

Deep Learning Framework for Physical Internet Hubs Inbound Containers Forecasting

El-Sayed Orabi Helmi¹, Osama Emam², Mohamed Abdel-Salam³

Dept. of Business Information Systems, Faculty of Commerce and BA, Helwan University, Cairo, Egypt^{1,3}

Dept. of Computer Science, Faculty of Computing and AI, Helwan University, Cairo, Egypt²

Abstract—This article presents a framework for physical internet hubs inbound containers forecasting based on deep learning and time series analysis. The inbound containers forecasting is essential for planning, scheduling, and resources allocation. The proposed framework consists of three main phases. First, the inbound historical transaction has been processed to find out the training window size (lags) using auto correlation function (ACF) and partial autocorrelation function (PACF). Second, the framework uses convolutional neural network (CNN) and recurrent neural network (RNN) as training networks for the historical time series data in two techniques. The proposed framework uses univariate and multivariate time series analysis to explore the maximum forecasting outcomes. Last, the framework measures the accuracy and compares the forecasting output using mean absolute error matrix (MAE) for both approaches. The experiments illustrated that RNN forecasts univariate inbound transaction with total 5.0954 MAE rather than 5.0236 for CNN. The CNN outperforms multivariate inbound containers forecasting with 0.7978 MAE. All the results has been compared with autoregressive integrated moving average (ARIMA) and support vector machine (SVR).

Keywords—Physical internet hubs (π hubs); deep learning; convolutional neural network (CNN); recurrent neural network (RNN); time series forecasting

I. INTRODUCTION

All Physical internet (π) is a global logistics system first introduced by [1] in the early of this decade. The main objective of the physical internet is to connect all the logistics partners (customers, suppliers, shippers) in an intelligent way. π Hub is one of the main milestones in the future logistics network. π hub is responsible for distributing goods and items through the logistics network. These hubs should be managed and controlled in an intelligent manner to perform the complex logistics challenges. Scheduling and resources localization are two of these challenges. Also, moving items in and out the hub requires clear vision and scalable solutions [2]. Machine learning gives the researchers the ability to contribute solutions in different research fields. Machine learning, especially deep learning proposed research outperformed in classification, clustering, and regression analysis [2].

In 2012 [3] proposed the main functions that should be in any railway π hub. These functions have been measured using key performance indicators. These indicators measure the performance of the proposed design of the π hub through three perspectives. The first is from the customer's perspective. The second is suppliers' point of view. The last indicators measure the railway worker's satisfaction. The researchers in this case

study faced some challenges such as determining the number of containers that will be inbound in a certain π hub. They assumed that 30% of containers on each train will be unloaded and reloaded by others in stock containers. Then they calculated the estimated unload and reload time to this assumption. They also calculated the required time to unload and reload all the containers for the entire train. Starting from this challenge we tried to forecast the actual or near actual containers flow through the π hub. The objective of this research is to integrate two deep learning training networks with a physical internet providing framework for inbound containers forecasting. This framework will forecast the flow of goods and items through the π hub based on historical time series data. Based on time series analysis and deep learning techniques we propose this framework to be a guide for π hub resource management system aiming to achieve high accuracy with minimum inbound forecasting error.

The prediction process is one of the most difficult operations because it is subject to several variables, which makes choosing the appropriate algorithms to solve this problem very important. Some of the current prediction algorithms lack self-learning, such as linear and non-linear systems and moving average. And because of the strength in the field of deep learning in a number of areas, especially machine learning, drones, autonomous cars, computer vision and other fields, the research team decided to use deep learning algorithms in this research.

This article is divided into several parts. The first part provides an overview of some concepts, previous studies and some current analysis methods for time series. The researcher also presents in the first part of the article some methods of deep learning. As for the second part of the article, it presents the proposed framework for forecasting future inbound containers quantities in the short term, based on some previous data. As for the third part, it presents the results of the established experiments, their analysis and comparison with some of the current methods of prediction.

II. THEORETICAL BACKGROUND

The following section discusses briefly the main concept of physical internet, time series analysis, recurrent neural network, and convolutional neural network respectively. Some related studies will be discussed in section.

