

Article

Forecasting the Demand for Container Throughput Using a Mixed-Precision Neural Architecture Based on CNN–LSTM

Cheng-Hong Yang ^{1,2,*} and Po-Yin Chang ^{1,*}¹ Department of Electronic Engineering, National Kaohsiung University of Science and Technology, Kaohsiung 80778, Taiwan² Ph.D. Program in Biomedical Engineering, Kaohsiung Medical University, Kaohsiung 80708, Taiwan

* Correspondence: chyang@nkust.edu.tw (C.-H.Y.); chang.po.yin@gmail.com (P.-Y.C.)

Received: 29 August 2020; Accepted: 14 October 2020; Published: 15 October 2020



Abstract: Forecasting the demand for container throughput is a critical indicator to measure the development level of a port in global business management and industrial development. Time-series analysis approaches are crucial techniques for forecasting the demand for container throughput. However, accurate demand forecasting for container throughput remains a challenge in time-series analysis approaches. In this study, we proposed a mixed-precision neural architecture to forecasting the demand for container throughput. This study is the first work to use a mixed-precision neural network to forecast the container throughput—the mixed-precision architecture used the convolutional neural network for learning the strength of the features and used long short-term memory to identify the crucial internal representation of time series depending on the strength of the features. The experiments on the demand for container throughput of the five ports in Taiwan were conducted to compare our deep learning architecture with other forecasting approaches. The results indicated that our mixed-precision neural architecture exhibited higher forecasting performance than classic machine learning approaches, including adaptive boosting, random forest regression, and support vector regression. The proposed architecture can effectively predict the demand for port container throughput and effectively reduce the costs of planning and development of ports in the future.

Keywords: container throughput forecasting; convolutional neural network (CNN); deep learning; long short-term memory (LSTM); recurrent neural network (RNN)

1. Introduction

Globalization has facilitated the rapid growth of international trade. The degree of containerization has emerged as a competitive advantage for countries in this globalized trading environment [1]. Island countries rely heavily on imports and are considerably affected by changes in the global economy. In this rapidly changing competitive environment, port operations, construction, and upgrading of port facilities are critical. Governments and industries accord considerable attention to these aspects. Container throughput is essential information for port infrastructure investment and construction, and it is a substantial irreversible investment. The process of construction of a new port or upgrading an already existing port requires considerable time, and availability of port facilities is considerably restricted during the construction process [2]. If the trade volume and container throughput of the seaport is unbalanced, then the performance and international competitiveness of the seaport is severely affected [3]. Therefore, failure to forecast the container throughput may result in substantial financial losses. Although researchers have developed various approaches for forecasting container throughput

based on a linear system, most methods are an extension of the classic time-series model, including autoregressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA) [4], exponential smoothing [5], gray forecasting [6], and regression analysis [7]. However, a container throughput time series consists of linear and nonlinear trends. Therefore, achieving effective forecasting is difficult. In contrast to previous studies, in our work, a forecasting method was developed based on a mixed-precision neural architecture to forecast future container throughput. This study focused on investigating the capability of the convolutional neural network (CNN) for learning the strength of the features of the container throughput and the effectiveness of long short-term memory (LSTM) to identify the crucial internal representation of time series depending on the strength of the features. In particular, the contribution of this study consists of three parts. First, the state-of-art method in a mixed-precision neural architecture was investigated for throughput forecasting. The proposed hybrid CNN–LSTM model exhibited considerable advantages of both CNN and LSTM. Finally, the robustness of this mixed-precision neural architecture was tested using the five-container throughput with different trends. This study is the first to use a mixed-precision neural network to forecast container throughput. This work provides a solid foundation for a novel neural network architecture in order to guide future studies to develop more effective approaches based on our foundation.

The remainder of this paper is organized as follows. Section 2 presents a brief review of previous works related to container throughput forecasting. Section 3 introduces the CNN and LSTM and presents the CNN–LSTM forecasting approach in container throughput forecasting. Section 4 presents the empirical results and a discussion. Finally, Section 5 summarizes the conclusions and future research directions.

2. Literature Review

Container throughput forecasting first emerged in the 1980s. Since then, considerable development has been achieved in the field [8,9]. Most forecasting methods, such as autoregressive integrated moving average (ARIMA), seasonal ARIMA (SARIMA) [4], exponential smoothing [5], gray forecasting [6], and regression analysis [7], assume a linear trend. However, a container throughput time series consists of linear and nonlinear trends. Therefore, achieving effective forecasting is difficult. An increasing number of studies have developed nonlinear forecasting approaches to address nonlinear systems.

Niu presented a novel hybrid decomposition ensemble method to decompose container transportation volume series into low-frequency and high-frequency components. This method involves the use of the moving-average model and support vector regression (SVR) to process the two components and generate the final throughput forecasting [10]. Chen used genetic programming (GP), decomposition methods, and SARIMA methods to predict Taiwan's container throughput and concluded that the GP model outperformed the other two methods [11]. Syafi'i et al. introduced a vector error correction method based on a vector autoregression to predict the container transportation volume [12]. Their method was promising and had considerable development potential. Guo et al. proposed an improved gray Verhulst model, which overcomes the problem of the increase in the forecast error when the container transportation volume exhibits an S-shaped trend. Their finding indicated that the proposed gray Verhulst model not only achieved high prediction accuracy but also retained the strengths and characteristics of the gray system model [13].

Gosasang et al. used multilayer perceptron (MLP) and linear regression (LR) forecasting models with multivariate data that collect data from public institutions to forecast cargo throughput; the models were compared in terms of their root mean squared errors (RMSEs) and mean absolute errors (MAEs). Their results suggested that MLP forecasting outperformed LR forecasting [14]. Geng et al. proposed a hybrid model with multivariate adaptive regression splines for selecting SVR input variables. Furthermore, SVR fails to provide reasonable performance without appropriate parameters. This study used the chaotic simulated annealing particle swarm optimization algorithm to determine the optimal SVR parameters and improve the model efficiency [15].

Deep learning has yielded promising results in many research areas and has attracted considerable academic and industry attention [16]. Deep learning has considerable potential for use in several applications, such as supervised (classification) and unsupervised learning (clustering) tasks, natural language processing, dimensionality reduction, biology, medicine, and motion modeling [17–22]. The advantage of deep learning is that it creates a mixture function set of numerous nonlinear transformations and useful expressions to produce more abstraction and consequently more benefits [17].

None of the aforementioned studies have involved considering a mixed-precision neural architecture to forecast container throughput. In this study, the univariate time-series forecasting method based on a mixed-precision neural architecture was proposed to forecast future container throughput.

3. Methods

3.1. Proposed Method

Among deep learning techniques, LSTM networks and CNNs are probably the most popular, efficient, and widely used deep learning techniques [23]. These models are typically used for time series because the appropriate use of different networks can achieve superior results. The LSTM model is inherently designed with a special storage mechanism that can effectively capture sequential schema information [24]. CNNs achieve superior data noise to extract important features and increase the performance of the final prediction model. However, a typical CNN is suitable for processing spatial autocorrelation data but cannot correctly handle complex and long-term time dependence [17]. Therefore, considering the strengths of the two deep learning techniques, we proposed a novel mixed-precision architecture based on the CNN–LSTM hybrid neural network that can better handle nonlinear systems and enhance the forecasting performance. The model architecture is presented in Figure 1. The proposed mixed-precision architecture primarily comprises an input, a CNN, LSTM, and a fully connected layer. The input layer loads time-series data as the input and transfers them to a CNN for the extraction of high-level features (the features extractor in Figure 1). The output is then sent to an LSTM recurrent neural network (the regressor in Figure 1) and then to a fully connected layer, which then generates the forecast for container throughput. The operations of the layers are briefly introduced in the subsequent sections.

