



Article

Deep Learning Forecasting for Supporting Terminal Operators in Port Business Development

Marco Ferretti ^{1,†}, Ugo Fiore ^{2,*,†}, Francesca Perla ^{1,†}, Marcello Risitano ^{1,†} and Salvatore Scognamiglio ^{1,†}

¹ Department of Management and Quantitative Studies, Parthenope University, 80132 Napoli, Italy; marco.ferretti@uniparthenope.it (M.F.); francesca.perla@uniparthenope.it (F.P.); marcello.risitano@uniparthenope.it (M.R.); salvatore.scognamiglio@uniparthenope.it (S.S.)

² Department of Computer Science, University of Salerno, 84040 Fisciano, Italy

* Correspondence: ufiore@unisa.it

† These authors contributed equally to this work.

Abstract: Accurate forecasts of containerised freight volumes are unquestionably important for port terminal operators to organise port operations and develop business plans. They are also relevant for port authorities, regulators, and governmental agencies dealing with transportation. In a time when deep learning is in the limelight, owing to a consistent strip of success stories, it is natural to apply it to the tasks of forecasting container throughput. Given the number of options, practitioners can benefit from the lessons learned in applying deep learning models to the problem. Coherently, in this work, we devise a number of multivariate predictive models based on deep learning, analysing and assessing their performance to identify the architecture and set of hyperparameters that prove to be better suited to the task, also comparing the quality of the forecasts with seasonal autoregressive integrated moving average models. Furthermore, an innovative representation of seasonality is given by means of an embedding layer that produces a mapping in a latent space, with the parameters of such mapping being tuned using the quality of the predictions. Finally, we present some managerial implications, also putting into evidence the research limitations and future opportunities.

Keywords: deep learning forecasting; machine learning; terminal operator; port management



Citation: Ferretti, M.; Fiore, U.; Perla, F.; Risitano, M.; Scognamiglio, S. Deep Learning Forecasting for Supporting Terminal Operators in Port Business Development. *Future Internet* **2022**, *14*, 221. <https://doi.org/10.3390/fi14080221>

Academic Editor: Davide Tosi

Received: 20 June 2022

Accepted: 22 July 2022

Published: 25 July 2022

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



Copyright: © 2022 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 (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, containerisation and ports have played important roles in international trade [1,2] for supporting freight traffics efficiently in worldwide competition [3,4]. Today, nearly all overseas shipments of products such as furniture, toys, footwear, clothing, auto parts, electronic components, and computers are usually made through containers on cargo ships. Furthermore, the amount of food goods shipped in refrigerated containers is also constantly growing [1], and this process has greatly stimulated the business development of terminal operators [5]. Therefore, shipping companies are using larger and larger ships that allow the transport of about 24,000 TEUs, minimizing unit transport costs and developing maritime economics of scale [6] to efficiently manage logistics activities both in port handling and in container storage on vessels through adoption of the so-called model of naval gigantism [7].

In this context, the importance of optimal management and planning of ports' development is unquestionable [8]. In particular, construction costs represent sunk costs for a port, and to ensure that these investments in the development of ports will be profitable, the number and layout of the terminals [9,10] in the port must be appropriately commensurate with the volume of container transport in the port itself [11]. It is therefore evident that, to avoid waste, port managers must be able to accurately predict the container transport volumes [12,13], as this is essential for planning and organizing port activity [14] and is also relevant for government departments dealing with transport at both the micro and macro levels [15].

The demand forecast for container traffic is a crucial success factor in the business development of port logistics which allows efficient management of resources by both container terminal operators (i.e., concessionaires) and port authorities. Moreover, the strategic management of these maritime logistic activities depends on port governance models [16] as well as on the level of vertical integration of container transport processes between terminal operators and shipping companies [17,18]. In this perspective, port authorities must forecast—through some concession clauses and the setting of target performance—the achievement of economic and operational outcomes by terminal operators [19], measurable in terms of traffic demand, logistics effectiveness, and efficiency managed by the “land-side”. Therefore, container terminal operators manage and plan critical logistic activities in accordance with port authority marketing strategies [20,21] to support port competitiveness in the worldwide scenario [4,22]. Technological advances—such as digitalisation and the Internet of Things—offer clear opportunities to streamline and synchronise operations, increase efficiency, and improve productivity toward the building of the so-called smart port [23,24]. However, automation and port call optimisation involve digital investments, both for initial deployment and for keeping abreast of recent advances, and these investments are linked to traffic volume monitoring to support the port’s competitive advantage [25].

Coherent with the aforesaid statements, this study comes at an opportune time when substantial variation can be observed in containerised shipment volumes, owing to factors such as trade tensions, protectionism, regionalisation of supply chains, the availability of mega container ships, and last but not least, the global pandemic [26]. In a scenario filled with uncertainty, port managers have to carefully rethink their development plans. Regulators and port authorities are also facing the challenges of a changing environment, especially insofar as they grant container terminal concessions. Finally, there is a fine-grained model capable of forecasting several categories of container traffic fits separately but simultaneously with the efforts to reduce carbon emissions due to shipping and the increased attention ports are giving to landside operations and logistics.

2. Related Work

Forecasting the container transport volume is a hot topic in the maritime transportation literature, and several approaches have been developed to improve the predictions. In this section, a brief review of the literature is conducted.

Given the time-dependent nature of the phenomenon, models typical of time series analysis have been applied. The autoregressive integrated moving average (ARIMA) [27] models represent the most common approach, and some interesting empirical studies that apply these models to container transport volume forecasting can be found in the work by Chan et al. [28].

Other authors have highlighted the presence of seasonal fluctuations in the container transport volume data [29,30] and suggested the use of models able to take these effects into account. In this direction, the Seasonal-ARIMA (SARIMA) models appeared to be the natural extension of the traditional ARIMA models. Schulze and Prinz [31] applied SARIMA models to predict the container transport volume in German ports.

The main limitation of the ARIMA and SARIMA models is the assumption of linearity, which could make them unable to capture the complexity present in some empirical evidence. The need for more accurate predictions directed scientific interest towards more complex and non-linear models. Some interesting examples are trigonometric regression models [30] and machine learning-based models, such as artificial neural networks (ANNs) [32] and support vector machines (SVMs) [33]. Focusing on an intercontinental liner service, Wang and Meng [34] proposed a method for forecasting the number of containers to be transported between two continents. They used piecewise linear regression, autoregression, and artificial neural networks. Using real-case booking data, they found that a combination of the three methods yielded forecasts of satisfactory precision. To cope with changing trends and limited data, Chen and Liu [35] developed a predictive method

based on a GM(1,1) grey model and generalised contour lines to forecast Chinese port cargo throughput. Fouries analysis was used in combination with a SARIMA model in an empirical study of cargo and container throughput for the ports of Hong Kong and Kaohsiung. Numerous other approaches, based on decomposition methods or hybrid models, have been proposed. The interested reader may refer to the classic book by Box et al. [27].

The mainstream literature, especially in the last period, has shown a growing interest in machine learning (ML). Barua et al. [36] provided an exhaustive review of the progress in ML for transportation management. The interesting point of forecasting with ANNs is that the optimal model is determined from the data itself. The first paper that employed ANNs in this field is dated 1999, when an ANN-based model was used to predict the volume of container trade in the port of Kaohsiung in Taiwan [37]. More recent studies are proposed by Gosasang et al. [38], who analysed data relating to the port of Bangkok, and Milenković et al. [39], who focussed on data from the port of Barcelona. Working on data relating to the port of Shanghai and Shenzhen, Xie et al. [40] proposed a hybrid approach to predicting container trading volume based on the least squares SVR (a regression version of the traditional least squares SVM method) and offered an interesting comparative analysis with some models, including ARIMA and SARIMA models.

Notable new approaches include the one by Wang et al. [41], who discussed a generative model based on an autoencoder coupled with an ANN for forecasting the quality of production, the one by Zhou et al. [42], who incorporated dynamic factors into a recurrent network, and the one by Du et al. [43], who combined variational mode decomposition to isolate multiple seasonal components and an extreme learning machine. Another relatively recent paper proposes a univariate long short-term memory (LSTM) network model to forecast the future volume of container transport [44]. LSTMs are a type of recurrent neural network [45] specifically designed to analyse data with sequential structures, and they have recently become extremely popular after the excellent results achieved in various fields such as handwriting and speech recognition and text translation [46,47].

This paper contributes to the literature in two directions. First, we explore two other ML-based algorithms specifically designed for processing sequential data: 1-D convolutional neural networks (CNNs) and gate recurrent unit networks (GRUs). We also introduce in the neural network architectures investigated an additional component that explicitly models the seasonal effect via embedding neural networks. Second, we develop a multi-variate forecasting approach able to predict the container traffic not only at the macro level (i.e., the entire port) but also on a micro level (e.g., the specific terminal operator).

3. Materials and Methods

3.1. Methods

This section briefly describes the neural network (NN) blocks used in this research. In particular, neural networks represent a class of algorithms replicating the human brain's learning process. They are constituted by a set of interconnected units arranged in layers that learn from experiences using training algorithms. The way these neurons are connected (i.e., through the synapses) configures different types of neural networks with different characteristics and properties [48]. NNs have been proven to be promising function approximators and are extensively used for complex forecasting tasks. Neural networks represent a promising tool for modelling the dynamics of container traffic volume, which is generally characterised by strong nonlinearity and seasonality effects. This is especially true when modern deep learning networks (DLN) designed explicitly for processing time-series data (such as recurrent and 1D-convolutional data) are considered. The mathematical descriptions of these kinds of neural networks, which were used in this research, are provided in the following subsections.

