Proceedings of CONF-MLA 2025 Symposium: Applied Artificial Intelligence Research
DOI: 10.54254/2755-2721/2025.BJ28430

Multi-Step Forecasting of U.S. Maritime Transportation Flows Using Hybrid ARIMA, PCR, CNN, and Recurrent Neural Network Models

Xinyue Zhu

University of Toronto, University College, Toronto, Canada
xzcindy.zhu@mail.utoronto.ca

Abstract. Maritime transportation plays an essential role in the global economy, supporting trade, supply chains and the development of countries. Forecasting maritime traffic and shipping connectivity between different ports in the United States can help us optimize trading strategy, improve global economic competition and promote technological innovation,. In this paper, six time series forecasting models, ARIMA, PCR, GRU, CNN, RNN and LSTM, are used to forecast the mode of maritime transportation trade in United State and compare between each model to discuss which fits well and further apply to future three years. This approach enables a comparative analysis of the relationship between port performance and geographic time-zone divisions. Applying data from The United Nations Conference on Trade and Development (UNCTAD) to the models demonstrates that the most suitable model of the data is PCR. With accurate forecasts, policymakers and stakeholders could formulate forward-looking strategies for risk mitigation.

Keywords: Time series forecasting, maritime transportation, Autoregressive Integrated Moving Average (ARIMA), Principal Component Regression (PCR), Convolutional Neural Network (CNN)

1. Introduction

The analysis in this study relies on the Liner Shipping Connectivity Index (LSCI), introduced by the United Nations Conference on Trade and Development in 2004, which serves as a key indicator of how effectively countries and ports are integrated into global shipping networks [1]. According to Beatriz Tovar's research, even small gains in connectivity can lead to significant improvements in port efficiency, with a strong positive relationship observed between the two [2]. The UNCTAD Policy Paper further concludes that the quality of maritime connectivity is a major factor in securing access to foreign markets and, consequently, in enhancing export performance [3]. This index is created in six components: (1) The number of scheduled ship calls per week; (2) Total scheduled annual deployed capacity (TEU); (3) The total count of scheduled liner shipping routes connecting to and from the country; (4) The number of liner shipping carries operating services to and from the country; (5) The size, in TEU, of the largest ship deployed on services from and to the country; (6)

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2025.BJ28430

The number of other countries that are connected to the country through direct liner shipping services [4].

In the previous literature, several time-series models have been adopted for U.S. container trade forecasting. For instance, a different index, the North American Container Trade (NACT) index, has been employed to contrast with alternative container trade indices, demonstrating that harbor transportation flows are closely associated with both the global business cycle and the U.S. business cycle [5]. Munim proposed a Prophet-based forecast combination approach to predict container throughputs for the ports of Shanghai, Busan, and Nagoya, revealing that the combined forecast model achieves superior accuracy for these ports [6]. Eskafi developed a Bayesian model using two types of Iceland port throughput data: containerized throughput in TEUs and non-containerized throughput [7]. Similarly, Rashed employed an ARIMA model to forecast the port of Antwerp's monthly container throughput [8]. Morales-Ramírez used VAR and SARIMA models to forecast Mexico's port cargo throughput, highlighting that past trends strongly influence future patterns and emphasizing operational efficiency [9].

Despite these contributions, only a limited number of studies have employed the same database of the U.S. as used in the present research. While some prior works adopt forecasting models similar to those applied herein, differences in data sources impede direct comparability and systematic evaluation. To address this gap, this study implements multiple deep learning architectures, alongside other benchmark models, to forecast the U.S. maritime transportation flows, thereby enabling a comprehensive comparative analysis of predictive performance across methodological approaches.

This study mainly focuses on six models to forecast the U.S. maritime transportation flows and systematically compare the model's performance in terms of MSE, MAE, RMSE, and MAPE to identify the most effective forecasting approach. Applying the model to real life can provide reliable short-term and medium-term forecasts that can guide authorities, shipping companies, and policymakers in capacity planning, scheduling, and resource allocation to improve shipping connectivity. Greater integration can lower domestic transportation costs, enhancing a country's competitiveness. Reduced per-unit transport costs also decreased carbon emissions, yielding notable environmental benefits [10].

