



BigLSTM: Recurrent neural network for the treatment of anomalous temporal signals. Application in the prediction of endotracheal obstruction in COVID-19 patients in the intensive care unit[☆]

Pablo Fernández-López^a, Patricio García Báez^b, Ylermi Cabrera-León^a,
Juan L. Navarro-Mesa^c, Guillermo Pérez-Acosta^d, José Blanco-López^d,
Carmen Paz Suárez-Araujo^a,*

^a Instituto Universitario de Cibernética, Empresa y Sociedad (IUCES), Universidad de Las Palmas de Gran Canaria, Campus Universitario de Tafira, Las Palmas de Gran Canaria, 35017, Canary Island, Spain

^b Departamento de Ingeniería Informática y de Sistemas, Universidad de La Laguna, Camino San Francisco de Paula, 19, San Cristobal de La Laguna, 38200, Canary Island, Spain

^c Instituto para el Desarrollo Tecnológico y la Innovación en Comunicaciones (IDeTIC), Universidad de Las Palmas de Gran Canaria, Campus Universitario de Tafira, Las Palmas de Gran Canaria, 35017, Canary Island, Spain

^d Complejo Hospitalario Universitario Insular Materno Infantil de Gran Canaria, Avenida Marítima del Sur s/n, Las Palmas de Gran Canaria, 35016, Canary Island, Spain

ARTICLE INFO

Keywords:

Recurrent neural network
Deep learning
Missing data
Irregular sampling
Forecasting
LSTM
Endotracheal obstruction
COVID-19

ABSTRACT

Real-world applications, particularly in the medical field, often handle irregular time signals (ITS) with non-uniform intervals between measurements. These irregularities arise due to missing data, inconsistent sampling frequencies, and multi-sensor signals from different sources. Predicting outcomes using ISMTS is complex, especially when missing data is involved.

This paper introduces the Binomial Gate LSTM (BigLSTM), a modular Recurrent Neural Network model designed to process ISMTS. Built on the LSTM network, BigLSTM integrates techniques for handling irregular time intervals and multiple sampling rates by injecting information redundancy. BigLSTM comprises five interconnected modules. Four are dedicated to information processing: Information Distribution, Central Computing, Predictive, and Time Axis Processing Modules. These modules ensure the redundancy of system, making it tolerant to missing data. The fifth module, LSTM Cells On/Off Control, manages the internal operations of the network.

BigLSTM was tested on a critical clinical problem: predicting endotracheal obstruction in COVID-19 patients in intensive care units using ventilatory signals from 96 patients. BigLSTM achieved a mean validation mean squared error (MSE) of 0.028 for patients with obstructions and 0.2 for the entire dataset.

Additionally, we analysed the prediction tendencies of the system, finding an advance trend of 3.87 days and a delay trend of 2.15 days for distant predictions (7 days), with shorter intervals for near predictions (48 h). BigLSTM provided an obstruction prediction, in the short-term, not earlier than the next 10.64 h, and not later than the next 6.8 days, with a confidence percentage of 95%, indicating its effectiveness in handling irregular time series data.

1. Introduction

Time series have been widely used in various applications and models [1], including health care [2–4], social sciences [5,6],

biology [7], finance and behaviour analysis [8–10], and wherever data are measured, calculated or predicted.

Most of these models have assumed that the data associated with the time series are uniform and complete. However, when working

[☆] This work was funded by the Consejería de Vicepresidencia Primera y de Obras Públicas, Infraestructuras, Transporte y Movilidad del Cabildo de Gran Canaria, Spain under Grant Number “23/2021”. The funding agency had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of this manuscript, or in the decision to publish the results.

* Corresponding author.

E-mail addresses: pablo.fernandezlopez@ulpgc.es (P. Fernández-López), carmenpaz.suarez@ulpgc.es (C.P. Suárez-Araujo).

<https://doi.org/10.1016/j.complbiomed.2025.110146>

Received 31 August 2024; Received in revised form 1 April 2025; Accepted 3 April 2025

Available online 23 April 2025

0010-4825/© 2025 Elsevier Ltd. All rights reserved, including those for text and data mining, AI training, and similar technologies.

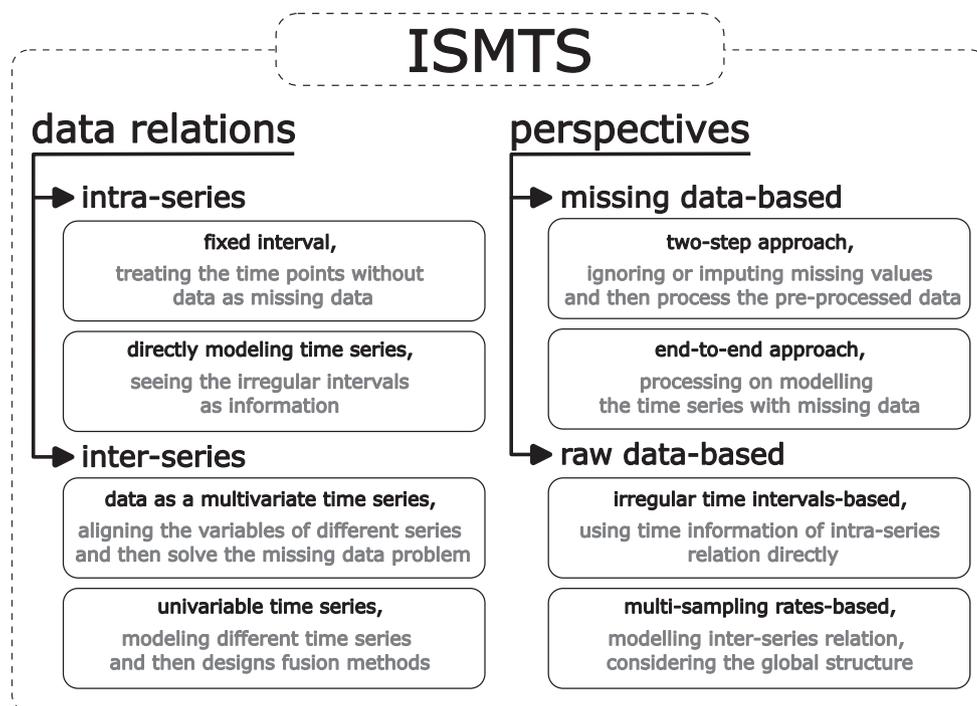


Fig. 1. ISMITS data relations and perspectives.

Source: Adapted from Sun C. et al. 2020 [1].

with real-world applications and models, time series observations tend to have non-uniform time intervals between successive measurements because of (1) time series with missing data due to possible breaks in device sensors, interrupted data transmissions, or damaged storage devices; (2) sampling devices that cannot maintain a constant sampling frequency; and (3) different time series that come from different sources with different sampling frequencies.

When we work with multiple time series with irregular sampling (ISTS), the data can present not only irregular time intervals between the observations of the same series but also different sampling frequencies between different series. Especially in the medical environment, where electronic medical records are widely used, it is common to find these irregularities; therefore, it is necessary to work with irregularly sampled medical time series (ISMITS) [1].

Sun et al. 2020 [1] catalogued the series involved in previous situations, identifying two types of relationships between the data that formed them: data relationships within a time series (intraseries), and data relationships between different time series (interseries).

These authors proposed to address the first of the problems (irregular intraseries sampling intervals) in two different ways: (1) determine a fixed sampling interval that assumes missing data are being produced at those points where there are no data, and (2) directly model the time series as they are and consider the irregular intervals presented as information (see Fig. 1).

The irregularities between series are mainly associated with the multiple and different sampling frequencies associated with the different time series. There are two ways to approach the multiple sampling problem: (1) consider the data as multivariate time series or (2) process multiple time series of a single variable separately. In the first approach, the variables of different series are aligned in the same dimension, and then the problem of missing data is solved. In the second approach, different time series are modelled simultaneously, and fusion methods are designed between them.

The identification and solution proposals associated with the irregularities of the ISMITS carried out by Sun et al. 2020 [1] can be summarized from the following two approaches: (1) an approach based

on missing data and (2) an approach based on modelling the data, as shown in Fig. 1.

In the missing data approach, all-time series are considered to have uniform time intervals, and irregular series are treated as having missing data at those time points where data are not available. This problem is usually solved by searching for an adequate level of precision in the data imputation process. In the approach based on modelling the raw data, the irregular data are used directly, considering that the irregular time itself is valuable information that should be used in the modelling and processing of the ISMITS.

Therefore, the first of the above approaches includes models that can address missing values in the time series and are divided into two subcategories: (1) two-step models and (2) end-to-end models. Two-step models can ignore or impute the missing values in the data to advance to the real process, which engages with data that are considered preprocessed (whether or not there is data imputation). End-to-end models carry out their process based on the modelling of time series with missing data.

The raw data approach works directly with the ISMITS as input. There are also two subcategories: (1) models based on irregular time intervals and (2) models based on multiple sampling rates. Models based on irregular time intervals model unevenly spaced data directly using temporal information from within-run relationships. Models based on multiple sampling frequencies can model the interseries relationship by considering the overall structure of the ISMITS.

