.9240**

paper. The network traffic data in the selected data set is used to complete the training and testing of the above model, and the performance of the model is compared according to the predicted results.

The results of multiple models for network traffic prediction are shown in Fig. 8. The unit of traffic is 100 000 000 bytes, and the result is 7 decimal places. According to the evaluation formula Eqs. (19), (20), (21), the model evaluation index values can be calculated and are shown in the following table (results are kept to 4 decimal places).

Table 4 shows the experimental results of the model, from which it can be seen that the ARIMA-SA-BPNN model proposed in this paper has smaller MSE, RMSE and MAE values, which shows that the predicted value of the model proposed in this paper is close to the actual network traffic value, the prediction accuracy is high, so the ARIMA-SA-BPNN model is more competitive for completing the task of real network traffic prediction.

It can be seen from Fig. 8 that the ARIMA–SA–BPNN model also has a better fitting effect when predicting a large change in network traffic.

5.5. Distribution forecast

In order to verify whether the prediction accuracy of the method proposed in this paper will fluctuate greatly with the time span, this paper uses ARIMA–SA–BPNN to predict network traffic again by changing the prediction step size. Since the network traffic selected in this article is based on the number of days(or weeks), single-step prediction means that the network traffic of the next day (second week) can be predicted, and N-step prediction means that the next N days (N Network traffic within a week). The change in model prediction accuracy obtained by changing the step size is shown in Fig. 9. In particular, the value of MAPE retains 4 decimal places.

Table 5
Evaluation index values.

	One step	Two steps	Four steps	Six steps	Eight steps	Ten steps
Data set 1	0.6750	2.5131	6.8853	7.4137	9.7071	9.9614
Data set 2	0.1461	0.2980	1.0901	1.1335	1.7088	1.9240

Under different step size, the MAPE variation range predicted by ARIMA–SA–BPNN models of the two data sets is shown in Table 5.

It can be seen from Fig. 9 that as the step size increases, the MAPE value predicted by the ARIMA–SA–BPNN model gradually increases, but the increase in MAPE is small, indicating that the increase in model prediction error is small, so the model can be better Describing the trend of phased changes in network traffic has certain practical application value.

6. Future direction

In future research, we will consider the impact of network users' online behavior characteristics with other random factors that are difficult to quantify on network traffic prediction, and further explore the characteristics of network traffic. We will also establish a more accurate network traffic prediction model, and explore an SDN system that can automatically open or terminate the virtual network function according to the predicted future network traffic change trend.

7. Conclusion

Network traffic possesses the features of long range dependence, non-linearity, time-varying, weak coupling, etc. When predicting the network traffic using traditional single model and hybrid model, prediction results have a rather low accuracy as the network traffic trends cannot be described objectively. Hence, a novel network traffic prediction method based on SA optimized ARIMA–BPNN is proposed. Our method utilizes the SA algorithm to optimize the network weights of the BPNN in the ARIMA–BPNN hybrid model, and improves its

