Information Systems and Economics (2023)
Clausius Scientific Press, Canada

DOI: 10.23977/infse.2023.040406
ISSN 2523-6407 Vol. 4 Num. 4

Research on Synergistic Enhancement of Sales Forecasting through Time Series and Neural Networks

Feng Wang[a], Joey Aviles[b,*]

Graduate School, Angeles University Foundation, 2009 Angeles City, Philippines
[a]wang.feng@auf.edu.ph, [b]aviles.joey@auf.edu.ph
**Corresponding author*

Keywords: Time series, neutral network, machine learning, sales, algorithm

Abstract: The objective of this study was to explore the combined use of time series and neural networks, employing the Narxnet algorithm, for predicting commodity sales volume. The Narxnet algorithm is a neural network model specifically designed for handling time series data, allowing for the utilization of historical data and external factors to forecast future sales volume. By harnessing the strengths of time series analysis and neural networks, we aimed to achieve more accurate predictions of commodity sales volume and assist businesses in formulating more effective supply chain management and marketing strategies. In this study, we investigated the performance of the Narxnet algorithm in predicting commodity sales volume, analyzing its predictive accuracy, model complexity, data requirements, and its adaptability to trends and seasonality. The obtained optimal performance was 85%, which, although lower than 90%, was attributed to the limited sample size. It is believed that a larger sample size would significantly enhance the algorithm's performance. This research provides insights into the field of commodity sales volume prediction, offering a perspective on the potential use of multiple algorithmic approaches to improve performance.

1. Introduction

Accurately predicting product sales is crucial in the retail and e-commerce industry for supply chain management, inventory planning, and sales strategies. However, product sales are influenced by various factors such as seasonal variations, promotional activities, and competitor behavior, making forecasting complex and challenging. Traditional time series methods such as ARIMA, exponential smoothing, etc., can handle basic time series patterns but have limited modeling capabilities for complex non-linear patterns and long-term dependencies. In such cases, neural networks, as powerful machine learning tools, can learn non-linear patterns and complex temporal dependencies from data, providing more accurate sales predictions. The combination of time series and neural networks enables effective capturing of hidden patterns and trends in product sales data. By introducing recurrent neural networks (RNNs) or long short-term memory networks (LSTMs), the networks can remember and utilize past sales data and make predictions based on historical patterns. This approach is adaptive to different data characteristics and capable of handling non-linear relationships. For example, in seasonal product sales forecasting, neural networks can capture cyclic variations on a yearly, monthly, or weekly basis, considering these seasonal factors in the prediction process. Additionally, neural networks can learn more subtle sales trends and patterns from extensive historical sales data, such as sales peaks, gradual growth or decline trends, etc.

By combining time series and neural networks, the accuracy of predicting product sales can be significantly improved. This approach not only adapts to changing sales environments and market factors but also flexibly responds to new sales trends and consumer behavior changes. It helps retailers and e-commerce businesses optimize inventory management, reduce supply chain risks, and better meet consumer demands.

2. Literature Review

Many scholars have done research on forecasting sales volume by combining neural network with time series data. Some scholars suggest that combining LSTM neural network with time series data can reduce the error in forecasting and improve the forecasting performance[1]. Compared with ARIMA model, BP neural network algorithm has more advantages and plays an outstanding role in improving the error correction ability of prediction results, which shows that the two algorithms are complementary and can be combined[2]. The combination algorithm of deep learning and time series superposition is used to predict the data, and good results are obtained[3]. An ARIMA-LSTM combined model is used to predict the data, and it is found that the prediction accuracy is higher and the error is the smallest[4].

3. Method

The method of combining time series data with neural network can make better use of time-dependent and nonlinear patterns in the data, thereby improving the accuracy and effect of commodity sales forecasting. This has important significance for enterprise decision-making, supply chain management and marketing. There are several commonly used methods when combining time series data with neural networks for product sales forecasting.