In the latter part of this paper, the Port Liner Shipping Connectivity Indec (PLSCI) is employed to provide a more detailed depiction of port-level characteristics. PLSCI indicates a port's standing within the global liner chipping network, where higher values represent stronger connectivity [11]. For analytical clarity, U.S. ports are classified into four groups based on their respective time zones: Eastern Time (ET), Pacific Time (PT), Central Time (CT), and others. In addition, key statistical metrics, mean, median, maximum, and the slope of the linear regression line, which represents the growth rate, are computed to identify the top ten ports in each category. This approach enables a comparative analysis of the relationship between port performance and geographic time-zone divisions.

2. Related work

In the research paper, six distinct time series forecasting models are applied to the U.S. LSCI in order to provide a comprehensive comparison to discover the model that best fit the data.

ARIMA model, first introduced by Box and Jenkins in the 1970s [12], has long been recognized as one of the most widely used statistical methods for time series analysis. It relies on autoregression, differencing, and moving average components to capture linear temporal dependencies, and has been widely applied in economics, such as short-term stock price forecasting

DOI: 10.54254/2755-2721/2025.BJ28430

to support investment decision-making [13]. PCR, described by Massy in 1965 [24], combines Principal Component Analysis (PCA) with regression. By transforming correlated predictors into orthogonal principal components before regression, PCR effectively reduces noise and multicollinearity. This makes it particularly suitable for forecasting in high-dimensional or strongly correlated datasets, such as macroeconomic or trade indices. RNN was introduced by Elman in 1990 [14] and marked a significant shift by introducing recurrent connections that allow models to capture sequential dependencies over time. This capacity to learn dynamic relationships across time steps [15] has made RNN highly effective in applications such as speech recognition, natural language processing [16], and financial forecasting. Building on this, Cho and colleagues proposed GRU in 2014 [17]. GRUs simplify the RNN architecture by introducing update and reset gates, reducing the number of parameters while maintaining the ability to model long-term dependencies. This makes GRU computationally efficient while retaining strong forecasting performance in domains such as finance and economics [18]. Another important advancement is LSTM, developed by Hochreiter and Schmidhuber in 1997 [19]. LSTM addressed the vanishing gradient problem of traditional RNN by introducing memory cells and gating mechanisms, enabling the model to capture nonlinear and long-range temporal patterns. LSTM have since been widely adopted in areas including energy demand forecasting, stock market prediction, and maritime transportation analysis [20]. Finally, CNN was originally developed by LeCun in 1989 for image recognition [21], have been adapted for time series analysis. By extracting local patterns and short-term dependencies from sequential data, CNN is particularly effective in detecting fluctuations in domains such as international trade [22] and electricity consumption forecasting [23].

To evaluate the predictive performance of the forecasting models, four widely used error metrics are employed: Mean Squared Error (MSE), Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and Mean Absolute Percentage Error (MAPE). These metrics provide complementary perspectives on model accuracy by measuring the deviation between predicted values ($\widehat{y_t}$) and actual observations (y_t) over n time steps. The formulas are given as follows:

$$MSE = \frac{1}{n}\sum_{t=1}^{n}\left(y_t - \widehat{y_t}\right)^2 \tag{1}$$

$$MAE = \frac{1}{n}\sum_{t=1}^{n}\left|y_t - \widehat{y_t}\right| \tag{2}$$

$$RMSE = \sqrt{\frac{1}{n}\sum_{t=1}^{n}\left(y_t - \widehat{y_t}\right)^2} \tag{3}$$

$$MAPE\,(\%) = \frac{100}{n}\sum_{t=1}^{n}\left|\frac{y_t - \widehat{y_t}}{y_t}\right| \tag{4}$$

3. Data processing

Before conducting the data analysis, it is necessary to preprocess the dataset to ensure its suitability for subsequent modeling. The required data were extracted and organized into a new table, in which LCSI values were placed in columns with corresponding time periods in rows, making them suitable for time-series forecasting. For the visualization of port LCSI across the four U.S. time zones, the full dataset was filtered according to these port lists by Python, and the selected data were plotted to compare temporal patterns between time zones.

DOI: 10.54254/2755-2721/2025.BJ28430

In order to capture recent patterns and model changing dynamics, a moving window was applied in the models. It is a convergent approximate method for time series optimization [25]. Initial time series can change over time, so older data might be less relevant. Therefore, finding a fixed length, named window size, is necessary to train in the data and predict the next point in the series. As shown in Figure1, add the true next point to the end of the window and drop the oldest point so the window remains the same size. Repeat the step until you reach the end of the series. The key parameters in the moving window are window size and lag, which is the number of past points that are used as predictors inside each window. During this research paper, the window size is set up to 10 with a lag equal to 5.