A. Physical Internet

Physical internet is a global open logistic system. Physical internet's main objective is to encapsulate interfaces and

protocols to manage physical, digital, and operational interconnectivity of the recent logistic functions through one global system. New containers, movers, nodes, and hubs have been proposed. Through a series of proposed standards and functions designs which replaced or integrated with the current logistics infrastructure that will replace the entire logistics system by 2050 [1]. The Physical Internet is structured in a similar way to data packets sent via the standard digital Internet. This notion fundamentally alters how commodities are designed, relocated, and distributed today. This approach, in which the items relocation process is known and implemented in an optimal and efficient manner at each relocation stage, is critical for all supply chain players. Prior to the start of the procedure, it was ensured that it would be transparent, efficient, and ecologically friendly [4]. The Physical Internet's goal entails enclosing commodities in smart, eco-friendly, and adaptable containers ranging in size from a shipping container to a little box. It therefore generalizes the marine container, which has shaped ships and ports to accommodate globalization, and extends containerization to logistical services in general. Instead of a warehouse or a truck, the Physical Internet relocates the private space's boundary to the inside of a container. These modular containers will be continuously monitored and directed using the Internet of Things to take advantage of their digital interconnection [1,4]. Each π container has a unique global identifier, such as the MAC address in the Ethernet network and the digital Internet, from an informational standpoint. This identifier is physically and digitally attached to each π container to ensure identification reliability and efficiency. Each π container has a smart tag attached to it that acts as its representative agent. Through the Physical Internet, it helps to ensure container identification, integrity, routing, conditioning, monitoring, traceability, and security. Smart tagging allows for the distributed automation of a wide range of handling, storage, and routing tasks.

B. Time Series Analysis

Before going further in our forecasting case study, it is essential to briefly illustrate the term time series data analysis which is the core foundation of our study. Time series data is recording of processes and observations varies over time. These observations can either be recorded in continuous points or as a set of discrete observations sequentially. These observations are exposed to trending, cyclical, seasonality and irregular variations. The trend of the observed data may be positive or negative in other word increasing or decreasing of data values over time. The cycle is a repetition of data behavior over a long time. The seasonality is a regular fluctuation of the observations at the same week, month or quarter every year. The irregularity in the time series data may happen for more than one reason. It could be because of noise, outliers, wrong data entry or sudden increase or decrease of observations value. Different analysis methods and techniques had been proposed over years. Time series consists of modeling mathematical descriptions estimating separately the four components independently. Time series analysis could be presented statistically in two approaches. The first approach is univariate analysis. The univariate approach is the analysis of single variable. The second approach is multivariate analysis approach. The multivariate approach is the analysis of two or

more variables. These variables may be dependent or independent variables. Univariate time series are subject to descriptive statistical analysis such as central tendency (mode, median, and median). It also, subject to dispersion analysis such as (variance, range and standard deviation). Multivariate analysis is more suitable for real life applications because of its high conclusion accuracy. Multivariate includes more than one factor of independent variables that influence the variability of dependent variables. Multivariate analysis is computational intensive. The researchers over years proposed significant methods and approaches. Those methodologies can be distinguished as ARIMA and nonARIMA methods. Several ARIMA stochastic models has been introduced, such as autoregressive (AR), moving average (MA), autoregressive moving average (ARMA), ARIMA, seasonal ARIMA (SARIMA), autoregressive fractionally integrated moving average (ARFIMA), and autoregressive conditional heteroscedasticity (ARCH) [5]. The ARIMA method has often been utilized for various types of univariate time series for many years. The ARIMA method has been well developed which made this method used in many research fields. Recently, many researchers developed nonARIMA methods with artificial intelligence [5, 6]. ARIMA model has some back draws such as it is computationally costly. It has poor performance in Long-term forecasting. Also, seasonal time series are not supported by ARIMA model. Today, the use of deep learning (DL) techniques has become the most popular approach for many machine learning problems, including time series forecasting. Deep neural networks have shown a great potential to map complex non-ARIMA feature interactions. Deep learning models are an alternative solution for forecasting because of their accuracy [7, 8].

Other researcher used support vector machine in regression analysis despite of it has some major disadvantages such as it is ineffective for large datasets. The SVM will underperform if the number of features for each data point exceeds the number of training data samples [9].

C. Recurrent Neural Network

A recurrent neural network (RNN) is a class of artificial neural networks (ANN) connections between nodes. RNN is made up of a set of nodes connected by edges, where the edges have a direction associated with them along with a temporal sequence. This allows it to exhibit temporal dynamic behavior. Derived from feed forward neural networks, RNNs can use their internal state (memory) to process variable length sequences of inputs [3]. RNNs are one of the most frequently utilized ANN architectures for time series prediction problems. They also become popular in natural language processing research. RNNs feedback architecture allows cells inherent the temporal sequence order and variables dependencies [9]. Long Short-Term Memory (LSTM) cell, Elman RNN cell, and the Gated Recurrent Unit (GRU) are the most popular RNN network architectures in time series modeling and forecasting [10].

