

Empirical Evaluation of the Time Series Forecasting Method by Combining ARIMA with RBFNN under the Additive Model

Thanh Son Nguyen^{1*}, Chi Cong Pham²

¹Ho Chi Minh City University of Technology and Education, Vietnam

²Ho Chi Minh City Open University, Vietnam

*Corresponding author. Email: sonnt@hcmute.edu.vn

ARTICLE INFO

Received: 14/01/2024
Revised: 20/02/2024
Accepted: 21/02/2024
Published: 28/02/2024

KEYWORDS

Time series;
Prediction model;
Time series prediction;
ARIMA;
RBFNN.

ABSTRACT

Time series data is a series of values observed through repeated measurements at different times. Time series data is a type of data present in almost all different fields of life. Time series prediction is an significant problem in time series data mining. Accurate forecasting is crucial to support decision making in many areas of life. Therefore, improving the precision of time series predicting is a interesting mission for experts in this field. Many models for predicting time series have been proposed from traditional time series models as Auto Regressive Integrated Moving Average (ARIMA) model to artificial neural network (ANN) models. ARIMA is a linear model therefore it can only take the linear characteristics in time series. In contrast, Radial Basis Function Neural Network (RBFNN) is a non-linear model therefore it can not predict effectively seasonal or trend changes in time series. To combine the strengths of these two models, in this study, we experimentally evaluate the hybrid method between ARIMA and RBFNN on real time series data from different fields. Experimental results demonstrate that the combined method outperforms each model used individually in terms of accuracy.

Doi: <https://doi.org/10.54644/jte.2024.1520>

Copyright © JTE. This is an open access article distributed under the terms and conditions of the [Creative Commons Attribution-NonCommercial 4.0 International License](https://creativecommons.org/licenses/by-nc/4.0/) which permits unrestricted use, distribution, and reproduction in any medium for non-commercial purpose, provided the original work is properly cited.

1. Introduction

Time series data is a sequence of real values observed through repeated measurements at different times. Time series data is a type of data present in almost all different fields of life. Many time series data in business, economics and other areas of life often exhibit seasonality and trends. Season is a recurring and cyclical form due to issues such as vacations, weather, promotions, etc. Although season is the most significant component in time series, trends are often accompanied by seasonal fluctuations and can have a major influence on forecasting methods. Exactly forecast of trend and seasonal time series data is very important to support decision making in many areas of life and therefore research to improve the effectiveness and efficiency of forecast methods has never stopped.

A lot of time series forecasting method have been introduced from traditional models to artificial neural network (ANN) models such as the method using clustering technique [1]. This technique was used to forecast electricity market [2]. The method based on clustering techniques using a brute force algorithm which was applied to stock market data [3], the method based on motif information [4], the methods of predicting short-term traffic flow and Gross Domestic Product (GDP) using k-NN model [5], [6], the method using the similar pattern search technique for predicting seasonal or trend time series [7], the method using a moving average model for predicting number of tourist [8]. Forecasting sale data and stock data using machine learning model [9], [10] and so on. Among traditional models, ARIMA has been broadly studied and used in the prediction field [11], [12]. This model reliably predicts future events, but unluckily, it is not very precise in many cases [13]. It is because ARIMA is a linear model therefore it can only take the linear characteristics in time series.

ANN has been used for the problem of forecasting time series data [14]-[18]. Because, this model is a non-linear model, it can not predict seasonal or trend changes effectively if the raw data is not preprocessed [19].

Recently, researchers often tend to use hybrid models in forecasting to combine the strengths of each model. Some combined models are also proposed for forecasting time series [20]-[31]. Most of these methods are researched to forecast a certain field, while others are studied for prediction problem on general time series. Some typical methods can be introduced such as: the method combining exponential smoothing and ANN is proposed for forecasting financial dataset [20], the method combining motif information and ANN is proposed for predicting time series [21], the method combining Winters' exponential smoothing technique and ANN for predicting seasonal and trend time series [22], the linear hybrid combination method between ANN and pattern matching using Euclidean distance is introduced in [23]. The ARIMA and HyFIS models were combined for predicting univariate time series [24], the hybrid model which combines ANN and least square SVM model for forecasting Monthly Streamflow Data [25], the model combining support vector machine and ANN for predicting stock data [26], the hybrid prediction model combining ARIMA and RBFNN models for examining water property [27], the hybrid method using machine learning and deep learning for predicting crop yield [28], the hybrid model between ARIMA and ANN for forecasting problems on monthly gold prices [29], the hybrid machine learning model which combines ANN-imperialist competitive algorithm (ANN-ICA) and ANN-gray wolf optimizer (ANN-GWO) method for predicting crop yield [30], the hybrid method using simulation models and machine learning to predict tea crop yield [31]