In recent years, there has been a significant increase in the development of specialized models and architectures for the prediction and treatment of time series with irregular sampling and a high level of missing data. Making predictions for an ISMITS with missing data is a difficult task. The importance of the temporal variable in the ISMITS makes prediction a complicated process compared to other forms of data analysis. Ignoring the missing values can destroy the continuity of an ISMITS. Replacing the missing values alters the original ISMITS and can seriously affect prediction reliability. This complicates the methods and mechanisms of prediction evaluation. Among the existing techniques that allow prediction models to work with ISMITSs with irregular sampling are those applicable in the data preprocessing phase

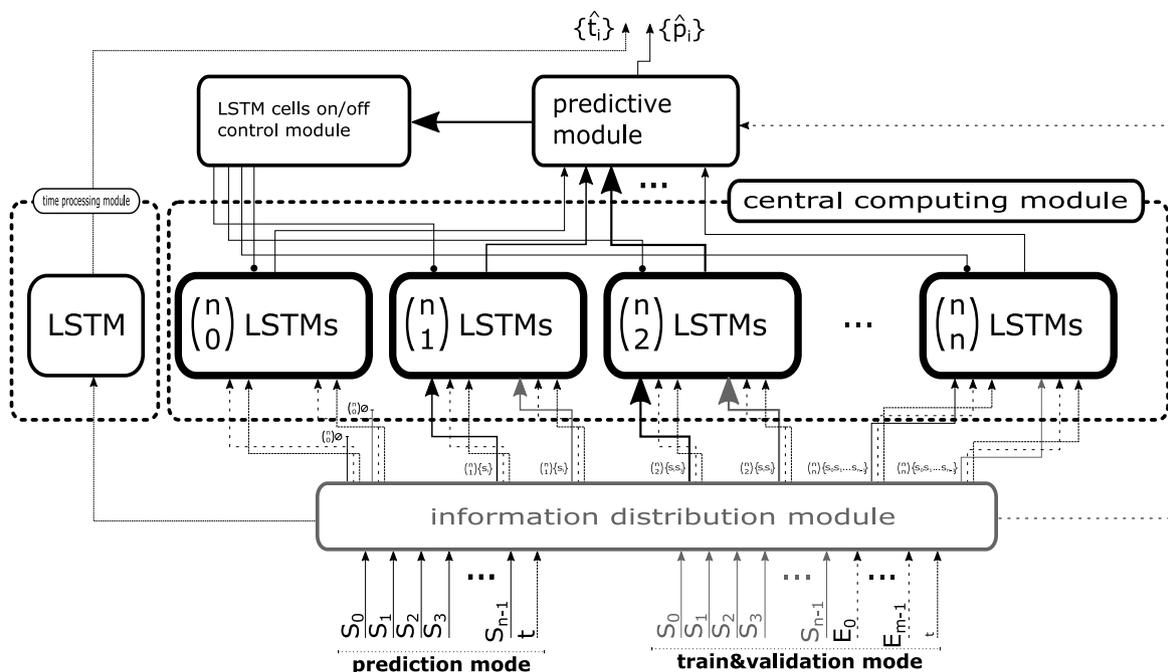


Fig. 2. Functional and modular diagram of BigLSTM.

and those that modify the models, from their structural and operational perspective, so that they can work with data that present irregular sampling. First, we can highlight the different techniques used to make the data present a regular sample, as well as to make it more appropriate to how the model in question needs it. The above processes include discretization, cutting and interpolation [11–13].

Conventional data imputation methods in the preprocessing phase use statistics to fill in the gaps corresponding to such missing data, and there are various techniques [14]. There are also fewer conventional methods in which specific algorithms are built to calculate the values that are used to fill in and condition an ISMTS [15–18]. Among the latter, we highlight those that use recurrent neural networks as well as computational architectures in the field of image processing using DL models to perform pertinent calculations of the missing data in question. Among the most commonly used, we highlight the use of adversarial generative networks [19,20] and context encoders [21].

In this work, we detail BigLSTM, a recurrent neural network model for treating anomalous temporal signals such as those found in an ISMTS. The BigLSTM is a modular system that incorporates techniques typical of models based on irregular time intervals (modelling the ISTS as they are and incorporating an effective mechanism to manage the time axis of the observations) and models based on multiple sampling rates (by injecting redundancy). From the perspective of the biological plausibility of the proposed model, the importance of redundancy in the brain has been recognized for different purposes [22], including its plastic capacity and for reconstructing input patterns, especially in information-lacking environments [23]. Therefore, BigLSTM is an architecture with biological plausibility that can capture the long-term temporal dependencies in ISMTSs. In this work, we analysed the performance of BigLSTM by examining a real problem in the clinical field: the clinical care of SARS-CoV-2 patients in intensive care units (ICUs). We used the BigLSTM model to predict the time at which endotracheal obstruction may occur in COVID-19 patients admitted to the intensive care unit (ICU) during the first and second waves of this disease in Spain.

We organized this work into two main sections, in addition to the introduction and conclusions. Section 2 presents the BigLSTM model in detail, formalizing all the aspects associated with its modular structure and operation. Section 3 presents the application of BigLSTM to a real

problem in the clinical setting and discusses the results obtained in the prediction of endotracheal obstruction in COVID-19 patients in the ICU. We end this work with the obtained conclusions and identify the methodologies of our future work.

2. BigLSTM

Using the context outlined in the introduction to solve a prediction problem in real-world applications, specifically in the health field, we propose a deep modular binomial gate long short-term memory (BigLSTM) neural architecture, a new computing architecture that is tolerant of missing data and able to work with ISMTSs that present irregular sampling (Fig. 2). BigLSTM is a deep modular neural architecture with a base information processing system comprising a long short-term memory (LSTM) neural network. BigLSTM incorporates information redundancy injection to predict observations that contain a large amount of missing data as an effective mechanism for managing time control.

The modular structure of BigLSTM is composed of five interconnected modules, with a specific objective, and works following an explicit decomposition scheme of their functions [24]. Four of the BigLSTM modules are information processing modules, and the fifth module is dedicated to controlling tasks in the internal functioning of the network. The function and responsibility of each module, as well as its style of information processing and transmission, are explained in the following subsections.

A regular time series (TS) is composed of a succession of observations, which are made with a constant sampling period (T). These observations correspond to the values that the study variables take over time.

In a context where we are working with time series, we say that we are making predictions when we calculate the future values of the observations, using the knowledge we have of the historical record of the TS.

We define the concepts of observation and prediction in the context of BigLSTM. These will help in the description of the modular architecture of BigLSTM.

Definition 1. An observation (O) is the set of values that take the signals that are inputs to the system at an instant of time t_i . These signals can be basic signs of the problem environment $\{S_j\}$ or prediction signs $\{E_k\}$. Therefore, $O_i = \{S_0, S_1, S_2, \dots, S_j, \dots, S_{n-1}; E_0, E_1, E_2, \dots, E_k, \dots, E_{m-1}; t_i\}$, where the values that S_j takes correspond to numbers belonging to \mathbb{R} or unknown values (or missing values that are indicated by *), E_k are also values belonging to \mathbb{R} , and t_i is the time of observation.

Definition 2. We define a prediction (P) as the set of values that are outputs of the system when it is presented with an O_i , which is composed only of basic signals from the environment of the problem $\{S_j\}$. Therefore, $P_i = \{p_0, p_1, p_2, \dots, p_k, \dots, p_{m-1}; t_i\}$, where p_k are values belonging to \mathbb{R} , which correspond to the predictions of the values $E_0, E_1, E_2, \dots, E_k, \dots, E_{m-1}$.

Note that we eliminate the time dependence of $\{S_j\}$, $\{E_k\}$ and $\{p_k\}$ to simplify the notation, leaving only that temporary character in the subscript i associated with O_i , and P_i .

2.1. LSTM

The basic element of BigLSTM is long short-term memory (LSTM). LSTM is a recurrent neural network (RNN) architecture that can be considered a deep learning architecture; it uses gates as elements that modify the data flows between its components. It is used to treat data sequences [25].

One can address the problem of gradient propagation instability present in recurring networks [26] by using a mechanism that acts as a constant error carousel (CEC).

LSTM was initially proposed by Sepp Hochreiter and Jürgen Schmidhuber [27]. Later, other authors [28–30] added new characteristics, forming what is known today as vanilla LSTM. Different versions have emerged from the previous version. However, none of these variants have significantly improved the standard LSTM architecture [31]. The vanilla LSTM version is derived from the original LSTM block and consists of three recurring submodules that act as a gate (entry, forgetting and exit gates), an entry block, a constant error carousel unit (CEC unit), an output activation function, and so-called peephole connections [29,30].

Fig. 3a shows a diagram of a counting cell of this version of LSTM. Inside the cell, the three gates regulate the flow of the input information, the output information, and the state of the cell, the latter having a long-term status characteristic of the cell (grey line in Fig. 3a). All these gates present the same recurrent dynamics, as shown in Fig. 3b.

The CEC unit can be considered responsible for the state of the cell and the final output of the cell. Its dynamics are divided into three phases (see Fig. 3c). In the first two phases, the state of the cell is processed and stored using feedback connections, and the information flows from the entry and forgetting gates. In the third phase, the final output of the cell is calculated. LSTM has the ability to remove or add information to the cell state. To do this, the flow of information to be incorporated into the state of the cell is controlled by the entry gate, and information to be eliminated from the state of the cell is controlled by the forget gate.

The output of the LSTM cell is generated using the information from the new state of the cell, which is modulated by the output activation function, to finally be controlled by the output gate.

The output of the LSTM cell is repeatedly connected to the input of the LSTM cell, becoming an input to all the gate units, which also receive the input information from the network and all of them with their own weight groups. The state of the cell, calculated by the CEC unit, is also recurrently connected to all the gate units through sight glass connections, shown by the dashed lines in Fig. 3.

