

IMPROVING THE EFFICIENCY OF TIME SERIES FORECASTING BY INTEGRATING ARTIFICIAL NEURAL NETWORKS AND GENETIC ALGORITHMS

Vo Viet Minh Nhat

Corresponding author, Hue University, Hue, Vietnam
vymnhat@hueuni.edu.vn

Le Van Hoa, Nguyen Thi Thuy Van and Hoang Thi Hue

Co-authors, School of Hospitality & Tourism, Hue University, Hue, Vietnam
levanhoah@hueuni.edu.vn, ngththvan@hueuni.edu.vn, hthue.kdl@hueuni.edu.vn

Abstract

Artificial neural networks (ANN) are known to be an effective forecasting method for time series data. However, a challenge of this method is how to determine an optimal set of parameters for ANN, such as number of inputs, number of hidden neurons, learning rate, set of weights, etc. so that ANN can achieve the most accurate forecasting results. Given the possible combinations of parameters, there is an explosion in the number of ANN architectures that need to be considered, so it is necessary to incorporate another intelligent technique, such as genetic algorithms (GA), to be able to find the best ANN. This paper analyzes the integration of ANN with GA, called the GANN model, in which ANNs are candidate solutions in the search space and GA tries to determine the best architecture for forecasting. Time series data on tourism demand is used to evaluate the effectiveness of the GANN model. Experimental results show that, ANN(12:14:1) is the best ANN architecture, with the minimum forecasting error (MAPE = 5.50). A comparison between ANN(12:14:1) and other architectures with the same inputs, but varied hidden neurons was also investigated and the results show that ANN(12:14:1) is the global optimal architecture.

Keywords: Forecasting; Time series data; Artificial neural network; Genetic algorithm; Tourism demand.

1. Introduction

A time series is a set of observations measured sequentially over time. Depending on the measurement, the time series can be discrete or continuous [1]. Time series forecasting is the process of forecasting future values based solely on past values. Based on the number of time series involved in the forecasting process, time series forecasting can be univariate (which is based on only one time series) or multivariate (which depends directly or indirectly on multiple time series). However regardless of its type, time series forecasting has become an important methods for decision-making, as it has been successfully applied in many fields such as economics, finance, management, engineering, and so on.

Traditionally, time series forecasting is often performed using statistical methods [2]. The main limitation of most statistical methods is that they consider the time series generated from a linear process. However, most time series generated in the real world often contain nonlinear variations in time (and possibly in space). Therefore, several intelligent computational approaches are used for time series forecasting. Among those intelligent methods, artificial neural networks (ANNs) are the most prominent and widely used because of some of its unique features. First, ANNs are data-driven self-correcting nonlinear methods that do not require prior specific assumptions about the underlying model. Second, ANNs have the ability to derive “relationships” between the “inputs” and “outputs” of a training dataset, which means they can self-learn. Finally, ANN is an universal approximation method that can approximate any nonlinear function with any fairly high degree of accuracy, making it applicable to complex models [3].

Due to the above characteristics, ANN has been widely used in time series forecasting. However, a dilemma is how to design an appropriate ANN architecture for a particular time series [4]. For example, what type of ANN (e.g. a multilayer Perceptron) will be chosen for a given forecasting issue; which architecture (number of inputs, number of hidden neurons, number of outputs, topology, matrix of weights, etc.) and which learning algorithm (e.g. back-propagation) are appropriate for a given time series dataset. However, even with a specific type of ANN is identified; different combinations of parameters can create an explosion in the number of ANN

architectures to be investigated. Finding an optimal architecture in the space of candidate solutions is clearly a challenge, and genetic algorithm (GA) has been proven the most suitable approach to solve this type of problem

This paper analyzes an ANN and GA integration model (briefly, GANN model), in which the ANN parameters are integrated into the GA chromosome. In the space of candidate ANNs, GA searches (using genetic techniques such as selection, crossover, mutation, etc.) for the best ANN. In order to evaluate the effectiveness of GANN model, a time series dataset on tourism demand is implemented.

The main contributions of the paper include:

- Modeling the time series forecasting problem into the problem of finding the optimal ANN architecture by using GA; An ANN and GA integration (GANN) model is then proposed;
- Analyze, normalize and convert time series data into input datasets for GANN model; and
- Implement and evaluate the effectiveness of GANN model with a time series dataset on tourism demand.