D. Convolutional Neural Networks

A convolutional neural network (CNN, or ConvNet) is a type of deep neural network that is most commonly used for image analysis [11]. Based on the shared-weight architecture of

the convolution kernels that scan the hidden layers and translation invariance properties, they are also known as shift invariant or space invariant artificial neural networks (SIANN). Multilayer perceptron are regularized variants of CNNs. Multilayer perceptron are completely linked networks in which each neuron in one layer communicates with all neurons in the subsequent layer. These networks' "complete connectivity" makes them vulnerable to data over fitting. Regularization methods commonly used include adjusting weights as the loss function is minimized and randomly trimming connections. CNNs take a different method to regularization: they take advantage of the hierarchical pattern in data and use smaller and simpler patterns embossed in the filters to assemble patterns of increasing complexity. As a result, CNNs are at the bottom end of the connectedness and complexity spectrum [12]. This is accomplished by running a filter (or weight matrix) over the input and computing the dot product between the two at each location (i.e. a convolution between the input and filter). Because of this structure, the model can learn filters that recognise specific patterns in the incoming data. The idea behind using CNNs to anticipate time series values is to learn filters that reflect certain recurrent patterns in the series and use them to forecast future values. CNNs may function well on noisy series because of their layered structure, which allows them to eliminate noise in each subsequent layer and extract just the important patterns, comparable to neural networks that use wavelet transformed time series [13].

III. PROPOSED FRAMEWORK

This framework consists of three phases. The first phase is data collection and preprocessing. In this task the framework collects, integrates, and preprocesses all the previous inbound transactions that have been made in the π hub. This task checks the data stationary. If the data is non-stationary data, the framework will use the difference technique to convert the data to be in stationary status. Section 4 discusses this phase. Deep learning is the second phase with 70% of the inbound transactions. In this phase the framework feeds the stationary data to the learning network (NN, RNN, and CNN) and computes the learning rate. The training happens with two approaches (univariate, multi-variate). The univariate approach is suitable for independent variables. The multi-variate is suitable for highly dependent dimensions. Testing and validation are the third phase. This phase tests and validates the inbound flow prediction against 30% off the collected data. Therefore, the framework calculates the accuracy of each learning network using mean absolute error technique. Fig. 1 illustrates the phases of the proposed framework.

As in Fig. 1, the entire data values must be stationary data to avoid the impact of the abnormal and outliers. Also, the framework calculates the statistical auto-correlation function (ACF) and partial auto-correlation function (PACF) to find out the target lag length. Those lags indicate the most appropriate forecasting window size, for example predicts 10 days flow ahead. Also, the framework ignores the calculation of variables independency by making the training in two techniques using univariate or multivariate analysis. Despite of the time consuming, the use of both techniques make the framework suitable for any time series analysis.