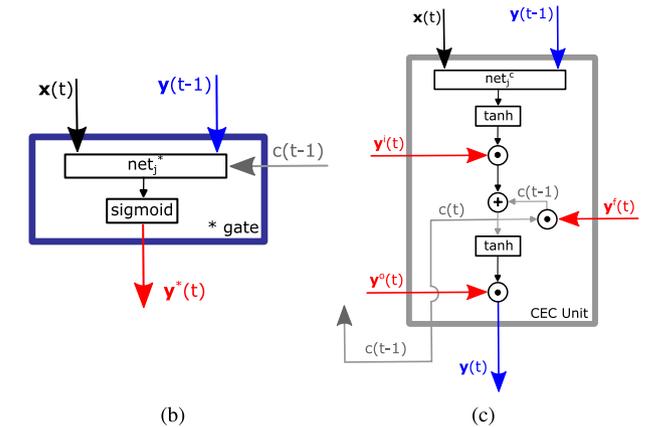
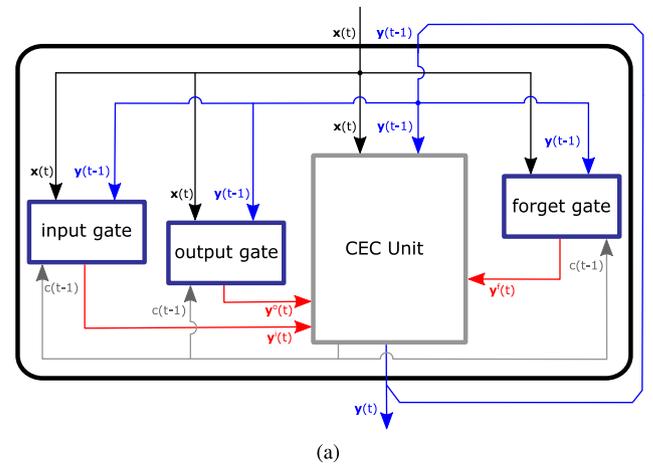


Fig. 3. (a) LSTM cell. (b) Gate scheme. (c) CEC unit scheme.

Let $x(t)$ and $y(t)$ be the input and output vectors at time t , respectively; the following weight vectors are associated with an LSTM cell j :

$$\begin{aligned}
 & \mathbf{w}_j^i, \mathbf{w}_j^f, \mathbf{w}_j^o, \mathbf{w}_j^c \\
 & \text{weights associated with the entrance } x(t) \text{ for all units in the cell} \\
 & \mathbf{u}_j^i, \mathbf{u}_j^f, \mathbf{u}_j^o, \mathbf{u}_j^c \\
 & \text{recurring weights associated with output } y(t-1) \text{ for all units in the cell} \\
 & \mathbf{b}_j^i, \mathbf{b}_j^f, \mathbf{b}_j^o, \mathbf{b}_j^c \\
 & \text{weights associated with bias for all units in the cell} \\
 & \mathbf{p}_j^i, \mathbf{p}_j^f, \mathbf{p}_j^o, \mathbf{p}_j^c \\
 & \text{associated weights of peephole connections}
 \end{aligned} \tag{1}$$

The activation of the gate units of LSTM cell j is calculated as follows:

$$y_j^i(t) = \sigma_i[net_j^i(t)] \tag{2}$$

For the inlet hatch, σ_i corresponds to a sigmoidal activation function, and $net_j^i(t)$ is calculated according to the following expression:

$$net_j^i(t) = \mathbf{w}_j^i x(t) + \mathbf{u}_j^i y(t-1) + \mathbf{b}_j^i + \mathbf{p}_j^i c_j(t-1) \tag{3}$$

$$y_j^f(t) = \sigma_f[net_j^f(t)] \tag{4}$$

For the floodgate of oblivion, σ_f corresponds to a sigmoidal activation function, and $net_j^f(t)$ is calculated according to the following expression:

$$net_j^f(t) = \mathbf{w}_j^f x(t) + \mathbf{u}_j^f y(t-1) + \mathbf{b}_j^f + \mathbf{p}_j^f c_j(t-1) \tag{5}$$

$$y_j^o(t) = \sigma_o[net_j^o(t)] \tag{6}$$

For the exit gate, σ_o corresponds to a sigmoidal activation function, and $net_j^o(t)$ is calculated according to the following expression:

$$net_j^o(t) = \mathbf{w}_j^o \mathbf{x}(t) + \mathbf{u}_j^o \mathbf{y}(t-1) + \mathbf{b}_j^o + \mathbf{p}_j^o \mathbf{c}_j(t-1) \quad (7)$$

The state $\mathbf{c}_j(t)$ of the LSTM cell j at time t is calculated as follows:

$$\mathbf{c}_j(t) = \sigma_c[net_j^c(t)] \mathbf{y}_j^c(t) + \mathbf{c}_j(t-1) \mathbf{y}_j^f(t) \quad (8)$$

where σ_c corresponds to a hyperbolic tangent activation function and $net_j^c(t)$ is calculated according to the following expression:

$$net_j^c(t) = \mathbf{w}_j^c \mathbf{x}(t) + \mathbf{u}_j^c \mathbf{y}(t-1) + \mathbf{b}_j^c \quad (9)$$

Finally, the output $\mathbf{y}_j(t)$ of LSTM cell j at time t is calculated as follows:

$$\mathbf{y}_j(t) = \sigma_y[\mathbf{c}_j(t)] \mathbf{y}_j^o(t) \quad (10)$$

where σ_y corresponds to a hyperbolic tangent activation function [31].

LSTM architectures are typically trained in a supervised manner using a descending gradient-based optimization algorithm. Backpropagation through time (BPTT) is often used [27,30] to calculate the necessary gradients during the optimization process. The use of full BPTT has the additional advantage that LSTM gradients can be verified using finite differences, which increases the reliability of practical implementations [31].

2.2. Information distribution module

An information distribution module (IDM) is formed by a binomial configuration of gates and is responsible for receiving the observations O_i at each instant of time t_i .

The dynamics of the IDM gates send the O_i , organized in indexed observation tracks, to the next computational phase, the central computing module (CCM).

Definition 3. We define an indexed observation Track (IOT) as a set of observations O_i grouped by a gate when all of them share the same pattern of no missing data, without losing their temporal reference:

$$IOT_w = \{(O_0, t_0); (O_1, t_1); \dots; (O_i, t_i); \dots; (O_h, t_h)\} \quad (11)$$

Different IOTs are generated following the BigLSTM information redundancy scheme, which is responsible for the O_i values that make up each IOT, as shown in Fig. 4.

Definition 4. The information redundancy scheme (IRS) determines the different IOTs and thus the way in which the information reaches the different LSTM computing cells of the BigLSTM architecture. The IRS is also responsible for the adequate redundancy of information in the system so that the prediction system is tolerant of missing data. Therefore, the IRS allows for determining the redundancy index with which the BigLSTM will work, which is a function of the redundancy factor of the observations that are established.

Definition 5. Associated with each observation, we define the observation's redundancy factor (ORF) as a parameter that measures the information redundancy index produced by the applied IRS. This factor will depend on the level of missing data that the observation presents and the IRS with which the system is working. Thus, for an IRS with maximum redundancy (see Fig. 4a), an O_i composed of n possible values of basic signals and where r corresponds to missing data, the ORF can be expressed as follows:

$$ORF(O_i) = 2^{n-r} (2^r - 1) \quad (12)$$

The ORF of an observation O_i corresponds to the number of computational cells that will process said O_i .

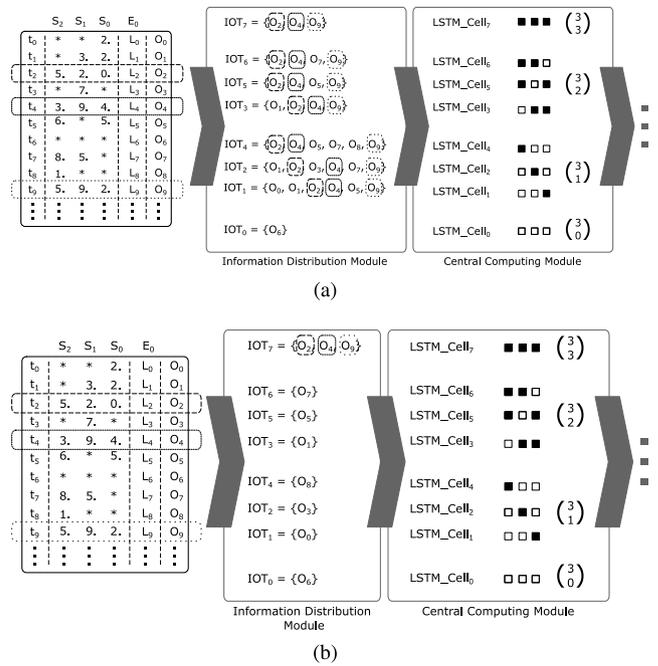


Fig. 4. IOT configuration process with an IRS of (a) maximum and (b) minimum information redundancy.

The system redundancy factor (SRF), when working with h observations, is defined according to the following expression:

$$SRF(\{O_i\}) = \sum_{i=0}^h 2^{n-r_i} (2^{r_i} - 1) \quad (13)$$

where r_i corresponds to the number of missing data points that O_i has.