The next contents of the paper include: Section 2 reviews and evaluates related works, which focus on models of integrating ANN and GA in different forecasting problems, including tourism demand forecasting. Section 3 describes in detail the ANN and GA integration model in which the steps to search for the optimal ANN are analyzed. Section 4 presents simulation results and analysis; the conclusion is in Section 5

2. Related works

Based on previous studies, there are two different ways of combining ANN and GA: support and collaboration. In support combination, ANN and GA are applied sequentially, where GA is mainly used to remove some nonconforming features of ANN. With collaborative combination (or integration), ANN and GA are applied simultaneously where GA is used to select the best parameter combination of ANN. This paper mainly focuses on the integrated approach.

Rahman and Setu [5] studied a collaborative approach based on combining GA with ANN for sonar image classification problem. To integrate GA and ANN into a single system, a set of ANNs is developed and the goal of the system is to find the optimal ANN. In this cooperative system, GA and ANN operate in parallel. Research results show that GA reduces training error after each iteration and is optimal over 100 generations.

Inthachot et al. [6] developed a model of combining ANN with GA to forecast the movement of the stock index SET50 and tested it on a large set of past stock trading data. The aim is to achieve better forecasting accuracy than the previous ANN model developed by these authors. Research results show that the combination model achieved an average improvement of 12.4011%, with an average forecasting accuracy of 63.60%.

Jeeananunta and Abeyrathna [7] propose a combination of ANN with GA to forecast electricity demand in short term. ANNs with 192 inputs and 48 outputs are trained by a model of combining ANN with GA, the results are compared with a backpropagation neural network (BPNN). The research results show that the model of combining ANN with GA takes longer (about 30 minutes) than BPNN (about 1 minute) to train and forecast demand in a day, but the forecast results of combination model is better. Therefore, these two models can be used based on demand: if time is important then it is better to use BPNN and if accuracy is more important than time then the combination model ANN with GA is the best way.

In the field of tourism, there have also been some studies integrating GA and ANN in tourism demand time series forecasting. Specifically, Noersasongko et al. [8] studied how to optimize the parameters of BPNN using GA. With the dataset on tourist arrivals to three cities Magelang, Surakarta and Wonosobo from 2010 to 2013, the results show that there is a similarity between the integrated model's forecast and the actual data obtained from the Java Center of Tourism Statistics, Indonesia. The root mean square error (RMSE) obtained in the integrated model is better than using BPNN alone. The research results also show additional attributes affecting the number of tourists visiting Central Java, including: number of tourist attractions, number of guests staying at hotels or motels, number of tourists arriving at airport, etc.

Huang and Hou [9] studied the application of ANN in combination with GA to establish a forecast model of air ticket sales. GA is used to determine the optimal number of inputs and hidden neurons of a feedforward ANN. The experimental results show that the mean relative error (MARE) of the combined model's forecasting and the actual value is 10.51% and the correlation coefficient is 0.913. This model clearly has good predictability and can provide travel agency operators with reliable and highly efficient analytical data.

Abas and Lasarudin [10] studied the forecast of domestic and foreign tourists to Gorontalo by using ANN combined with GA. By incorporating the parameters determined by GA in training ANN, the analysis results show that the optimization performed by GA can reduce RMSE. The study also proves that the model of combining ANN with GA will help predict more accurately than the case without using GA.

Wang et al. [11] built a tourism demand forecasting model based on BPNN and GA. The obtained experimental results include: the relative error of the number of domestic and international tourists is 12.23% and 1.84%; and that of income from tourism of these two groups of tourists is 4.95% and 1.82%. The research results also show that the tourism demand forecasting model based on BPNN and GA has good adaptability to the installation scenarios. At the same time, it provides a smart and efficient method in forecasting the demand

for high-speed rail service of tourists, contributing to orientation and decision-making support for the development of this type of transport service.

Obviously, in the above studies, the designed ANN with GA integration models are specific to each given time series dataset and no study considers all the ANN parameters simultaneously. Some studies focus only on the number of inputs and hidden neurons, while others focus on training weights and learning rates. The construction of an ANN and GA integration model depends on some forecasting need. Therefore, the optimal selection of parameters is an open problem that the researchers need to choose correctly to deploy for each specific case.