3.1 RNN

RNN is a type of neural network widely used in modeling sequence data. It is characterized by the ability to process time-dependent data. In merchandise sales forecasting, RNNs can be used to capture temporal relationships and trends among sales data. Commonly used RNN variants include Long Short-Term Memory (LSTM) and Gated Recurrent Unit (GRU)[5].

The basic formula for RNN:

❿ Input:

x_t: Input at time step t

h_{t-1}: Hidden state from the previous time step t-1

Hidden state update:

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h) \tag{1}$$

❿ Output:

$$y_t = f\left(W_{hy}h_t + b_y\right) \tag{2}$$

Where:

W_{xh}: Weight matrix from input to hidden layer

W_{hh}: Weight matrix from hidden layer to hidden layer

W_{hy}: Weight matrix from hidden layer to output layer

b_h: Bias vector of the hidden layer
b_y: Bias vector of the output layer
f: Activation function (e.g., sigmoid, tanh, or ReLU)

The primary purpose of RNN is to handle time dependencies in sequential data. It captures context information across different time steps by using the recurrent connections, allowing it to make predictions based on the context. RNN can handle sequences of varying lengths and effectively process inputs and outputs at different time steps.

RNN finds wide applications in natural language processing, speech recognition, machine translation, and more. Some specific use cases include:

Language modeling: Training RNN to generate coherent text sequences, such as generating articles, dialogues, or poems.

Machine translation: Translating sequences from one language to another.

Sentiment analysis: Analyzing the sentiment state at each time step for a series of text data, such as determining the sentiment (positive, negative, or neutral) of comments.

Time series forecasting: Using RNN to predict time-dependent data, such as stock prices, weather data, or sales data.

Speech recognition: Converting audio signals into text sequences, for example, speech recognition in voice assistants.

RNN is a powerful tool for processing sequential data, capturing time dependencies, and finding applications in various fields for sequence modeling and prediction tasks.

3.2 CNN

Although CNN is mainly used in the field of image processing, it also has certain application value in time series data. By viewing time-series data as an image-like two-dimensional structure, CNNs can be used to extract local patterns and features of the data. These features can help predict trends and periodicities in merchandise sales.

The formulas:

➓ Convolutional layer:

$$h_i = f\left(\sum_j \left(W_j * x_{i+j}\right) + b\right) \tag{3}$$

➓ Pooling layer:

$$y_i = \mathrm{pool}(x_i) \tag{4}$$

➓ Fully connected layer:

$$y = f(Wx + b) \tag{5}$$

Where:
x_i: a local region of the input data (e.g., a small patch of an image)
W_j: convolutional kernel (weights) used for convolution operation on the input data
h_i: output feature map of the convolutional layer
b: bias term
y_i: output of the pooling layer
W: weight matrix of the fully connected layer
x: input data
y: output result
f: activation function (e.g., ReLU, sigmoid, or tanh)
pool: pooling operation (e.g., max pooling or average pooling)

The main purpose of CNN is to extract local patterns and features from the input data and perform classification or regression predictions using fully connected layers[6]. Through the convolution operation, CNN captures the spatial structure information of the input data, and pooling layers downsample the data to reduce the number of parameters and computational complexity. This makes CNN highly effective for processing grid-like data, such as images.

CNN finds extensive applications in the field of computer vision. Some specific application scenarios include:

Image classification: Training CNN to recognize and classify objects or scenes in images.

Object detection: Locating and recognizing multiple objects in an image, providing bounding boxes and class labels.

Image segmentation: Dividing an image into different regions or pixel-level labels for image analysis and processing.

Feature extraction: Utilizing the convolutional layers of CNN to extract feature representations from images for tasks like image retrieval or image generation.

Video analysis: Extending CNN to the temporal dimension for action recognition, behavior analysis, and other video sequence-related tasks.

CNN is a powerful tool for processing grid-like data, extracting local patterns and features through convolution and pooling operations, and making predictions using fully connected layers. It has wide applications in computer vision tasks such as image analysis, classification, and object detection.