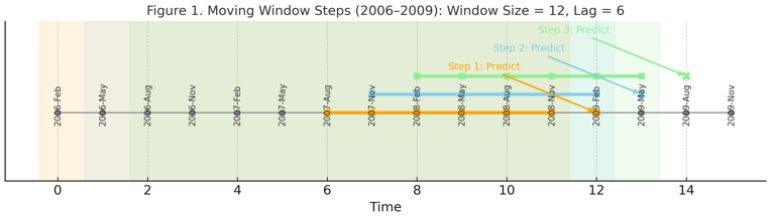

Figure 1. Moving window steps (2006-2009): window size = 12, Lag = 6

4. Methods

4.1. Autoregressive Integrated Moving Average (ARIMA)

ARIMA is a statistical model to forecast and analyze time series data which combines both Autoregressive (AR) process and Moving Average (MA) processes into a unified time series model, with an additional differencing component to address non-stationarity. As indicated by its notation ARIMA (p,d,q), the model consists of three key parameters:

• AR(p): Describes the relationship between the current observation and its p lagged values. The AP(p) process is expressed as:

$$x_t = c + \sum_{i=1}^{p} \emptyset_i x_{t-i} + \varepsilon_t \tag{5}$$

where x_t is the stationary time series, c is a constant, $\emptyset_i$ are autoregressive coefficients, and ε_t is Gaussian white noise with mean zero and variance σ_ε^2 .

• I(Integrated): Refers to differencing the series d times to achieve stationarity, removing trends or other forms of non-stationarity.

• MA (Moving Average): Represents the dependency between the current observation and the q lagged forecast errors. The MA(q) process is written as:

$$x_t = \mu + \sum_{i=0}^{q} \theta_i \varepsilon_{t-1} \tag{6}$$

where μ is the expected value of x_t , and θ_i are moving average coefficients.

The general ARIMA(p,d,q) model combines the AR and MA structures after differencing, and can be compactly expressed as:

$$\emptyset_p(B)(1-B)^d x_t = \theta_q(B)\varepsilon_t \tag{7}$$

DOI: 10.54254/2755-2721/2025.BJ28430

Where B is the backshift operator ($Bx_t = x_{t-1}$), $\emptyset_p(B)$ is the autoregressive polynomial of order p, and $\theta_q(B)$ is the moving average polynomial of order q.

4.2. Recurrent Neutral Networks (RNN)

RNN have traditionally been employed in sequence modeling, achieving strong performance across various natural language processing tasks. Due to their natural ability to interpret time series as ordered sequences of inputs and targets, numerous RNN-based architectures have been developed for temporal forecasting applications.

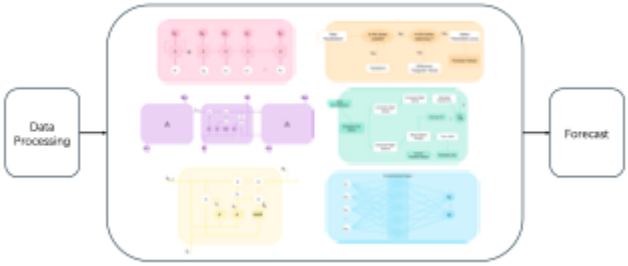

Figure 2. Flowcharts of CNN, LSTM, GRU, ARIMA, PCR, CNN

RNN cells inherently keep an internal hidden state that summarizes past information, updating it recursively with each new input at every time step as shown in the Figure 2 (pink), as expressed by:

$$z_t = v(z_{t-1}, y_t, x_t, s) \tag{8}$$

where $z_t \in \mathbb{R}^{\mathscr{H}}$ denotes the hidden state, and $v(\bullet)$ is the learned memory update function.

A simple example is the Elman RNN, which computes the output and updates the hidden state as follows:

$$y_{t+1} = \gamma_y(W_y z_t + b_y) \tag{9}$$

$$z_t = \gamma_z(W_{z1} z_{t-1} + W_{z2} y_t + W_{z3} x_t + W_{z4} s + b_z) \tag{10}$$

Here, W and b represent the network's weights and biases, while $\gamma_y(\bullet)$ and $\gamma_z(\bullet)$ are activation functions.