In this paper, we experimentally evaluate the method combining ARIMA and RBFNN on time series in different fields that may have trend or seasonal changes. The evaluation is based on two criteria: execution time and accuracy. Experimental results indicate that the hybrid method gives better results than each model used individually in terms of accuracy.

The rest of the paper is organised as follows. In section 2 we describe the background knowledge on ARIMA and RBFNN. Section 3 describes the combination of ARIMA and RBFNN in a additive model for time series forecasting. Section 4 shows our experimental results on real datasets. We present some conclusions in Section 5.

2. Fundamental knowledge on ARIMA, RBFNN and combining ARIMA with RBFNN

2.1. Auto Regressive Integrated Moving Average (ARIMA) model

The ARMA (p, q) model derived from two autoregressive models AR (p), moving average model MA (q) is proposed for univariate time series prediction.

The autoregressive model AR(p), where p is the order of autoregressive model, is defined in the following form:

$$y_t = c + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} + \varepsilon_t \quad (1)$$

Where, y_t is the prediction value at time period t , ε_t is the estimated residual at each time period t , $\varphi_i, i = 1..p$ are the estimated weights representing the influence of y_{t-i} values on y_t and c is a constant. For simplicity, the constant c is sometimes omitted.

The moving average model MA(q), where q is the order of moving average model, is defined in the following form:

$$y_t = \mu - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} + \varepsilon_t \quad (2)$$

Where, μ is the mean of time series, $\theta_i (i=1..q)$ are the coefficients that estimate the influence of ε_{t-i} on y_t , and ε_t is the estimated residual at each time period t .

Combining the two autoregressive models AR (p) and moving average model MA (q), we can represent the ARMA(p, q) model as follows [32], [33]:

$$y_t = c + \varepsilon_t + \varphi_1 y_{t-1} + \varphi_2 y_{t-2} + \dots + \varphi_p y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (3)$$

To form a stationary time series, the simplest method is to take the difference. d -order difference of the series is calculated as follows:

The 1st order difference: $I(1) = \Delta(y_t) = y_t - y_{t-1}$

The d^{th} order difference: $I(d) = \Delta^d(y_t) = \underbrace{\Delta(\Delta(\dots\Delta(y_t)))}_{d \text{ times}}$

Normally the time series will become a stationary series after the differentiation process $I(0)$ or $I(1)$. The ARIMA(p, d, q) model can be represented as:

$$y_t = \varepsilon_t + \varphi_1 \Delta y_{t-1} + \varphi_2 \Delta y_{t-2} + \dots + \varphi_p \Delta y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (4)$$

Where Δy_t is the d^{th} -order difference of y_t , ε_t is white noise, φ_i ($i=1..p$) and θ_i ($i=1..q$) are the coefficients of AR(p) and MA(q) components, respectively.

2.2. Radial Basis Function Neural Network (RBFNN)

The simplest structure of RBFNN includes three different layers with feedforward architecture: an input layer, a hidden layer and an output layer. The nodes within each layer are fully connected to the previous layer as shown in figure 1. The input layer is a set of r units, which receives a r -dimensional input feature vector. Nodes in the hidden layer (called RBF units) use a radial basis function. The output layer is linear and serves as a summation unit. The output of the i^{th} RBF unit in the hidden layer is calculated as follows [34]:

$$R_i(X) = R_i \left(\frac{\|X - C_i\|}{\sigma_i} \right) \quad i = 1, 2, \dots, u \quad (5)$$

Where $\|\cdot\|$ implies the Euclidean norm on the input vector, X is an r -dimensional input vector, C_i is a vector with the same dimension as X , u is a number of nodes in the hidden layer, $R_i(\cdot)$ is the output of the i^{th} radial basis function unit and σ_i is the width of the i^{th} RBF unit.