3.3 Combination of Neural Network and ARIMA Model

ARIMA is a classic time series model. Combining the neural network with the ARIMA model can make full use of the neural network's ability to learn complex patterns, while retaining the ARIMA model's ability to model time series trends and seasonality[5,7]. The usual approach is to use neural networks to capture nonlinear patterns, which are then combined with the residuals from the ARIMA model.

When combining neural networks with ARIMA models, the following equations are typically used:

➓ ARIMA model part:

$$y_t = \phi_0 + \phi_1 y_{t-1} + \phi_2 y_{t-2} + \cdots + \phi_p y_{t-p} + \theta_1 \varepsilon_{t-1} + \theta_2 \varepsilon_{t-2} + \cdots + \theta_q \varepsilon_{t-q} + \varepsilon_t \tag{6}$$

➓ Neural network model part:

$$y_t = f(W_{xh} x_t + W_{hh} h_{t-1} + b_h) \tag{7}$$

Where:

y_t: The observed or predicted value at time step t.

ϕ_i: Autoregressive coefficients of the AR part.

θ_i: Moving Average coefficients of the MA part.

ε_t: Error term.

W_{xh}: Weight matrix from input to hidden layer.

W_{hh}: Weight matrix from hidden layer to hidden layer.

b_h: Bias vector of the hidden layer.

f: Activation function.

3.4 Deep Neural Network Ensemble

Ensemble methods improve accuracy and stability by combining the predictions of multiple models. In merchandise sales forecasting, multiple deep neural network models such as multi-layer perceptron (MLP), LSTM, GRU, etc. can be used and their prediction results averaged or weighted summed. This ensemble approach reduces the bias of individual models and reduces forecast volatility

3.5 Narxnet

A neural network model used for modeling and predicting nonlinear dynamic systems. It captures system dynamics by incorporating external inputs and time delays, and leverages the nonlinear modeling capability of neural networks for prediction. The NARXNET model incorporates time delays, allowing it to capture system dynamics and historical dependencies. By using external inputs and the previous time step's hidden state as inputs, the NARXNET model can better capture the nonlinear dynamics of the system. The model can be trained using backpropagation algorithm, updating the weights and biases based on the difference between predicted and actual values[8].

4. Result

4.1 Dataset Preparation

A supermarket's commodity sales data, from January 1, 2021 to December 31, 2022, includes sales time, season, payment method, satisfaction, weather, discount, retail price, sales volume and other data, totaling 730 pieces of data. Among them, payment method, satisfaction, weather and discount are obtained through customer survey and are subjective factors of customers. The following table 1 is in data form.

Table 1: Numeric field.

Sale_Time	Season	Payment Method	Satisfaction	Weather	Discount	Price($)	Quantity
2021/1/1	1	1	3	2	2	20.5	11
2021/1/2	1	1	3	2	1	20.5	9
2021/1/3	1	3	3	3	3	20.5	9
2021/1/4	1	5	4	4	5	20.5	15
2021/1/5	1	3	3	3	2	20.5	7
2021/1/6	1	4	3	4	5	20.5	15
2021/1/7	1	3	5	4	5	20.5	14
:	:	:	:	:	:	:	:
2022/12/31	4	3	3	4	5	19	26

4.2 Model Performance

In Figure 1, the parameters are specified as follows: the number of hidden neurons is 10, and the number of delays is 2.