4.3. Gated Recurrent Unit (GRU)

GRU is a streamlined RNN variant that merges the input and forget gates into a single update gate. It uses update and reset gates to efficiently control memory flow, enabling faster training with performance similar to LSTM. It is illustrated in Figure 2(yellow), processes a sequence x as follows:

$$z_t = \sigma(x_t U^z + h_{t-1} W^z) \tag{11}$$

$$r_t = \sigma(x_t U^r + h_{t-1} W^r) \tag{12}$$

$$s_t = \tanh(x_t U^s + (r_t \odot h_{t-1} W^s) \tag{13}$$

DOI: 10.54254/2755-2721/2025.BJ28430

$$h_t = (1 - z_t) \odot s_t + z_t \odot h_{t-1} \tag{14}$$

Here, $x_t \in \mathbb{R}^d$ is the token embedding at position t, $h_t \in \mathbb{R}^h$ is the hidden state, and z_t and r_t are the update and reset gates, respectively. All $U \in \mathbb{R}^{d \times h}$ and $W \in \mathbb{R}^{d \times h}$ are trainable parameters. The update gate controls the balance between keeping past information and incorporating new input, while the reset gate decides how much past information to forget.

4.4. Long Short-Term Memory (LSTM)

LSTM is a specialized form of RNN designed with additional mechanisms to retain sequential information over extended periods. The internal structure of an LSTM cell is demonstrated in Figure 2 (purple) and operates as follows:

$$i_t = \sigma\left(x_t U^i + h_{t-1} W^i + b_i\right) \tag{15}$$

$$f_t = \sigma\left(x_t U^f + h_{t-1} W^f + b_f\right) \tag{16}$$

$$o_t = \sigma(x_t U^o + h_{t-1} W^o + b_o) \tag{17}$$

$$q_t = \tanh(x_t U^q + h_{t-1} W^q + b_q) \tag{18}$$

$$p_t = f_t * p_{t-1} + i_t * q_t \tag{19}$$

$$h_t = o_t * \tanh(p_t) \tag{20}$$

The LSTM architecture contains three gates: the input gate i_t, forget gate f_t, and output gate o_t. Each gate applies a sigmoid activation to a linear transformation of the current input x_t and the previous hidden state h_{t-1}. A candidate state q_t is generated through a tanh nonlinearity, which is then integrated with the previous cell state p_{t-1} by the input and forget gates to form the updated cell state p_t. Finally, the output gate regulates the exposure of p_t to produce the hidden state h_t, enabling the network to control which information is stored, discarded, or passed to the next time step.

4.5. Convolutional Neural Network (CNN)

CNN traditionally is designed for image datasets and extract local patterns that remain consistent across spatial dimensions.

4.5.1. Input layer

Given a sequence x with n entries, each element is represented by a d-dimensional dense vector. As a result, the input sequence forms a feature map with dimensions d $\times$ n. Figure 2(blue) depicts this input layer as the lower rectangle containing multiple columns.

4.5.2. Convolution layer

The convolution layer learns representations from sliding w-grams. For an input sequence $x_1, x_2, \ldots, x_n$, each w-gram is constructed by concatenating the embeddings of w consecutive

DOI: 10.54254/2755-2721/2025.BJ28430

entries, denoted as $c_i \in \mathbb{R}^{wd}$, with zero-padding applied when i < 1 or i > n. The representation for each w-gram is computed as:

$$p_i = \tanh(W \bullet c_i + b) \tag{21}$$

where $W \in \mathbb{R}^{d \times wd}$ is the convolution weight matrix and $b \in \mathbb{R}^d$ is the bias.

4.5.3. Maxpooling

All w-gram representations p_i are aggregated via maxpooling across each feature dimension j, producing the sequence-level representation:

$$x_j = \max(p_{1,j}, p_{2,j}, \ldots), \quad j = 1, \ldots, d \tag{22}$$

4.6. Principal Component Regression (PCR)

PCR is a two-stage procedure that combines Principal Component Analysis (PCA) with linear regression. PCA first transforms the original predictor matrix X into uncorrelated principal components by solving:

$$X^T X v_k = \lambda_k v_k \tag{23}$$

where v_k is the k-th eigenvector, and λ_k is its corresponding eigenvalue representing the variance explained by the k-th component. The transformed principal components are obtained as:

$$Z = XV \tag{24}$$

where V contains the selected eigenvectors as columns.