The output $y_j(X)$ of the j^{th} output node of an RBF neural network is calculated by

$$y_j(X) = \sum_{i=1}^N w_{ji} * R_i(X) + b(j) \quad (6)$$

Where w_{ji} is the weight between the i^{th} RBF unit and the j^{th} output node, and $b(j)$ is the bias of the j^{th} output node. To reduce the network complexity, the bias is usually not considered.

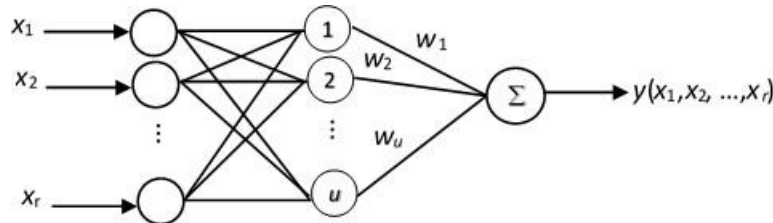


Figure 1. RBFNN structure

3. Combining ARIMA with RBFNN under additive model

The additive sequential hybrid model is proposed by Zhang et al. [35]. This model assumes the time series y_t is the sum of two linear and nonlinear components. It means $y_t = \hat{y}_t + e_t$, where \hat{y}_t , the linear components, are predictions of y_t and e_t , the nonlinear parts, are residual sequences (prediction errors). In this hybrid model, the ARIMA model is used to predict the linear component \hat{y}_t of y_t and RBFNN is used to predict the nonlinear part \hat{e}_t of e_t . The hybrid model between ARIMA and RBFNN according to additive model is described in Figure 2.

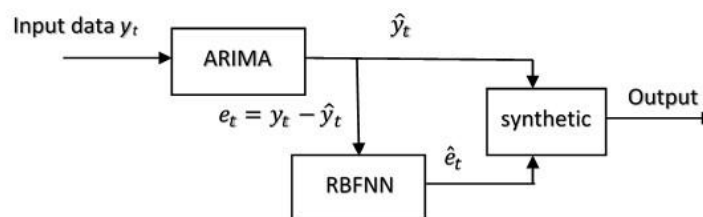


Figure 2. The hybrid predicting model

The time series data is fed into ARIMA model. The input time series data y_t is converted into stationary series by applying the d^{th} order difference of data points (normally d is chosen to be 0 or 1). Then prediction values \hat{y}_t are obtained from ARIMA by:

$$\hat{y}_t = c + \varepsilon_t + \varphi_1 \Delta y_{t-1} + \varphi_2 \Delta y_{t-2} + \dots + \varphi_p \Delta y_{t-p} - \theta_1 \varepsilon_{t-1} - \theta_2 \varepsilon_{t-2} - \dots - \theta_q \varepsilon_{t-q} \quad (7)$$

Where Δy_t is the d^{th} -order difference of y_t .

Nonlinear components (also known as residual series) are calculated by using the original series y_t subtracting the linear predictions \hat{y}_t from the ARIMA model as:

$$e_t = y_t - \hat{y}_t \quad (8)$$

Then, the residual series are fed into RBFNN to predict nonlinear components \hat{e}_t using the following formula:

$$\hat{e}_t = \sum_{j=1}^N w_{tj} * R_j + b(t) \quad (9)$$

Where w_{ij} is the weight between the j^{th} RBF unit and the i^{th} output node, and $b(t)$ is the bias of the i^{th} output node.

The final result \hat{y}_{hybrid} of the time series y_t of the hybrid model is calculated by synthesizing the forecast results of the two models ARIMA and RBFNN by the following formula:

$$\hat{y}_{\text{hybrid}} = \hat{y}_t + \hat{e}_t \quad (10)$$

4. Experimental evaluation

In this section, we represent the empirical comparison of ARIMA, RBFNN and hybrid ARIMA-RBFNN model according to additive model. The hybrid model used in this study is similar to the model introduced by Zhang et al. [27]. However, the method suggested in [27] is used for water quality analysis purposes in which there is no trend and changes with the seasons, while our experiments are carried out on time series in different fields that may have trend or seasonal changes.

4.1. The datasets used in experiment and experimental environment

We compare empirically ARIMA, RBFNN and hybrid ARIMA-RBFNN model according to additive model on four real datasets: AirPassengers, Sunspots, Dentists, City_temperature. For each dataset, we chose a ratio of 70% for training and 30% for testing.