We define the IRS with minimal information redundancy when the IOTs are configured as follows: if two observations belong to the same IOT, both share the same pattern of missing data. Fig. 4a shows the IOT configuration process for a case in which there are 3 basic signals S_n ($n = 1, 2$ and 3), 1 prediction signal E_m ($m = 1$), and an IRS with maximum information redundancy. Fig. 4b shows the details of an IOT configuration when BigLSTM works with an IRS with minimal redundancy. In this case, we have $ORF(O_i) = 1, \forall i$ and thus an $SRF(\{O_i\}) = h$ (the latter corresponding to the number of observations).

2.3. Central computing module

The CCM is formed by an array of LSTM cell layers. The inputs to the LSTM cells follow a temporal window process in the observations belonging to their IOTs. Each LSTM cell processes its corresponding IOT, and the set of predictions that the cell learns is in direct correspondence with the number of observation windows that have been created.

The objective of each LSTM cell is the association of basic signals $\{S_0, S_1, S_2, \dots, S_j, \dots, S_n\}$, corresponding to the IOT observations, with prediction signals $\{E_0, E_1, E_2, \dots, E_k, \dots, E_m\}$ at different instants of time t_j . This parameter is the number of observations in the IOT, together with the size of the window, which determines the number of predictions per LSTM cell (see Fig. 5). We have already indicated that the temporal variable associated with the observations intervenes in the association process carried out by each cell. This time information is expanded with a set of values calculated expressly to reciprocally locate all the observations on said time axis (a_z values at the input of the LSTM cell; see Fig. 5). Therefore, the system can work with observations taken during irregular sampling periods.

Therefore, the input of an LSTM cell is made up of a set of windows of values associated with the basic signals $\{S_j\}$, the values associated

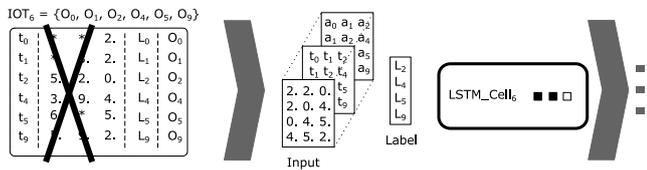


Fig. 5. Process of creating observation windows, from a determined IOT formed by 6 observations, to be the input data to an LSTM cell of the CCM and using 3-time units as window sizes.

with the temporality of the previous signals $\{t_i\}$ and $\{a_z\}$, and the values associated with the signals for prediction $\{E_k\}$ (L_k values of the label vector; see Fig. 5).

The number of LSTM cells that are necessary in this module depends on the IRS that is used and the index of missing data that is included in the observations. Likewise, and as we will see in a later section, BigLSTM provides control intelligence for turning the cells on and off, depending on what they contribute to the predictions.

The learning process of this module is carried out using the LSTM algorithm [17,20,26,32], as described in Section 2.1.

The output of this module is an essential part of the system output, that is, of the prediction sought. We must bear in mind that, depending on the IRS with which we are working, at the same time t_i , there may be different output values produced by different LSTM cells associated with the same prediction. These output values, which we call partial predictions (PP) and follow a strategy of combined predictions, will provide the final prediction, that is, the prediction given by BigLSTM. This is done by a predictive module (PM). The possible parameters of this module, which can be adjusted and/or defined in a specific problem, are the number of internal processing units (neurons) of LSTM cells and the size of the time window. The latter is related to the size of the temporary memory desired in BigLSTM.

2.4. Predictive module

The PM takes the output of the CCM and calculates the final prediction of the system $\{\hat{p}_i\}$, which, together with the prediction of the temporal moments $\{t_i\}$ performed by the time axis processing module (TAPM; Section 2.6.), forms the output of the system. This module also sends information to the LSTM cell on/off control module (Section 2.5), depending on the contribution that each of the LSTM cells makes in the predictions.

The operating dynamics of this module are conditioned by the predictive convergence strategy, which is selected and adjusted by the IRS.

Definition 6. We define the predictive convergence strategy (PCS) as a function Φ through which all the partial predictions that are generated in the LSTM cells are processed to convert them into the final prediction associated with an observation.

As we have already mentioned, the LSTM cells that are involved in the prediction of an observation depend on the IRS with which the IOTs are configured. Thus, with an IRS of minimal redundancy, the operational dynamics of the PM are reduced to converting the only predictions that the CCM sends into definitive predictions of the system. However, when working with an IRS that injects maximum redundancy, the predictions sent by the CCM must be processed so that they contribute fairly to the definitive predictions of the system.

Therefore, the final predictions calculated by the system depend on both the defined PCS (dynamics performed by this module to recombine all the partial predictions that have been calculated in the CCM; see Fig. 6) and the IRS. Thus, working with an IRS that injects maximum redundancy, the predictions sent by the CCM are

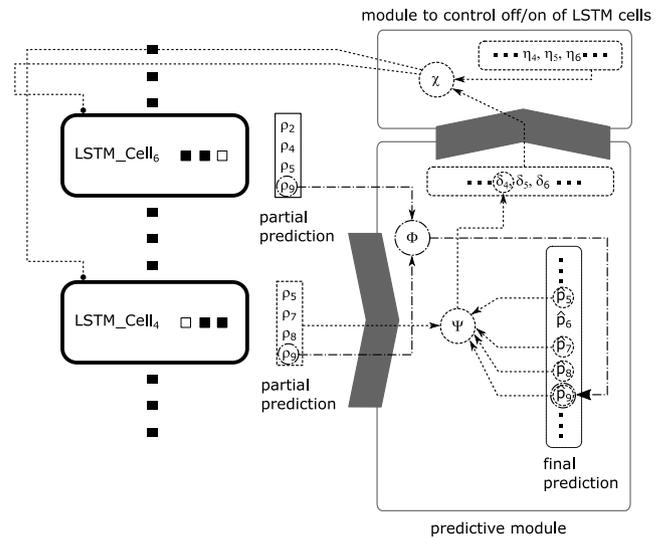


Fig. 6. Predictive convergence strategy (PCS) carried out by the PM module, and the diagram and process of switching on/off the different LSTM cells carried out by the CM on/off module.

processed to contribute to the final predictions of the system based on the following expression of Φ :

$$\Phi(\dots, \{\rho_k\}_h, \dots) = \{\hat{p}_i\} \quad (14)$$

where $\{\rho_k\}_h$ corresponds to the set of all partial predictions ρ_k performed by cell LSTM h and $\{\hat{p}_i\}$ the set of all final predictions, calculated according to the following expression:

$$\hat{p}_i = \frac{\sum_{k=0}^{\text{ORF}(O_i)} \rho_k}{\text{ORF}(O_i)} \quad (15)$$

where O_i corresponds to observation i for which we are calculating its prediction, and $\text{ORF}(O_i)$ corresponds to the redundancy factor of said observation.

2.5. LSTM cell on/off control module

The on/off control module (on/off CM) of LSTM cells verifies the contribution to the final prediction of each of the LSTM cells. It works in coordination with the PM and calculates the difference between the partial predictions made by each LSTM cell and the final predictions.

$$\Psi(\dots, (\{\rho_k\}_h, \{\hat{p}_k\}_h) \dots) = (\dots, \delta_h \dots) \quad (16)$$

where $\{\rho_k\}_h$ corresponds to the set of all partial predictions ρ_k performed by cell LSTM h , $\{\hat{p}_k\}_h$ corresponds to the subset of the final predictions to which the LSTM cell h contributes, and Ψ is the function that receives the set of previous tuples from all the LSTM cells of the CCM and calculates the difference vector $(\dots, \delta_h \dots)$ according to the following expression:

$$\delta_h = \sqrt{\sum_k (\rho_k - \hat{p}_k)^2} \quad (17)$$

Finally, the on/off CM makes the decision to turn off, or keep on, each of the LSTM cells of the CCM based on the value taken by the $(\dots, \delta_h \dots)$ in relation to the vector of threshold values $(\dots, \eta_h \dots)$, following Eq. (18).

$$\chi(\dots, \delta_h, \dots; \dots, \eta_h, \dots) = \begin{cases} \text{cell}_h \text{ off} & \delta_h < \eta_h \\ \text{cell}_h \text{ on} & \delta_h \geq \eta_h \end{cases} \quad (18)$$

Fig. 6 shows the operating diagram of this module. The expressions that define its dynamics analyse the importance/relevance of the different cells in the calculation of the final predictions made by the system. This module analyses these situations and has the ability to cancel the count, partially or totally, of the LSTM cells.

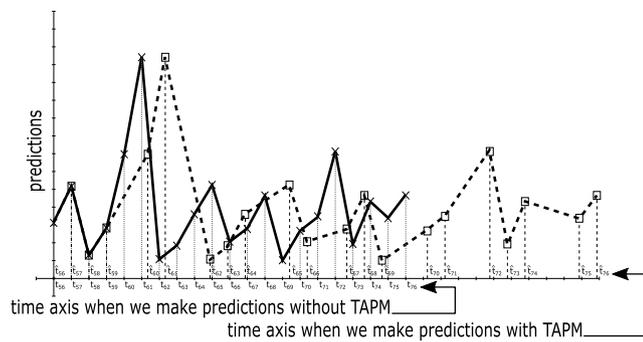


Fig. 7. Predictions of two signals when not working with TAPM (continuous line) and when working with TAPM (dashed line).

2.6. Time axis processing module

The time axis processing module (TAPM) is a critically important feature in BigLSTM that provides this neural architecture with the ability to work with temporal signals with an irregular sampling period.