In the second stage, a standard linear regression model is fitted using the first m principal components Z_m as predictors:

$$y = Z_m \beta + \varepsilon \tag{25}$$

where β is estimated via Ordinary Least Squares (OLS), and ε denotes the residual term.

DOI: 10.54254/2755-2721/2025.BJ28430

5. Results

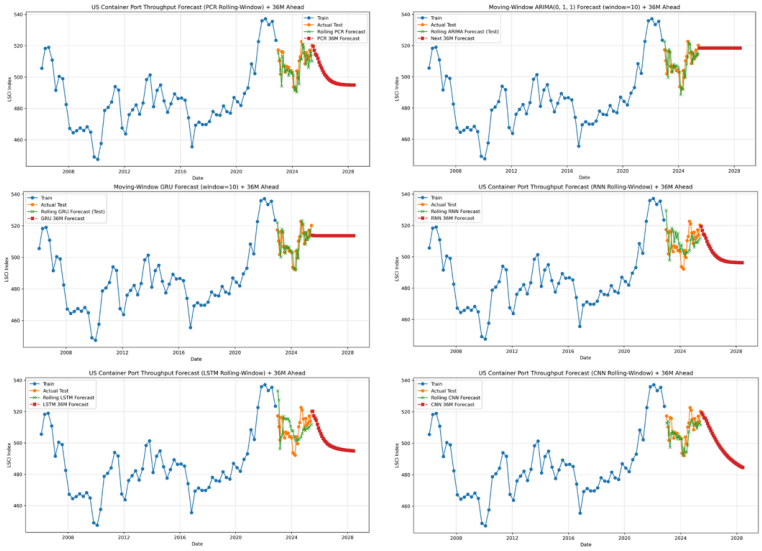

Figure 3. Plots of six models forecasting results

Table 1. Error metrics of six models forecast

	MSE	MAE	RMSE	MAPE (%)
ARIMA	41.2433	5.2460	6.4221	1.0297
GRU	39.5656	4.9713	6.2901	0.9743
PCR	48.7138	4.8033	6.9795	0.9396
CNN	41.3229	4.8982	6.4283	0.9586
RNN	90.3499	8.1972	9.5053	1.6101
LSTM	46.5053	4.9756	6.8195	0.9736

Figure 3 exhibits the results of six models, actual tests compared to forecast outcomes. Together with error metrics in Table 1, for MSE and RMSE, GRU shows the best accuracy. Nevertheless, PCR demonstrates better in both MAE and MAPE for 4.80 and 0.94%. Therefore, it can be inferred that PCR is the most suitable model to forecast U.S. LSCI with more stable and accurate predictions.

Although CNN has shown strong performance in various forecasting tasks, their advantage relies heavily on the availability of large-scale datasets and the presence of prominent local temporal patterns. In the case of the LSCI series, the dataset exhibits strong long-term trends and relatively limited local fluctuations, which are more effectively captured by regression-based models, such as PCR and sequence models like GRU. Additionally, CNN performance is sensitive to kernel size, stride, and the number of convolutional layers, and suboptimal hyperparameter settings may have further constrained its predictive capability in this study.

The three-year forecasts indicate an overall downward trajectory of U.S. maritime connectivity, with fluctuations around the declining trend. Classical models yield smoother and more gradual decreases, while neural network models capture more pronounced short-term variations superimposed on the long-term decline. These results suggest a weakening in LSCI values over the forecast horizon, highlighting potential challenges for sustaining maritime transport connectivity in the coming years.