3.1.1. Fully Connected Neural Networks

In feedforward networks, the information propagates along the synapses from the input to the output layer in only one direction. Each layer transforms the data, and successive transformations are chained to produce a composition of functions.

We denote by $\mathbf{x} = (x_1, x_2, \dots, x_d)^\top \in \mathcal{X}$ the vector of features. Mathematically, a $q \in \mathbb{N}$ -dimensional feedforward (or fully connected) neural network layer can be seen as a vectorial function that maps the training samples in a new space. The coordinates of the data in the new space are the output of the neurons in the layer, and each unit computes the nonlinear function $z_i(\mathbf{x})$ as follows:

$$z_i(\mathbf{x}) = \phi \left(w_{i,0} + \sum_{l=1}^d w_{i,l} x_l \right) \quad i = 1, 2, \dots, q,$$

where $w_{i,l} \in \mathbb{R}$ are coefficients to learn and $\phi(\cdot)$ denotes the activation function. The most popular choices of the activation function in the machine learning literature are the sigmoid, hyperbolic tangent, or rectified linear units, where the latter is known as a ramp function outside the machine learning parlance. In matrix form, the output $\mathbf{z}(\mathbf{x}) = (z_1(\mathbf{x}), z_2(\mathbf{x}), \dots, z_q(\mathbf{x}))' \in \mathbb{R}^q$ can be also expressed as

$$\mathbf{z}(\mathbf{x}) = \phi(\mathbf{w}_0 + W\mathbf{x}), \quad (1)$$

where $\mathbf{w}_0 = \{w_{i,0}\}_{1 \leq i \leq q} \in \mathbb{R}^q$ and $W = \{w_{i,l}\}_{1 \leq i \leq q, 1 \leq l \leq d} \in \mathbb{R}^{q \times d}$.

If the network has a single layer, the new features $\mathbf{z}(\mathbf{x})$ are used directly to calculate the output. Otherwise, multiple layers are stacked. The output of each layer represents the input of the next one and so on for the following layers. A network with several layers is called a deep network. Considering a deep network architecture with $m \in \mathbb{N}$ layers, the mechanism behind can be formalised as follows:

$$\begin{aligned} \mathbf{z}^{(1)}(\mathbf{x}) &= \phi_1 \left(\mathbf{w}_0^{(1)} + W^{(1)}\mathbf{x} \right) \\ \mathbf{z}_j^{(2)}(\mathbf{x}) &= \phi_2 \left(\mathbf{w}_0^{(2)} + W^{(2)}\mathbf{z}^{(1)}(\mathbf{x}) \right) \\ &\dots \\ \mathbf{z}^{(m)}(\mathbf{x}) &= \phi_m \left(\mathbf{w}_0^{(m)} + W^{(m)}\mathbf{z}^{(m-1)}(\mathbf{x}) \right), \end{aligned}$$

where q_k indicates the number of units in the k -th layer (with $q_0 = d$), $\mathbf{w}_0^{(k)} \in \mathbb{R}^{q_k}$, $W^{(k)} \in \mathbb{R}^{q_k \times q_{k-1}}$ are the coefficients to optimise, and $\phi_k(\cdot)$ is the activation function applied by the k -th layer. The network learning is carried out by optimising the coefficient matrices and vectors according to a specific loss function. It poses the minimisation of a high-dimensional non-convex function, and the backpropagation algorithm is generally used. The coefficients are iteratively adjusted by minimising the error of the network outputs with respect to some reference values.

An extensive description of neural networks and backpropagation can be found in [32].

3.1.2. Embedding Neural Networks

Sometimes, information is available in the form of categorical variables. In the statistical literature, the standard procedures for dealing with categorical variables are one-hot and dummy encoding. Nevertheless, these coding schemes produce high-dimensional sparse vectors, which could present potential reliability problems in the obtained estimations when there are many categorical features or one of the variables presents many levels. Embedding is an innovative technique to analyse categorical variables that appeared for the first time in the natural language processing context [49]. With embedding, the ability of the neural network to learn representations has been used to encode the levels of

categorical variables into numerical vectors. The coordinates of the levels of the categorical variables in the new space are the network's parameters learned during the training process. Furthermore, the Euclidean distance in the new real-valued space of the levels indicates that they are similar concerning the target variable.

Let $\mathcal{B} = \{b_1, b_2, \dots, b_{n_b}\}$ be a discrete set of categories. An embedding layer is a function

$$z_b : \mathcal{B} \mapsto \mathbb{R}^{q_b}$$

where $q_b \in \mathbb{N}$ is a hyperparameter defined by the modeller describing the size of each embedding layer. This function is equivalent to applying different q_b -dimensional FCN layers to the levels of the categorical variable after one-hot encoding. In this setting, the output of each neuron will be equal to the weight multiplied by the only element equal to one. Consequently, they can be trained in the same way as the other network parameters related to the other layers [50]. The output of these layers can be seen as the coordinates of the levels of the categorical variables in the new q_b -dimensional space. Furthermore, the Euclidean distance in the new space of the levels indicates that they are similar concerning the target variable.

Denoting with $n_b = |\mathcal{B}|$ the number of levels of the categorical feature, the number of weights that must be learned in the embedding layer during training is $n_b \times q_b$.

3.1.3. Recurrent Neural Networks

Recurrent neural networks (RNNs) provide additional synapses that connect neurons cyclically [45]. The result of the previous elaborations is reprocessed as input in the following sessions. In this way, previsions are influenced by what has been learnt from the past. The recurrent nature of these architectures makes them powerful tools for processing sequential data [51,52].

Let $\mathbf{x}^{(t)} \in \mathbb{R}^d$, $0 \leq t \leq T$ be multivariate time series. Then, the output of an RNN layer with $q \in \mathbb{N}$ can be formulated as:

$$\mathbf{z}(\mathbf{x}) = \phi\left(\mathbf{w}_0 + W\mathbf{x}^{(t)} + U\mathbf{z}^{(t-1)}\right) \quad (2)$$

where $U = \{u_{j,k}\}_{1 \leq j,k \leq q} \in \mathbb{R}^{q \times q}$ represents the weights associated with the recurrent synapses (with an initial value $\mathbf{z}^{(0)} = \mathbf{0}^q$). The RNN training is carried out using backpropagation through time, which generally converges slowly and has vanishing gradient issues.

In order to overcome these problems, two more sophisticated cell-based architectures were proposed: long short-term memory (LSTM) and gated recurrent unit (GRU) networks.

3.1.4. Long Short-Term Networks

Long short-term memory (LSTM) networks, one of the earliest approaches to RNN architectures with nontrivial state control [46], in addition to the recurrent synapses containing the short-term memory, has an additional memory cell that stores and releases the long-term information through a system of subnets called gates. Combining these two types of memory, LSTMs have achieved excellent results in several fields. Greff et al. [53] offered an exhaustive overview of LSTM architectures.

Mathematically, denoting with $W^{(p)} \in \mathbb{R}^{q \times d}$, $U^{(p)} \in \mathbb{R}^{q \times q}$, and $\mathbf{w}_0^{(p)} \in \mathbb{R}^q$ the weights associated to each subnet $p \in \{i, o, f, z\}$, the output of the LSTM layer with $q \in \mathbb{N}$ can be described as follows:

$$\mathbf{i}^{(t)} = \sigma\left(\mathbf{w}_0^{(i)} + W^{(i)}\mathbf{x}^{(t)} + U^{(i)}\mathbf{z}^{(t-1)}\right) \quad (3)$$

$$\mathbf{o}^{(t)} = \sigma\left(\mathbf{w}_0^{(o)} + W^{(o)}\mathbf{x}^{(t)} + U^{(o)}\mathbf{z}^{(t-1)}\right) \quad (4)$$

$$\mathbf{f}^{(t)} = \sigma\left(\mathbf{w}_0^{(f)} + W^{(f)}\mathbf{x}^{(t)} + U^{(f)}\mathbf{z}^{(t-1)}\right) \quad (5)$$

$$\mathbf{z}^{(t)} = \tanh\left(\mathbf{w}_0^{(z)} + W^{(z)}\mathbf{x}^{(t)} + U^{(z)}\mathbf{z}^{(t-1)}\right) \quad (6)$$

$$\mathbf{c}^{(t)} = \mathbf{c}^{(t-1)} \odot \mathbf{f}^{(t)} + \mathbf{i}^{(t)} \odot \mathbf{z}^{(t)} \quad (7)$$

$$\mathbf{z}^{(t)}(\mathbf{x}) = \phi(\mathbf{c}^{(t)}) \odot \mathbf{o}^{(t)} \quad (8)$$

where $\sigma(\cdot) : \mathbb{R} \mapsto (0, 1)$ and $\tanh(\cdot) : \mathbb{R} \mapsto (-1, 1)$ are the sigmoid and the tanh activation function, respectively, and $\mathbf{c}^{(t)}$ is the state memory cell at time t (with an initial value $\mathbf{c}^{(0)} = \mathbf{0}^q$). The forget input and output gates regulate the mechanism of storing and releasing the information.

Chronologically, the forget gate is the first gate to act on the memory cell (Equation (5)). It uses the sigmoid function to delete a percentage of information considered obsolete. Next, the input gate (Equation (3)) adds to the output of the forget gate (Equation (7)) the new useful information extracted from the data input. Finally, the state of the memory cell $\mathbf{c}^{(t)}$ is combined with the output gate (Equation (6)) to compute the output value of the LSTM cell (Equation (8)).

The number of weights in an LSTM layer with q units is equal to $4 \times q \times (d + q + 1)$. A graphical illustration of the LSTM cell is showed on the left side of Figure 1.