The TAPM works in parallel with the rest of the modules and receives the sequence of times in which the observations have been taken. The TAPM predicts the time intervals that are associated with the predictions. Therefore, the output values of this module index the predictions on the time axis, forming part of the final output of BigLSTM, as seen in Eq. (19).

$$\text{OUTPUT}_{\text{BigLSTM}} = \{(\hat{p}_t, \hat{t}_t)\} \tag{19}$$

Fig. 7 shows how this module conditions the shape of the prediction signal. The solid line in the graph represents a signal constructed based on the prediction values, where the sampling interval is assumed to be regular and constant (although the observations used to make the prediction show irregular sampling), and the dotted line in the graph represents the same values of the previous predictions but relocated on the time axis, as the TAPM module predicts the times associated with the observations with which the system has worked. BigLSTM was developed in Python 3.10.13 (www.python.org) and uses TensorFlow 2.10.0 libraries (www.tensorflow.org) and Keras 2.10.0 (www.keras.io). The BigLSTM source code that we used in all the experiments in this article is available for download on GitHub ([BigLSTM](https://github.com)).

3. BigLSTM for endotracheal obstruction prediction in COVID-19 patients in the ICU: Development, results and discussions

The modular BigLSTM neural architecture has been tested and validated with different datasets. In this section, we show the high capacity of the BigLSTM model for solving a real clinical problem, namely, the prediction of endotracheal obstruction in COVID-19 patients in the ICU. Many of these patients are on assisted respiration using invasive mechanical ventilation (IMV) devices, and necrotizing trachea bronchitis can cause airway obstruction, with necrotic and haemorrhagic remnants obstructing the trachea and bronchi. The mortality rate in this situation can approach 30% [33]. The information environment is plagued by missing data and highly irregular temporal sampling. The functional and structural organization of BigLSTM is designed for these characteristics, providing an efficient computational solution for ISMTs in ICU environments.

In a study in Germany during 2020 and 2021 [32], of 561,379 hospitalized COVID-19 patients, 24.54% were admitted to the ICU. Hospital mortality was 16.69%, and 33.36% in the ICU group.

Predicting endotracheal obstruction in advance is crucial for patient prognoses and preventing deaths and viral transmission. Our proposal uses IMV device signals and BigLSTM to predict when endotracheal obstruction may occur in COVID-19 patients.

Table 1
Demographic data and information regarding the length of stay in the ICU.

Age	Weight (kg)	Height (cm)	BMI	ICU (days)
62.04 ± 12.72	87.38 ± 20.36	170.9 ± 11.45	29.2 ± 4.33	21.02 ± 16.44

3.1. Related work

Missing data have a huge impact on the quality of data sets, and thus on classification processes, and they can lead to unstable predictions and other unpredictable effects. On the other hand, missing data can also be generated when processing sampling irregularity in time series to convert them into regularly sampled series. There are different approaches to address the existence of missing data in time series, such as techniques that perform data imputation by substituting the missing values with reference statistic values [28]. Such techniques, called conventional techniques, are the most commonly used. And among them, we can cite those based on: ‘mean imputation’, ‘regression imputation’, ‘stochastic regression imputation’, ‘last observation carried forward (LOCF) imputation’, ‘baseline observation carried forward (BOCF) imputation’, etc.

Another approach, less conventional and more related to models based on neural computing techniques in general and Deep Learning in particular, is based on using the raw data, being the neural models or architectures those that have the ability to process the missing data directly (the BigLSTM is an example of them). In these methods, imputation is not necessary to be used. Instead, this irregularity is used as additional, and useful, information that requires to be processed.

There are works with different approaches to address the prediction of the evolution of clinical pathologies in patients. We present some of these works related to our proposal and developments. Li Yan et al. 2020 [34] used a database of blood samples from COVID-19 patients in the Wuhan region (China) to identify predictive biomarkers for mortality from the disease. Among the most important, the authors identify the following: lactic dehydrogenase (LDH), lymphocyte and high-sensitivity C-reactive protein (hs-CRP). A prediction algorithm based on the use of XGBoost is used to predict mortality more than 10 days in advance, achieving an accuracy of more than 90%. This is the result of classifying patients as patients who die at N days and patients who do not die at N days. The best results were obtained in the correct prediction when N is 11.95 days.

Anne Chen et al. 2021 [35] used univariate logistic regression models to study and identify the predictive power of the following 7 variables: lactate dehydrogenase, lymphocytes, procalcitonin, D-dimer, C-reactive protein, respiratory rate, and white blood cells, and to construct a predictive model of COVID-19 associated mortality. The data used in this work corresponds to data from the medical records of more than 1600 patients from Stony Brook University Hospital (New York, USA), as well as from Tongji Hospital (Wuhan, China). The performance achieved by this predictive model remains below 0.8 (AUC value) until 4 days before the death of the patients, being 0.7 (AUC value) on day 10. The scheme used to obtain the above results is again based on classifying patients on the basis of those who die at N days versus those who do not die at N days.

Kai Zhou et al. 2021 [36] identified a panel of eleven routine clinical factors for the classification of COVID-19 severity prediction using a dataset of 144 patients diagnosed in Taizhou hospital (China), and where the observations used consist of 124 types of measurements/variables over 52 days. The Machine Learning (ML) model to predict disease progression in a patient, based on ‘severe state’ and ‘non-severe state’, uses Genetic Algorithms in the feature selection phase, identifying the following eleven as the most relevant: oxygenation index, basophil counts (BASO#), aspartate aminotransferase (AST), gender, magnesium (Mg), gamma-glutamyl transpeptidase (GGT), platelet counts, activated partial thromboplastin time (APTT), oxygen saturation (SaO2), body temperature and days of symptom onset. In this

case, the authors do describe the process performed with the data with regard to the treatment of missing data, applying a basic data imputation mechanism based on the mean value of the characteristic for all patients. On the other hand, in the prediction phase, Support Vector Machines (SVM) were used, achieving a prediction performance of 0.8 (AUC value).

Maryam Seyedtabib et al. 2024 [37] conducted a retrospective study in a large hospital in Abadan city (Iran) to identify predictive factors associated with COVID-19 mortality in a dataset composed of demographic, clinical, comorbid, treatment, baseline vital signs, symptomatic and laboratory test attributes. The authors used different ML algorithms to analyse the predictive power of these variables, finding the highest scores for age, comorbidities (hypertension, diabetes), specific treatments (antibiotics, remdesivir, favipiravir, vitamin zinc) and clinical indicators (heart rate, respiratory rate, temperature). They also identify a great importance of certain specific symptoms (productive cough, dyspnea, delirium) in predicting outcomes, as well as laboratory values (D-dimer, ESR).

Krishnaraj Chadaga et al. 2024 [38] used ML and deep learning (DL) algorithms to classify the severity of COVID-19 patients and perform early prediction of COVID-19 severity using clinical and laboratory markers. The dataset in this case is based on 899 patients (599 non-severe disease stage patients, and 300 severe patients) monitored during the months of September to December 2021, in two hospitals in Manipal (India). Multiple feature selection algorithms and three different types of deep architectures were used to perform the classification. The most important criteria identified were c-reactive protein, basophils, lymphocytes, albumin, D-dimer and neutrophils. The three architectures used in the classification were a deep Feed Forward network, a one-dimensional Convolutional network and a LSTM type network, which achieved a value of 0.97, 0.94 and 0.88 in AUC, respectively. The scheme used to obtain the above results is again based on classifying patients based on their severity status. Another prediction problem similar to the one solved in this work, also applicable to the ICU setting and allowing to streamline resource allocation and provide personalized interventions for patients, is the one worked by Zhijiang Yu et al. 2024 [39] to predict mortality of patients with Sepsis-3. Here, various ML tools are used to predict the 30-day mortality rate of ICU patients with Sepsis-3 using the MIMIC-III database [40]. In the data preparation and preprocessing phase, these authors started from a dataset associated with 9118 patients and used a decision tree model and entropy analysis to select the final features used, which were also contrasted with clinical practitioners. Prediction was performed with the Light Gradient Boosting Machine (LightGBM) model, which was able to obtain a 0.983 AUC value, again using a prediction scheme that classifies patients by virtue of their 30-day mortality.

In contrast to the previous contributions, which are representative of the current state of the art in predicting the occurrence of situations in COVID-19 patients in the ICU, our BigLSTM model predicts the time in which a critical situation, such as an endotracheal obstruction, may occur in a patient. To do this we use the following set of signals: Peak Pressure, Average Pressure, Plateau Pressure, Compliance and Resistance, which are associated with the patient's ventilatory mechanics. The main output of our system is a numerical prediction of the time to elapse until the patient becomes obstructed. This prediction is made based on the current functional state of the patient's respiratory system, which is collected in the longitudinal evolution of the previous signals, which present very high levels of missing data and irregularity in their sampling/capture. Instead of applying data imputation, BigLSTM uses an information redundancy injection mechanism. Finally, our results are not measured in AUC values but in mean squared error (MSE) levels.

3.2. COVID-19 dataset

Our study was carried out with 96 patients with COVID-19 who were hospitalized in the ICU of the Insular University Hospital Complex for Mothers and Children (CHUIMI) between March 8, 2020, and February 4, 2021. This period covers the first and second waves of COVID-19 in Spain.

Table 1 shows patient demographic data, age, weight, height, and body mass index (BMI), as well as the length of time they spent in the ICU. For such data, the mean and standard deviation ($\bar{x} \pm \sigma$) of all the patients included in the study were calculated.