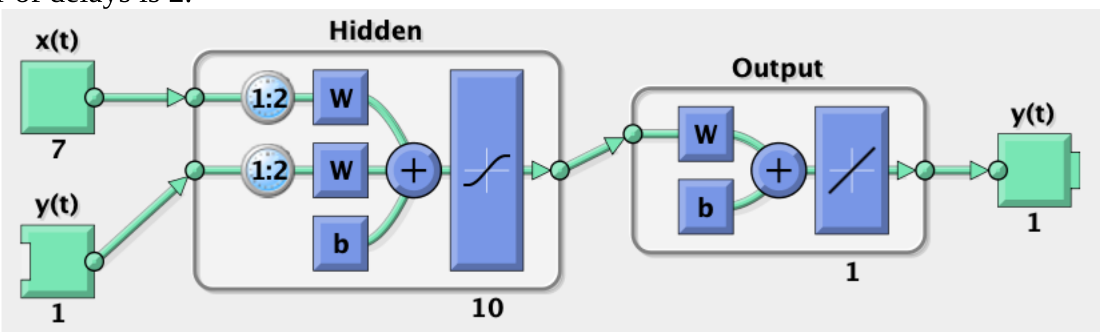

Figure 1: Neural network.

The performance results obtained by the algorithm in Table 2 are 86% for training, 83% for validation and 82% for testing.

Table 2: Algorithm's performance results.

	Target Values	MSE	R
Training	510	12.00329e-0	8.60213e-1
Validation	110	15.67056e-0	8.33925e-1
Testing	110	14.06204e-0	8.27618e-1

4.3 Diagnostic Evaluation

Look at the Figure 2. The blue, green, and red lines represent the mean square error of the training set, validation set, and test set, respectively. The dashed line indicates the best result, which is 15.6706 achieved during the 9th epoch of training, and it has been highlighted with small circles. The subsequent three lines of different colors are closely grouped together, indicating the relative stability of the predictions for this dataset.

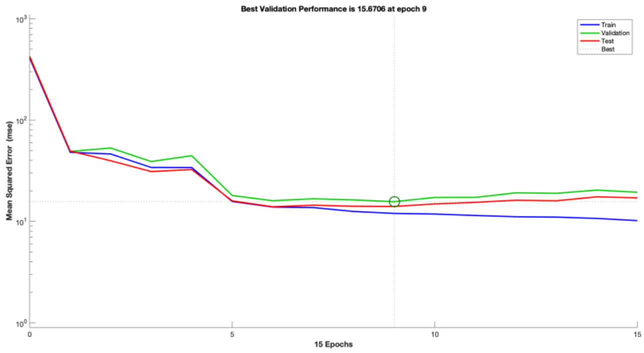

Figure 2: Best validation performance.

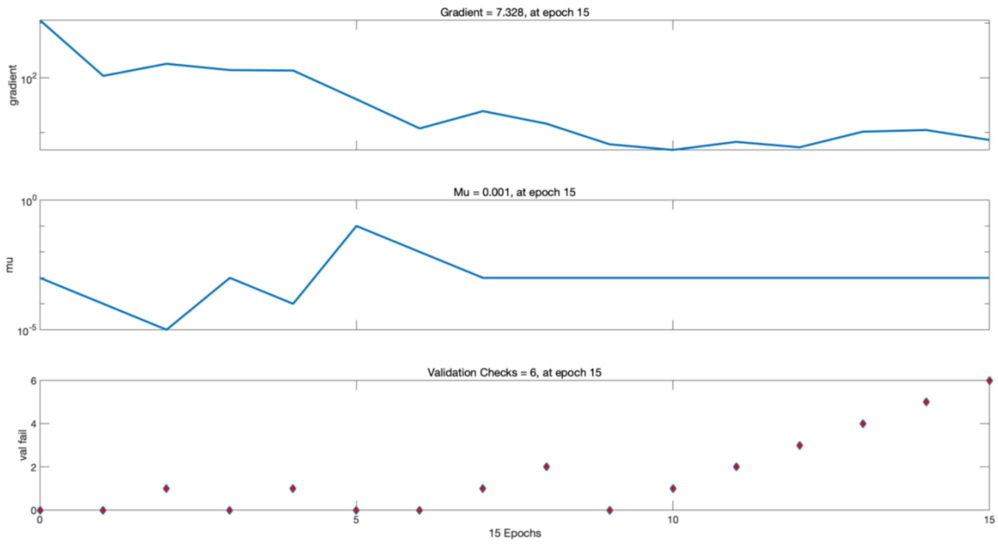

Figure 3: The Magnitude of Gradient, Damping Factor (Mu), and Validation Checks.