3. Integrated model of ANN and GA

3.1. Background

An artificial neural network is a computational model that simulates the way biological neural networks work. It is made up of a collection of processing units (neurons) that communicate with each other by sending analog signals. These signals travel through weighted links between neurons. Each neuron sums the weighted inputs and generates the output through an activation function (Fig. 1). This output can be the input of the neurons of the next layer or can be the final output of the neural network [12].

The general structure of a neural network, specifically considered here a multilayer perceptron (MLP), is depicted in Figure 2, where the inputs are responsible for receiving the features of the input pattern, the neurons of hidden layer only connects to the inputs and outputs without any interaction with external environment, and the outputs produce results after performing the calculations. The neurons of one layer all have weighted connections to the neurons of the next layer (Fig. 2), whose matrix of weights defines the “relationship” between “inputs” and “outputs” of the training dataset.

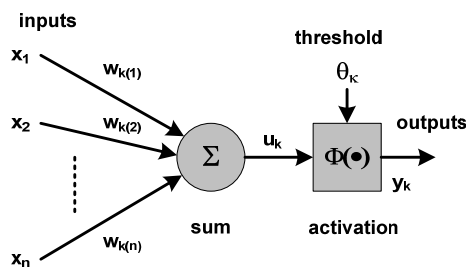


Fig. 1. Model of single neuron

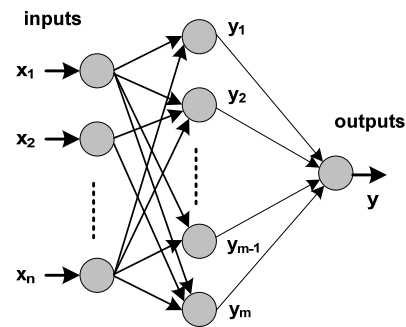


Fig. 2. Structure of a 3-layer MLP

Genetic algorithm (GA) is a type of search algorithm that simulates the evolution of biological populations by applying genetic techniques. Based on the theory of genetics and survival of organisms in their environment, GA has been developed as shown in Fig. 3, where genetic techniques (including selection of individuals for reproduction, crossover, mutation, and creation of population for the next generation) is implemented at each cycle (or generation) with the goal that the individuals in the next generation are always better or at least equal to the individuals in previous generations. After a number of generations or meeting a certain stopping condition, the genetic algorithm will converge and the best individual is found [13].

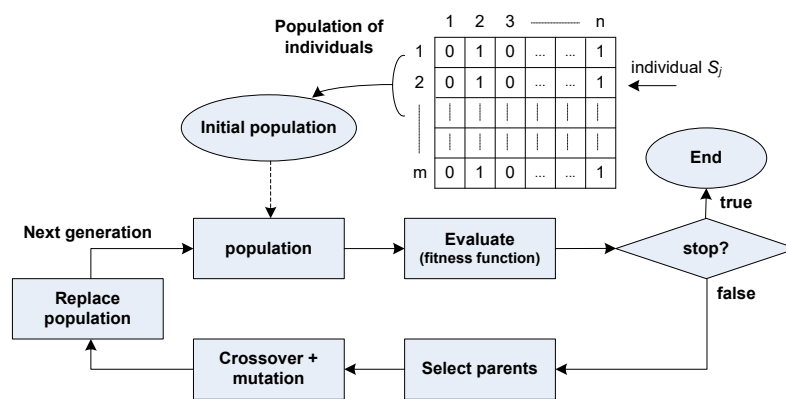


Fig. 3. Schema of genetic algorithm

3.2. Proposed integration model

ANNs have some limitations such as long training time, unwanted convergence to a local rather than global optimal solution, and large number of parameters. Therefore, there have been attempts to overcome these limitations by combining ANN with another method. A method commonly combined with ANNs is GA, where GA is used to find the best combination of ANN's parameters.

In the ANN and GA integration (GANN) model proposed in this paper, each ANN is considered as a candidate solution in the space of all possible ANNs. Each ANN in the phenotype space is mapped to a chromosome in the genotype space on which GA searches to determine the best chromosome. From there, the corresponding best ANN is determined. GANN model thus performs through 10 operational steps as follows (Fig. 4).