The dataset consists of records of the mechanical ventilation signals associated with the patients. The mechanical ventilation applies positive pressure breaths and depends on airway compliance and resistance, affecting the pressure needed for a given Tidal Volume (TV) - the air volume entering the lung during inhalation [41]. These records correspond to the parameters captured from the IMV devices or calculated on the previous devices. The set of records for each parameter corresponds to a time series of each captured signal (Ppeak, Pave, Pplat, R and C), where:

- Ppeak (Peak Pressure): Maximum pressure at end inspiration, including elastic and resistive components. Modified by factors like endotracheal tube diameter [42].
- Pave (Average Pressure): Mean value of Ppeak over time.
- Pplat (Plateau Pressure): Measured during an inspiratory pause, equal to alveolar pressure when airflow is zero.
- R (Airway Resistance): Difference between Ppeak and Pplat divided by airflow. Normal values do not exceed 15–20 cmH₂O/L/s under controlled mechanical ventilation [43].
- C (Compliance): Change in volume per unit change in pressure, indicating the system's elastic resistance.

These signals are with irregular sampling and high missing data rates. They are accessed by consulting the electronic medical records in the PICIS system [44].

The format of these records is a data matrix in which each row is an observation for each record in direct correspondence with an instant in time and a column for each value measured for each recorded ventilatory mechanics signal. An exhaustive analysis of these signals together with the CHUIMI clinical team allowed a marking procedure to be carried out on the signals to identify the normal areas, the pre-obstruction areas and the exact moment in which the obstruction occurred in each patient. These results allowed us to identify these areas in only 37 of the 96 patients for whom data were available. The level of missing data is high in the two waves and is greater in the second wave, as shown in Fig. 8. Of the 96 patients, 20 had almost all data missing and were excluded from this study (white levels in Fig. 8). Another set of patients with high levels of missing data had data only for Ppeak signals (Fig. 8). After cleaning, 76 patients remained: 37 with obstructions and 39 without (Fig. 8).

The length of time patients spent in the ICU varied significantly, with a standard deviation (σ) of more than 16 days (Table 1). The number of observations per patient showed no correlation with ICU stay length, and the sampling period of the observations is highly irregular (Fig. 9).

In Fig. 9, several descriptive statistics for patients with obstructions quantify the irregular sampling period of the signals. All patients showed notable differences in all quartiles and the interquartile range, indicating considerable irregularity. The relative position of the median suggests non-normal distribution, and the mean and standard deviation show considerable variations. Similar statistics for non-obstructed patients are in Figure A1.1 of Annex 1 (Supplementary Material). Ppeak is a key IMV signal and had the most data of all observations for the 76 patients. Fig. 10 shows an irregular dispersion for Ppeak in patients with obstructions using descriptive statistics. There was also a non-normal distribution and a high number of outliers.

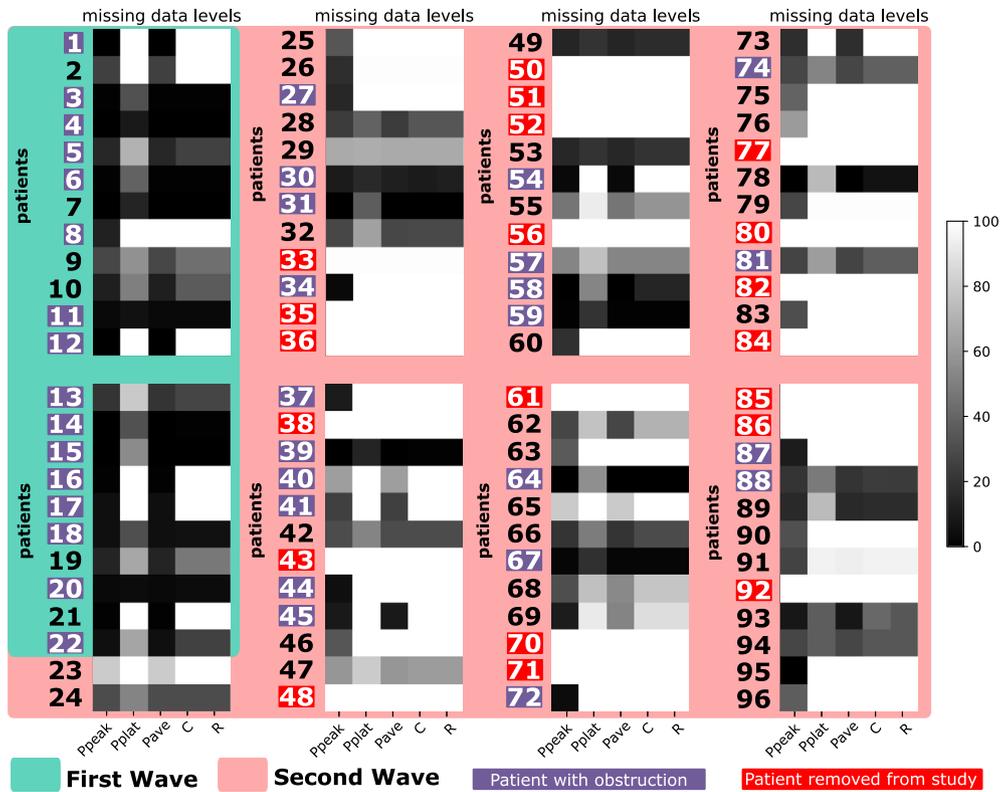


Fig. 8. Level of missing data in the signals associated with ventilatory mechanics (the white colour corresponds to 100% absence of data).

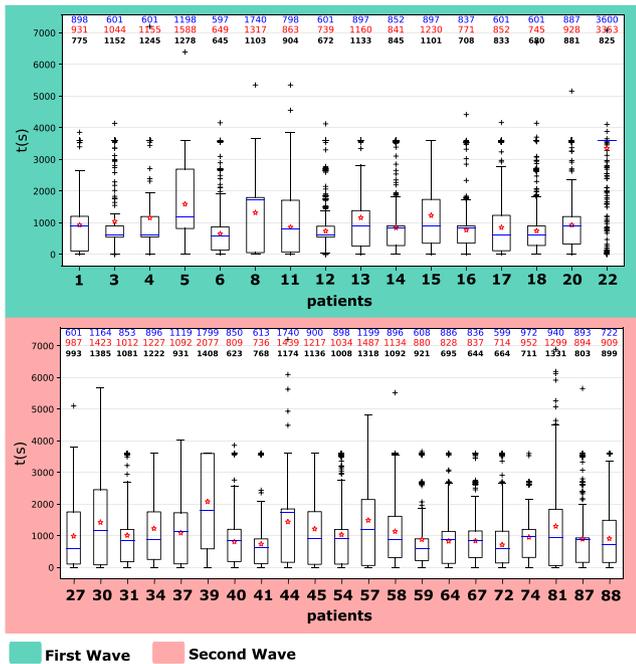


Fig. 9. Descriptive statistics concerning the irregular sampling period of the signals. The 1st, 2nd and 3rd quartiles, the interquartile range, the median (blue colour), the mean (red colour) and the standard deviation (black colour).

Other signals (Pave, Pplat, R, and C) showed similar statistical variations, though some had missing data in many patients. Detailed statistics for these signals are in Figures A1.3, A1.4, A1.5, and A1.6 in Annex 1 (Supplementary Material), covering both obstructed and non-obstructed patients. We characterize our dataset with a representation

of the reference signal, corresponding with the time remaining for the patient to produce an obstruction. This must be defined in a way that also allows us to analyse the performance of BigLSTM when there are patients who have not presented an obstruction. That representation is based on the following exponential function:

$$f(x) = e^{-\frac{1}{x}} \tag{20}$$

Patients who present with obstruction have a reference signal defined according to Eq. (20) (Fig. 11), and patients who do not present with obstruction have a constant value of 1 for all observations. The purpose of the above is that for patients whose profiles and evolution of IMV signals do not converge during obstruction, our system predicts an output very close to a value of 1. This means that, by virtue of the evolution presented by the patient, the obstruction is very distant in time or may never occur.

3.2.1. Alignment of observations

The need to align observations prior to system training is also important. This alignment involves temporally ordering the observations of all patients and calculating all temporal references based on a single temporal reference point (TRP). Several ways exist to establish the TRP:

- i. The TRP is calculated based on the time at which patients entered the ICU.
- ii. The TRP is calculated based on the time associated with the occurrence of the obstruction.
- iii. The TRP is calculated based on the average time to obstruction (ATO).

The ATO is a statistical value calculated using all patient evolutions with obstruction. Fig. 12 shows the evolution of COVID-19 patients from ICU admission until the first obstruction, highlighting differences in the duration of each patient’s evolution. These evolutions can be characterized by five concepts: moment of entry into the ICU, past evolution of the patient (PEP) up to a certain moment, state of the

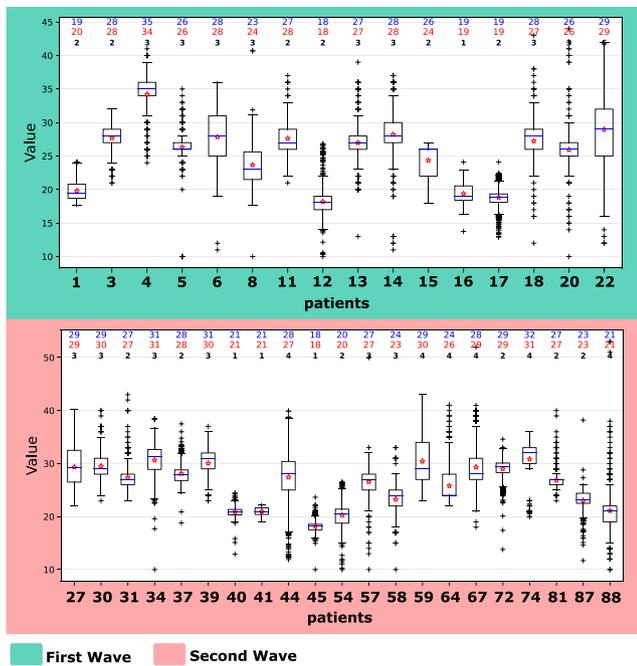


Fig. 10. Characterization of the Ppeak data by median (blue line), mean (red star), interquartile range (height box) and data dispersion for all patients with obstruction (see Figure A1.2 in Annex 1, Supplementary Material, for patients without obstruction).