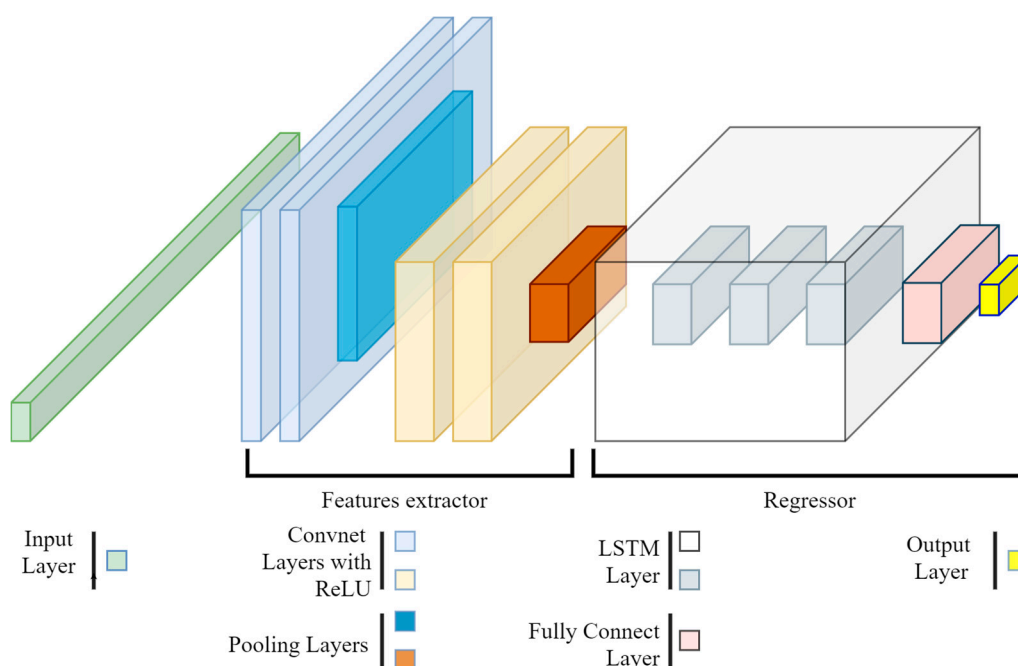


Figure 1. Overall mixed-precision architecture of the CNN–LSTM neural network.

3.1.1. Input Layer

Given a sequence of numbers for a time-series data set, to apply machine learning methods to the data set, we restructure the time-series data as a supervised learning problem. Hence, the first layer of the network restructures time-series data as a supervised learning problem by using the value at the previous time step to predict the value at the next time step.

3.1.2. CNN Layer

A classic CNN is composed of two specially designed data preprocessing, convolutional, and pooling layers. Their characteristics can filter input data and extract useful information. In general, the fully connected network layer followed the CNN and received the filtered input data as input data. The goal of the convolution layer is to use convolution operations to process image data to generate new feature values [16]. A convolution kernel (filter) is a unit used for convolution operation. The form of a filter can be considered the size of an area formed by a matrix, and this area slides to spread over the input data. The feature values specified by the coefficient value and the size of the applied filter are obtained on convolution. By applying different filters to the input data, multiple convolved feature maps can be generated. These feature maps are typically more useful than the original input data and considerably improve the performance of the model.

In CNNs, the convolution layer is followed by a pooling layer. This technique is a subsampling technique that extracts specific values from features maps and generates a low-dimensional matrix. Similar to the convolution layer operation, the pooling layer uses a small sliding window, which takes the value of each side of the convolution feature as the input and outputs a new value. The value is typically max pooling and average pooling, which are the maximum and average values of each moving range, respectively. The new matrix generated by the pooling layer can be considered a concise version of the convolution features generated by the convolution layer. Pooling can improve the system and make it more robust because small changes in the input do not change the output value of the pool.

3.1.3. LSTM Layer

For effective extraction of the temporal relationship between time lags, an LSTM layer was established followed the CNN layer. To address the vanishing gradient concerns of a recurrent neural network, the memory scheme in the LSTM network comprises three gate types: forget gates, input gates, and output gates [24]. The operation of the LSTM is illustrated in Figure 2. Table 1 presents a summary of the parameters used in the LSTM layer.

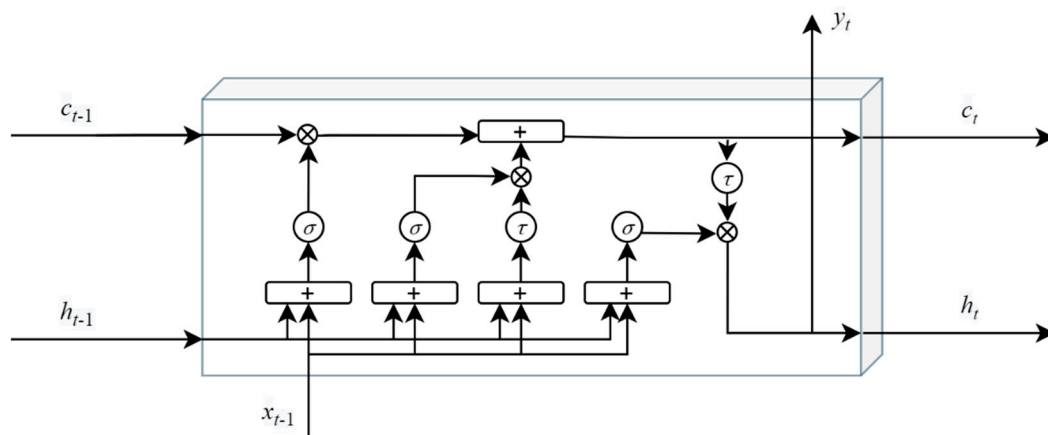


Figure 2. Structure of an LSTM neural network. Here, x_{t-1} is the input, h_{t-1} is the previous cell output, y_t is the output, C_{t-1} is the memory of the previous cell, h_t is the current cell output, C_t is the current cell memory, F_t is the forget gate, I_t is the input gate, and O_t is the output gate.

Table 1. Notations of the variables in LSTM.

Notation	Explanation
LSTM Input	
x_t	Current input vector from the deep features extracted from the CNN
c_{t-1}	Memory cell state from the previous LSTM unit
h_{t-1}	Previous hidden state output from the previous LSTM unit
LSTM Output	
c_t	New updated memory
h_t	Current hidden state output
y_t	Current output
LSTM Nonlinearities	
σ	Activation function: Sigmoid
τ	Activation function: Tanh
LSTM Vector Operation	
\otimes	Pointwise multiplication
\oplus	Pointwise addition
Weights	
W_f, W_i, W_o, W_c	Recurrent weight vector for the forget gate, input gate, output gate, and memory cell
U_f, U_i, U_o, U_c	Input weight vector for the forget gate, input gate, output gate, and memory cell
b_f, b_i, b_o, b_c	Input bias vector for the forget gate, input gate, output gate, and memory cell
Gates	
F_t	Forget gate: responsible for determining whether the old information is remembered
I_t	Input gate: responsible for determining whether the input is adopted
O_t	Output gate: responsible for determining whether the output is adopted
Loss Function	
θ	The parameters need to learning from LSTM
y	Ground truth value
\hat{y}	Predicted value
y_i	Denotes 12 month container throughput data that must be predicted, where $i = 1, 2, \dots, 12$

The gates, hidden outputs, and cell states can be represented as follows [25]:

$$F_t = \sigma(h_{t-1}W_f + x_tU_f + b_f) \quad (1)$$

$$I_t = \sigma(h_{t-1}W_i + x_tU_i + b_i) \quad (2)$$

$$O_t = \sigma(h_{t-1}W_o + x_tU_o + b_o) \quad (3)$$

$$\hat{C}_t = \tau(h_{t-1}W_c + x_tU_c + b_c) \quad (4)$$

$$C_t = F_t \otimes C_{t-1} \oplus I_t \otimes \hat{C}_t \quad (5)$$

$$h_t = O_t \otimes \tau(C_t) \quad (6)$$

The mean squared error loss is calculated as follows:

$$\text{loss}(y, \hat{y}) = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2 \quad (7)$$

The loss function is represented as follows:

$$\operatorname{argmin}_{\theta} L(\theta) = \sum_{i=1}^N \operatorname{loss}(y_i, \hat{y}_i) \quad (8)$$

$$\theta = [U_f, U_i, U_c, U_o, W_f, W_i, W_c, W_o, b_f, b_i, b_c, b_o] \quad (9)$$

where σ is the sigmoid function, which is an activation function that converts an input value into a value between 0 and 1. Moreover, τ is the *Tanh* function that converts the output value to a value in the range of -1 to 1 . Then, the value can be used to determine which specific information should be excluded or retrained. The terms x_t , h_t , and C_t denote the input value, hidden value, and cell state, respectively, at time step t , respectively; the terms h_{t-1} and C_{t-1} are the hidden value and cell state at time step $t-1$, respectively. The legends \otimes and \oplus are pointwise multiplication and addition, respectively. Here, U_f , U_i , U_c , and U_o are their input weights; W_f , W_i , W_c , and W_o are their recurrent weights; and b_f , b_i , b_c and b_o are their biases, which must be determined through learning in training processes.