From top to bottom, Figure 3 shows the overall downward trend of the gradient, indicating that the change of the objective function relative to the parameters is very small at epoch 15. Usually, a smaller gradient value indicates that the model is close to the local optimal solution or converges to the global optimal solution. Mu=0.001, which is very close to zero. This means that in epoch 15, the optimization algorithm uses a very small damping factor to control the convergence speed and stability of the training process. When reaching epoch 15, two verification checks have been made. This means that before the epoch 15, the performance of the model has been evaluated twice using the verification set.

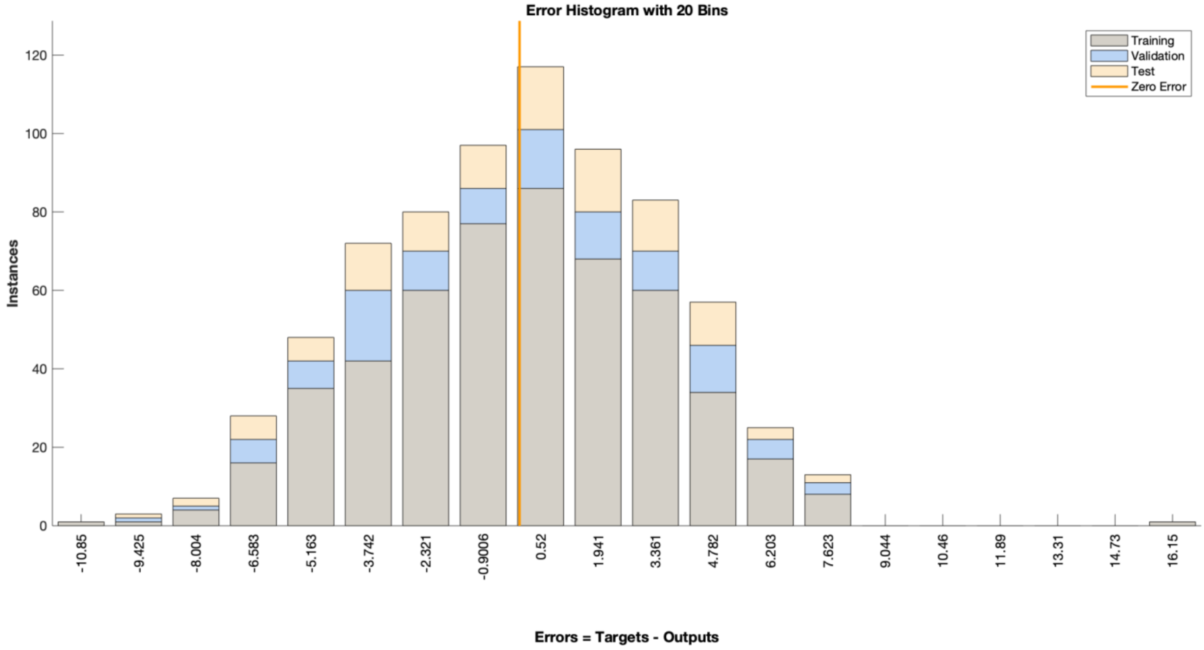

Figure 4: Error histogram.

Figure 4 presents the error histogram, where the gray, blue, and yellow bars represent the training set, validation set, and test set, respectively. The x-axis represents the median value of the error interval, while the y-axis represents the number of samples falling within each error interval. In an ideal scenario, the error would be entirely 0, resulting in a single row of 0 values in the center. In a typical case, the majority of samples are concentrated near 0, with fewer samples as the absolute value of the error increases. This error distribution exhibits a certain degree of normality.