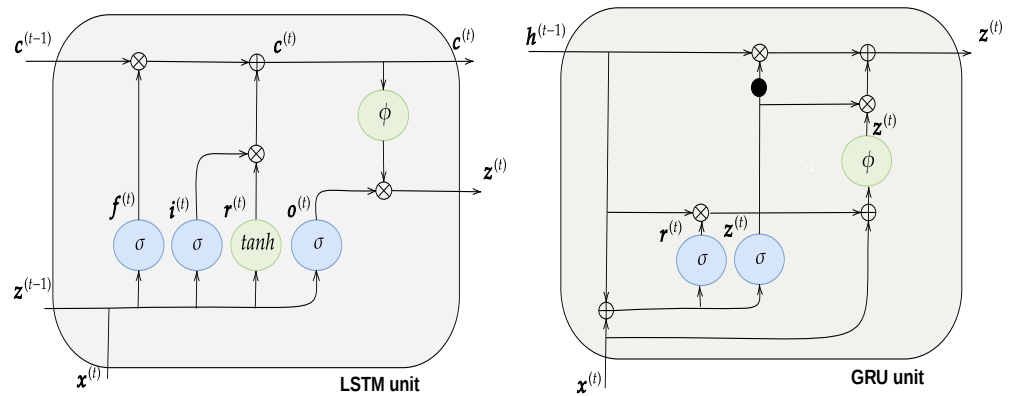


Figure 1. A pictorial representation of the LSTM unit (left) and GRU unit (right).

3.1.5. Gated Recurrent Unit Networks

Gated recurrent unit (GRU) networks [54] seem to be one of the most promising alternatives to LSTM. Being recently developed, their architecture does not provide a memory cell but includes two gates called the update gate and reset gate, which govern the flow of information. In fact, GRU represents a simplified version of LSTM and allows avoiding the gradient vanishing using a lower number of parameters. Using the previously adopted notation, the number of weights in a GRU layer with q units equal to $3 \times q \times (d + q + 1)$ is lower than an LSTM layer.

Let $W^{(p)} \in \mathbb{R}^{q \times d}$, $U^{(p)} \in \mathbb{R}^{q \times q}$, and $\mathbf{w}_0^{(p)} \in \mathbb{R}^q$ be the weights indexed by $p \in \{o, r, c\}$. The output of the q -dimensional GRU layer can be formalised as follows:

$$\mathbf{o}^{(t)} = \sigma \left(\mathbf{w}_0^{(o)} + W^{(o)} \mathbf{x}^{(t)} + U^{(o)} \mathbf{z}^{(t-1)} \right) \quad (9)$$

$$\mathbf{r}^{(t)} = \sigma \left(\mathbf{w}_0^{(r)} + W^{(r)} \mathbf{x}^{(t)} + U^{(r)} \mathbf{z}^{(t-1)} \right) \quad (10)$$

$$\mathbf{c}^{(t)} = \tanh \left(\mathbf{w}_0^{(c)} + W^{(c)} \mathbf{x}^{(t)} + U^{(r)} (\mathbf{r}^{(t)} \odot \mathbf{z}^{(t-1)}) \right) \quad (11)$$

$$\mathbf{z}^{(t)}(\mathbf{x}) = (1 - \mathbf{o}^{(t)}) \odot \mathbf{z}^{(t-1)} + \mathbf{o}^{(t)} \odot \mathbf{c}^{(t)} \quad (12)$$

The reset gate (Equation (10)), using the sigmoid activation function, determines when the hidden state has to ignore the previous hidden state and uses the input data (Equation (11)). The update gate (Equation (9)) helps the RNN store long-term information, determining how much of the previous memory should be kept. In fact, it realises a task similar to that performed by the memory cell in the LSTM network. Finally, the output of the GRU cell is computed by combining the outputs of both gates (Equation (12)).

Generally, a GRU layer contains several units, each having separate reset and update gates. This mechanism is used to learn the dependencies over different time scales. Some units will tend to have reset gates that are frequently active and therefore learning short-term dependencies, while other units will have most active update gates, and so they will be able to capture longer-term dependencies.

The number of weights in a GRU layer with q units equal to $3 \times q \times (d + q + 1)$ is lower compared with the number of parameters in an LSTM layer with the same number of units. A graphical representation of the GRU cell is shown on the right side of Figure 1.

3.1.6. Convolutional Neural Networks

Convolutional neural networks (CNN)s, proposed in [55], are feedforward NNs that make use of the mathematical operator called *convolution* to extract new feature maps from data. They work by multiplying subsets of the input data matrix by another matrix, called a filter, to extract useful latent features. CNNs typically process multidimensional arrays, and the number of dimensions to which the convolution operator is applied distinguishes 1D, 2D, and 3D convolutional networks.

The main characteristics that differentiate CNNs from the traditional fully connected ones are basically twofold [32]. The first is *local connectivity*, where nodes are only connected to a smaller region of the input, and each component of the feature maps is obtained by processing only a subset of the input data. The second is *weights sharing*, where each member of the kernel is used at every position of the input, meaning that all the nodes in the output detect exactly the same pattern.

Let $\mathbf{x}^{(t)} \in \mathbb{R}^d$, $0 \leq t \leq T$ be a multivariate time series of input variables. A 1D-CNN layer with filters $W^{(j)} = (\mathbf{w}_{j,1}, \mathbf{w}_{j,2}, \dots, \mathbf{w}_{j,m}) \in \mathbb{R}^{d \times m}$ and biases $w_{j,0} \in \mathbb{R}$ for $j = 1, \dots, q$ extracts a new feature map $\{z_{s,j}(\mathbf{x})\}_{0 \leq s \leq T+1-m, 1 \leq j \leq q}$ from the input data, where each component is a nonlinear function of some input data and is given by

$$z_{s,j}(\mathbf{x}) = \phi \left(w_{j,0} + (W^{(j)} * \mathbf{x})_s \right) = \phi \left(w_{j,0} + \sum_{l=1}^m \langle \mathbf{w}_{j,l}, \mathbf{x}^{(m+s-l)} \rangle \right).$$

where $*$ indicates the convolution operator and $\langle \cdot, \cdot \rangle$ denotes the scalar product in $\mathbb{R}^d$. It must be remarked that the index j refers to the different filters. Each filter $W^{(j)}$ produces a set of new features. The second index s refers to the different features extracted from a given filter.

3.2. The Model

This section describes the neural network models developed. Let $\mathcal{S} = \{s_1, s_2, \dots\}$ be the ordered set of the different seasons considered (i.e., quarterly data have $|\mathcal{S}| = 4$ and monthly data $|\mathcal{S}| = 12$) and $\mathcal{Y} = \{y_1, y_2, \dots\}$ be the set of years under investigation. We

denote by $\mathbf{r}^{(t)} \in \mathbb{R}^l$ the multivariate time series of the container traffic volume with a given level of detail at time t . The index t is supposed to vary in $\mathcal{T} = \{t \in \mathbb{N} : t \leq |\mathcal{Y} \times \mathcal{S}|\}$ such that each integer $t \in \mathcal{T}$ uniquely identifies a pair $(y, s) \in \mathcal{Y} \times \mathcal{S}$ according to the chronological order of the seasons and the years.

We model $\mathbf{r}^{(t)}$ as a function of the season $s \in \mathcal{S}$ and its previous $|\mathcal{S}| \in \mathbb{N}$ time lags:

$$\mathbf{r}^{(t)} = f(s, \mathbf{r}^{(t-1)}, \mathbf{r}^{(t-2)}, \dots, \mathbf{r}^{(t-|\mathcal{S}|)}). \quad (13)$$

We modelled the function $f(\cdot)$ using neural networks since they are known to be universal approximators [56]. Our neural network architecture combines some of the network architecture described in Section 3.1.

The season s is modelled using embedding networks. When choosing an embedding size $q_s \in \mathbb{N}$, an embedding layer maps

$$\mathbf{z}_s : \mathcal{S} \rightarrow \mathbb{R}^{q_s}, \quad p \mapsto \mathbf{z}_f(s). \quad (14)$$

where $\mathbf{z}_f(s)$ is a set of new features that summarises the information concerning the seasonal effect and represents s in a way that is optimised with respect to the desired output in a q_s real-valued dimensional space. Information related to the previous $|\mathcal{S}|$ time lags of $\mathbf{r}^{(t)}$ is processed by neural networks layers specifically designed to process sequential data such as LSTM, GRU, or CNN layers. For illustrative purposes, we will consider the LSTM networks, but an interesting comparison among these different neural networks architecture will be carried out in the numerical experiments section.

Let $R^{(t)} = \{\mathbf{r}^{(t_0)}\}_{t-|\mathcal{S}| \leq t_0 < t} \in \mathbb{R}^{l \times |\mathcal{S}|}$ be the matrix that contains the previous $|\mathcal{S}|$ observations of $\mathbf{r}^{(t)}$. An LSTM layer with $q_r \in \mathbb{N}$ units processes it as illustrated in equations $R^{(t)}$, and the output is

$$\mathbf{z}_r(r) = f(R^{(t)}).$$

The set of new features $\mathbf{z}_r(r) \in \mathbb{R}^{q_r}$ summaries the information related to the container traffic volume of the previous $|\mathcal{S}|$ time lags. Let $\mathbf{z} = (\mathbf{z}_s, \mathbf{z}_r)' \in \mathbb{R}^{q_s + q_r}$ be the full set of features obtained concatenating the output of the embedding and LSTM layer. The quantity of interest $\mathbf{r}^{(t)}$ can be calculated by processing $\mathbf{z}$ with a fully connected layer:

$$\mathbf{r}^{(t+1)} = \phi(\mathbf{b}_0 + W\mathbf{z}). \quad (15)$$

The embedding weights $(\mathbf{z}_s, \forall s \in \mathcal{S})$, the LSTM weights $(W^{(p)}, U^{(p)}, \mathbf{w}_0^{(p)}, \forall p \in \{r, z, c\})$, and the weights of the fully connected layer $(\mathbf{b}_0$ and $W)$ must be properly optimised in order to minimise the error of the network outputs with respect to the reference values. A graphical representation of the proposed neural network model is provided in Figure 2.