The three stages of LSTM to control three gates are illustrated in Figure 1. First, for the previous node h_{t-1} , x_t , content selected for forgetting is generated through (1) for the calculation of F_t , in which 0 represents complete erasure and 1 represents complete retention. Next, the system must update the cell status with (5). Eventually, the final output is confirmed using (8) and (9). The output of LSTM is passed as the input of fully connection layer, then generate the final container throughput forecasts. The unfolded chain structure of LSTM in an input flow $[x_1, x_2, \dots, x_k]$ is depicted in Figure 3, $\{h_j, c_j\}$ where $j = 1, 2, \dots, k$. h_j is the hidden state at time j and c_j is the cell state at time j . In the recurrent regression, the LSTM unit uses previous state (h_{j-1}, c_{j-1}) and compute container throughput $\{y_j\}$ where $j = 1, 2, \dots, k$. Using this method, the past data can be transferred recursively in the whole loop of LSTM.

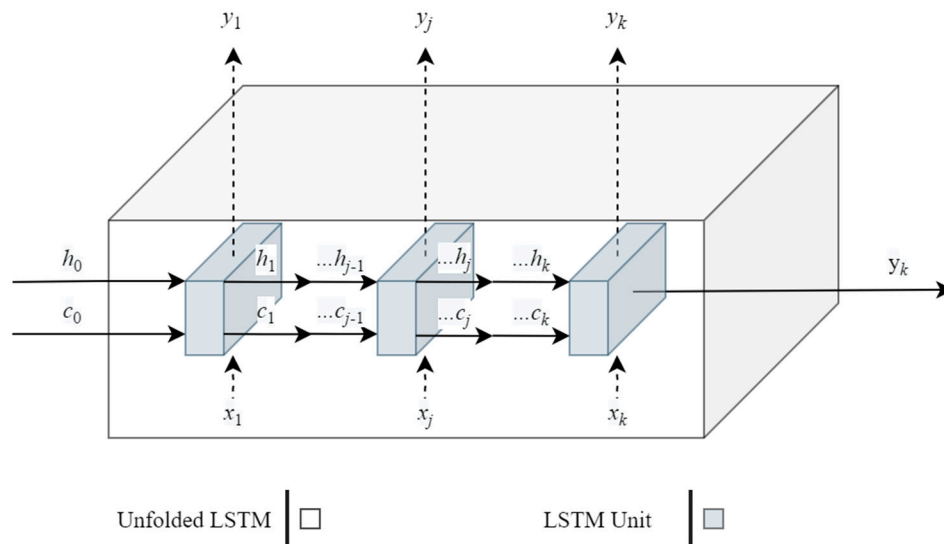


Figure 3. The unfolded chain structure of LSTM in time sequence.

3.2. Baseline Models

This section introduces the baseline models, including statistical learning and machine learning methods.

3.2.1. Random Forest Regression (RFR)

RF is an ensemble method in machine learning for classification and regression problems [26]. Typical RF implementation incorporates some set of binary decision trees. Forecasting is conducted

using most of the trees (in the classification) to vote or to average their outputs (in the regression). In this paper, container throughput forecasting is a regression problem. Therefore, RFR was selected as a baseline model.

3.2.2. Linear Regression (LR)

LR is a standard statistical technique, which enables researchers to investigate a relationship considering the roles that multiple independent variables play in variance in a single dependent variable [27]. The aim of the model fitting process in LR is to identify the coefficients $W = (w_1, \dots, w_n)$ that can minimize the residual sum of squares between the observed samples in the data set.

3.2.3. Adaptive Boosting (AdaBoost)

In AdaBoost, a boosting method proposed by Freund and Schapire [28], the samples misclassified by a previous basic classifier were strengthened, and the weighted total samples were used to train the next basic classifier. A novel weak classifier was added in each round until the error rate was sufficiently low to satisfy a predetermined threshold or until a predetermined maximum number of iterations is performed.

3.2.4. SVR

Vapnik et al. proposed support vector machines (SVMs) to address classification problems [29]. For decades, SVMs have delivered promising results in several research areas such as solar forecasting, ECG signal preprocessing, and information security [30]. Drucker proposed SVR as an SVM-based solution for regression problems. The basic principle in SVR is to calculate a nonlinear function that maps training data to a high-dimensional feature space [31].

3.3. Rolling Forecasting

Rolling forecasting is used as a forecasting scheme and for time-order maintenance in time-series models and machine learning models. In this scheme, a fixed window was used, and the value was updated in the fixed window with each newly predicted value. This process involved removing historical data and adding future data so that the fixed window always retained the same amount of time-series data. First in/first out (FIFO) is an updating strategy in rolling forecasting; this type of strategy is referred to as a continuous or recursive strategy. An example of FIFO for rolling forecasting is displayed in Figure 4.

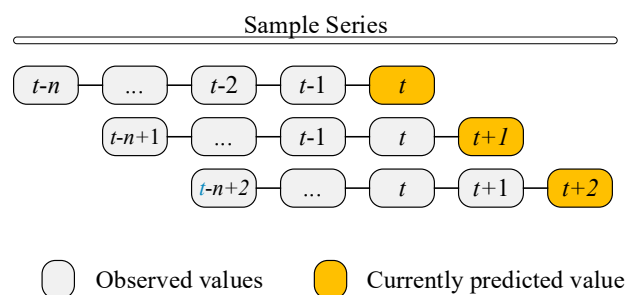


Figure 4. Sequences for observed value and predicted value in rolling-based forecasting.

3.3.1. Parameter Settings

The predictive capability of a forecasting method depends on whether appropriate parameters are used to formalize the model. For machine learning approaches, such as RFR, ADABOOST and MLP, the default parameter settings in scikit-learn were adopted [32]. For the statistical learning method SVR, the kernel function radial basis function is used. However, the initial parameters are crucial for the performance of SVR. Furthermore, a grid search method is used to determine the appropriate

parameter settings for C and σ . SVR increments the parameters exponentially; thus, the search spaces of SVR are set as $C = [2^0, 2^1, \dots, 2^{22}]$, $\sigma = [2^{-10}, 2^{-9}, \dots, 2^0]$.

Evaluation of search parameters for deep learning can be considerably expensive—the predetermined architecture and parameter settings were determined empirically. The CNN–LSTM model consists of two convolutional layers of 64 and 128 filters with sizes of 2, respectively. They were followed by one max pooling layer with a size of 2, a LSTM layer of 100 units, a fully connected layer of 32 neurons, and a neuron as the output layer. The model was trained using the Adam optimizer [33] and a ReLU activation function [34] for 1000 epochs with early stopping. Each data set was divided into a training set and testing set. The training set used for training the models; it ranged from 2001 of monthly data until the end of 2018. The testing set was used for testing the forecast accuracy; it consisted of the monthly data for 2019.

3.3.2. Forecasting Performance Criteria

Root mean square error (RMSE) and mean absolute percentage error (MAPE) are two well-known statistical measures. These measures were performed in the study for assessing the deviation of forecasted values from actual values. RMSE is obtained by first summing the squares of the residual and then square root of the value. This indicator is more sensitive to outliers than MAPE. MAPE divides each error value by the actual value, which results in a tilt: if the actual value at a certain time point is low but the error is large, the MAPE value will be greatly affected. RMSE and MAPE are described in (10) and (11), respectively.

$$\text{RMSE} = \sqrt{\frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2} \quad (10)$$

$$\text{MAPE} = \frac{1}{N} \sum_{i=1}^N \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (11)$$

where y_i represents the actual value, \hat{y}_i the forecasted value, and N the sample size. Lower RMSE and MAPE values exhibit higher forecasting accuracy.