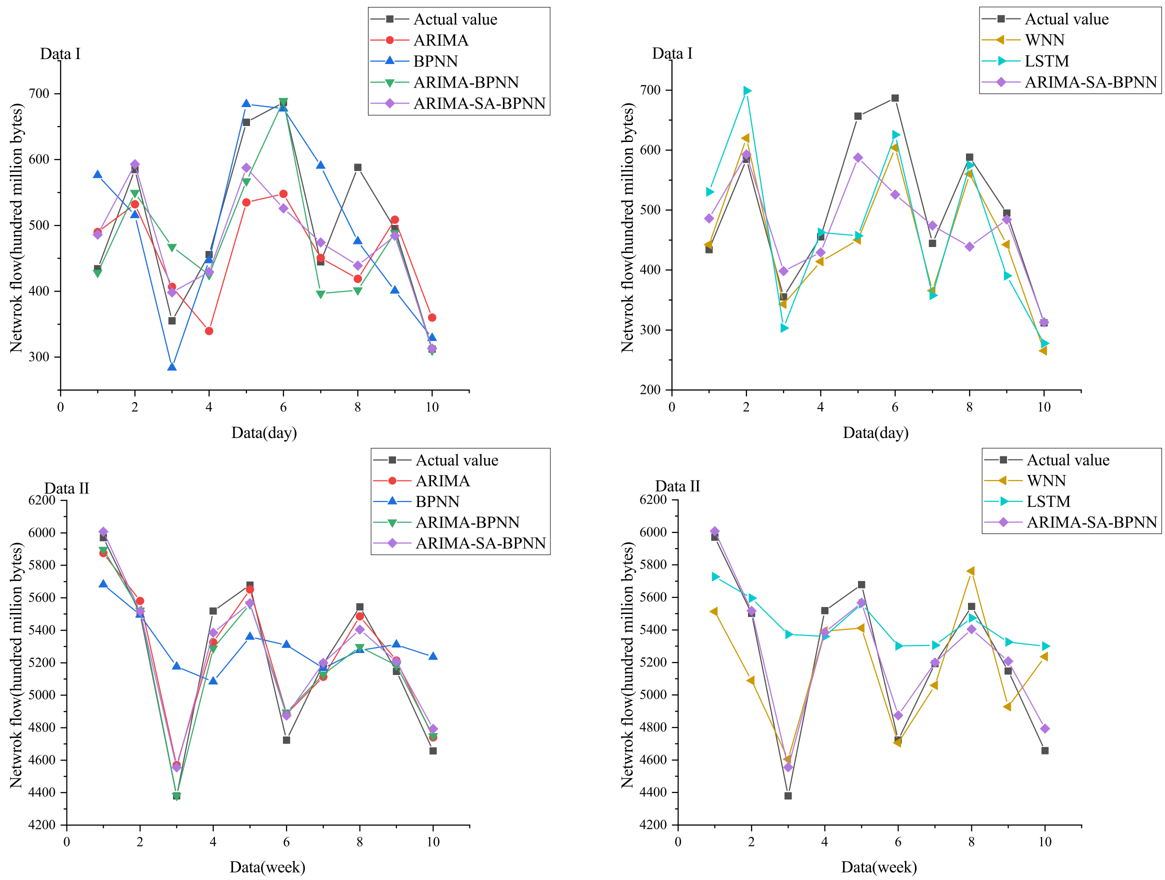
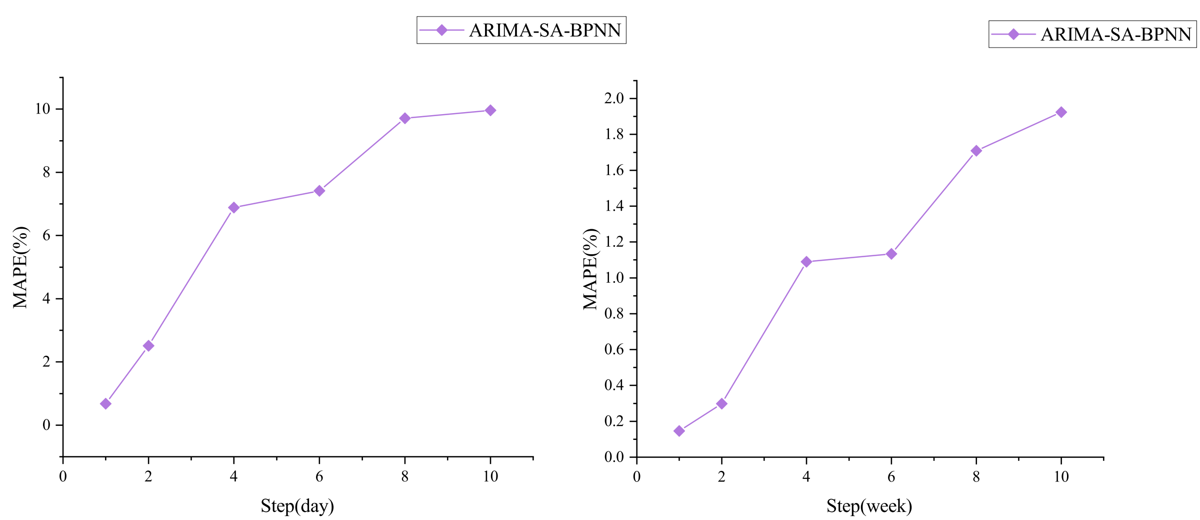


Fig. 8. Network traffic prediction results.



(a) Data I (b) Data II

Fig. 9. Changes of MAPE under different step size.

global optimization ability. This paper uses the same network traffic data set and compares simulation experiments with other network traffic prediction models to verify the superiority of the network traffic prediction method based on SA optimized ARIMA–BPNN, and obtain appealing network traffic prediction results. Therefore, our prediction method can be promoted and applied, and it can play an important role in the fields of equipment fault diagnosis, economic prediction, and non-linear prediction with more complexity.

CRediT authorship contribution statement

Hanyu Yang: Conceptualization, Methodology, Software, Writing - original draft. **Xutao Li:** Software, Validation, Investigation, Visualization, Writing - original draft. **Wenhao Qiang:** Writing - original draft. **Yuhan Zhao:** Writing - review & editing. **Wei Zhang:** Supervision, Project administration, Funding acquisition. **Chang Tang:** Supervision, Project administration, Funding acquisition.