3.3. Data

In order to validate the effectiveness of the proposed methodology, we carried out some experiments on data downloaded from the website of the port of Barcelona. These data were also used in [39]. This port is one of the most important maritime logistics ports in the Mediterranean, with a land area of 10.653 square kilometers. Barcelona is the third largest port for container activity in Spain (after Valencia and Algeciras) and ninth in Europe. The port is managed by the Barcelona Port Authority. As one of the main ports in Spain—where it adopted the landlord governance model [16]—the Port of Barcelona is coordinated by Ports of the State, a state-owned organisation responsible for the meta-management of the 28 main Spanish port authorities that are in national seaports of general interest. In the port of Barcelona, there are two terminal operators specialised in container handling: “APM Terminals Barcelona” and “BEST—Barcelona Europe South Terminal”. Both these container operators manage international terminals—linked with rail facilities—with more than 3000 m of berthing line and 17 container cranes.

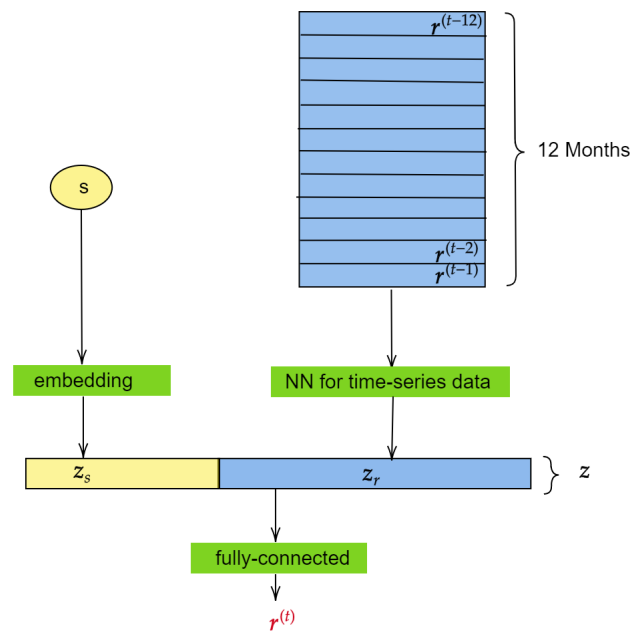


Figure 2. A graphical diagram of the proposed neural network model.

The website of the Port of Barcelona provides multivariate time series data related to container traffic. The dataset includes monthly observations from January 2010 to December 2020 ($|\mathcal{S}| = 12$, $|\mathcal{Y}| = 11$, and $\mathcal{T} = \{t \in \mathbb{N} : t \leq 132\}$) related to the count of containers, which are broken down into six categories: loaded and unloaded foreign, loaded and unloaded domestic, empty, and transit. A separate analysis of the different categories is also especially interesting in light of the reshaping of global supply chains, with a thrust toward regionalisation that is being amplified by attempts to build resilience in response to the COVID-19 outbreak.

First, explanatory data analysis was carried out. Table 1 reports some statistics (mean, coefficient of variation (CoV), and range) of our dataset. We observe that the time series under analysis presented very different magnitudes. For example, the volume of full unloaded domestic containers was the smallest, while the volume of containers in transit was the largest.

Table 1. Data statistics.

Type	Mean	CoV	Min	Max	Range
empty	48.3668	0.2611	27.5010	86.2680	58.7670
full loaded domestic	7.5032	0.1888	3.7720	10.4460	6.6740
full loaded foreign	52.8784	0.1842	26.3830	71.5410	45.1580
full transit	47.7746	0.6771	14.0320	126.9690	112.9370
full unloaded domestic	1.7183	0.2168	1.0070	2.8020	1.7950
full unloaded foreign	40.4105	0.1722	26.4280	56.8010	30.3730

Figure 3 provides pairwise scatter plots (below the main diagonal) and Pearson correlation coefficients (above the main diagonal) among the time series of interest, arranged in matrix form. From both, we noted moderate positive correlations. The 3 stars next to the correlation coefficients indicate that the corresponding p-values were less than 0.001. This evidence suggests that a multivariate model could exploit the information among the different series and produce more accurate predictions.

Figure 4 shows the box plots for the different container types grouped by month. The red boxes refer to the warm months, while the blue ones represent the cold months.

Some common elements can be observed among the different series. First, we observed that container traffic increased in the warm weather months, but this effect was more pronounced in some series and less so in others.

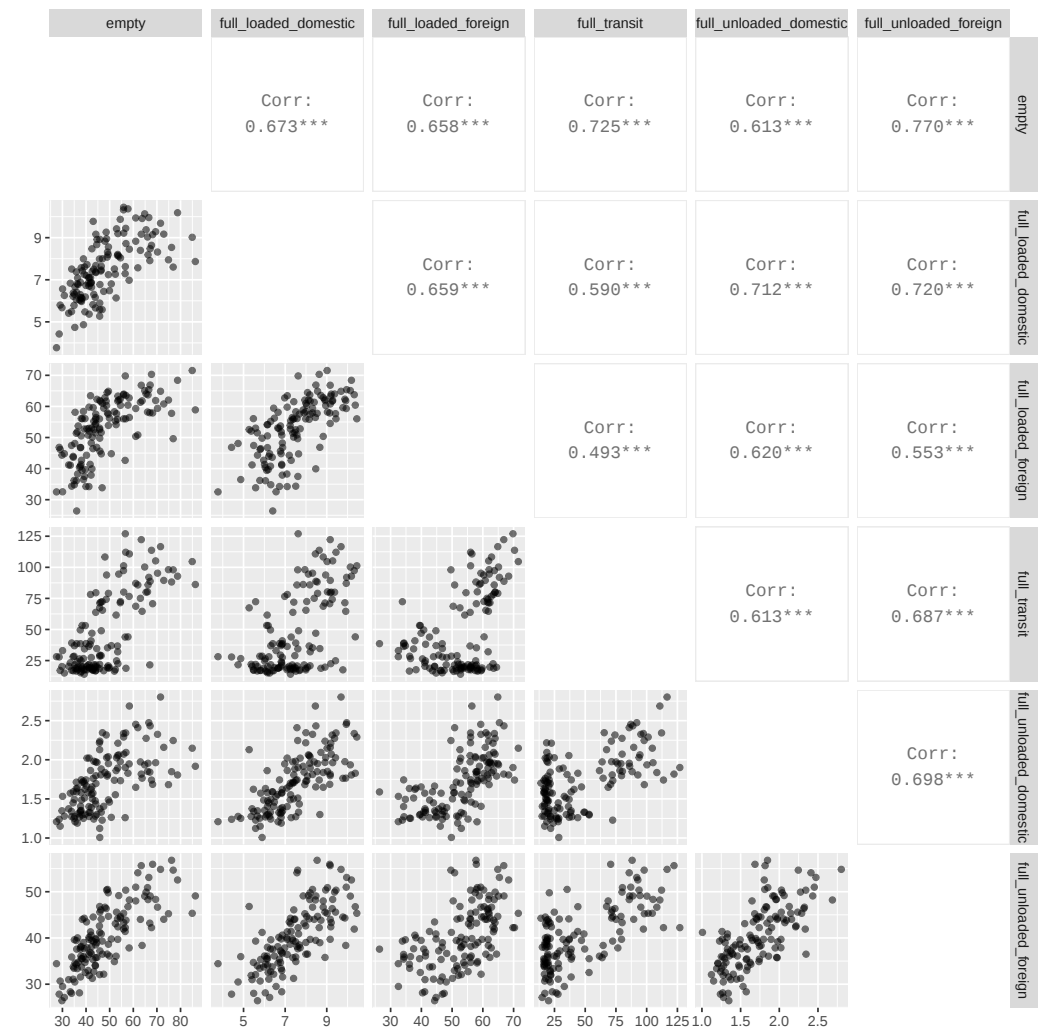


Figure 3. Pairwise scatter plots of the container traffic volume data. Correlation coefficients are marked with *** if the p -value of them being zero is less than 0.001.