Mean absolute scale error (MASE) is a scale-free error measure that compares each error with the average error at baseline. The advantage of MASE is that it never represents an undefined or infinitely large value. Hence, MASE is a suitable metric for intermittent demand series when there is zero demand in the forecast. MASE is described in (12).

$$\text{MASE} = \frac{\frac{1}{J} \sum_j |e_j|}{\frac{1}{T-1} \sum_{t=2}^T |y_t - y_{t-1}|} \quad (12)$$

where e_j is the forecast error for a given period J , which is the number of forecasts. $e_j = y_j - f_j$, y_j is the ground truth value, and f_j is the forecast value. The denominator is the mean absolute error of the one-step “naive forecast method” on the training set ($t = 1, 2, \dots, n$) [35].

Symmetric mean absolute percentage error (SMAPE) is an accuracy metric based on percentage (or relative) errors. SMAPE is defined as follows [36]:

$$\text{SMAPE} = \frac{1}{N} \sum_{i=1}^N \frac{2 * |y_i - \hat{y}_i|}{|y_i| + |\hat{y}_i|} \quad (13)$$

where y_i represents the ground truth value, \hat{y}_i denotes the predicted value, and N denotes the sample size. The absolute difference between y_i and \hat{y}_i is multiplied by 2 and divided by the absolute values of the ground truth y_i and the predicted value \hat{y}_i . The value is summed for every fitted sample, which is then divided by N . Chen [36] used absolute values in the denominator of the SMAPE equation to avoid obtaining a negative value; this approach was also used by Hyndman and Koehler [37].

Mean arctangent absolute percentage error (MAAPE) is used when ground truth is zero. MAAPE is defined as follows:

$$\text{MAAPE} = \frac{1}{N} \sum_{i=1}^N \arctan\left(\left|\frac{y_i - \hat{y}_i}{y_i}\right|\right) \quad (14)$$

where y_i represents the ground truth value, \hat{y}_i denotes the predicted value, and N denotes the sample size. Unlike the regular absolute percentage error (APE), for MAAPE, the arctangent absolute error approaches $\pi/2$ when division by zero occurs.

Mean bounded relative absolute error (MBRAE) is derived from two measures, relative absolute error (RAE) and bounded RAE (BRAE) [36], and is defined as follows:

$$\text{RAE} = \left| \frac{e_t}{e_t^*} \right| \quad (15)$$

where $e_t^* = y_i - \hat{y}_i^*$, where \hat{y}_i^* denotes the forecast at time t by using the benchmark method. Because RAE has no upper bound, it can be excessively large or undefined when $|e_t^*|$ is small or equal to zero. This problem can be easily addressed through the addition of $|e_t|$ to the denominator of the RAE equation, which introduces bounded RAE (BRAE), defined as follows:

$$\text{BRAE} = \frac{|e_t|}{|e_t| + |e_t^*|} \quad (16)$$

Based on BRAE, the metric MBRAE can be defined as follows:

$$\text{MBRAE} = \frac{1}{N} \sum_{i=1}^N \frac{|e_t|}{|e_t| + |e_t^*|} \quad (17)$$

Each measure has its characteristics—the use of multiple measures is helpful to understand the performance of the model in all aspects.

4. Results and Discussion

4.1. Data Source

The container throughput data of five ports—Anping, Hualien, Kaohsiung, Taichung, and Suao—in Taiwan were used in the study. The seasonal adjusted monthly levels were retrieved from Taiwan International Ports Corporation. Because of limited data availability, the samples were collected from 2001M1 to 2019M12. Container throughput capacity is defined as the estimated total cargo that can be processed [2]. For containers, it can be expressed as twenty-foot equivalent unit (TEU) containers in a period. The container throughput data are displayed in Figure 5. As illustrated in the figure, the container throughput series of all ports exhibited a different time trend. Kaohsiung port throughput exhibited a steady trend, Hualien, and Suao ports revealed downward trends, Anping port's throughput grew from 2001 to its peak in 2010, and then returned to similar throughput in 2001.

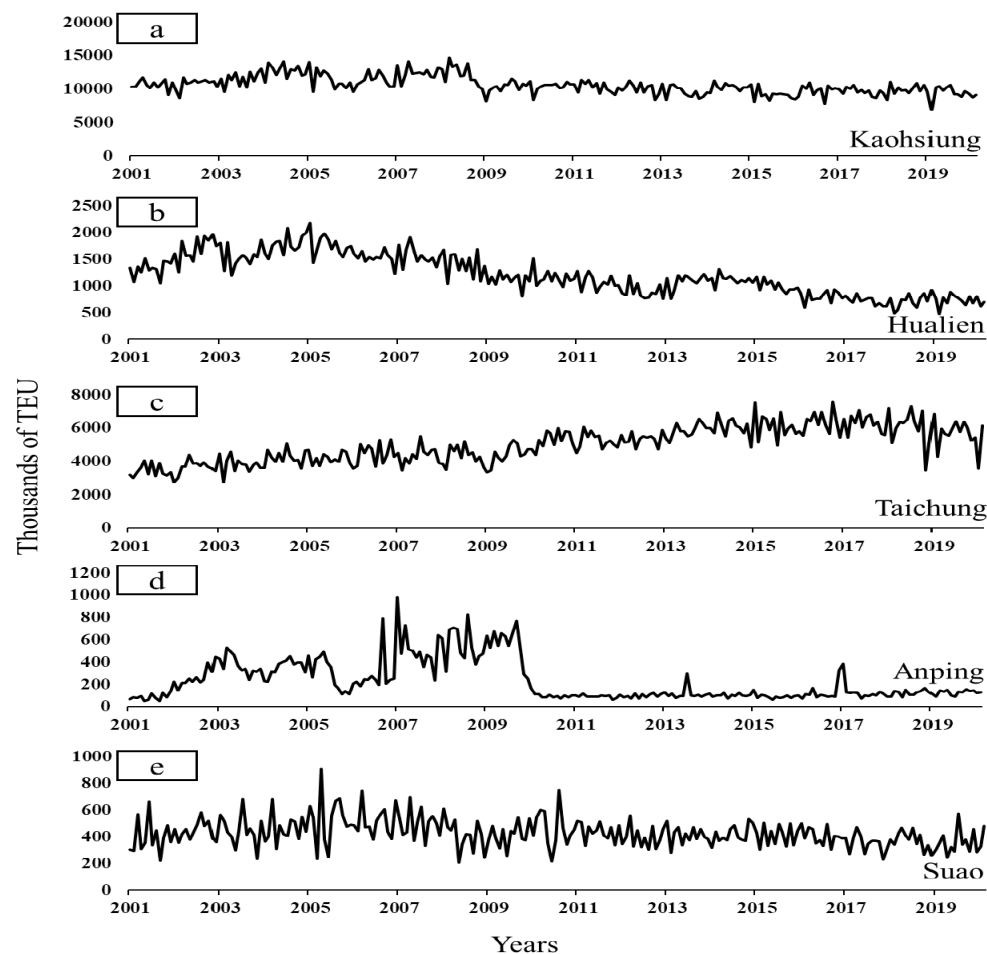


Figure 5. Raw data of monthly container throughput from 2001 to 2019: (a) Kaohsiung, (b) Hualien, (c) Taichung, (d) Anping, and (e) Suao. Unit: thousands of twenty-foot equivalent units (TEU).

Overall, each series of data exhibited varying degrees of variations, which hampered forecasting based on fundamentals. Table 2 presents the descriptive statistics of monthly container throughput for the five ports from 2001 to 2019, in all 228 samples for each port were collected. The table displayed the following information of the ports: maximum (Max), minimum (Min), mean, median, first quartile (Q1), third quartile (Q3), interquartile range (IQR), standard deviation (SD), and coefficient of variance (CV, %). For container throughput in Table 1, Anping exhibited the highest degree of variation in terms of its CV value (approximately 81.4%). Taichung exhibited the lowest CV value (19.2%) of all ports, implying a low variability in the container throughput of Taichung. These results revealed that the five ports exhibited different variations and trends.

Table 2. Descriptive statistics for five ports in Taiwan.