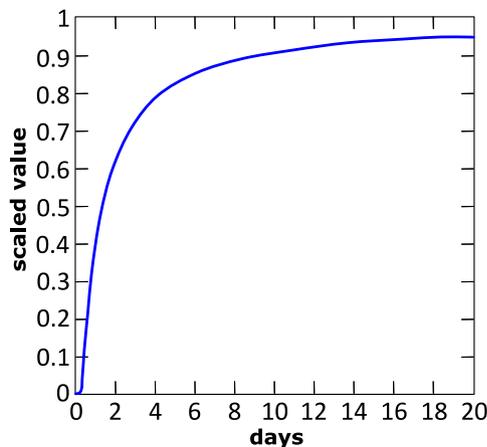


Fig. 11. Function used in the normalization and representation of the label signal of all patients.

patient at that specific moment (SP), future evolution of the patient (FEP), and occurrence of obstruction. Therefore, once trained, BigLSTM must be able to implement our system (S), outputting the duration of the EFP when the inputs are the EPP and SP at a certain time, i.e., $S(EPP, SP) = \text{the duration of the EFP}$, this situation is shown in Fig. 13. Calculating the temporal references according to TRP from ii is only feasible when the evolution of the patients is fully known. That situation is not ever in the real clinical environment, working with real ICU patients. Therefore, BigLSTM will be validated and tested (as detailed in Section 3.3.) using a TRP calculated as established in points i and iii.

3.3. Results and discussions

The performance, effectiveness and sensitivity of the modular recurrent neural architecture, BigLSTM, in predicting the time at which an ICU patient will reach endotracheal obstruction, its tendency to

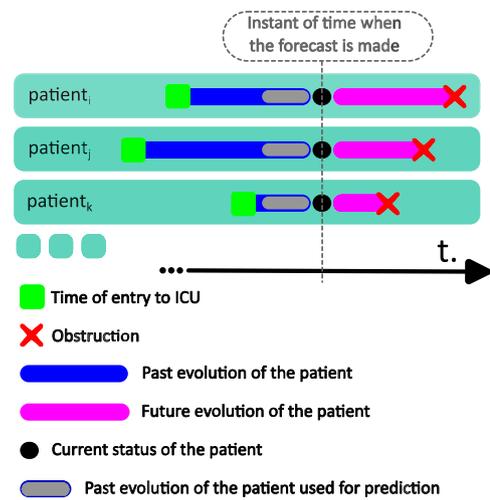


Fig. 12. Diagram of ICU stays of COVID-19 patients.

advance or delay the prediction, and its sensitivity when working with data associated with patients who have presented with obstruction and with patients who have not presented with obstruction will be evaluated.

Several simulations are carried out for different sizes (15, 30 and 45 process units) of each LSTM cell that make up the CCM module of the BigLSTM architecture, as well as for different sizes of working windows of said LSTM cells (15, 30 and 45 values of the signal). As a result of the above, it was observed that the best validation MSE values were obtained for the following configurations: 15–15, 30–30 and 45–45 (see 3.3.1 Sensitivity Analysis of the BigLSTM and Annex 4 in Supplementary Material). In all the simulations, an IRS is used that injects the maximum redundancy, and the on/off CM keeps all the LSTM cells on. These parameters are used to catalogue the different configurations of the BigLSTM model, and the results for configurations 15–15, 30–30 and 45–45 correspond with the size small, medium and large, respectively, of the LSTM cell, and thus of BigLSTM.

All configurations of the BigLSTM model have been set with the following configuration of the LSTM cells: internal units in correspondence with the configurations discussed above, “tanh” and “sigmoid” as activation and recurrent activation functions respectively, “glorot_uniform” and “orthogonal” for kernels and recurrent initializers and no dropout level set. An “adam” optimizer has been used in the learning process with the following configuration: a learning rate value of 0.001, and values of 0.8 and 0.9 for the exponential decay rates for the 1st and 2nd moments estimates (beta_1 and beta_2 parameters). The size of dataset can determine the optimal configuration in which our system reaches, in medium and limited terms, effective and reliable performance. We are working with a very small dataset, in order to mitigate the possible effect of this dataset characteristic we use a cross-validation procedure, specifically leave-one-out cross-validation (LOOCV) [45], where the work item is the patient (see Figure A2.1 of Annex 2, Supplementary Material). Working in scenarios where the TRP has been established based on the time of admission to the ICU and only in patients with obstructions, the BigLSTM has reached high quality results as we can see in the Fig. 14. The best validation MSE value is 0.058 in epoch 2 for a 15–15 configuration, from which the system enters an overfitting regime. This best value is higher than that obtained in previous works [46].

The results given by the BigLSTM working with a TRP based on ATO were also studied and they were very good results, Fig. 15. In that analysis we worked only in patients with obstruction. The values achieved in this case (see Fig. 15) are lower than those achieved by the system that works with a TRP based on the entry of patients into

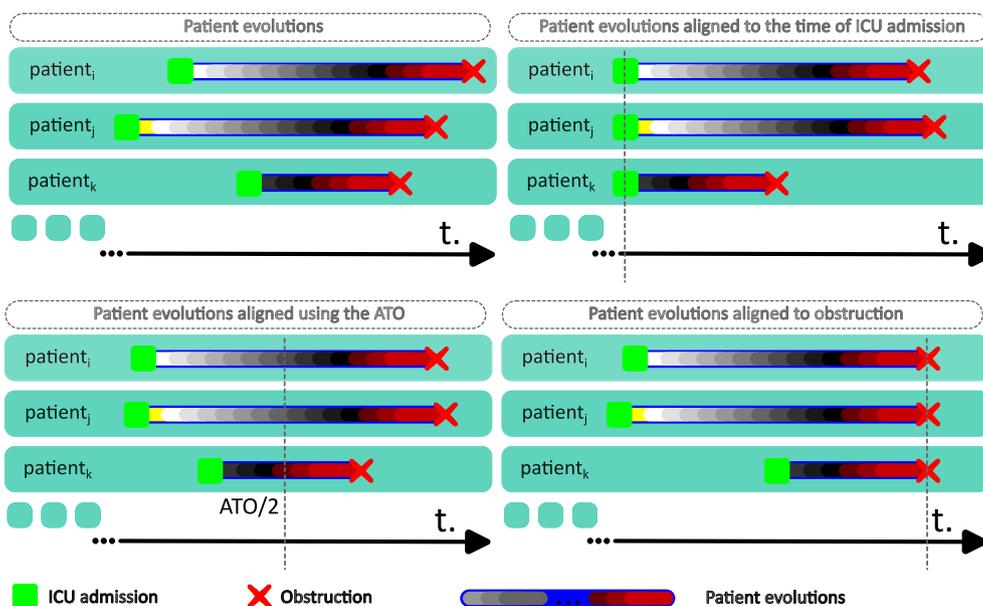


Fig. 13. Simplified diagram of the evolutions of the patients in ICU aligned according to the moment of time of entry into the ICU and occurrence of the first obstruction. ATO = Average time to obstruction.

the ICU, although somewhat higher than what was obtained in our previous work, Fernández-López, P. et al. 2021 [46] for a 15–15 A better value of 0.028 is achieved for the validation MSE in epoch 9 and for a 45–45 configuration. The system does not enter an overfitting regime. The BigLSTM is also capable to manage patients where the obstruction process is very distant in time or does not occur. For that BigLSTM must work with the data of all the patients (with and without obstruction) in training and validation processes. The evolution of the training and validation MSEs through 9 epochs, as shown in Fig. 16, allows us to see how all the configurations go into the overfitting regime from epoch 1, and the best value of the validation MSE is 0.09 for a single configuration at 30–30. A more detailed analysis of the predictions allows us to conclude that patients without obstruction causes a delayed prediction trend in BigLSTM. The second analysis carried out with patients without obstruction, the BigLSTM was trained just in patients with obstruction, a TRP calculated according to the ATO. The validation process was carried out for patients without obstruction, Fig. 17. The mean MSE oscillates at approximately 0.20. In both analyses carried out with patients without obstruction, the main difference between patients who presented with obstruction and patients who did not was in the mean duration of the progression of the patients in the ICU. The mean duration of evolution in patients with obstruction was 11.015 days, while in patients without obstruction, it was 47.2 days.

3.3.1. Tendency to advance or delay prediction, and prediction interval

Another aspect that we consider necessary to evaluate and discuss in relation to the functioning of our system is related to the tendency to advance or delay the prediction that it presents. In addition, it is known which of these predominates throughout the predictions made by the system since both do not have the same predictive importance. A system with a tendency towards delay would present predictions of occurrence subsequent to the real event, which is not admissible in clinical prediction systems. We believe it is necessary to know how these trends evolve since the output of our system in confidence factor mode must be able to be calculated from these values/variables.