DOI: 10.54254/2755-2721/2025.BJ28430

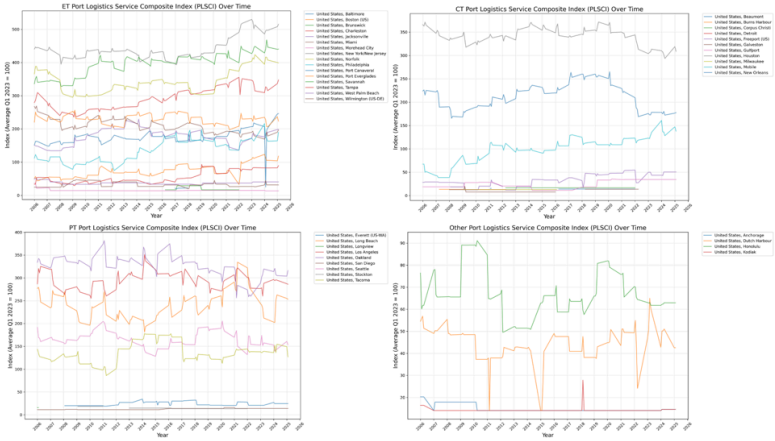

Figure 4. Four time zones PLSCI plots overtime

To conduct a more nuanced analysis of U.S. port connectivity, time plots were constructed for individual ports within each of the four time zones in Figure 4. These visualizations allow for the examination of temporal variations in connectivity patterns, enabling a clearer comparison of trends across regions and facilitating the identification of potential spatial patterns. To be more specific, the time series analysis of PLSCI across the four U.S. time zones reveals notable spatial and temporal disparities. Ports in the Eastern and Pacific zones exhibit consistently higher and more stable connectivity levels, while those in the Central and Other zones tend to have lower values with greater volatility. This pattern reflects the geographical concentration of major trade corridors along the East and West Coasts, highlighting the uneven distribution of port connectivity nationwide.

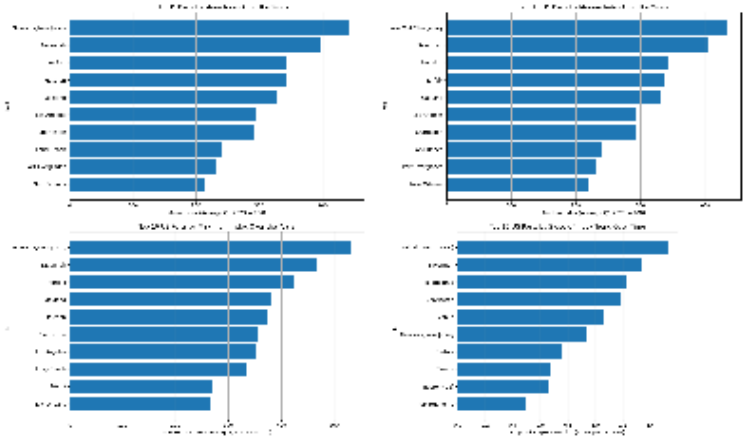

Figure 5. Top 10 ports by mean, median, maximum and growth rate

Figure 5 reveals that Los Angeles, Long Beach, and New York/New Jersey lead the top ten ports by average PLSCI, serving as the dominant, consistently high-performing gateways for trans-Pacific and trans-Atlantic trade. Mid-ranked ports such as Savannah and Norfolk have improved steadily, while Houston and Charleston display greater variability, indicating higher sensitivity to market and supply chain fluctuations.

DOI: 10.54254/2755-2721/2025.BJ28430

Overall, the U.S. maritime network exhibits a highly centralized structure, with the top ten ports accounting for a disproportionately large share of national connectivity. While this concentration enhances the strategic value and resilience of major gateways, it also introduces systemic risks, as disruptions at one or more key nodes could trigger widespread network effects.

6. Conclusion

The analysis of U.S. port connectivity using the PLSCI index, combined with the evaluation of six forecasting models, ARIMA, PCR, CNN, RNN, GRU, and LSTM, under a rolling-window framework, reveals both spatial disparities and structural concentration in the national maritime network. Based on MSE, MAE, RMSE, and MAPE, PCR achieves the highest overall accuracy, highlighting the strength of dimensionality reduction in combination with linear modeling for capturing key patterns in maritime connectivity time series. These results emphasize the value of robust statistical approaches in improving connectivity monitoring and informing strategies to strengthen the resilience of the U.S. port system. Furthermore, Eastern and Pacific zones maintain higher and more stable connectivity than Central and Other zones, especially LA and NY, consistently ranking among the top ten ports.

The findings provide valuable guidance for policymakers, port authorities, and logistics stakeholders in prioritizing infrastructure investment and capacity expansion at high-impact hubs, while also implementing targeted support for lower-connectivity regions to balance national network resilience. By leveraging accurate forecasts from models such as PCR, decision-makers can develop proactive risk-mitigation strategies to minimize the impact of potential disruptions at critical nodes and sustain the stability of maritime trade flows.