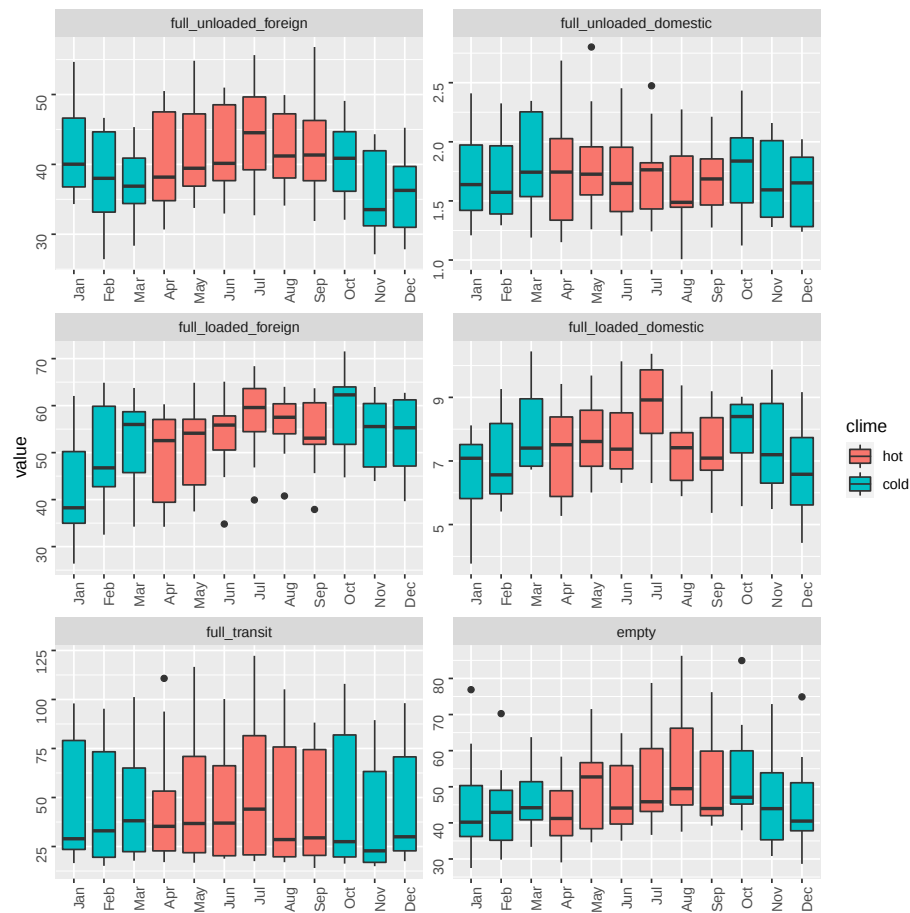


Figure 4. Seasonal box plots of the container traffic volume data.

4. Results

In this section, we present the results of some numerical experiments carried out on the data described in Section 3.3. A comparative analysis of the out-of-sample accuracy of some models in predicting the container traffic volume is carried out. The aim is to find the model able to more accurately forecast, out of sample, the container traffic volumes. We evaluated and compared the out-of-sample performance of the different methods using the mean absolute percentage error (MAPE):

$$\frac{1}{n} \cdot \sum_i \left| \frac{\hat{y}_i - y_i}{y_i} \right| * 100.$$

The comparison includes the standard SARIMA models, the deep learning model described in Section 3 based on LSTM and embedding networks, and two variants that use GRU and 1D-convolutional networks (in place of the LSTM network) to process the time-specific component.

Let $\mathcal{T}$ be the full set of the time points available. We considered two different forecasting scenarios:

- In the first scenario, the models were calibrated using data up to December 2018, and forecasts were made for the calendar year 2019. In other words, data were partitioned in the training set, including data related to $\mathcal{T}_{train}^{(1)} = \{t \in \mathcal{T} : t \leq 108\}$ (from January 2010 to December 2018) and a testing set $\mathcal{T}_{test}^{(1)} = \{t \in \mathcal{T} : 108 < t \leq 120\}$ (from January 2019 to December 2019) for measuring the out-of-sample performance.
- In the second scenario, the models were calibrated using data up to December 2019, and forecasts were made for the calendar year 2020. In this case, we defined the

training set and testing set as $\mathcal{T}_{train}^{(2)} = \{t \in \mathcal{T} : t \leq 120\}$ (from January 2010 to December 2019) and $\mathcal{T}_{test}^{(2)} = \{t \in \mathcal{T} : 120 < t \leq 132\}$ (from January 2020 to December 2020), respectively.

We treated these two years separately since 2020 presented very different features from other years.

First, we discuss the calibration of all the models investigated. As far as the SARIMA models are concerned, in order to find the most suitable model specification, we used the “auto.arima” algorithm available in the R package “forecast” [57]. The “auto.arima” algorithm explores the ARIMA model space, selecting a recommended model based on the Akaike information criterion (AIC) and unit root tests. Table 2 lists the models selected by the “auto.arima” algorithm for the different time series and for both forecasting scenarios. The autoregressive (AR), differentiation (I), and moving average (MA) orders for non-seasonal and seasonal components (denoted by the lower index 12) are reported with the resulting AIC quality index.

Table 2. Best SARIMA models selected.

	Type	Non-Seasonal Orders			Seasonal Orders			AIC
		AR	I	MA	AR ₁₂	I ₁₂	MA ₁₂	
2019 Forecasts	full unloaded foreign	0	1	2	0	1	1	477.7100
	full unloaded domestic	0	1	2	0	0	0	−73.9716
	full loaded foreign	5	1	0	1	0	1	648.0511
	full loaded domestic	2	1	0	1	0	0	224.9120
	full transit	0	1	1	0	0	2	792.7719
	empty	1	1	1	2	0	0	713.5212
2020 Forecasts	full unloaded foreign	0	1	2	1	1	2	540.6166
	full unloaded domestic	0	1	1	0	0	1	−58.0310
	full loaded foreign	0	1	2	0	0	2	717.8455
	full loaded domestic	1	1	1	1	0	0	252.6635
	full transit	0	1	1	1	0	0	882.7890
	empty	1	1	1	2	0	0	794.4201

As expected, in almost all cases, the algorithm selected models that provided seasonal components. Only for the full unloaded domestic data in the 2019 forecasts did the best model results have only non-seasonal terms.

Regarding the deep learning models, some details must be provided. According to machine learning best practices, preliminary pre-processing is performed. The time-specific data concerning the volume of container traffic were scaled to $[0, 1]$ via min-max scaling, while the month names were coded into integer values. We considered the minimum and maximum values in the training set in the min-max scaling. Furthermore, the minimum value were reduced by 10% of the variation range, and the same percentage increases were applied to the maximum value to ensure the testing set data were in $[0, 1]$ after the scaling. In all the architectures investigated, we set the size of the month embedding layer to $q_s = 5$. For the LSTM (GRU) model, we used an LSTM (GRU) layer with 64 units, while in the case of the CONV model, we used 64 filters of a 6×6 size. In the latter case, the filter matrices slid along the data matrix backwards along the time dimension. The filter weights were multiplied by the input data matrix (lag values of the container traffic volume) and combined in a non-linear fashion (via an activation function) to extract the optimal features with respect to the quantity of interest. In addition, three different activation functions were considered to process the time-specific component:

- The *linear* activation $\phi(x) = x$;
- The *tanh* $\phi(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$;
- The *relu* function $\phi(x) = \max(0, x)$.

The fitting of the neural networks was performed using the Adam optimiser [58], setting the parameters to the default values. The network weights were recursively adjusted, aiming to minimise the mean absolute error between the predicted and actual values. To

avoid overfitting, we used leave-out validation by using 10% of the training set as the validation set. The update of the weights was calculated using the remaining 90% of the data, and at each epoch, the loss function was evaluated on the training and validation sets. To avoid overfitting, we saved the weight configuration that produced the lowest value of the loss function on the validation set. This was a set of data that the neural network did not use during training. The performance obtained on this set could be a reliable estimate of the model's forecasting performance. A review of other possible validation techniques can be found in [59]. The results of training a neural network are sensitive to some components of the training process: the starting point of the optimisation algorithm, the random sampling of batches of training data employed to calculate the network weight updates, and so on. Then, analysing the results of a single run could be not sufficient to make a judgment. We fit each architecture 10 times and compared the performance on the testing set in terms of accuracy and stability.

Figure 5 shows the box plots of the outsample MAPE for all three networks (LSTM, GRU, and CONV) for the different activation functions. This analysis shows the sensitivity of the different models concerning the starting point of the optimisation algorithm and the random sampling of batches. Some evidence may be noted.

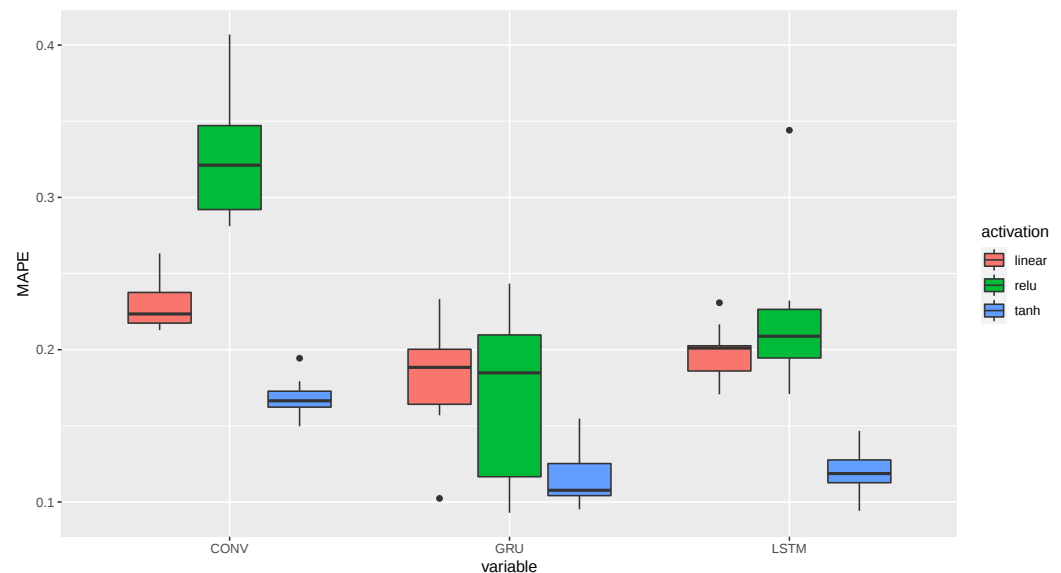


Figure 5. Forecasting MAPE box plots for CONV, GRU, and LSTM.