Port	Max	Min	Mean	Median	Q1	Q3	IQR	SD	CV (%)
Kaohsiung	189.126	29.504	71.283	53.171	53.171	113.914	60.743	36.385	51.0%
Hualien	166.882	28.328	84.094	75.604	56.852	107.491	50.638	32.222	38.3%
Taichung	67.146	25.114	44.416	43.797	38.408	48.918	10.510	8.531	19.2%
Anping	97.937	4.878	23.047	13.194	9.621	33.245	23.624	18.764	81.4%
Suao	24.467	3.118	12.357	12.201	9.649	14.848	5.199	3.968	32.1%

Unit: data are represented in thousands of twenty-foot equivalent units (TEU).

The throughput data are visualized in the form of the box and whisker plots in Figure 6 to present an informative representation of data distribution. In particular, the middle line of the box is the median. The median splits the data set into a bottom half and a top half. The bottom line of the box is the median of the bottom half or Q1. The top line of the box represents the median of the top half or Q3. The whiskers (vertical lines) extend from the ends of the box to the minimum value and maximum value, respectively. The IQR is defined as the distance between the Q1 and Q3: $IQR = Q3 - Q1$. The data points are considered outliers if they exceed 1.5 times the IQR below the Q1 or 1.5 times the IQR above Q3. The container throughput of five ports in order is Kaohsiung, Taichung, Hualien, Suao, and Anping. The maximum and minimum, Q1, Q3 mean, median, and outliers for each port are displayed in Figure 6.

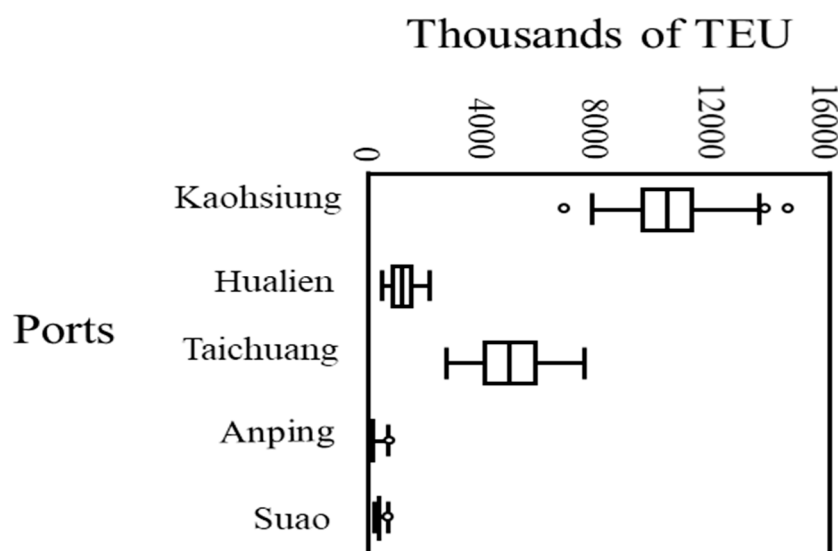


Figure 6. Box and whisker plot of monthly container throughput of five ports in Taiwan from 2001 to 2019. Unit: data are represented in thousands of twenty-foot equivalent units (TEU).

4.2. Comparison of Models

The metrics MAPE, RMSE, MASE, SMAPE, MAAPE, and MBRAE were used for each method to determine the predictive capabilities of the two approaches for developing forecasting models: The results are listed in Table 3. CNN-LSTM demonstrated performance superior to that of the other methods.

The advantage of CNN-LSTM over time-series models is that CNN-LSTM does not determine whether data are stationary and these models do not consider whether other statistical tests should be used. The characteristics associated with the training data were learned through CNN-LSTM. Furthermore, CNN-LSTM outperformed other machine learning methods in handling forecasting problems in most cases. The overall forecasting of the various models is presented in Figure 7. The histogram in the figure displays the mean MAPE and mean RMSE for exports and imports in each model. The MAPE and RMSE averages of the CNN-LSTM model were lower than those of other models, which indicates that the CNN-LSTM model outperformed the other models.

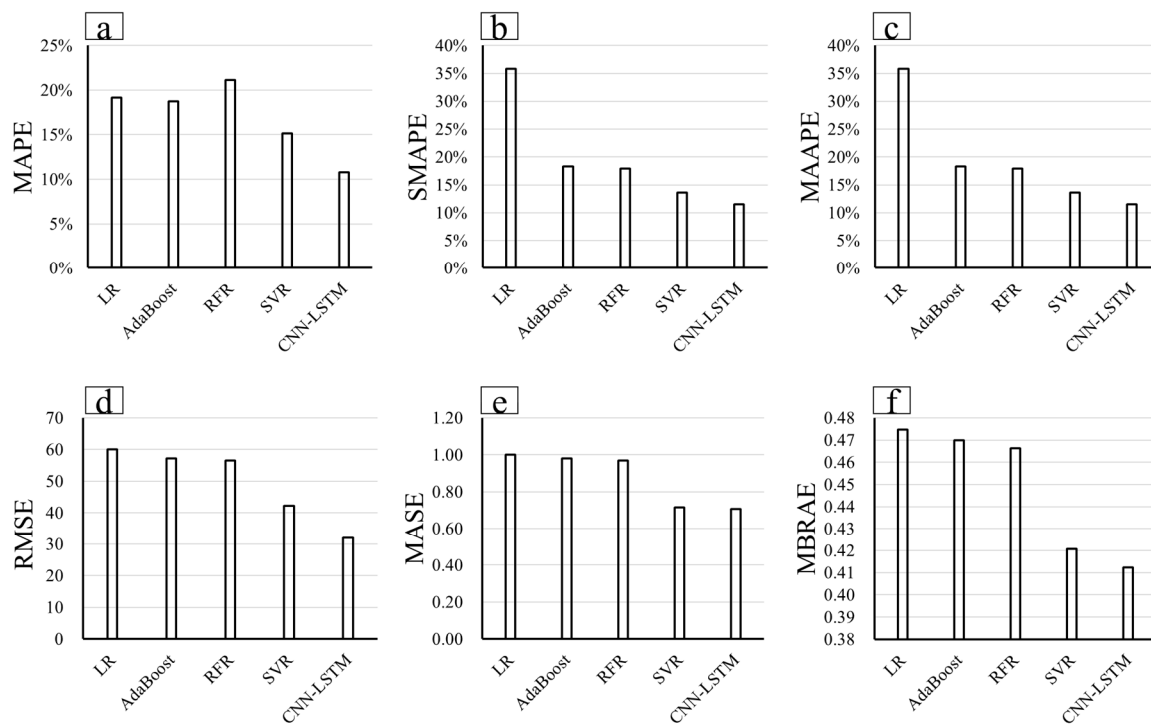
Diebold and Mariano [38] and Derrac et al. [39] revealed that the Wilcoxon signed-rank test [40] and the Friedman test [41] are reliable statistical benchmarks. These benchmarks have been widely applied in studies of machine learning models [42–44]. We used both methods to compare the forecasting accuracy of the applied CNN-LSTM and the performances of the LR, AdaBoost, RFR, and SVR. Both statistical tests were simultaneously implemented with a significance level $\alpha = 0.05$. The results in Table 3 revealed that the CNN-LSTM model could provide a significantly better outcome in terms of forecasting performance than other models.

Table 3. Model comparison and ranking using the metrics MAPE, RMSE, MASE, SMAPE, MAAPE, and MBRAE obtained for the five ports in Taiwan.