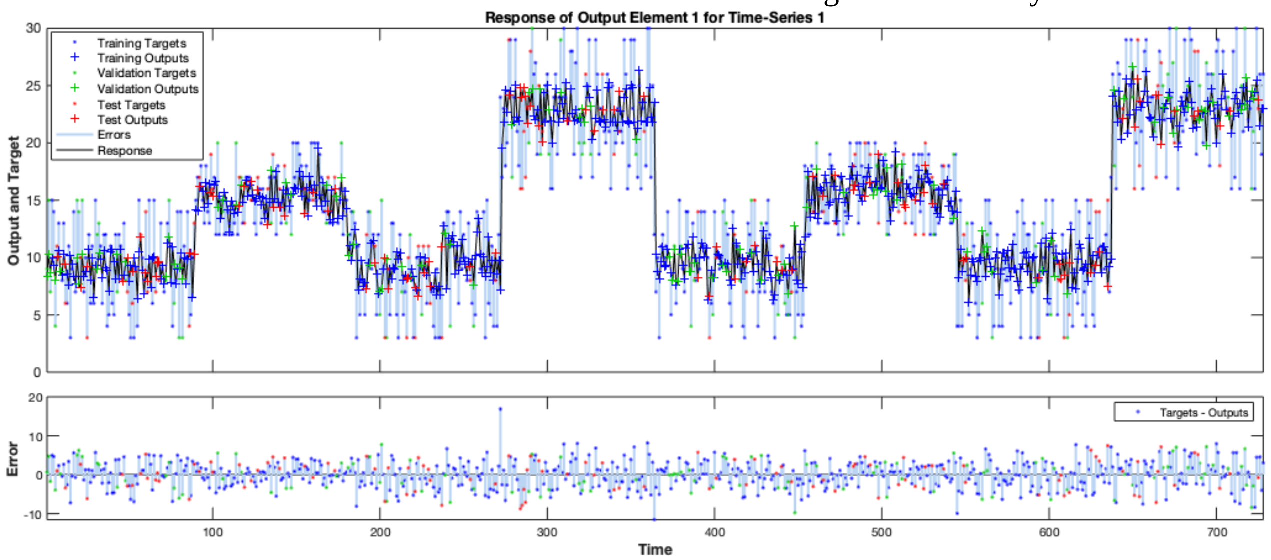

Figure 5: Time-series response.

The Figure 5 represents the response of a specific output element, which is element 1, for a particular time-series, which is time-series 1. In a time-series analysis or prediction context, the response refers to the output or predicted values generated by a model or system for a given input or time-series data. Each element of the output represents a specific variable or feature of interest. In this case, the plot focuses on element 1 of the output, indicating the response or predicted values associated with that particular element. The time-series 1, on the other hand, represents a specific input sequence or data series.

By examining this plot, you can analyze the behavior, patterns, or trends of the response of element 1 for time-series 1, gaining insights into how the model or system is predicting or generating output for that specific element and input sequence.

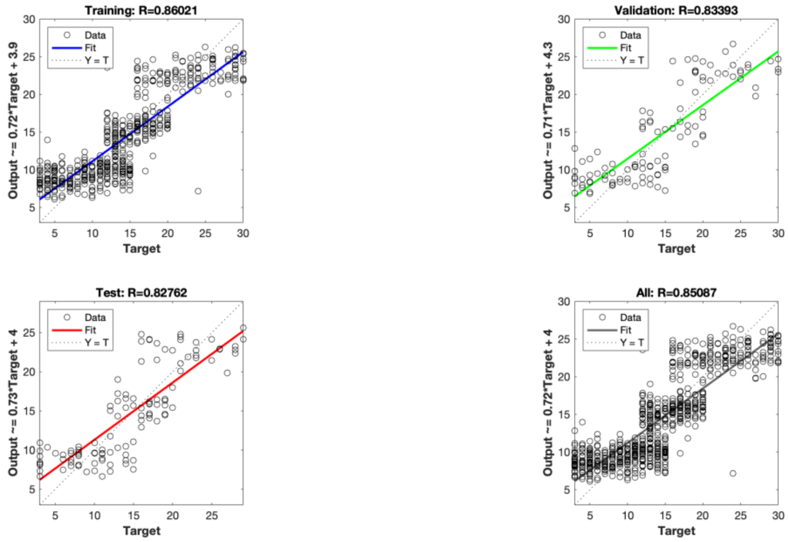

Figure 6: Regression.