First, we observed that the tanh activation (blue boxes) function produced the best performance for all the network architectures investigated. The linear function (red boxes) was the runner-up, while the relu activation (green boxes) produced the lowest out-of-sample accuracy in our tests. Analysis of the variability of the results confirmed this ranking: the tanh activation produced less variable results except for the GRU model, while relu activation was associated with less stable results. According to these results, we can conclude that the tanh activation function appeared to be the best activation function for the investigated phenomenon.

Table 3 compares the results of the SARIMA models against the LSTM, CONV, and GRU models with tanh activation in both forecasting scenarios. For the deep learning models, we considered the ensemble prediction, obtained as the average of the forecasts in 10 different runs. The MAPEs obtained by the different methods were reported at the single time series level and at the aggregate level, where all values were summed. First, we observed that the errors increased dramatically when moving from the 2019 forecasting scenario to the 2020 forecasts. This is quite reasonable, since container traffic volume showed a highly irregular pattern over the last year due to the COVID-19 outbreak. In addition, it is visible that the LSTM and GRU methods produced a lower global MAPE

than the SARIMA models in both forecasting scenarios. On the contrary, the CONV model had poorer performances. This probably suggests that the evolution of container traffic can be better modelled by using recurrent neural network architectures than convolutional ones. Considering both scenarios, the GRU model appeared to be the best model. A further comparison between the performances of the GRU and SARIMA models is depicted in Figure 6 using barplots. Looking at the performance on individual series, in the 2019 forecasts, we observed that the improvement induced by the GRU models was especially large for the “full transit” and “empty” container volumes, which were two of the largest series in terms of size. In particular, the training of the deep learning models was carried out simultaneously on all data and pursued the minimisation of the global MAE in the training set. In this setting, it is not surprising that the learning algorithm put more weight on the larger series, which could produce larger residues, than the smaller ones. At the same time, the SARIMA approach appeared to be highly competitive on both “full loaded” series. The comparison between the forecasts obtained by the GRU and SARIMA models in this forecasting scenario is also graphically represented in Figure 7. In the 2020 forecasts, the GRU and LSTM models still overperformed the SARIMA model at the overall level, but the improvement appeared to be smaller. In this scenario, the only case in which the SARIMA approach yielded a lower MAPE than the GRU was the “full transit” container traffic volume. This evidence is in contrast with that observed in the first forecasting scenario. However, when analysing in greater detail the behaviour of the “full transit” container volume traffic, we observed that this series presented a large downward peak in the initial months of 2020 and a rebound in the summer months of 2020. All of the models investigated failed to predict the evolution of the “full transit” series, and this is not surprising given that these troughs were probably due to the COVID-19 outbreak. The reasons why the effects of the pandemic on the predictive ability of models were large in this particular series have yet to be clarified, and we plan to shed some light on this theme in a forthcoming work.

Table 3. Forecasting MAPE results for SARIMA, CONV, LSTM, and GRU models in 2019 and 2020.

	Type	SARIMA	CONV	LSTM	GRU
2019 Forecasts	empty	14.97%	19.94%	9.51%	11.83%
	full loaded domestic	7.43%	8.62%	11.27%	13.37%
	full loaded foreign	6.96%	8.00%	8.19%	11.34%
	full transit	23.48%	9.10%	10.14%	9.05%
	full unloaded domestic	14.86%	42.85%	11.99%	9.83%
	full unloaded foreign	7.56%	7.27%	6.27%	5.49%
	on aggregate	12.54%	15.96%	9.56%	10.15%
2020 Forecasts	empty	29.75%	19.51%	22.09%	21.13%
	full loaded domestic	20.28%	14.48%	19.60%	19.55%
	full loaded foreign	10.84%	10.58%	8.72%	9.43%
	full transit	19.08%	23.45%	28.42%	25.85%
	full unloaded domestic	12.39%	43.13%	9.36%	7.79%
	full unloaded foreign	13.10%	11.79%	14.46%	13.09%
	on aggregate	17.57%	20.49%	17.11%	16.14%

Figure 8 shows the standardised residuals obtained by the SARIMA and GRU model, distinguished by month for all the time-series data. We observed that both models presented positive and negative residuals in all months. This evidence suggests that the SARIMA and the GRU model did not have a systematic error component. Furthermore, we note that the amplitude of the errors was generally stable over the different months.

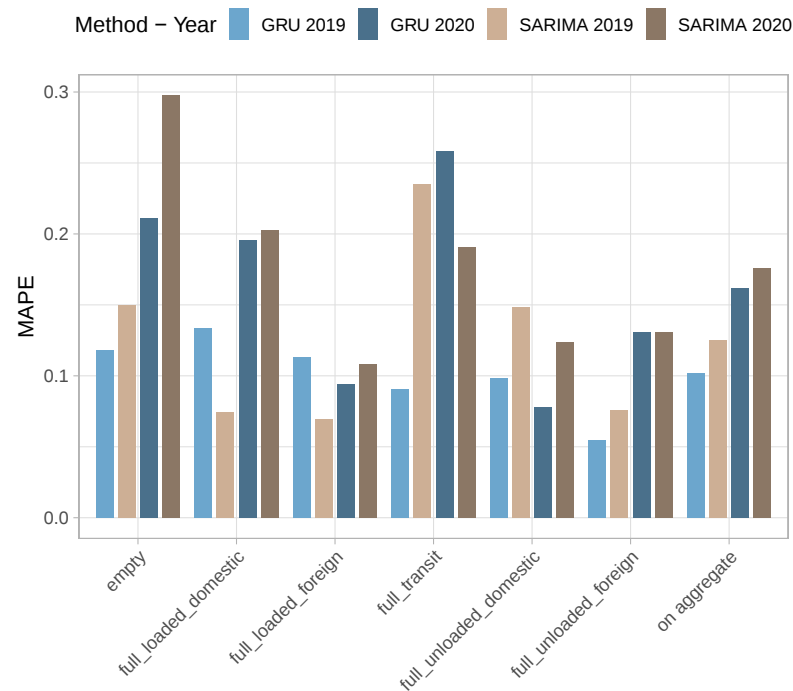


Figure 6. Performance of the SARIMA and GRU models expressed as MAPE for forecasting years 2019 (lighter) and 2020 (darker).

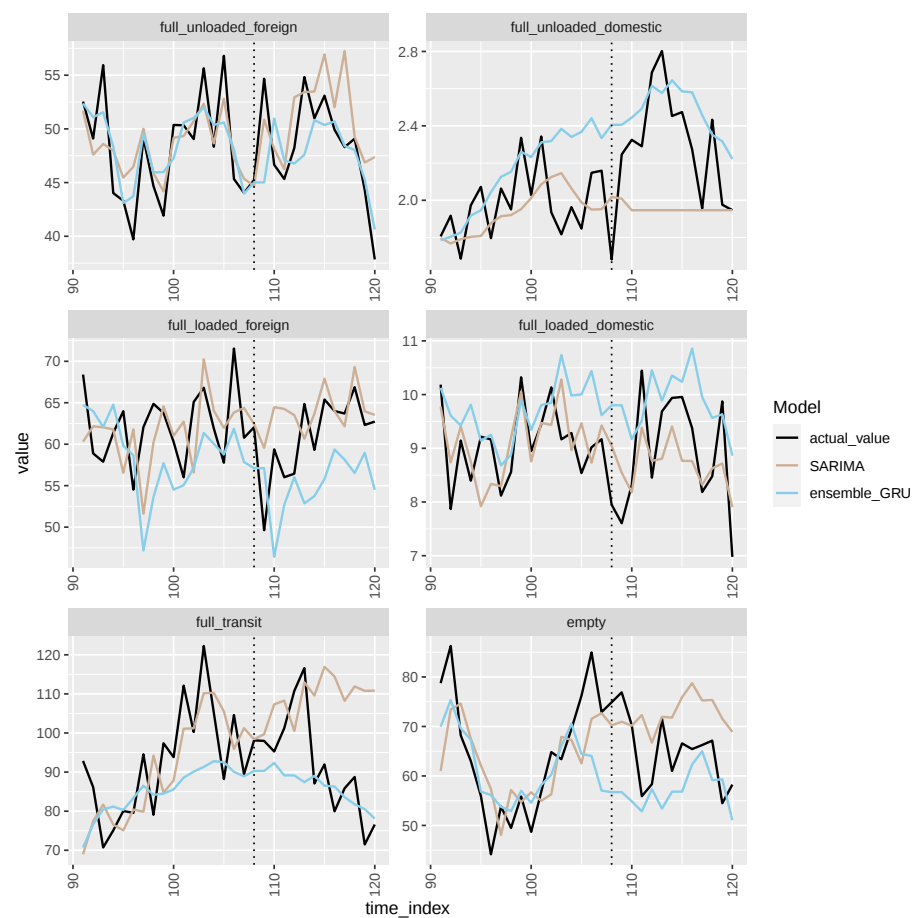


Figure 7. Time series predictions of GRU and SARIMA models.