- The AirPassengers dataset provides information on the number of passengers flying per month from 1949 to 1960 in US.
- The Sunspots dataset provides information on monthly mean total sunspot number from 1749 to July 2018.
- The Dentists dataset provides information on world health statistics on the number of dentists available per 10,000 population.
- The City_temperature dataset provides information on daily average temperature values recorded in major cities of the world.

All four datasets above are taken from the web: <https://www.kaggle.com/datasets>. Figure 3 shows the plots of the above datasets.

We experimentally compare the performance of the hybrid approach under the additive model with those of RBFNN model and ARIMA model which are used separately. The three prediction models ARIMA, RBFNN and hybrid ARIMA-RBFNN model according to additive model are experimentally compared on all segments of the testing dataset. Then the arithmetic mean of errors in the predictive duration are calculated. The methods are implemented with Python and experiments are conducted on a Dell Inspiron 15 computer, Inter® core™ i5-5300U CPU @ 2.3 GHz, 16 GB RAM, Windows 10 operating system.

To estimate the parameters p, d, q for the ARIMA(p, d, q) model, the auto_arma function in python is used. We run this function with the training dataset. The auto_arma function considers all possible models for the time series and selects the parameters that minimizes the Akaike information criterion (AIC). The AIC value for a model is calculated as:

$$AIC = 2k - 2\ln(\hat{L}) \quad (11)$$

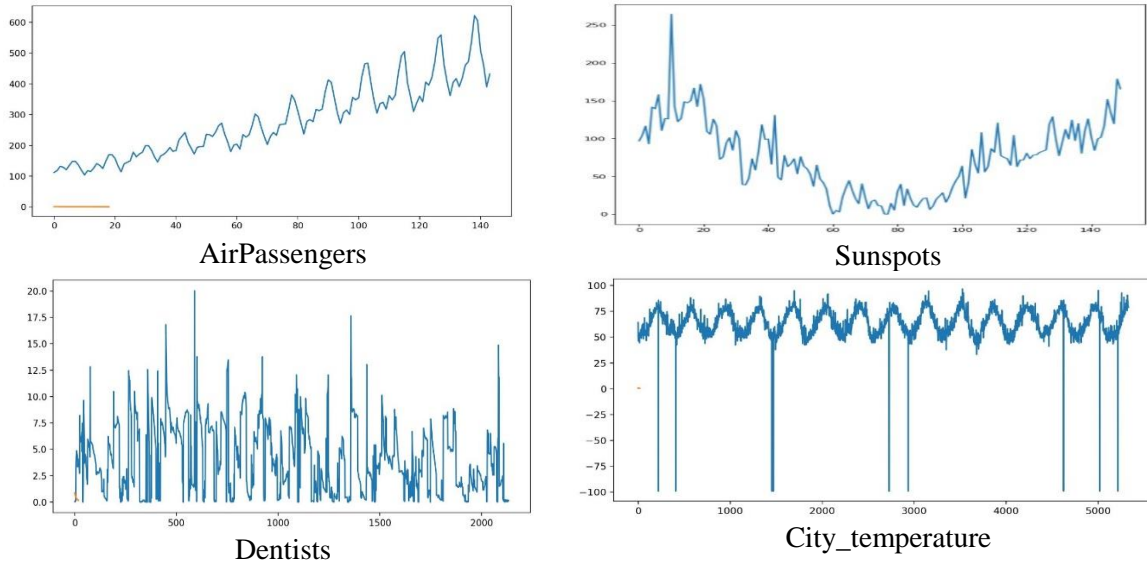


Figure 3. The plots of four different datasets.

Where, k is the number of estimated parameters in the model and \hat{L} is the maximum value of the likelihood function for the model. Through the test, the best Arima model obtained for each set of data as follows: ARIMA(0,1,0) for AirPassengers dataset, ARIMA(0,1,1) for Sunspots dataset, ARIMA(1,0,0) for Dentists dataset and ARIMA(3,0,2) for City_temperature dataset.

For RBFNN, we use the three layers model: input, hidden and output layers. The number of input and output nodes for RBFNN is the same and is configured as the length of ARIMA output. To determine the number of hidden nodes of RBFNN, we use the algorithm creating dynamically nodes to insert nodes to the hidden layer in training phase until RBFNN meets the specified mean squared error target.