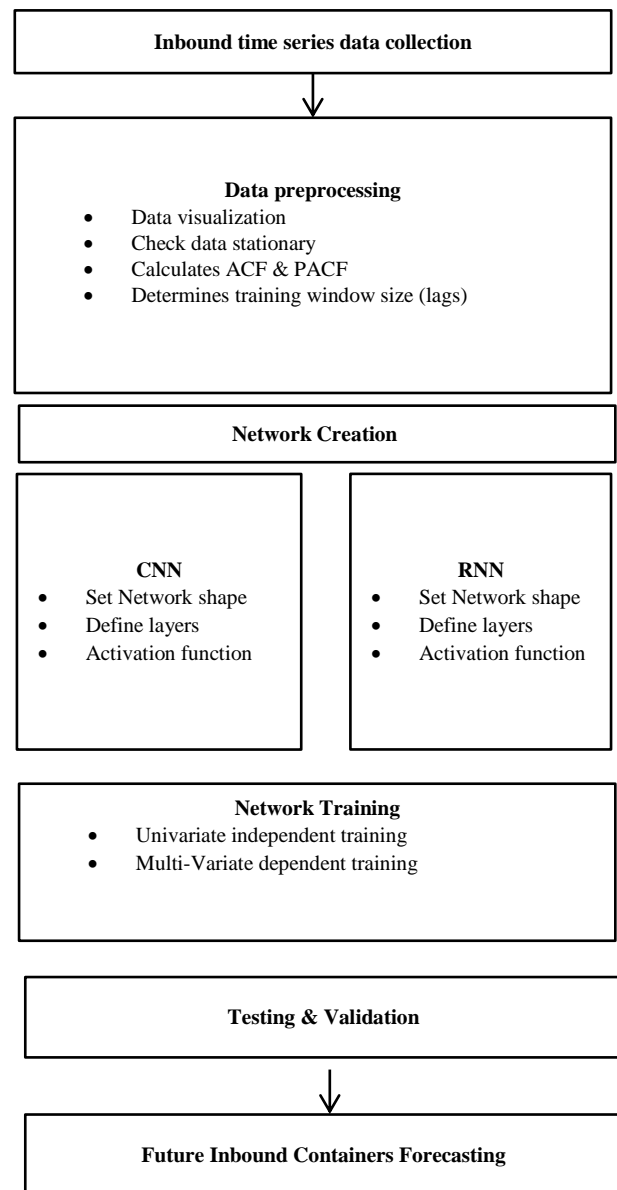


Fig. 1. Physical Internet Hubs Inbound Containers Forecasting Proposed Framework.

IV. DATA PRE-PROCESSING

Regarding the lack of real-life π hubs, we use a store item demand forecasting challenge dataset offered by Kaggle [14]. Then we select 6 random variables from the dataset to be present container volume. Some data preprocessing has been made to meet the proposed design of [3] which proposed a design for railway π hub. The inbound containers in their proposed π hub had 6 main volumes. The container volumes are (1.2, 2.4, 3.6, 4.8, 6 and 12 meters). These containers are the current intermodal containers. Table I shows the number of data points (count), the arithmetic mean, the standard deviation, the 1st quartile, the 2nd quartile, the 3rd quartile, the minimum, the maximum, interquartile range (IQR) and outlier values for each container volume used in this study.

TABLE I. THE INBOUND HUB CONTAINERS TIME SERIES DATASET SUMMARY (NUMBERS IN 1000S)

	Container size in meter					
	1.2 m	2.4 m	3.6 m	4.8 m	6 m	12 m
Count	2922	2922	2922	2922	2922	2922
Mean	30.95	57.43	36.99	35.6	32.4	55.23
S.D	17.38	16.67	11.96	23.2	22.9	15.46
Min	4	13	8	4	3	17
Q1	18	45	28	18	15	44
Q2	25	56	36	25	21	55
Q3	42	68	44	54	52	65
Max	100	115	81	120	113	108
IQR	24	23	16	36	37	21
outlier	>78	>102.5	>68	>108	>107.5	>96.5

As shown in Table I, the training data set contains time series data for 6 different containers. Each container has 2922 observations. It also, shows that the training data is normally distributed for all variables with different IQRs and outlier values. Although the training dataset is normally distributed, it is non-stationary data. Fig. 2 illustrates the non-stationary status of the training dataset.

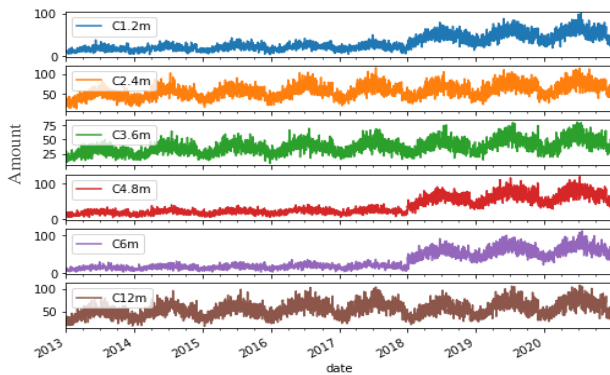


Fig. 2. Non-stationary Time Series Historical Inbound Containers Data.

As shown in Fig. 2, some fluctuations were observed in the training data. It also shows that some repeat behavior (cycle) in the data especially for the 2.4 meters, 3.6 meters and 12 meters volume containers.