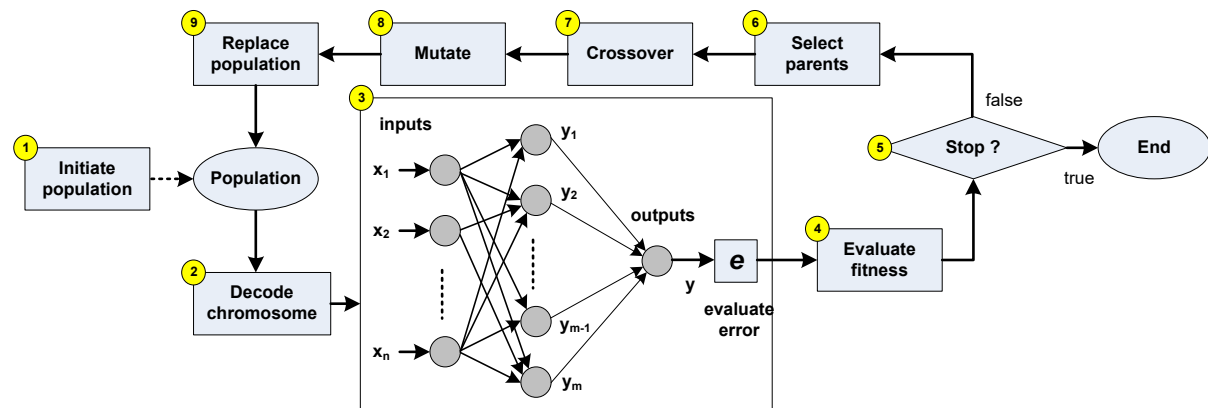


Fig. 4. Operational steps of GANN model

Step 1. (Initiate population): Create a population of initial chromosomes that are gene sequences with randomly generated values.

Step 2. (Decode chromosome): Decode the chromosome to determine which parameters should be selected for constructing ANN.

Step 3. (Run ANN): Run the ANN with the corresponding prepared training dataset. The result of ANN running is the minimum forecasting error value (or maximum forecasting accuracy).

Step 4. (Evaluate fitness): Calculate the chromosome fitness based on the obtained forecasting error.

Step 5. (Stop training). Determine whether to continue or stop GA. The stopping criterion can be the maximum number of generations or the threshold of forecasting error (or accuracy).

Step 6. (Select parents): Select parents to crossover with a certain selected crossover technique. The selection techniques can be roulette wheel, ranking, tournament etc.

Step 7. (Crossover): Apply a crossover technique to combine the two chromosomes linearly. Crossover techniques can be single-point, multi-point, uniform crossover etc.

Step 8. (Mutate): Insert new genes into one of several chromosomes in population using a selected mutation technique. Types of mutation can include single-bit or multi-bit flip.

Step 9. (Replace population). Replace parent chromosomes with better offspring chromosomes for the next generation. The replacement can be complete, partial or by elite individuals.

Step 10. (Repeat). Go back to Step 2.

4. Experimental design

GANN model is a combination of the advantages of both ANN and GA techniques. Following are the experimental design steps of GANN model for a time series data of tourism demand.

4.1. Data collection

The first step in designing a tourism demand forecasting model is to collect data. Tourism demand data, specifically the number of tourists to Thua Thien Hue from January 2017 to December 2019, is collected from the website of the Department of Tourism of Thua Thien Hue province, Vietnam. This is statistical data on the number of monthly tourists to Thua Thien Hue, Vietnam and has not been normalized (see Table 1).

Year	2017				2018	2019			
Month	1	2	...	12	...	1	2	...	12
Num of tourists	229.800	265.934	...	327.095	...	392.611	407.719	...	370.628

Table 1. Raw data on tourists to Thua Thien Hue, Vietnam from 1/2017 to 12/2019.

4.2. Data normalization

ANNs have improved performance when the data has been normalized. Using the original (unnormalized) data for training can cause unwanted convergence, such as a local minimum [2]. Thus, all data has been converted to values from 0 to 1 by dividing the difference of the actual value $x(t)$ and the minimum value x_{min} by the difference of the maximum value x_{max} and the minimum value x_{min} of the normalized data in (1).

$$x_{norm} = \frac{x(t) - x_{min}}{x_{max} - x_{min}} \tag{1}$$

The purpose of data normalization, combined with weight initialization, is to allow the activation function to work at least at beginning of the training phase. Therefore, the derivative (decreasing gradient) of the nonlinear function is always non-zero. When training is finished, the output is converted to the original data format. With the normalization, the normalized data is shown in Table 2.