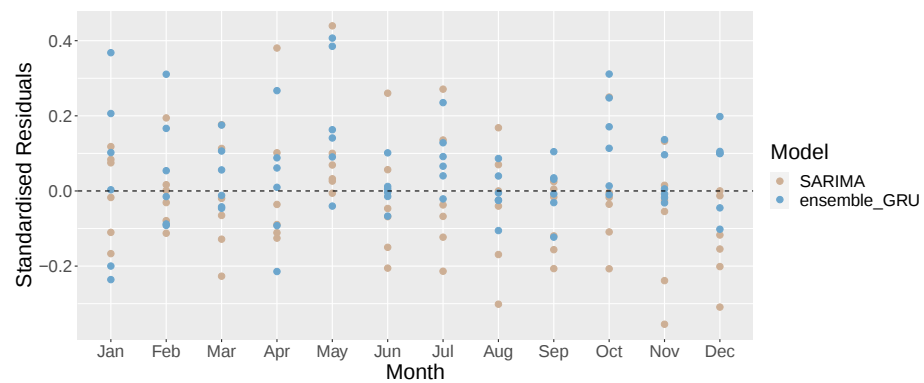


Figure 8. Standardised residuals of the SARIMA and GRU models for the different months.

We conclude this paragraph by analysing the representation of the seasonal components learned by the network during the training process. We focus on the GRU model in this part. The embedding layer, as described in Section 3.2, mapped the elements of the set $\mathcal{S}$ (the months of the year) in a $q_s = 5$ -dimensional space, in which they were represented in an optimal way with respect to the outcome variable. The Euclidean distance of the levels in the new space is a measure of their similarity with respect to the target variable. In our case, the distance measured the similarity of the months to the container traffic volume. Since the visualisation of a month embedded in a five-dimensional space was complex, we performed principal component analysis (PCA) on the embedding features to reduce the dimensionality of the data with a minimum loss of information. In particular, we extracted the first two (of five) components, and the months were mapped into a new two-dimensional space $\mathbb{R}^5 \mapsto \mathbb{R}^2$. In our numerical experiments, the first two principal components explained about 95% of the total variability and represented the months of the year in a new bivariate space that allowed a simple graphical representation. Figure 9 plots the months of the year in this reduced-size space. We observed that the months characterised by similar (average) temperatures tended to be close to each other in the new space. In particular, the warm weather months (in red) were concentrated at the new space's bottom, while the cold weather months (in blue) were located in the remaining part of the new space. This suggests that the network recognises similar patterns in data referring to times of the year characterised by a similar climate and assigned similar weights to months with similar characteristics.

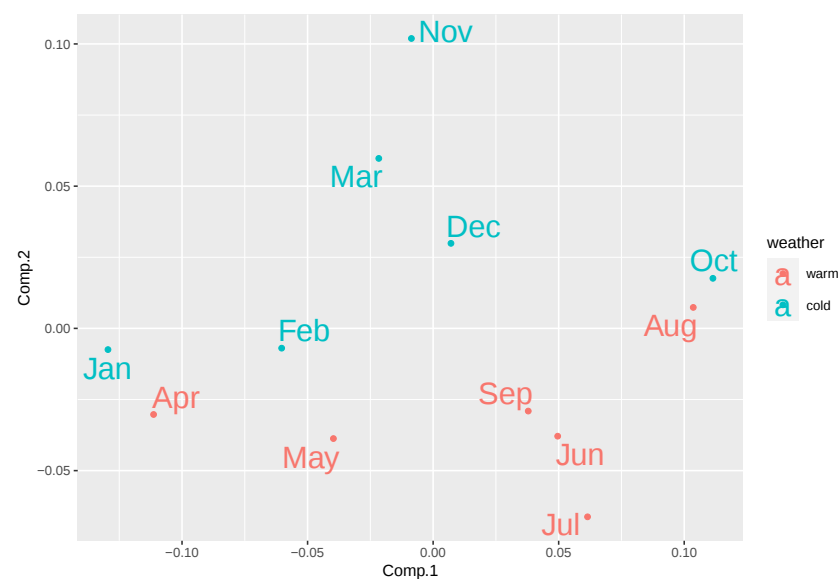


Figure 9. Graphical representation of the months embedding $z_s(s) \in \mathcal{R}^5$ after the dimensionality reduction (from 5 to 2) via principal component analysis.

5. Conclusions

The availability of multivariate predictive models for container throughput with a breakdown for domestic and foreign freight will be beneficial for ports, shippers, and terminal operators, as well as for national regulators and authorities. Bearing in mind the increased variability and the quick evolution of the scenario, further accelerated as various interventions are implemented to mitigate the pandemic and its fallouts, adding forecasting models based on neural networks to the arsenal of tools available has not only the potential to yield better estimates but to complete and reduce the variation when used in combination with other methods. The ability to rapidly identify trends will have effects on directing efforts towards building capacity, and this applies equally to human resources training and recruitment. Therefore, the adoption of deep learning forecasting models turns out to be useful not only for the container terminal operators (e.g., “APM Terminals Barcelona” and “BEST—Barcelona Europe South Terminal”) but also for the public organisations responsible for port management at the national level (e.g., Ports of the State) and local level (e.g., Barcelona Port Authority). In fact, the demand estimation for maritime container services allows an efficient management of ports from the perspective of global competitiveness. Moreover, a correct estimate of these data allows one to strategically sustain the stable development of vertical alliances between container terminal operators and shipping companies.

This paper sheds some light on the potential of predictive models based on neural networks to forecast the volume of containerised freight. The two recurrent neural network models outperformed SARIMA and the CNN in both of the forecasting scenarios analysed, achieving an absolute reduction in the MAPE—for the aggregate data—of about 3 percentage points in the first scenario and 1.5 percentage points in the second. The embedding realised by a neural network showed a clear spatial separation between the warmer and cooler months. The proposed approach can be extended, keeping in mind external regressors so that additional available information can be integrated. Future work will also include a refinement of the embedding used to build a meaningful representation of seasonality so that additional insights can be gained, as well as an investigation of hierarchical models.

Author Contributions: Conceptualisation, all authors; methodology, U.F., F.P. and S.S.; validation, all authors; data curation, S.S.; writing—original draft preparation, S.S.; writing—review and editing, U.F., M.R. and S.S.; visualisation, U.F. and S.S.; supervision, M.F. and F.P. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable, the study does not report any data.

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

Abbreviations

The following abbreviations are used in this manuscript:

AIC	Akaike information criterion
ANN	Artificial neural network
ARIMA	Autoregressive integrated moving average
CNN	Convolutional neural network
GRU	Gated recurrent unit
LSTM	Long short-term memory
RNN	Recurrent neural network
TEU	Twenty-foot equivalent unit