		LR	AdaBoost	RFR	SVR	CNN-LSTM
Kaohsiung	MAPE (%)	10.93	9.92	9.85	8.40	5.78 *
	SMAPE (%)	96.90	9.20	9.20	7.50	5.70 *
	MAAPE (%)	10.70	9.70	9.70	8.10	5.60 *
	RMSE	128.21	114.39	113.90	108.05	72.55 *
	MASE	0.96	0.87	0.87	0.72	0.87
	MBRAE	0.51	0.51	0.52	0.48	0.51
Hualien	MAPE (%)	14.27	14.15	12.87	12.43	7.46 *
	SMAPE (%)	14.00	13.60	12.30	11.00	9.60 *
	MAAPE (%)	14.00	13.80	12.60	11.90	9.70 *
	RMSE	11.71	11.65	10.81	10.57	6.67 *
	MASE	0.68	0.67	0.61	0.54	0.62
	MBRAE	0.39	0.40	0.37	0.36	0.37
Taichung	MAPE (%)	19.26	19.02	32.15	11.18	9.39 *
	SMAPE (%)	18.90	18.50	19.10	10.70	10.40 *
	MAAPE (%)	18.10	17.90	18.40	10.90	10.80 *
	RMSE	143.50	143.21	141.35	79.73	68.98 *
	MASE	1.45	1.43	1.47	0.84	0.75 *
	MBRAE	0.53	0.53	0.53	0.47	0.43 *
Anping	MAPE (%)	22.23	21.85	21.86	16.78	12.47 *
	SMAPE (%)	20.20	20.70	19.90	14.70	11.50 *
	MAAPE (%)	20.90	20.60	20.70	15.90	12.30 *
	RMSE	3.16	3.27	3.09	2.42	1.85 *
	MASE	1.05	1.079	1.04	0.76	0.72 *
	MBRAE	0.45	0.46	0.46	0.38	0.39
Suao	MAPE (%)	29.01	28.69	28.88	26.88	18.65 *
	SMAPE (%)	29.30	29.50	29.20	24.20	20.10 *
	MAAPE (%)	27.30	26.90	27.20	25.50	18.50 *
	RMSE	13.43	13.79	13.42	10.55	9.67 *
	MASE	0.85	0.85	0.85	0.72	0.57 *
	MBRAE	0.46	0.45	0.46	0.42	0.35 *
Avg	MAPE (%)	19.14	18.73	21.12	15.14	10.75 *
	SMAPE (%)	35.86	18.30	17.94	13.62	11.46 *
	MAAPE (%)	18.20	17.78	17.72	14.46	11.38 *
	RMSE	60.00	57.26	56.52	42.26	31.94 *
	MASE	1.00	0.98	0.97	0.71	0.70 *
	MBRAE	0.48	0.47	0.47	0.42	0.41 *

The values for MAPE, SMAPE, and MAAPE are display as percentage. LR: linear regression, AdaBoost: adaptive boosting, RFR: random forest regression, and SVR: support vector regression units. Data are represented in thousands of twenty-foot equivalent units (TEU). Bold indicates the best results. * Indicates a significant statistical difference between LSTM and other models.

The outcomes obtained through metrics and statistical tests represent the performance of different models. In Figure 8A, the x axis denotes the time (January–December, 2019), and the y axis denotes container throughput, presented in thousands of twenty-foot equivalent units (TEU). The cross-trend line denotes the ground truth. Prediction outcomes of different models are indicated as follows: CNN-LSTM (triangle), SVR (circle), RFR (plus), AdaBoost (rectangle), and LR (diamond). We visualized the forecasted value and the actual value to understand the difference between the predicted and actual trends. The subfigures (A) Kaohsiung, (B) Hualien, (C), Anping, (D) Suao, and (E) Taichung highlight that the value predicted by CNN-LSTM (denoted by the red line) is the closest to the actual value and trend. Although other methods yielded similar trends, they were far from the actual trend.



Methods

Figure 7. Comparison of the average performance of different models using metrics. (a) MAPE, (b) SMAPE, and (c) MAAPE are percentage metrics. (d) RMSE, (e) MASE, and (f) MBRAE are relative metrics. LR: linear regression, AdaBoost: adaptive boosting, RFR: random forest regression, and SVR: support vector regression units, and CNN-LSTM: convolutional neural network and long short-term memory.

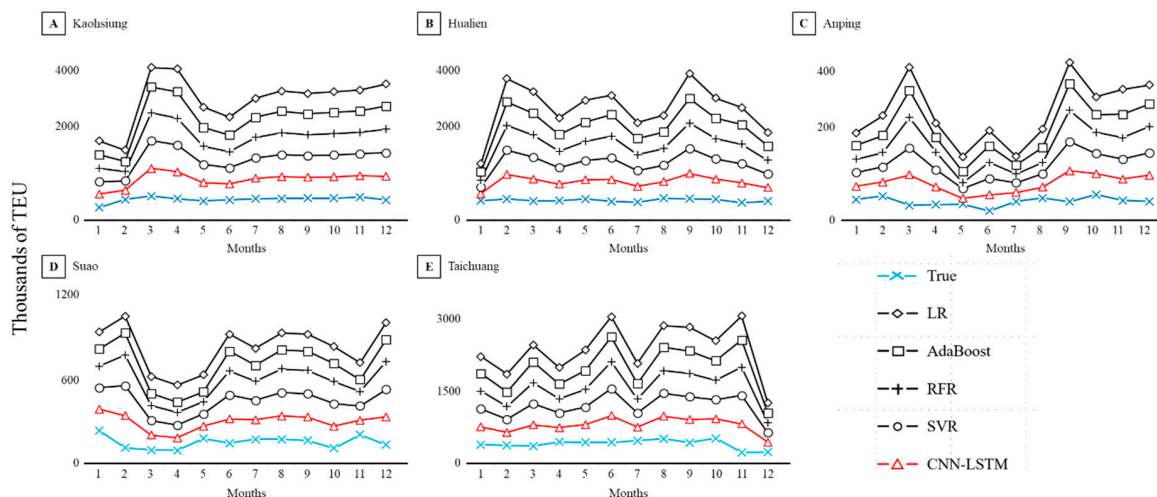


Figure 8. Container throughput forecasting for five ports. LR: linear regression, AdaBoost: adaptive boosting, RFR: random forest regression, and SVR: support vector regression units, and CNN-LSTM: convolutional neural network and long short-term memory.

The data distribution of residuals (the deviation between the actual and predicted value) is visualized as a box and whisker plot (Figure 9) to distinguish the model's predictive performance. The box and whisker plot helps observe the data distribution and the distance between median and zero in the residual distribution. The smaller box implies that the residuals were distributed in a more concentrated manner. A median (middle line) closer to 0 in the y axis indicates a more accurate

prediction by the model. In Figure 9, the residual distribution of CNN–LSTM (the box on the extreme right) is smaller and more concentrated than that of other methods. Thus, the forecasting capability of CNN–LSTM method is more stable. Furthermore, the median of the CNN–LSTM model is closer to 0 except Kaohsiung, which indicated that the percentage of residuals of the CNN–LSTM model is also smaller than others in most cases. Therefore, the forecasting capability of the CNN–LSTM is more stable and robustness than other models.

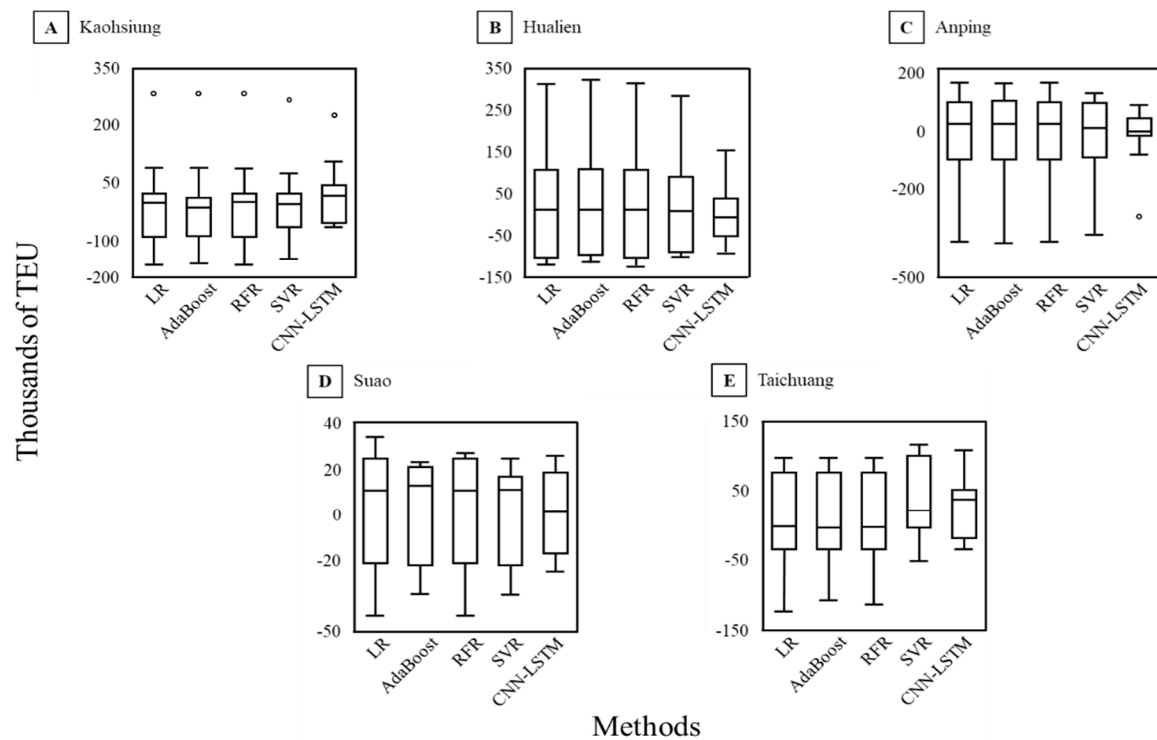


Figure 9. Box and whisker plot of the distribution of the residual for each model.