Year	2017				2018	2019			
Month	1	2	...	12	...	1	2	...	12
Num of tourists (normalized)	0,142256	0,305165	...	0,383041	...	0,172897	0,640971	...	0,44322

Table 2. Normalized data on tourists to Thua Thien Hue, Vietnam from 1/2017 to 12/2019.

4.3. Size of the sliding window

The work of an ANN is focused on forecasting the future evolution of a time series from the values of x up to the present time. Formally, this can be stated as finding a function $f(x): N \rightarrow R$ to get the forecasting of x at time $t + 1$, from N steps back in time from time t , such that:

$$x(t + 1) = f(x(t), x(t - 1), \dots, x(t - N + 1)) = f(y(t)) \tag{2}$$

where $y(t)$ is the N -dimensional vector of backward values.

The ANN method in time series forecasting can be viewed as a function that converts a set of N input values into a single output (target). As depicted in Fig. 5, the used ANN architecture has as inputs the set of x values in a sliding window and the target is the next value of x .

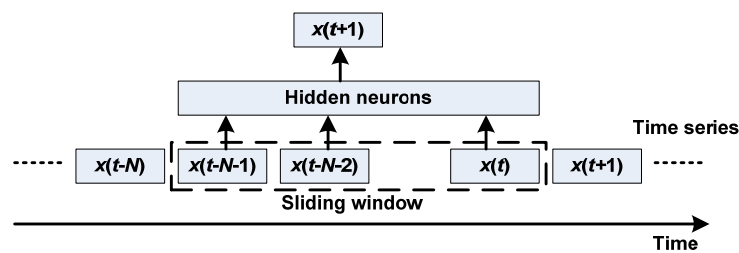


Fig. 5. Operation steps of GANN model

This method is called sliding window technique with inputs as a set of N values x that can slide over the entire training dataset. Given the seasonality of tourism, the sliding window size needs to be large enough, at least 6 months. Therefore, the sliding window size for the tourism demand forecasting problem here is chosen as $N = 6, 8, 10$ and 12 .

4.4. Running GANN

GANN model is the result of the integration of ANN and GA, where ANNs are the individuals in the phenotype space for which GA searches to determine the best ANN. GA's search is done over generations. Fig. 7 depicts one execution cycle of GANN.

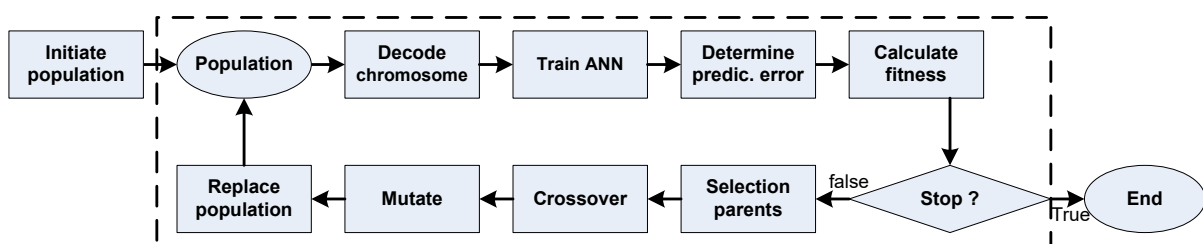


Fig. 6. One execution cycle of GANN

a. Chromosome representation

The chromosome representation determines how the genetic algorithm works. In the problem of tourism demand forecasting, the number of inputs and hidden neurons play an important role in designing ANN, where the number of inputs is the sliding window size and the number of hidden neurons determine the classification of ANN. If the number of hidden neurons is high, it may lead to overfitting, while if this number is small, the separation between classes becomes unclear and therefore the forecasting accuracy may be reduced. In GANN model, the number of inputs and hidden neurons is encoded into binary sequences in chromosomes. Specifically, with 4 sliding window sizes of 6, 8, 10 and 12, only 2 bits are needed to represent these 4 window sizes, where 00, 01, 10 and 11 correspond to 6, 8, 10 and 12 inputs, respectively. The reason for considering only a few cases with a discrete number of inputs is that it is necessary to prepare different training datasets corresponding to the number of inputs, as shown in Fig. 7.