4.2. Valuation measures

In this work, we compare the hybrid method with the RBFNN method and the ARIMA method used separately according to two criteria: execution time and accuracy. The execution time is calculated from the time the model starts until the model produces forecast results. To measure the prediction accuracy, we use the mean absolute error (MAE) and the mean squared error (MSE). They are defined as follows.

$$MAE = \frac{1}{n} \sum_{i=1}^n |Y_{obs,i} - Y_{model,i}| \quad (12)$$

$$MSE = \frac{1}{n} \sum_{i=1}^n (Y_{obs,i} - Y_{model,i})^2 \quad (13)$$

Where Y_{obs} is real values and $Y_{model,i}$ is value at time i of the model.

4.3. Experimental results

After the models have been built, we performed a evaluation of the models on the testing dataset. The experimental results of the test on AirPassengers are shown in Table 1. The experimental results on Sunspots dataset are presented in Table 2. Table 3 presents the experimental results on Dentists dataset. Table 4 shows the experimental results on City_temperature dataset.

From the experimental results presented in the above tables, it can be seen that the forecast error of the predicting method which combines ARIMA and RBFNN is better than the forecasting method using ARIMA and the forecasting method based on RBFNN. That means that for all four datasets, the hybridization forecasting model is the best of the three models. That is because the combined model has integrated the advantages of the two models ARIMA and RBFNN to create a synergistic result and hence improve forecast effectiveness. But, the execution time of the combined model is the longest

among the three models because it has to execute both models and then synthesize the results of the two models to produce the final prediction.

Table 1. *Experimental results from the AirPassengers dataset*

	Time (s)	MSE	MAE
ARIMA	468.168	1467.689	31.918
RBFNN	41.589	38478.000	234.451
Hybrid ARIMA and RBFNN	646.053	1386.966	30.163

Table 2. *Experimental results from the Sunspots dataset*

	Time (s)	MSE	MAE
ARIMA	114.958	515.157	16.764
RBFNN	23.79	1275.811	29.486
Hybrid ARIMA and RBFNN	142.835	486.823	15.842

Table 3. *Experimental results from the Dentists dataset*

	Time (s)	MSE	MAE
ARIMA	1685.423	1.566	0.634
RBFNN	66.738	24.542	4.037
Hybrid ARIMA and RBFNN	2325.265	1.480	0.599

Table 4. *Experimental results from the City_temperature dataset*

	Time (s)	MSE	MAE
ARIMA	6247.267	82.603	3.099
RBFNN	72.175	8064.835	88.161
Hybrid ARIMA and RBFNN	16959.25	78.060	2.929

5. Conclusions

In our work, we experimentally evaluate the combined model between ARIMA and RBFNN according to the additive model for forecasting time series in different fields that may have trend or seasonal changes. Experimental results on four real time series datasets from different fields show that the combined model performs better than ARIMA and RBFNN in forecasting time series in terms of predictive accuracy. This result also suggests that the ARIMA method and the RBFNN model can complement each other in time series prediction. The limitation of this study is that we have not compared of this hybrid model with other traditional methods. In the future, we will conduct an experimental comparison of this combined model with other traditional methods. In addition, we will also research combining ARIMA and RBFNN in a parallel model.

Conflict of Interest

The authors declare no conflict of interest.

REFERENCES

- [1] A. T. Lora, J. C. Riquelme, J. L. M. Ramos, J. M. R. Santos, and A. G. Expósito, "Influence of kNN-Based Load Forecasting Errors on Optimal Energy Production," *Progress in Artificial Intelligence*, Lecture Notes in Computer Science, vol. 2902, pp. 189-203, 2003. ISSN: 0302-9743.
- [2] F. M. Alvarez, A. T. Lora, J. C. Riquelme, and J. S. A. Ruiz, "Energy Time Series Forecasting Based on Pattern Sequence Similarity," *IEEE Trans. on Knowledge and Data Engineering*, vol. 23, no. 8, pp. 1230-1243, 2011.
- [3] R. Nayak and T. Braak, "Temporal Pattern Matching for the Prediction of Stock Prices," in *Proceedings 2nd International Workshop on Integrating Artificial Intelligence and Data Mining (AIDM 2007)*, vol. 84, pp. 95-103.
- [4] Y. Jiang, C. Li, and J. Han, "Stock temporal prediction based on time series motifs," in *Proceedings of 8th International Conference on Machine Learning and Cybernetics*, 2009, vol. 6, pp. 3550-3555.