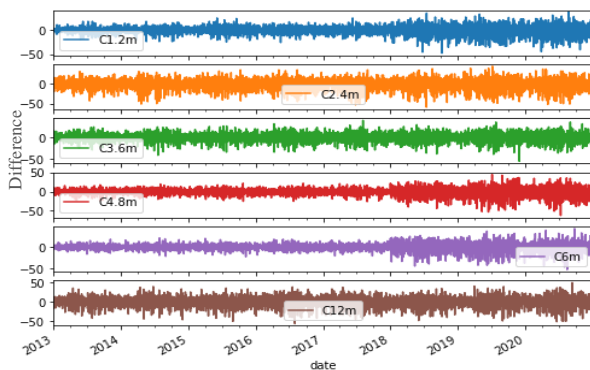


Fig. 3. Time Series Stationary Training Dataset.

Fig. 2 illustrates regular and predictable changes that recur every calendar year (seasonality) in the time series dataset. It also shows trend fluctuation at some data point. The difference technique has been used to convert the time series to stationary status. Fig. 3 shows the stationary data which has been used for training the proposed framework. The data was converted to the stationary status using the difference method. This phase was essential to avoid any bias in the training data, which gives better judgment of the forecasting output.

The proposed framework uses an 8 years stationary dataset to perform the learning phase. The network uses 70% of the dataset for training, 20% for testing and 10% for validation. Furthermore, the autocorrelation function and partial autocorrelation function had been used to determine the lags length. Although this step can be dispensed with, the researcher believes that it may be a good start and is governed by a statistical basis that enables the proposed framework to start the training process effectively. According to the results of ACF and PACF, the lags length of our training was 7, 24 days for narrow and wide window forecasting, respectively. The next section discusses the network learning, testing and validation experiments for CNN and RNN in two approaches. These approaches are univariate and multivariate time series forecasting.

V. EXPERIMENTS AND RESULTS

This section discusses in detail the performed experiments. The framework has been implemented using python and TensorFlow. The learning networks have been developed for both univariate and multivariate with two different input sizes to maximize the forecasting outcomes of the proposed framework. Also, this section discusses the different shapes of the implemented neural networks for narrow and wide input window as deep learning univariate and multivariate time series forecasting.

A. Narrow Window Univariate Inbound Containers Forecasting

Univariate time series refers to a time series that consists of single (scalar) observations recorded sequentially over equal time increments. The proposed implementation of CNN uses the previous 6 days to predict the 7th day in the time series inbound transaction. Fig. 4 and 5 show the building structure of CNN and RNN in the proposed framework experiments.

As shown in Fig. 4, the implemented CNN consists of 4 fully connected dense layers and 1 convolutional layer. Each layer uses the relu activation function. The relu function has been used to maximize the non-linearity behavior of the proposed network. This implementation forecasts the container's flow for each container volume one by one independently. The proposed implementation of RNN uses the long-short term memory (LSTM) for prediction. Fig. 5 illustrates the 6 layers RNN structure.

As shown in Table II, the total absolute error for the 4 algorithms is almost the same. But RNN (LSTM) outperforms with total MAE 5.0236. It also performs the training of 6 meters volume container better than SVR by 6 % and CNN by 10%.

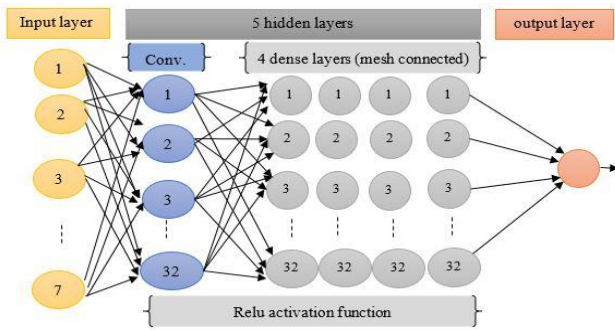


Fig. 4. CNN Structure for Multi Layers Forecasting Model.

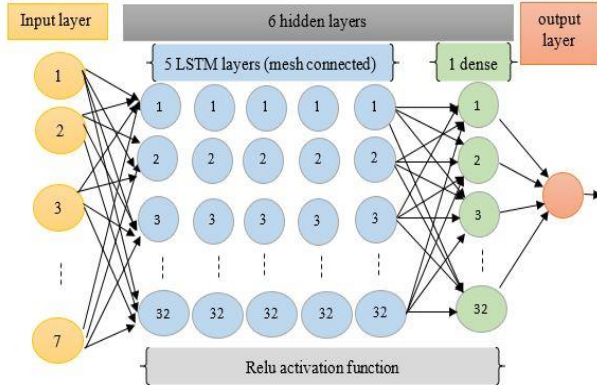


Fig. 5. RNN Proposed Multilayer Model Implementation.