N = 6	N = 8	N = 10	N = 12
X ₁ , X ₂ , ..., X ₆	X ₁ , X ₂ , ..., X ₈	X ₁ , X ₂ , ..., X ₁₀	X ₁ , X ₂ , ..., X ₁₂
X ₂ , X ₃ , ..., X ₇	X ₂ , X ₃ , ..., X ₉	X ₂ , X ₃ , ..., X ₁₁	X ₂ , X ₃ , ..., X ₁₃
...
X ₃₁ , X ₃₂ , ..., X ₃₆	X ₂₉ , X ₃₀ , ..., X ₃₆	X ₂₇ , X ₂₈ , ..., X ₃₆	X ₂₅ , X ₂₆ , ..., X ₃₆

Fig. 7. Different prepared training datasets corresponding to 6, 8, 10 and 12 inputs

Based on the determined number of inputs, the number of hidden neurons is chosen from 3 to 18, as recommended in [11]. The purpose of choosing this range of hidden neurons is to be able to represent 16 (2⁴) different states with 4 bits, where 0000, 0001, 0010, ..., 1111 corresponds to 3, 4, 5, ..., 18 hidden neurons, respectively. The result is a chromosome that encodes for an ANN with 10 inputs and 11 hidden neurons represented by the binary sequence 101011 (Fig. 8).

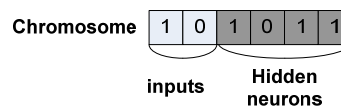


Fig. 8. An example of chromosome representation

b. Population Initialization

In the first step of GANN model, two problems need to be solved: determining the population size and initializing the population. Population size is the number of chromosomes present in a population at any given time. Population size depends on the number of possible chromosomes in the phenotype space. Determining the population size affects the rate of convergence and the computational cost per generation. In GANN model, the population size is chosen to be 5% of the possible chromosomes in the phenotype space [13]. The initial population can be randomly generated or directed. In randomization, the alleles of chromosomes need to be initialized so that the chromosomes are evenly distributed in the genotype space. This even distribution means that GA can find the global, instead of only local optimal. With directed initialization, the initialization is usually based on some known good solution. In GANN model, random initialization is preferred.

c. Decoding Chromosomes

The chromosome must be decoded into the corresponding ANN to perform the training. With the chromosomal structure where the first 2 bits are for inputs and the next 4 bits for hidden neurons, the corresponding ANN is easily constructed so that it can be trained by a prepared dataset. The ANNs considered here all have only one output, so the ANN corresponding to a particular chromosome, such as 101011, will have 10 inputs and 11 hidden neurons which are denoted as ANN(10:11:1).

d. Training ANN and calculating the forecasting error

Each constructed ANN is trained with a corresponding dataset. The ANN considered here is a 2-layer MLP and the chosen learning algorithm is backpropagation. The considered error function is mean absolute percentage error (MAPE) and the training objective is to minimize this error function. After a finite number of training rounds or MAPE reaches a given minimum threshold, training is stopped.

MAPE is defined as follows:

$$MAPE = \frac{100}{p} \sum_{t=1}^p \frac{|y(t) - \hat{y}(t)|}{|y(t)|} \tag{3}$$

individuals and the individuals that can create the diversity for the next generation. The process is repeated many times to find the best solution.

5. Simulation and result analysis

The GANN model is implemented with a dataset of monthly tourists to Thua Thien Hue (from January 2017 to December 2019). The setting parameters are described in Table 3. The results of the forecasting error of ANNs over the generations are shown in Fig. 10, where the best fitness always decrease over the generations. Starting from the 6th generation, the best fitness no longer increases and remains the same through the following generations (although only shown up to the 10th generation in Fig. 10). That proves that GANN converges quite quickly and achieves the best fitness.