- [5] L. Zhang *et al.*, "An Improved K-nearest Neighbor Model for Short-term Traffic Flow Prediction," in *Intelligent and Integrated Sustainable Multimodal Transportation Systems Proceedings from the 13th COTA International Conference of Transportation Professionals (CICTP2013)*, vol. 96, pp. 653–662.
- [6] G. Rigopoulos, "Univariate Time Series Forecasting Using k-Nearest Neighbors Algorithm: A Case for GDP," *International Journal of Scientific Research and Management (IJSRM)*, vol. 10, no. 09, pp. EM01, Sep. 2022, doi: 10.18535/ijrm/v10i9.em01.
- [7] N. T. Son, N. H. Le, and D. T. Anh, "Time Series Prediction Using Pattern Matching," in *Proc. of International Conference on Computing, Management and Telecommunications (ComManTel 2013)*, Ho Chi Minh City, Vietnam, Jan. 21–24, 2013, pp. 401–406.
- [8] Z. Ivanovski, A. Milenkovski, and Z. Narasnov, "Time Series Forecasting Using a Moving Average Model for Extrapolation of Number of Tourist," *UTMS Journal of Economics*, vol. 9, no. 2, pp. 121–132, 2018.
- [9] M. Sreemalli, P. Chaitanya, and K. Srinivas, "Comparative Analysis of Machine Learning Techniques on Stock Market Prediction," in *Vol. 14 ICETCSE 2016 Special Issue International Journal of Computer Science and Information Security (IJCSIS)*, ISSN 1947-5500 [Online]. Available: <https://sites.google.com/site/ijcsis/>.
- [10] B. M. Pavlyshenko, "Machine-Learning Models for Sales Time Series Forecasting," *Data*, vol. 4, no. 1, p. 15, Jan. 2019, doi: 10.3390/data4010015.
- [11] J. C. Brocklebank, D. A. Dickey, and B. S. Choi, *SAS for Forecasting Time Series*, 3rd ed. Cary, NC: SAS Institute Inc., 2018.
- [12] J. H. Kim, "Forecasting autoregressive time series with bias corrected parameter estimators," *International Journal of Forecasting*, vol. 19, pp. 493–502, 2003.
- [13] D. C. Montgomery, C. L. Jennings, and M. Kulahci, *Introduction to Time Series Analysis and Forecasting*. John Wiley & Sons, 2015.
- [14] S. D. Balkin and J. K. Ord, "Automatic neural network modeling for univariate time series," *International Journal of Forecasting*, vol. 16, pp. 509–515, 2000.
- [15] E. Cadenas and W. Rivera, "Short-term wind speed forecasting in La Venta, Oaxaca, México, using artificial neural network," *Renewable Energy*, vol. 34, pp. 274–278, 2009.
- [16] M. Ghiassi, H. Saidane, and D. K. Zimbra, "A dynamic artificial neural network for forecasting series events," *International Journal of Forecasting*, vol. 21, pp. 341–362, 2005.
- [17] S. Heravi, D. R. Osborn, and C. R. Birchenhall, "Linear versus neural network forecasting for European industrial production series," *International Journal of Forecasting*, vol. 20, no. 3, pp. 435–446, 2004.
- [18] G. Tkacz, "Neural network forecasting of Canadian GDP growth," *International Journal of Forecasting*, vol. 17, pp. 57–69, 2001.
- [19] G. P. Zhang and M. Qi, "Neural network forecasting for seasonal and trend time series," *European Journal of Operational Research*, vol. 160, no. 2, pp. 501–514, 2005.
- [20] K. Lai *et al.*, "Hybridizing exponential smoothing and neural network for financial time series prediction," in *Proceedings of 6th International Conference on Computational Science (ICCS'06)*, vol. 4, pp. 493–500, 2006.
- [21] C. D. Truong, H. N. Tin, and D. T. Anh, "Combining motif information and neural network for time series prediction," *Int. J. Business Intelligence and Data Mining*, vol. 7, no. 4, pp. 318–339, 2012.
- [22] D. N. Bao, N. D. K. Vy, and D. T. Anh, "A hybrid method for forecasting trend and seasonal time series," in *Proc. of 2013 IEEE RIVF International Conference on Information and Communication Technologies*, Hanoi, Vietnam, Nov. 10–13, 2013, pp. 203–208.