References

[1] "Transportation Statistics Annual Report" (2024). https: //doi.org/10.21949/e0kq-gf72
[2] Tovar-Minguez, C., Wang, Q., & Jiang, Z. (2022). The relationship between port-level maritime connectivity and port efficiency. Transportation Research Part A: Policy and Practice, 158: 123–137.
[3] UN Conference on Trade and Development. (2023). Maritime Connectivity and Trade: Assessing the Role of LSCI in Trade Costs. UNCTAD Policy Paper No. 72.
[4] UN trade & development. Liner shipping connectivity index, quarterly (analytical), https: //unctadstat.unctad.org/datacentre/reportInfo/US.LSCI
[5] Lutz Kilian, Nikos Nomikos, and Xiaoqing Zhou. (2021) Container Trade and the U.S. Recovery, Working Paper 2108. June 17. https: //doi.org/10.24149/wp2108
[6] Munim, Z.H., Fiskin, C.S., Nepal, B., Chowdhury, M.M.H., 2023. Forecasting container throughput of major Asian ports using the Prophet and hybrid time series models. The Asian Journal of Shipping and Logistics 39 (2): 67–77.
[7] Eskafi, M., Kowsari, M., Dastgheib, A. et al. (2021) A model for port throughput forecasting using Bayesian estimation. Marit Econ Logist, 23, 348–368.
[8] Rashed, Y., Meersman, H., Van de Voorde, E. et al. (2017) Short-term forecast of container throughout: An ARIMA-intervention model for the port of Antwerp. Marit Econ Logist, 19, 749–764.
[9] Dionicio Morales-Ramírez, Maria D. Gracia, Julio Mar-Ortiz. (2025) Forecasting national port cargo throughput movement using autoregressive models. Case Studies on Transport Policy, Volume 19, March, 101322. https: //doi.org/10.1016/j.cstp.2024.101322
[10] Durmaz, Y., Kilic, H., & Nutzman, R. (2024). A causality analysis between container throughput and the liner shipping connectivity index (LSCI). Acta Scientiarum, 5(2): 136–149.
[11] UN trade & development. Port liner shipping connectivity index, quarterly (analytical). (n.d.) https: //unctadstat.unctad.org/datacentre/reportInfo/US.PLSCI
[12] Box, G. E. P., & Jenkins, G. M. (1970). Time Series Analysis: Forecasting and Control. San Francisco: Holden-Day.
[13] A. A. Ariyo, A. O. Adewumi and C. K. Ayo. (2014) Stock Price Prediction Using the ARIMA Model, 2014 UKSim-AMSS 16th International Conference on Computer Modelling and Simulation, Cambridge, UK, pp. 106-112.

DOI: 10.54254/2755-2721/2025.BJ28430

[14] Elman, J. L. (1990). Finding structure in time. Cognitive Science, 14(2): 179–211.
[15] Sherstinsky, A. (2020). Fundamentals of Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network. Physica D: Nonlinear Phenomena, 404, 132306.
[16] Yin, W., Kann, K., Yu, M., & Schütze, H. (2017). Comparative Study of CNN and RNN for Natural Language Processing. ArXiv, abs/1702.01923.
[17] Cho, K., Merrienboer, B.V., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., & Bengio, Y. (2014). Learning Phrase Representations using RNN Encoder–Decoder for Statistical Machine Translation. Conference on Empirical Methods in Natural Language Processing.
[18] R. Dey and F. M. Salem. (2017) Gate-variants of Gated Recurrent Unit (GRU) neural networks, 2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS), Boston, MA, USA, pp. 1597-1600.
[19] Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. Neural Computation, 9(8): 1735–1780.
[20] Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). LSTM network: A deep learning approach for short-term traffic forecast. IET Intelligent Transport Systems, 11(2): 68-75.
[21] LeCun, Y., Boser, B., Denker, J. S., et al. (1989). Backpropagation applied to handwritten zip code recognition. Neural Computation, 1(4): 541–551.
[22] Jia, H., Tang, W., Jin, X., Mu, Y., Ai, D., Yu, X., & Wei, W. (2024). Optimization of a Bayesian game for Peer-to-Peer trading among prosumers under incomplete information via a CNN-LSTM-ATT. Energy and AI, 18, 100437.
[23] Lou, B., Tang, J., Hu, L., & Ye, J. (2025). Multi-source data-driven short-term remaining driving range prediction for electric vehicles: A hybrid CNN-transformer framework. Energy, 334, 137564.
[24] Massy, W. F. (1965). Principal components regression in exploratory statistical research. Journal of the American Statistical Association, 60(309): 234–256.
[25] Green, D. S. S., & Klimenko, A. Y. (2025). A moving window method for time series optimization, with applications to energy storage and hydrogen production. Energy, 318, 134616.