In the model training process, a validation set can be used to evaluate whether model overfitting occurred. Generally, after the performance of the validation set becomes stable, if the training continues, the loss of the training set will continue to improve, but the efficiency of the loss of the validation set will deteriorate, which indicates that the model is overfitting. Hence, the validation set can also be used to determine when to stop training. We considered 10% of the training set as the validation set. Figure 10 shows that the trends of the validation set of the container throughput data are stable according to the training result obtained using the proposed method. In most cases (Kaohsiung (Figure 10a), Hualien (Figure 10b), Suao (Figure 10d), and Taichung (Figure 10e)), the trend of the validation set was stable before 100 epochs. For Anping (Figure 10c), the trend of the validation set was stable at approximately 600 epochs.

This study revealed that the mixed-precision model based on CNN and LSTM has a superior predictive capability for container throughput data. The experiment results revealed that the CNN and LSTM layer led to the superior performance of the mixed-precision model. Therefore, using a convolutional layer for the feature extraction of time-series data is effective, and modeling the time-dependent relationship through LSTM is a feasible solution. However, for throughput prediction, the method proposed in this study is limited to univariate time-series data. To address the more complex multidimensional data, a more complex feature extraction architecture must be constructed. In the future, researchers should consider learning from the various convolution layers of image processing, such as dilated convolution, depth separable convolution, and deconvolution, which might further improve the model performance.

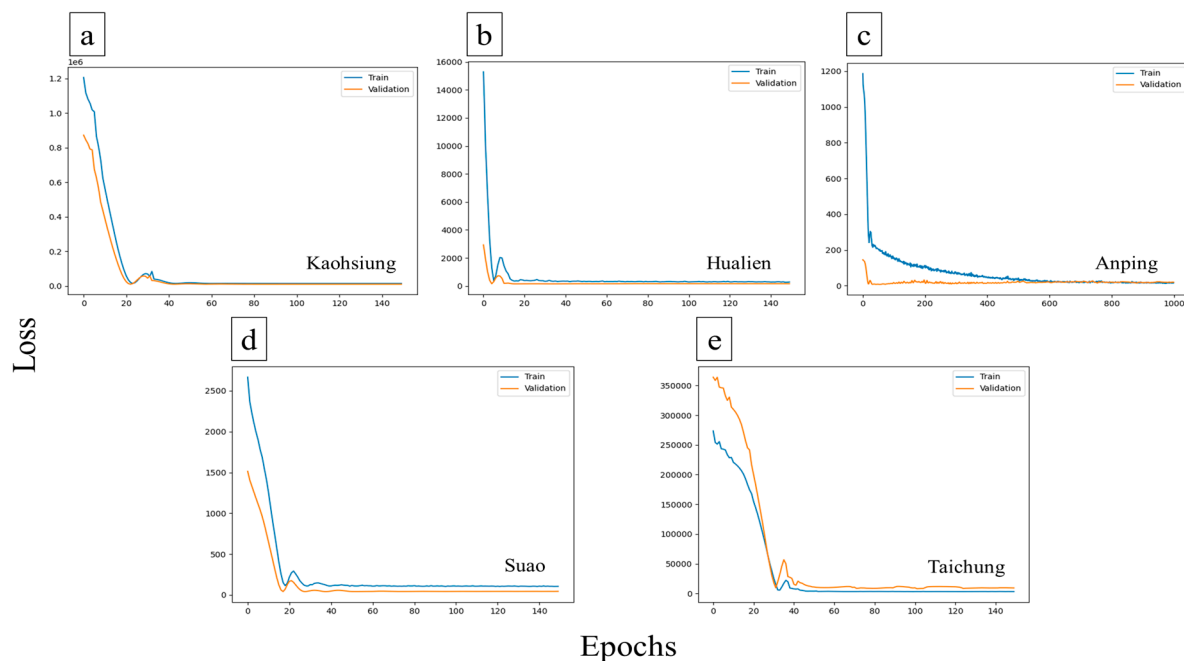


Figure 10. Validation set trends for five ports data with CNN-LSTM. (a) Kaohsiung, (b) Hualien, (c) Anping, (d) Suao, and (e) Taichung.

5. Conclusions and Future Works

The forecast for container throughput is a crucial issue in the operation of the port, and there is still room for improvement in existing forecasting methods. A mixed-precision neural architecture for container throughput forecasting was proposed in this study. Our mixed-precision neural architecture combined the strengths of CNNs and LSTM. The matrix operations in the CNN-LSTM model combined the abstracted feature and modeling time dependences from both the CNN and LSTM models and exhibited more comprehensive feature extraction for overall data to yield enhanced forecasting performance compared with either the CNN model or the LSTM model. Our mixed-precision neural architecture offers the advantage of simultaneously combining feature extraction in the CNN-LSTM model and providing precise time dependences results. The use of multidimensional and high-frequency data in port development has received considerable interest recently. Future studies can further explore the advances in neural network architectures to process multivariate time-series data and provide more comprehensive multistep throughput forecasting.

Author Contributions: Conceptualization, C.-H.Y. and P.-Y.C.; formal analysis, P.-Y.C.; methodology, P.-Y.C.; writing—original draft, P.-Y.C.; writing—review and editing, C.-H.Y. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported in part by the Ministry of Science and Technology, Taiwan, under grant 108-2221-E-992-031-MY3 and grant 108-2221-E-214-019-MY3.

Conflicts of Interest: The authors declare no conflict of interest.

References

- Guerrero, D.; Rodrigue, J.-P. The waves of containerization: Shifts in global maritime transportation. *J. Transp. Geogr.* **2014**, *34*, 151–164. [[CrossRef](#)]
- Bassan, S. Evaluating seaport operation and capacity analysis—Preliminary methodology. *Marit. Policy Manag.* **2007**, *34*, 3–19. [[CrossRef](#)]
- Jeevan, J.; Ghaderi, H.; Bandara, Y.M.; Saharuddin, A.; Othman, M. The Implications of the Growth of Port Throughput on the Port Capacity: The Case of Malaysian Major Container Seaports. *Int. J. E-Navig. Marit. Econ.* **2015**, *3*, 84–98. [[CrossRef](#)]