References

- Petering, M.E. Effect of block width and storage yard layout on marine container terminal performance. *Transp. Res. Part E Logist. Transp. Rev.* **2009**, *45*, 591–610. [\[CrossRef\]](#)
- Coto-Millán, P.; Baños-Pino, J.; Castro, J.V. Determinants of the demand for maritime imports and exports. *Transp. Res. Part E Logist. Transp. Rev.* **2005**, *41*, 357–372. [\[CrossRef\]](#)
- Cullinane, K.; Wang, T.F.; Song, D.W.; Ji, P. The technical efficiency of container ports: Comparing data envelopment analysis and stochastic frontier analysis. *Transp. Res. Part A Policy Pract.* **2006**, *40*, 354–374. [\[CrossRef\]](#)
- Parola, F.; Risitano, M.; Ferretti, M.; Panetti, E. The drivers of port competitiveness: A critical review. *Transp. Res.* **2017**, *37*, 116–138. [\[CrossRef\]](#)
- Parola, F.; Notteboom, T.; Satta, G.; Rodrigue, J.P. Analysis of factors underlying foreign entry strategies of terminal operators in container ports. *J. Transp. Geogr.* **2013**, *33*, 72–84. [\[CrossRef\]](#)
- Parola, F.; Satta, G.; Panayides, P.M. Corporate strategies and profitability of maritime logistics firms. *Marit. Econ. Logist.* **2015**, *17*, 52–78. [\[CrossRef\]](#)
- Haralambides, H.E. Gigantism in container shipping, ports and global logistics: A time-lapse into the future. *Marit. Econ. Logist.* **2019**, *21*, 1–60. [\[CrossRef\]](#)
- Notteboom, T.; Pallis, A.; Rodrigue, J.P. Port Economics. In *Management and Policy*; Routledge: New York, NY, USA, 2020.
- Chu, C.Y.; Huang, W.C. Determining container terminal capacity on the basis of an adopted yard handling system. *Transp. Res.* **2005**, *25*, 181–199. [\[CrossRef\]](#)
- Gharehgozli, A.; Zaerpour, N.; de Koster, R. Container terminal layout design: Transition and future. *Marit. Econ. Logist.* **2019**, *22*, 610–639. [\[CrossRef\]](#)
- Yap, W.Y.; Lam, J.S. Competition dynamics between container ports in East Asia. *Transp. Res. Part A Policy Pract.* **2006**, *40*, 35–51. [\[CrossRef\]](#)
- Anderson, C.M.; Opaluch, J.J.; Grigalunas, T.A. The demand for import services at US container ports. *Marit. Econ. Logist.* **2009**, *11*, 156–185. [\[CrossRef\]](#)
- Tavasszy, L.; Minderhoud, M.; Perrin, J.F.; Notteboom, T. A strategic network choice model for global container flows: specification, estimation and application. *J. Transp. Geogr.* **2011**, *19*, 1163–1172. [\[CrossRef\]](#)
- Yap, W.Y.; Loh, H.S. Next generation mega container ports: Implications of traffic composition on sea space demand. *Marit. Policy Manag.* **2019**, *46*, 687–700. [\[CrossRef\]](#)
- Fung, M.K. Forecasting Hong Kong's container throughput: An error-correction model. *J. Forecast.* **2002**, *21*, 69–80. [\[CrossRef\]](#)
- Verhoeven, P. A review of port authority functions: Towards a renaissance? *Marit. Policy Manag.* **2010**, *37*, 247–270. [\[CrossRef\]](#)
- Franc, P.; Van der Horst, M. Understanding hinterland service integration by shipping lines and terminal operators: A theoretical and empirical analysis. *J. Transp. Geogr.* **2010**, *18*, 557–566. [\[CrossRef\]](#)
- Zhu, S.; Zheng, S.; Ge, Y.E.; Fu, X.; Sampaio, B.; Jiang, C. Vertical integration and its implications to port expansion. *Marit. Policy Manag.* **2019**, *46*, 920–938. [\[CrossRef\]](#)
- Ferretti, M.; Parola, F.; Risitano, M.; Vitiello, I. Planning and concession management under port co-operation schemes: A multiple case study of Italian port mergers. *Res. Transp. Bus. Manag.* **2018**, *26*, 5–13. [\[CrossRef\]](#)
- Parola, F.; Pallis, A.A.; Risitano, M.; Ferretti, M. Marketing strategies of Port Authorities: A multi-dimensional theorisation. *Transp. Res. Part A Policy Pract.* **2018**, *111*, 199–212. [\[CrossRef\]](#)
- Mandják, T.; Lavissière, A.; Hofmann, J.; Bouchery, Y.; Lavissière, M.C.; Faury, O.; Sohier, R. Port marketing from a multidisciplinary perspective: A systematic literature review and lexicometric analysis. *Transp. Policy* **2019**, *84*, 50–72. [\[CrossRef\]](#)
- Kaliszewski, A.; Kozłowski, A.; Dkabrowski, J.; Klimek, H. Key factors of container port competitiveness: A global shipping lines perspective. *Mar. Policy* **2020**, *117*, 103896. [\[CrossRef\]](#)
- Castellano, R.; Fiore, U.; Musella, G.; Perla, F.; Punzo, G.; Risitano, M.; Sorrentino, A.; Zanetti, P. Do Digital and Communication Technologies Improve Smart Ports? A Fuzzy DEA Approach. *IEEE Trans. Ind. Inform.* **2019**, *15*, 5674–5681. [\[CrossRef\]](#)
- Molavi, A.; Lim, G.J.; Race, B. A framework for building a smart port and smart port index. *Int. J. Sustain. Transp.* **2020**, *14*, 686–700. [\[CrossRef\]](#)
- Ferretti, M.; Parmentola, A.; Parola, F.; Risitano, M. Strategic monitoring of port authorities activities: Proposal of a multi-dimensional digital dashboard. *Prod. Plan. Control* **2017**, *28*, 1354–1364. [\[CrossRef\]](#)
- Nikolopoulos, K.; Punia, S.; Schäfers, A.; Tsinopoulos, C.; Vasilakis, C. Forecasting and planning during a pandemic: COVID-19 growth rates, supply chain disruptions, and governmental decisions. *Eur. J. Oper. Res.* **2021**, *290*, 99–115. [\[CrossRef\]](#) [\[PubMed\]](#)
- Box, G.E.; Jenkins, G.M.; Reinsel, G.C.; Ljung, G.M. *Time Series Analysis: Forecasting and Control*; John Wiley & Sons: Hoboken, NJ, USA, 2015.
- Chan, H.K.; Xu, S.; Qi, X. A comparison of time series methods for forecasting container throughput. *Int. J. Logist. Res. Appl.* **2019**, *22*, 294–303. [\[CrossRef\]](#)
- Peng, W.Y. The comparison of the seasonal forecasting models: A study on the prediction of imported container volume for international container ports in Taiwan. *Marit. Q.* **2006**, *25*, 21–36.
- Peng, W.Y.; Chu, C.W. A comparison of univariate methods for forecasting container throughput volumes. *Math. Comput. Model.* **2009**, *50*, 1045–1057. [\[CrossRef\]](#)
- Schulze, P.M.; Prinz, A. Forecasting container transshipment in Germany. *Appl. Econ.* **2009**, *41*, 2809–2815. [\[CrossRef\]](#)

32. Goodfellow, I.; Bengio, Y.; Courville, A.; Bengio, Y. *Deep Learning*; MIT Press Cambridge: Cambridge, MA, USA, 2016; Volume 1.
33. Vapnik, V. *The Nature of Statistical Learning Theory*; Springer Science & Business Media: Berlin/Heidelberg, Germany, 2013.
34. Wang, Y.; Meng, Q. Integrated method for forecasting container slot booking in intercontinental liner shipping service. *Flex. Serv. Manuf. J.* **2019**, *31*, 653–674. [\[CrossRef\]](#)
35. Chen, Y.; Liu, B. Forecasting Port Cargo Throughput Based on Grey Wave Forecasting Model with Generalized Contour Lines. *J. Grey Syst.* **2017**, *29*, 51.
36. Barua, L.; Zou, B.; Zhou, Y. Machine learning for international freight transportation management: a comprehensive review. *Res. Transp. Bus. Manag.* **2020**, *34*, 100453. [\[CrossRef\]](#)
37. Wei, C.; Yang, Y. A study on transit containers forecast in Kaohsiung port: Applying artificial neural networks to evaluating input variables. *J. Chin. Inst. Transp.* **1999**, *11*, 1–20.
38. Gosasang, V.; Chandraprakasikul, 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. [\[CrossRef\]](#)
39. Milenković, M.; Milosavljević, N.; Bojović, N.; Val, S. Container flow forecasting through neural networks based on metaheuristics. *Oper. Res.* **2021**, *21*, 965–997. [\[CrossRef\]](#)
40. Xie, G.; Wang, S.; Zhao, Y.; Lai, K.K. Hybrid approaches based on LSSVR model for container throughput forecasting: A comparative study. *Appl. Soft Comput.* **2013**, *13*, 2232–2241. [\[CrossRef\]](#)
41. Wang, G.; Ledwoch, A.; Hasani, R.M.; Grosu, R.; Brintrup, A. A generative neural network model for the quality prediction of work in progress products. *Appl. Soft Comput.* **2019**, *85*, 105683. [\[CrossRef\]](#)
42. Zhou, X.; Dong, P.; Xing, J.; Sun, P. Learning dynamic factors to improve the accuracy of bus arrival time prediction via a recurrent neural network. *Future Internet* **2019**, *11*, 247. [\[CrossRef\]](#)
43. Du, P.; Wang, J.; Yang, W.; Niu, T. Container throughput forecasting using a novel hybrid learning method with error correction strategy. *Knowl.-Based Syst.* **2019**, *182*, 104853. [\[CrossRef\]](#)
44. Shankar, S.; Ilavarasan, P.V.; Punia, S.; Singh, S.P. Forecasting container throughput with long short-term memory networks. *Ind. Manag. Data Syst.* **2019**, *120*, 425–441. [\[CrossRef\]](#)
45. Elman, J.L. Finding structure in time. *Cogn. Sci.* **1990**, *14*, 179–211. [\[CrossRef\]](#)
46. Hochreiter, S.; Schmidhuber, J. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [\[CrossRef\]](#) [\[PubMed\]](#)
47. Gers, F.A.; Schmidhuber, J.; Cummins, F. Learning to Forget: Continual Prediction with LSTM. *Neural Comput.* **2000**, *12*, 2451–2471. [\[CrossRef\]](#)
48. Graziani, S.; Xibilia, M.G. Innovative Topologies and Algorithms for Neural Networks. *Future Internet* **2020**, *12*, 117. [\[CrossRef\]](#)
49. Bengio, Y.; Ducharme, R.; Vincent, P. A Neural Probabilistic Language Model. Available online: <https://proceedings.neurips.cc/paper/2000/file/728f206c2a01bf572b5940d7d9a8fa4c-Paper.pdf> (accessed on 19 June 2022).
50. Guo, C.; Berkhahn, F. Entity embeddings of categorical variables. *arXiv* **2016**, arXiv:1604.06737.
51. Sermpinis, G.; Karathanasopoulos, A.; Rosillo, R.; de la Fuente, D. Neural networks in financial trading. *Ann. Oper. Res.* **2019**, *297*, 293–308. [\[CrossRef\]](#)
52. Kumar, A.; Singh, J.P.; Dwivedi, Y.K.; Rana, N.P. A deep multi-modal neural network for informative Twitter content classification during emergencies. *Ann. Oper. Res.* **2020**, 1–32. [\[CrossRef\]](#)
53. Greff, K.; Srivastava, R.K.; Koutník, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2016**, *28*, 2222–2232. [\[CrossRef\]](#)
54. 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. *arXiv* **2014**, arXiv:1406.1078.
55. LeCun, Y.; Boser, B.E.; Denker, J.S.; Henderson, D.; Howard, R.E.; Hubbard, W.E.; Jackel, L.D. Handwritten digit recognition with a back-propagation network. In Proceedings of the Advances in Neural Information Processing Systems, Denver, CO, USA, 26–29 November 1990; pp. 396–404.
56. Hornik, K.; Stinchcombe, M.; White, H. Multilayer feedforward networks are universal approximators. *Neural Netw.* **1989**, *2*, 359–366. [\[CrossRef\]](#)
57. Hyndman, R.J.; Khandakar, Y. *Automatic Time Series for Forecasting: The Forecast Package for R*; Number 6/07; Monash University, Department of Econometrics and Business Statistics: Melbourne, Australia, 2007.
58. Kingma, D.P.; Ba, J. Adam: A method for stochastic optimization. *arXiv* **2014**, arXiv:1412.6980.
59. Hastie, T.; Tibshirani, R.; Friedman, J.H.; Friedman, J.H. *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*; Springer: Berlin/Heidelberg, Germany, 2009; Volume 2.