- [23] N. T. Son and D. T. Anh, "Hybridizing pattern matching and neural network for time series prediction," in *Proc. of 2013 World Congress on Information and Communication Technologies (WICT 2013)*, Hanoi, Vietnam, pp. 19–24.
- [24] S. A. Mangai *et al.*, "Hybrid ARIMA-HyFIS model for forecasting univariate time series," *International Journal of Computer Applications*, vol. 91, no. 5, pp. 38–44, 2014.
- [25] S. M. Pandhiani and A. B. Shabri, "Time series forecasting by using hybrid models for monthly streamflow data," *Applied Mathematical Sciences*, vol. 9, no. 57, pp. 2809–2829, 2015.
- [26] J. S. Vaiz and M. Ramaswami, "A Hybrid Model to Forecast Stock Trend Using Support Vector Machine and Neural Networks," *International Journal Of Engineering Research And Development*, vol. 13, no. 9, pp. 52–59, Sep. 2017.
- [27] L. Zhang, G. X. Zhang, and R. R. Li, "Water quality analysis and prediction using hybrid time series and neural network models," *J. Agr. Sci. Tech.*, vol. 18, pp. 975–983, 2018.
- [28] S. Agarwal and S. Tarar, "A Hybrid Approach For Crop Yield Prediction Using Machine Learning And Deep Learning Algorithms," *J. Phys.: Conf. Ser.*, vol. 1714, p. 012012, 2021.
- [29] A. A. Alsuwaylimi, "Comparison of ARIMA, ANN and Hybrid ARIMA-ANN Models for Time Series Forecasting," *Inf. Sci. Lett.*, vol. 12, no. 2, pp. 1003–1016, 2023.
- [30] S. Nosratabadi *et al.*, "Hybrid Machine Learning Models for Crop Yield Prediction," *arXiv:2005.04155 [cs.NE]*, 2020.
- [31] D. Batool *et al.*, "A Hybrid Approach to Tea Crop Yield Prediction Using Simulation Models and Machine Learning," *Plants*, vol. 11, no. 9, p. 1925, Sep. 2022, doi: 10.3390/plants11151925.
- [32] J. H. Cochrane, "Time Series for Macroeconomics and Finance," Graduate School of Business, University of Chicago, Spring 1997.
- [33] K. W. Hipel and A. I. McLeod, *Time Series Modelling of Water Resources and Environmental Systems*, Amsterdam, Elsevier, 1994.
- [34] Z. Q. Zhao and D. Huang, "A mended hybrid learning algorithm for radial basis function neural networks to improve generalization capability," *Applied Mathematical Modelling*, vol. 31, no. 7, pp. 1271–1281, 2007.
- [35] P. G. Zhang, "Time series forecasting using a hybrid ARIMA and neural network model," *Neurocomputing*, vol. 50, pp. 159–175, 2003.



Nguyen Thanh Son received his B.S. in Information Technology from Faculty of Information Technology, Ho Chi Minh City University of Natural Sciences, Vietnam where he also received his Master degree in the same branch. He is currently Ph.D. student in Faculty of Computer Science and Engineering, Ho Chi Minh City University of Technology, Vietnam. His main research interest is include artificial intelligence, machine learning, deep learning, time series data mining and association rule mining. He can be contacted at email: sonnt@hcmute.edu.vn

ORCID:  <https://orcid.org/0000-0003-0191-9150>



Pham Chi Cong received the B.S. degree in Information Technology Engineer at Saigon Technology University in 2008, the M.S. degree in Economic management at Hanoi University of Mining and Geology in 2013, and the M.S. degree in Computer Science at HCMC University of Technology and Education in 2021. From 2008 to 2022, he was a lecturer in Computer Science at the Southeast Asia Institute for Cultural and Educational Development Research. From 2022 up to now, he is a lecturer in Computer Science at the Ho Chi Minh City Open University. His research interest is time series datamining. He can be contacted at email: cong.pc@ou.edu.vn