Figure 6 shows the training set, verification set, test set and regression diagram of all data. The x-axis represents the actual value of the sample, while the y-axis represents the estimated value calculated by the model. The correlation coefficient (r) indicates the degree of linearity between the target value and the predicted value. The higher the r value, the stronger the linear relationship and the better the result. In this case, the r value of the training set is 86%, the r value of the verification set is 83%, the r value of the test set is 82%, and the r value of all data combinations is 85%. If the model obtained from this training process is not satisfactory, it may need to be recalculated. The recalibration process includes adjusting the number of neurons and the number of samples. However, in this case, the final R values of the training set, verification set and test set all exceed 80%. This shows that the performance of the model still has room for improvement. You can increase the sample size to make the performance better.

5. Conclusions

Narxnet algorithm can usually perform well in the prediction of commodity sales. It can capture the nonlinear relationship in time series data and the influence of external factors on sales volume. It is a relatively complex neural network model with multiple hidden layers and many parameters. This may require a long training time and a larger data set to optimize the model performance. In order to obtain better prediction results, this algorithm needs time series with sufficient historical data and related external inputs. Longer time series and more features can provide more information to train and improve the model. The sales volume of this data set has a certain trend and seasonal pattern. It can help to capture these patterns and make corresponding predictions. However, it should be noted that the actual results may be affected by many factors, including data quality, feature selection and model parameter adjustment. Therefore, when using this algorithm to predict the sales volume of commodities, the actual conclusions may be different for specific data sets and problems. In this experiment, the final performance is not very ideal because of the small number of samples. However, the algorithm combining time series and neural network is of great significance to the prediction of commodity sales, which can improve the accuracy of prediction, capture long-term dependence, consider external factors, and have good flexibility and expansibility. This will help enterprises to make more accurate sales forecasts, optimize supply chain management and formulate more effective marketing strategies, thus improving business performance and competitiveness.

References

[1] Raubitzek S, Neubauer T. Taming the Chaos in Neural Network Time Series Predictions [J]. Entropy, 2021, 23(11): 1424.

[2] Ali M, Khan D M, Aamir M, et al. Predicting the Direction Movement of Financial Time Series Using Artificial Neural Network and Support Vector Machine [J]. Complexity, 2021: 1–13.

[3] Bento P M R, Pombo J A N, Calado M R A, et al. Stacking Ensemble Methodology Using Deep Learning and ARIMA Models for Short-Term Load Forecasting [J]. Energies, 2021, 14(21): 7378.

[4] Li C, Fang X, Yan Z, et al. Research on Gas Concentration Prediction Based on the ARIMA-LSTM Combination Model [J]. Processes, 2023, 11(1): 174.

[5] Li M, Ji S, Liu G. Forecasting of Chinese E-Commerce Sales: An Empirical Comparison of ARIMA, Nonlinear Autoregressive Neural Network, and a Combined ARIMA-NARNN Model [J]. Mathematical Problems in Engineering, 2018: 1–12.

[6] Zain Z M, Alturki N M. COVID-19 Pandemic Forecasting Using CNN-LSTM: A Hybrid Approach [J]. Journal of Control Science and Engineering, 2021: 1–23.

[7] Wang J. Analysis of Sports Performance Prediction Model Based on GA-BP Neural Network Algorithm [J]. Computational Intelligence and Neuroscience, 2021: 1–12.

[8] Yang Z, Mehmed E E. Artificial neural networks in freight rate Forecasting [J]. Maritime Economics & Logistics, 2019, 21(3): 390–414.