4. Schulze, P.M.; Prinz, A. Forecasting container transshipment in Germany. *Appl. Econ.* **2009**, *41*, 2809–2815. [\[CrossRef\]](#)
5. Wang, J.; Jia, R.; Zhao, W.; Wu, J.; Dong, Y. Application of the largest Lyapunov exponent and non-linear fractal extrapolation algorithm to short-term load forecasting. *Chaos Solitons Fractals* **2012**, *45*, 1277–1287. [\[CrossRef\]](#)
6. Akay, D.; Atak, M. Grey prediction with rolling mechanism for electricity demand forecasting of Turkey. *Energy* **2007**, *32*, 1670–1675. [\[CrossRef\]](#)
7. Guo, Y.; Nazarian, E.; Ko, J.; Rajurkar, K. Hourly cooling load forecasting using time-indexed ARX models with two-stage weighted least squares regression. *Energy Convers. Manag.* **2014**, *80*, 46–53. [\[CrossRef\]](#)
8. Xiao, J.; Xiao, Y.; Fu, J.; Lai, K.K. A transfer forecasting model for container throughput guided by discrete PSO. *J. Syst. Sci. Complex.* **2014**, *27*, 181–192. [\[CrossRef\]](#)
9. Mo, L.; Xie, L.; Jiang, X.; Teng, G.; Xu, L.; Xiao, J. GMDH-based hybrid model for container throughput forecasting: Selective combination forecasting in nonlinear subseries. *Appl. Soft Comput.* **2018**, *62*, 478–490. [\[CrossRef\]](#)
10. Niu, M.; Hu, Y.; Sun, S.; Liu, Y. A novel hybrid decomposition-ensemble model based on VMD and HGWO for container throughput forecasting. *Appl. Math. Model.* **2018**, *57*, 163–178. [\[CrossRef\]](#)
11. Chen, S.-H.; Chen, J.-N. Forecasting container throughputs at ports using genetic programming. *Expert Syst. Appl.* **2010**, *37*, 2054–2058. [\[CrossRef\]](#)
12. Syafi'i, K.K.; Takebayashi, M. Forecasting the Demand of Container Throughput in Indonesia. *Mem. Constr. Eng. Res. Inst.* **2005**, *47*, 69–78.
13. Guo, Z.; Song, X.; Ye, J. A verhulst model on time series error corrected for port throughput forecasting. *J. East. Asia Soc. Transp. Stud.* **2005**, *6*, 881–891.
14. Gosasang, V.; Chandraprakaikul, W.; Kiattisin, S. A Comparison of Traditional and Neural Networks Forecasting Techniques for Container Throughput at Bangkok Port. *Asian J. Shipp. Logist.* **2011**, *27*, 463–482. [\[CrossRef\]](#)
15. Geng, J.; Li, M.-W.; Dong, Z.-H.; Liao, Y.-S. Port throughput forecasting by MARS-RSVR with chaotic simulated annealing particle swarm optimization algorithm. *Neurocomputing* **2015**, *147*, 239–250. [\[CrossRef\]](#)
16. Bengio, Y. Learning Deep Architectures for AI. *Found. Trends Mach. Learn.* **2009**, *2*, 1–127. [\[CrossRef\]](#)
17. Bengio, Y.; Courville, A.; Vincent, P. Representation Learning: A Review and New Perspectives. *IEEE Trans. Pattern Anal. Mach. Intell.* **2013**, *35*, 1798–1828. [\[CrossRef\]](#)
18. Schmidhuber, J. Deep learning in neural networks: An overview. *Neural Netw.* **2015**, *61*, 85–117. [\[CrossRef\]](#)
19. Yang, C.-H.; Moi, S.-H.; Ou-Yang, F.; Chuang, L.-Y.; Hou, M.-F.; Lin, Y.-D. Identifying Risk Stratification Associated With a Cancer for Overall Survival by Deep Learning-Based CoxPH. *IEEE Access* **2019**, *7*, 67708–67717. [\[CrossRef\]](#)
20. Stoean, R.; Stoean, C.; Atencia, M.; Rodríguez-Labrada, R.; Joya, G. Ranking Information Extracted from Uncertainty Quantification of the Prediction of a Deep Learning Model on Medical Time Series Data. *Mathematics* **2020**, *8*, 1078. [\[CrossRef\]](#)
21. Tsai, Y.-S.; Hsu, L.-H.; Hsieh, Y.-Z.; Lin, S.-S. The Real-Time Depth Estimation for an Occluded Person Based on a Single Image and OpenPose Method. *Mathematics* **2020**, *8*, 1333. [\[CrossRef\]](#)
22. Jinsakul, N.; Tsai, C.-F.; Tsai, C.-E.; Wu, P. Enhancement of Deep Learning in Image Classification Performance Using Xception with the Swish Activation Function for Colorectal Polyp Preliminary Screening. *Mathematics* **2019**, *7*, 1170. [\[CrossRef\]](#)
23. Fawaz, H.I.; Forestier, G.; Weber, J.; Idoumghar, L.; Muller, P.-A. Deep learning for time series classification: A review. *Data Min. Knowl. Discov.* **2019**, *33*, 917–963. [\[CrossRef\]](#)
24. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#)
25. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to forget: Continual prediction with LSTM. In Proceedings of the 9th International Conference on Artificial Neural Networks (ICANN 1999), Edinburgh, UK, 7–10 September 1999.
26. Breiman, L.; Last, M.; Rice, J. Random forests. In *Machine Learning*; Springer: Berlin/Heidelberg, Germany, 2001; Volume 45, pp. 5–32.
27. Nathans, L.L.; Oswald, F.L.; Nimom, K. Interpreting multiple linear regression: A guidebook of variable importance. *Pract. Assess. Res. Eval.* **2012**, *17*, 1–19.
28. Freund, Y.; Schapire, R.; Abe, N. A short introduction to boosting. *J. Jpn. Soc. Artif. Intell.* **1999**, *14*, 771–780.

29. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: New York, NY, USA, 2013.
30. Zendejboudi, A.; Baseer, M.; Saidur, R. Application of support vector machine models for forecasting solar and wind energy resources: A review. *J. Clean. Prod.* **2018**, *199*, 272–285. [[CrossRef](#)]
31. Drucker, H.; Burges, C.J.; Kaufman, L.L.; Smola, A.J.; Vapnik, V. Support vector regression machines. In Proceedings of the Advances in Neural Information Processing Systems 9 (Nips 1996), Denver, CO, USA, 2–5 December 1997; pp. 155–161.
32. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O. Scikit-learn: Machine learning in python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
33. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *arXiv* **2014**, arXiv:1412.6980.
34. Nair, V.; Hinton, G.E. Rectified linear units improve restricted boltzmann machines. In Proceedings of the 27th International Conference on Machine Learning (ICML-10), Haifa, Israel, 21–24 June 2010; pp. 807–814.
35. Hyndman, R.J.; Athanasopoulos, G. *Forecasting: Principles and Practice*, 3rd ed.; OTexts: Melbourne, Australia, 2013.
36. Chen, C.; Twycross, J.; Garibaldi, J.M. A new accuracy measure based on bounded relative error for time series forecasting. *PLoS ONE* **2017**, *12*, e0174202. [[CrossRef](#)]
37. Hyndman, R.J.; Koehler, A.B. Another look at measures of forecast accuracy. *Int. J. Forecast.* **2006**, *22*, 679–688. [[CrossRef](#)]
38. Diebold, F.X.; Mariano, R.S. Comparing Predictive Accuracy. *J. Bus. Econ. Stat.* **2002**, *20*, 134–144. [[CrossRef](#)]
39. Derrac, J.; García, S.; Molina, D.; Herrera, F. A practical tutorial on the use of nonparametric statistical tests as a methodology for comparing evolutionary and swarm intelligence algorithms. *Swarm Evol. Comput.* **2011**, *1*, 3–18. [[CrossRef](#)]
40. Wilcoxon, F. Individual Comparisons by Ranking Methods. In *Breakthroughs in Statistics*; Springer: Berlin, Germany, 1992; pp. 196–202.
41. Siegel, S.; Castellan, N.J. *Nonparametric Statistics for the Behavioral Sciences*; McGraw-Hill: New York, NY, USA, 1956; Volume 7.
42. Hong, W.-C.; Li, M.-W.; Geng, J.; Zhang, Y. Novel chaotic bat algorithm for forecasting complex motion of floating platforms. *Appl. Math. Model.* **2019**, *72*, 425–443. [[CrossRef](#)]
43. Fan, G.-F.; Peng, L.-L.; Hong, W.-C. Short term load forecasting based on phase space reconstruction algorithm and bi-square kernel regression model. *Appl. Energy* **2018**, *224*, 13–33. [[CrossRef](#)]
44. Dong, Y.; Zhang, Z.; Hong, W.-C. A Hybrid Seasonal Mechanism with a Chaotic Cuckoo Search Algorithm with a Support Vector Regression Model for Electric Load Forecasting. *Energies* **2018**, *11*, 1009. [[CrossRef](#)]

Publisher’s Note: MDPI stays neutral with regard to jurisdictional claims in published maps and institutional affiliations.



© 2020 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<http://creativecommons.org/licenses/by/4.0/>).