Declaration of competing interest

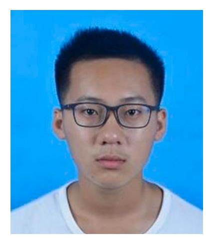
The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

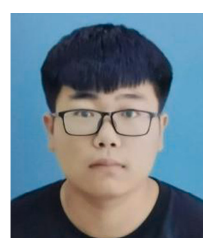
This work was supported by the National Natural Science Foundation of China (No.61802233, 62076228), Natural Science Foundation of Hubei Province (No.2020CFB644) and Natural Science Foundation of Shandong Province (No.ZR2020LZH010). Wei Zhang and Chang Tang are the corresponding authors.

References

[1] Y. Yang, A new network traffic prediction approach in software defined networks, Mob. Netw. Appl. (2019) 1–10.
[2] N. Krishnaswamy, M. Kiran, K. Singh, B.B.T. Mohammed, HPDC '20: The 29th International Symposium on High-Performance Parallel and Distributed Computing, Data-driven learning to predict WAN network traffic, 2020, pp. 11–18.
[3] G. Zhang, D.B.T. Huang, International Conference on Intelligent Networking and Collaborative Systems, Short-term network traffic prediction with ACD and particle filter, 2013, pp. 189–191.
[4] Y. Zhou, Y. Zhang, C. Ma, S. Chen, O.O. Odegbile, Generalized sketch families for network traffic measurement, Proc. ACM Meas. Anal. Comput. Syst. 3 (3) (2019) 1–34.
[5] K. Liu, Chaotic prediction of network traffic based on particle swarm algorithm optimization and support vector machine, Modern Electron. Tech. 42 (02) (2019) 120–123.
[6] C. Liu, Prediction of network traffic based on hybrid function Gaussian process, Laser J. 35 (11) (2014) 105–108.
[7] J. Fan, D. Mu, Y. Liu, Research on network traffic prediction model based on neural network, in: 2019 2nd International Conference on Information Systems and Computer Aided Education (ICISCAE), 2019, pp. 554–557.
[8] Z.Y. Tian, Theory and application of time series analysis, Higher Education Press, 2003, pp. 214–246.
[9] Z. Wu, H. Li, L. Liu, J. Zhang, M. yue, J. Lei, Detection of ldos attacks based on wavelet energy entropy and hidden semi-Markov models, J. Softw. v.31 (05) (2020) 313–326.
[10] M. Yuan, Jitter buffer control algorithm and simulation based on network traffic prediction, Int. J. Wirel. Inform. Netw. 26 (3) (2019) 133–142.
[11] Q. Chen, W. Guo, C. Li, Railway passenger volume forecast by GA-sa-BP neural network, in: 2009 International Workshop on Intelligent Systems and Applications, 2009, pp. 1–4.
[12] M. Saadatpour, A. Afshar, H. Khoshkam, S. Prakash, Equilibrium strategy based waste load allocation using simulated annealing optimization algorithm, Environ. Monit. Assess. 192 (9) (2020) 5875–5888.
[13] H. Li, Network traffic prediction of the optimized BP neural network based on Glowworm Swarm Algorithm.
[14] Y. Hou, L. Zhao, H. Lu, Fuzzy neural network optimization and network traffic forecasting based on improved differential evolution, Future Gener. Comput. Syst. 81 (2018) 425–432.
[15] H. Yang, Wavelet neural network with SOA based on dynamic adaptive search step size for network traffic prediction, Optik 224 (2020) 165322.
[16] N. Ramakrishnan, T. Soni, Network traffic prediction using recurrent neural networks, in: 2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA), 2018, pp. 187–193.
[17] Q.Y.W. Shihao, Z. Qinzheng, Y. Han, L.I. Qianmu, A network traffic prediction method based on LSTM, ZTE Commun. 17 (02) (2019) 19–25.
[18] Y. Tokuyama, R. Miki, Y. Fukushima, Y. Tarutani, T.B.T. Yokohira, ICIET 2020: 2020 8th International Conference on Information and Education Technology, Performance evaluation of feature encoding methods in network traffic prediction using recurrent neural networks, 2020, pp. 279–283.
[19] C.-W. Huang, C.-T. Chiang, Q. Li, A study of deep learning networks on mobile traffic forecasting, in: 2017 IEEE 28th Annual International Symposium on Personal, Indoor, and Mobile Radio Communications (PIMRC), IEEE, 2017, pp. 1–6.
[20] M. Li, Y. Wang, Z. Wang, H. Zheng, A deep learning method based on an attention mechanism for wireless network traffic prediction, Ad Hoc Netw. 107 (2020) 102258.
[21] M. Gao, C. Wu, B. Li, B. Sheng, Y. Huang, Design and implementation of a placido disk-based corneal topographer optical system based on aberration theory and simulated annealing algorithm, Opt. Commun. 475 (2020) 126097.
[22] J. Zhang, Deformation monitoring data prediction method based on simulated annealing optimized BP neural network, Meas. Control Technol. (2020) 1–6.
[23] S. Singh, K.S. Parmar, S.J.S. Makkhan, J. Kaur, S. Peshoria, J. Kumar, Study of ARIMA and least square support vector machine (LS-SVM) models for the prediction of SARS-CoV-2 confirmed cases in the most affected countries, Chaos Solitons Fractals 139 (2020) 110086.
[24] F. Wang, Y. Zou, H. Zhang, H. Shi, House price prediction approach based on deep learning and arima model, in: 2019 IEEE 7th International Conference on Computer Science and Network Technology (ICCSNT), 2019, pp. 303–307.
[25] Z.-J. Wang, S.-S. Liu, L. Feng, Y.-Q. Xu, BNNmix: A new approach for predicting the mixture toxicity of multiple components based on the back-propagation neural network, Sci. Total Environ. 738 (2020) 140317.
[26] A. Dalgkitsis, M. Louta, G.T.B.T. Karetsos, the 22nd Pan-Hellenic Conference, Traffic forecasting in cellular networks using the LSTM RNN, New York, pp. 28–33.
[27] H. Shen, Z. Wang, C. Gao, J. Qin, F. Yao, W. Xu, Determining the number of BP neural network hidden layer units (Ph.D. thesis), 2008.
[28] J. Zhao, Combining ARIMA model and BP neural network to forecast coer volume (Ph.D. thesis), Donghua University, 2016.
[29] Z. Han, L. Hong, J. Meng, Y. Li, Q. Gao, Temperature drift modeling and compensation of capacitive accelerometer based on AGA-BP neural network, Measurement 164 (2020) 108019.
[30] M. Zou, L. Xue, H. Gai, Z. Dang, S. Wang, P. Xu, Identification of the shear parameters for lunar regolith based on a GA-bp neural network, J. Terramech. 89 (2020) 21–29.
[31] F.Y.S. Du, Z. Xu, M.A. Tao, A survey of network traffic prediction based on neural network models, Radio Commun. Technol. 46 (02) (2020) 216–222.
[32] H. Akaike, A new look at the statistical model identification, IEEE Trans. Automat. Control 19 (6) (1974) 716–723.