	Parameters	Values
ANN	Number of inputs	6, 8, 10 and 12
	Number of hidden neurons	3, 4, 5, ..., 18
	Number of outputs	1
	Learning rate	0.05
GA	Population size	4
	Selection technique	Roulette wheel
	Crossover technique	at 2 points
	Mutation probability	0.01
	New population	Best individuals

Table 3. Parameters

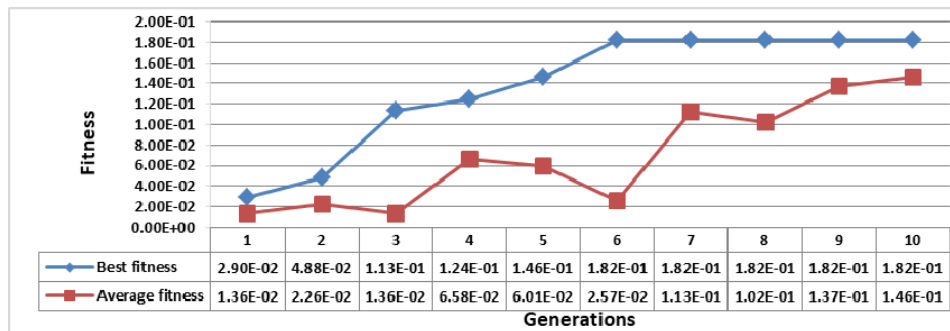


Fig. 10. Average and Best fitness across generations

ANNs is changed over generations in search of the best architecture. From the randomly initiated individuals in the first generation, the better ANNs are selected through each generation (see Table 4). From the 6th generation and later, the best neural network is always ANN(12:14:1), which is an architecture with 12 inputs, 14 hidden neurons and 1 output. In another way, GANN has converged to ANN (12:14:1).

Generation	1	2	3	4	5	6	7	8	9	10
Inputs	6	12	12	12	12	12	12	12	12	12
Hidden neurons	4	4	9	15	7	14	14	14	14	14
Best ANN	(6:4:1)	(12:6:1)	(12:9:1)	(12:15:1)	(12:7:1)	(12:14:1)	(12:14:1)	(12:14:1)	(12:14:1)	(12:14:1)

Table 4. The best ANN changes over generations and converges to ANN(12:14:1)

However, a question is whether ANN(12:14:1) is the best neural network in term of the global optimization, rather than just local optimization. Therefore a investigation of ANNs which are similar to ANN(12:x:1), i.e. same number of inputs (12) but varied number of hidden neurons ($x = 4, 6, \dots, 18$), is performed to evaluate the level of their optimization. The results in Figure 11 show that, there is a variation in forecasting error (MAPE) for different ANNs, but ANN(12:14:1) is still the best neural network with the minimal MAPE. This confirms that the GANN model is effective in time series forecasting.

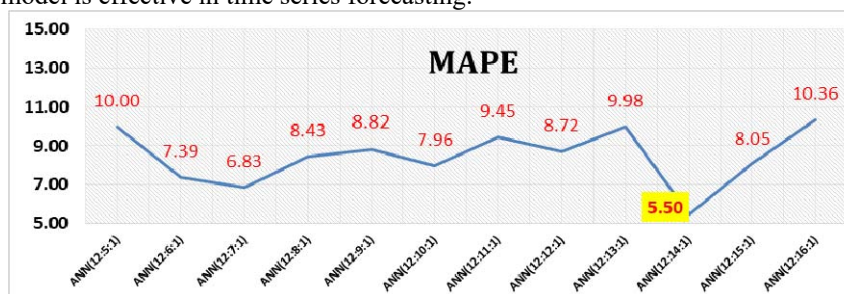


Fig. 11. A comparison of ANN(12:14:1) with similar ANNs

6. Conclusion

The paper analyzed the integration of ANN and GA to improve forecasting efficiency for time series data. The results showed that GA helped find the best ANN architecture, which is ANN(12:14:1) with 12 inputs, 14 hidden neurons and 1 outputs. Although the implementation was done with a relatively small data, that is the number of monthly tourist to Thua Thien Hue, Vietnam in 3 years, but the detailed analysis and descriptions of the implementation showed that GANN model is general and can be implemented well for any time series data model.