TABLE II. UNIVARIATE INDEPENDENT INBOUND NARROW WINDOW FORECASTING NETWORKS MEAN ABSOLUTE ERROR

Container volume	Algorithm			
	ARIMA	SVR	CNN	RNN (LSTM)
1.2m	1.2321	0.9522	0.9073	1.0219
2.4m	1.1201	0.7816	0.8157	0.8055
3.6m	1.3210	0.8194	0.8504	0.8207
4.8m	1.2421	0.9451	0.9316	0.8458
6m	1.4124	0.8414	0.8794	0.7902
12m	1.3410	0.7470	0.7110	0.7395
Sum of (MAE) error	7.6687	5.0867	5.0954	5.0236

B. Univariate Wide Window Inbound Containers Forecasting

In this series of experiments, the framework uses multi-steps output forecasting. Those experiments had been carried out with the same CNN and RNN previous architecture as in Fig. 4 and 5. The only difference in these experiments is that we predict 24 days in future rather than one day. The network shape is (32, 24, 1). Where 32 are number of neurons is the input layers, 24 is output size and 1 is number of features to be predicted. These experiments use the historical inbound transaction of the 6 container sizes independently to predict the flow of each container size individually. Table III illustrates the forecasting accuracy measurement.

C. Multivariate Narrow Window Inbound Containers Forecasting

The possibility of a dependency relationship that could exist between different container volumes, especially as stated in the proposed design of railway warehouses and the method of transporting containers using trains. In order to, give the proposed framework realistic and relevant real-life applications and our desire to improve the forecast. In these experiments series, we used a multivariate time series analysis technique. We used the same lag length of narrow univariate forecasting window and the structure of CNN and as in Fig. 6. The only difference here was using the entire day observations of the 6 containers volume as one input vector. Also, the output was a vector of 6 features each feature represent one of the containers volume. The experiments showed that, MAEs were 0.8921, 0.7934 & 0.9231 for CNN, RNN and SVR model respectively.

D. Multivariate Wide Window Inbound Containers Forecasting

The framework has been trained to forecast the future flow dependently at the same time. The output shape for both CNN and RNN is (32, 24, 6). Fig. 7 shows the structure of RNN-LSTM network.

As shown in Fig. 7, the RNN (LSTM) network consists of 6 hidden fully connected feed forward layers. The performed experiments proved that CNN outperformed. The mean absolute error for SVR, CNN and RNN (LSTM) multivariate forecasting is 0.9176, 0.7978 and 0.9151, respectively. These experiments showed that CNN proposed architecture performs multivariate forecasting better than RNN.

TABLE III. UNIVARIATE INDEPENDENT WIDE WINDOW INBOUND FORECASTING NETWORKS MEAN ABSOLUTE ERROR

Container volume	Algorithm			
	ARIMA	SVR	CNN	RNN (LSTM)
1.2m	1.3251	0.9904	0.89105	0.8802
2.4m	1.2134	1.014	0.9003	0.8605
3.6m	1.3421	0.9948	0.9102	0.7912
4.8m	1.1012	0.9853	0.8807	0.8014
6m	1.2115	0.9947	0.8904	0.8114
12m	1.2341	0.9862	0.8926	0.7812
Sum of (MAE) error	7.4274	5.9654	5.36525	4.9259

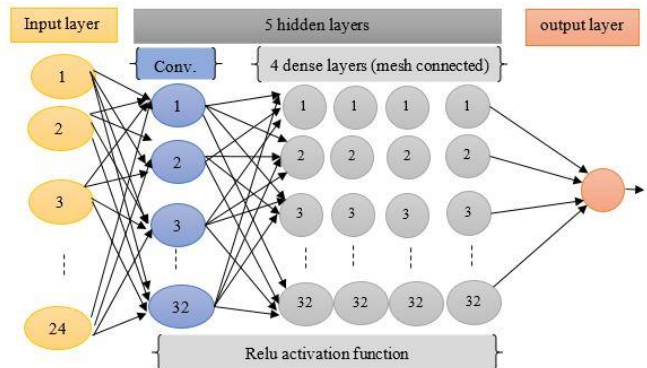


Fig. 6. CNN Multivariate Inbound Forecasting Structure.