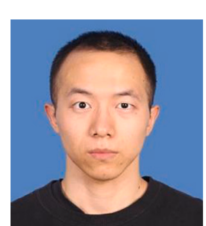


Hanyu Yang is currently studying for a bachelor's degree at the School of Cyberspace Security, Qilu University of Technology (Shandong Academy of Sciences). His research interests include data mining and deep learning.



Xutao Li is currently studying for a bachelor's degree at the School of Cyberspace Security, Qilu University of Technology (Shandong Academy of Sciences). His research interests include network security and data mining.

Wenhao Qiang is currently studying for a bachelor's degree at the School of Cyberspace Security, Qilu University of Technology (Shandong Academy of Sciences). His research interests include network security and mobile computing.



Yuhan Zhao received the B.E. degree in Cyber Security and Law Enforcement from Jiangsu Police Institute in 2018. He is currently pursuing the M.E. degree in computer application at the Qilu University of Technology (Shandong Academy of Sciences), Jinan, Shandong Province, P.R. China.

His research interests include fog computing, 5G network architecture, and Internet of Things.

Wei Zhang received the B.E. degree from Zhejiang University in 2004, the M.S. degree from Liaoning University in 2008, and the Ph.D. degree from Shandong University of Science and Technology in 2018. He is currently an Associate Professor with the Shandong Computer Science Center (National Supercomputer Center in Jinan), Qilu University of Technology (Shandong Academy of Sciences). His research interests include future generation network architectures, edge computing and edge intelligence.

Chang Tang received his Ph.D. degree from Tianjin University, Tianjin, China in 2016. He joined the AMRL Lab of the University of Wollongong between Sep. 2014 and Sep. 2015. He is now a professor at the School of Computer Science, China University of Geosciences, Wuhan, China. Dr. Tang has published 50+ peer-reviewed papers, including those in highly regarded journals and conferences such as IEEE T-PAMI, IEEE T-MM, IEEE T-KDE, IEEE T-HMS, ICCV, CVPR, IJCAI, AAAI and ACM MM, etc. He serves as an associate editor of BioMed Research International and young editor of CAAI Transactions on Intelligence Technology and Computer Engineering. He often serves on the Technical Program Committees of some top conferences such as NIPS, CVPR, ICCV, ECCV, IJCAI, ICME and AAAI. His current research interests include machine learning and computer vision.