References

- [1] Adhikari, R.; Agrawal, R. K. (2013): *An Introductory Study on Time Series Modeling and Forecasting*. LAP Lambert Academic Publishing.
- [2] Mahmoud, E. (1984): Accuracy in forecasting: A survey, *J. Forecast.*, 3(2), pp. 139–159, doi: 10.1002/for.3980030203.
- [3] Tealab, A. (2018): Time series forecasting using artificial neural networks methodologies: A systematic review, *Futur. Comput. Informatics J.*, 3(2), pp. 334–340, doi: 10.1016/j.fcij.2018.10.003.
- [4] Crone, S. F.; Kourentzes, N. (2010): Feature selection for time series prediction - A combined filter and wrapper approach for neural networks, *Neurocomputing*, 73(10–12), pp. 1923–1936, doi: 10.1016/j.neucom.2010.01.017.
- [5] Mijanur Rahman, Md.; Setu, T. A. (2015): An Implementation for Combining Neural Networks and Genetic Algorithms, *Int. J. Comput. Sci. Technol.*, 6(3), pp. 218–222.
- [6] Inthachot, M.; Boonjing, V.; Intakosum, S. (2016): Artificial Neural Network and Genetic Algorithm Hybrid Intelligence for Predicting Thai Stock Price Index Trend, *Comput. Intell. Neurosci.*, 2016, doi: 10.1155/2016/3045254.
- [7] Jeenanunta, C.; Abeyrathna, K. D. (2019): Neural network with genetic algorithm for forecasting short-term electricity load demand, *Int. J. Energy Technol. Policy*, 15(2–3), pp. 337–350, doi: 10.1504/ijetp.2019.10019649.
- [8] Noersasongko, E.; Julfia, F. T.; Syukur, P. A.; Premunendar, R. A.; Supriyanto, C. (2016): A Tourism Arrival Forecasting using Genetic Algorithm based Neural Network, *Indian J. Sci. Technol.*, 9(4), pp. 3–7, doi: 10.17485/ijst/2016/v9i4/78722.
- [9] Huang, H. C.; Hou, C. I. (2017): Tourism Demand Forecasting Model Using Neural Network, *Int. J. Comput. Sci. Inf. Technol.*, 9(2), pp. 19–29, doi: 10.5121/ijcsit.2017.9202.
- [10] Ilyas Abas M.; Lasarudin, A. (2019): Prediction of arrival domestic and foreign tourists based on regions using neural network algorithm based on genetic algorithm, *J. Phys. Conf. Ser.*, 1175(1), pp. 1–8, doi: 10.1088/1742-6596/1175/1/012045.
- [11] Wang, M.; Zhang, H.; Wu, Z. (2019): Forecast and Application of GA Optimization BP Neural Network Tourism Demand in High-speed Railway Era, *IOP Conf. Ser. Mater. Sci. Eng.*, 569(4), doi: 10.1088/1757-899X/569/4/042053.
- [12] Peretto, P. (1992): *An Introduction to the Modeling of Neural Networks*, Cambridge: Cambridge University Press.
- [13] Sivanandam, S.N.; Deepa, S. N. (2008): *Introduction to Genetic Algorithms*, Berlin, Heidelberg: Springer Berlin Heidelberg.

Authors Profile



Vo Viet Minh Nhat received his PhD in Cognitive Informatics from the University of Quebec in Montreal, Canada, in 2007. He is currently an associate professor at Hue University, Vietnam. His research interests include optical packet/burst-based switching networks, mobile RFID/sensor systems, soft computing, neural networks, and evolutionary computation. Email: vmmnhat@hueuni.edu.vn



Le Van Hoa received his PhD in Computer Science from Hue University, Vietnam, in 2020. He is currently a lecturer at Hue University, Vietnam. His research interests include optical packet/burst-based switching networks, mobile RFID/sensor systems, fairness, quality of service, e-commerce, e-Tourism, soft computing. Email: levanhhoa@hueuni.edu.vn



Nguyen Thi Thuy Van received her MsC in Travel and Tourism Management from Hospitality and Tourism School of Hue University, Vietnam, in 2018. She is currently a lecturer at Hospitality and Tourism School of Hue University, Vietnam. Her research interests include digital marketing, smart tourism, e-commerce in tourism, sustainable tourism... Email: ngththuyvan@hueuni.edu.vn



Hoang Thi Hue received her Master's degree in International Tourism Management from Auckland University of Technology in Auckland, New Zealand, in 2019. She is currently a lecturer at Hue University, Vietnam. Her research interests include tourism marketing, tourism innovation, destination management, sustainable tourism. Email: hthue.kdl@hueuni.edu.vn