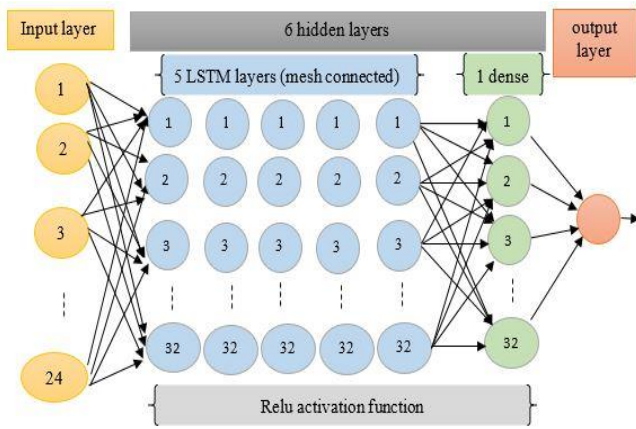


Fig. 7. Multivariate RNN (LSTM) Forecasting Network Structure.

VI. CONCLUSION

The proposed framework forecasts π_{hub} inbound containers using CNN and RNN deep learning networks for both univariate and multivariate forecasting approaches. ACF and PACF have been used to determine better forecasting window size based on the status of the training data. The difference technique has been used to overcome the non-stationary training data. All the forecasting results have been compared to time series forecasting ARIMA and SVR algorithms. It has been found that RNN forecasts the univariate independent container flow for short term rather than CNN. While CNN performs univariate independent containers flow better than RNN and SVR for long term forecasting. On other hand in has been found that CNN outperforms forecasting for multivariate analysis for both short and long time forecasting.

REFERENCES

[1] Montreuil, Benoit, "toward a Physical Internet: meeting the global logistics sustainability grand challenge", Springer, Vol. 3, pp. 71-87, 2, 2011.
[2] Martin Christopher, Logistics and supply chain management (5th Ed.). London: Pearson Education 2016.

[3] Dupond, Samuel, "A thorough review on the current advance of neural network structures", Annual Reviews in Control, Annual Reviews in Control, Vol. 14, pp. 200-230, 2019.
[4] Eric Ballot, Benoit Montreuil, Collin Thivierge, "Functional Design of Physical Internet Facilities: A Road-rail Hub". Gardanne, France : s.n.12th IMHRC Proceedings. pp. 13, 2012.
[5] Kolinski Adam, "The Impact of Eco-efficiency in Production on Availability of Machines and Equipment", Efficiency in Sustainable Supply Chain, springer, pp. 161-177, 2017.
[6] Hendikawati1, "A survey of time series forecasting from stochastic method to soft computing", Journal of Physics: Conference Series, pp. 20-30, 2020.
[7] Anggraeni W, Boga K and Mahananto F., "The Performance of ARIMAX Model and Vector Autoregressive (VAR) Model in Forecasting Strategic Commodity Price in Indonesia", Procedia Computer Science , pp. 189-196, 2018.
[8] Pedro Lara-Ben'itez, Manuel Carranza-Garc'ia, " An Experimental Review on Deep Learning Architectures". International Journal of Neural Systems, Vol. 31, pp. 201, 2021.
[9] Najafi G, Ghobadian B, Moosavian A, Yusaf T, Mamat R, Kettner M , Azmi W H. (2016). SVM and ANFIS for prediction of performance and exhaust emissions of a SI engine with gasoline ethanol blended fuels. Applied Thermal Engineering, pp.186-203, 2016.
[10] Hansika Hewamalage, Christoph Bergmeir, Kasun Bandara, "Recurrent Neural Networks for Time Series Forecasting: Current Status and Future Directions", Elsevier, pp. 1-56, 2020.
[11] Cho, K., van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., Bengio, Y. "Learning phrase representations using RNN Encoder-Decoder for statistical machine translation". Stroudsburg, PA, USA: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Association for Computational Linguistics. pp. 1724-1734, 2014.
[12] Zhang, Wei. "Shift-invariant pattern recognition neural network and its optical architecture". Proceedings of Annual Conference of the Japan Society of Applied Physics, pp. 90, 1988.
[13] T. Fisher, C. Krauss, "Deep learning with Long Short-Term Memory networks for financial market predictions". FAU Discussion papers in Economics, pp. 102, 2017.
[14] Store Item Demand Forecasting Challenge. Kaggle. Online|12 June 2021 [Cited: 18 March 2022 .]https://www.kaggle.com/c/demand-forecasting-kernels-only.