To do this, we focus on the evolution of these trends on a 10-day scale, which corresponds to the approximate average time (according to the calculated ATO value) that elapses from when patients are admitted to the ICU until the obstruction is presented.

We carry out this analysis by dividing the set of predictions into two sets: distant predictions (>3 days) and close predictions (between 3 days and 0 days). Corresponding to these two types of predictions, Figs. 18 and 19 show the evolution of the trend towards the lag (in red) and the trend towards the lead (in blue) of the system for a 45–45 configuration and with a calculated PRT based on an ATO of 11.02 days. The calculation of these trends is based on the statistical values associated with the following error variables:

- $errorAnticipation = y_{Patient} - predictedY_{Patient}$ (calculated when $y_{Patient} > predictedY_{Patient}$),
- $errorLag = |y_{Patient} - predictedY_{Patient}|$, (calculated when $y_{Patient} \leq predictedY_{Patient}$).

where $y_{Patient}$ corresponds to the reference signal (real times until the moment of obstruction) and $predictedY_{Patient}$ corresponds to the prediction of the system. Fig. 18 shows these trends, in anticipation and lag, for the distance predictions. When our system was able to make a prediction of 7 days for the obstruction, this prediction (in terms of average) tended to increase to 3.87 days, a delay of 2.15 days. The above prediction intervals between Days 4.85 (7 – 2.15) and 10.87 (7 + 3.87) were established from the moment the prediction was made.

Fig. 19 shows the calculation of these trends when we work with close predictions, in such a way that when BigLSTM predicts that the time of obstruction is between 48 and 49 h previously, ‘2 days, 00:00:00 to 2 days, 00:59:59’, the trend towards advancement of BigLSTM is circumscribed to the values of 1.88 and 1.13 days, meaning that some of the actual obstructions of the patients occurred at most 1.88 days after the prediction. On the other hand, the tendency to delay, which we consider to be more critical, is between 0.53 and 0.51 days, meaning that some of the actual obstructions of the patients occurred 12 h earlier than those predicted by BigLSTM. Therefore, when the BigLSTM model predicts that the obstruction of patients will occur within 48 h, the actual obstruction will occur no earlier than 32 h and no later than 3.88 days.

Figures A3.1 and A3.2 of Annex 3 (Supplementary Material) show the leading and lagging trends that the BigLSTM model commits for the two configurations (15–15, 30–30) in addition to the one presented in this section, and all of them show that, on average, the network never commits delays in the predictions that may be higher than the prediction itself.

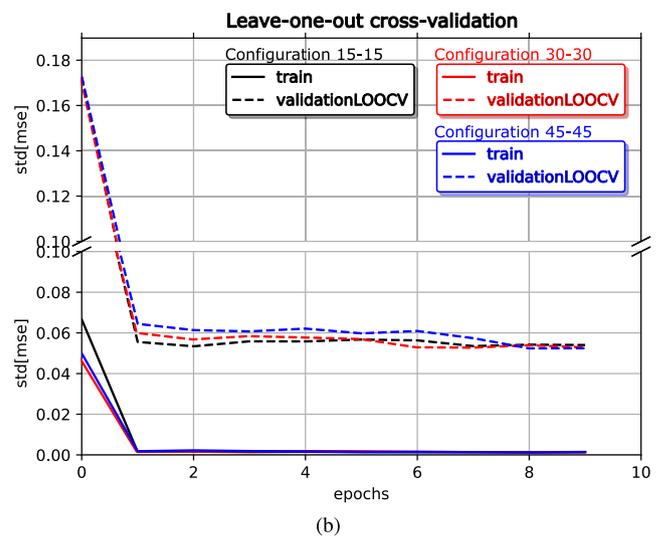
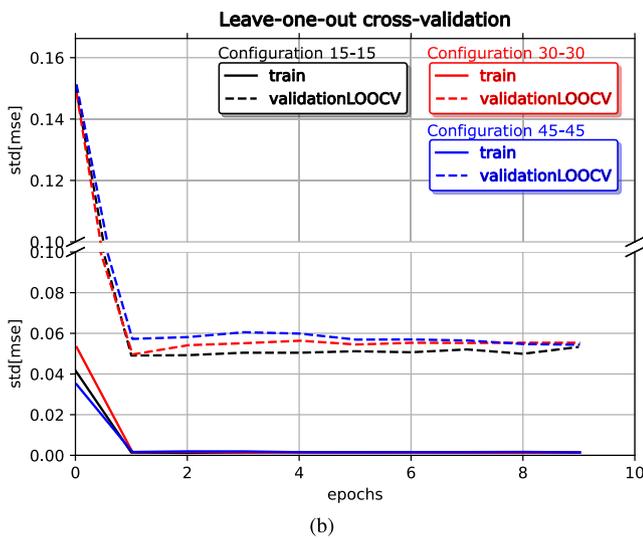
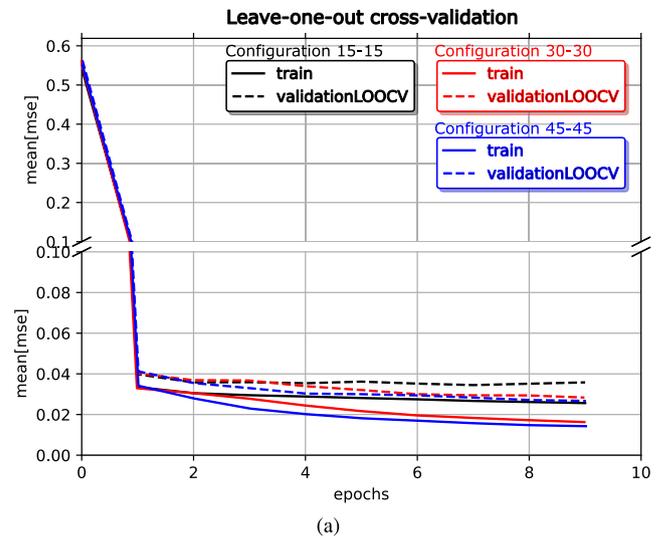
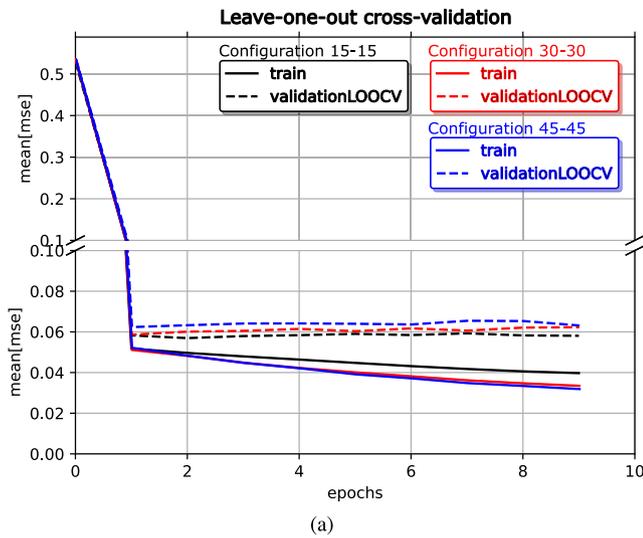


Fig. 14. Evolution of the (a) Mean MSE (training and validation) and the (b) standard deviation according to epochs when we work with alignment at the entrance to the ICU, in patients with obstructions.

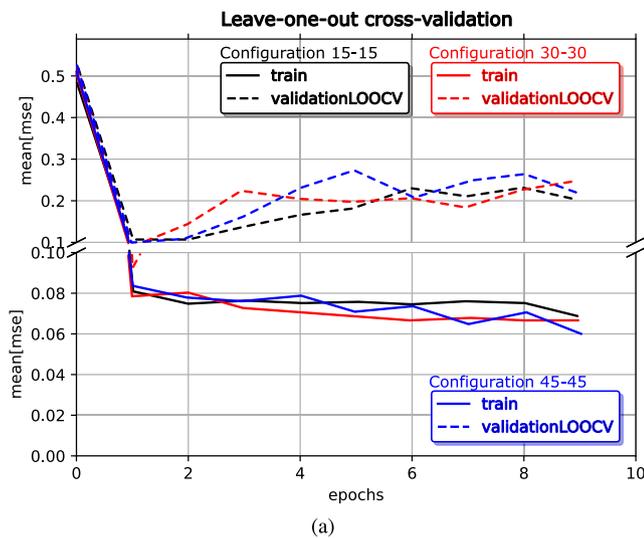
Fig. 15. Evolution of the (a) Mean MSE (training and validation) and the (b) standard deviation according to epochs when we work with alignment according to the ATO, in patients with obstructions.

The final objective in our predictions corresponds to the calculation of a prediction interval, which is a function of the output that the network has at a given moment and whose limits can represent, on average, the interval of time in which the patient will experience an actual obstruction. In this way and taking as an example the results that we observe in Fig. 18, we can conclude the following: when the network predicts an exit that places the patient's obstruction within 7 days, we take as the final exit the interval between 4.85 days and 10.87 days, meaning that the actual obstruction of the patient may occur no earlier than 4.84 days and no later than 10.87 days. This will allow us to determine the percentage of patients for whom the obstruction occurred within the prediction interval (exit from the network) in real time. In such a way, the higher this percentage is for each of the study intervals, the better the performance of the network (see Fig. 20). Additionally, if we organize the real occurrences of the obstructions of the patients around the value predicted by the network, we obtain the form (value of their limits) in which the intervals must be obtained so that the confidence factor of the prediction is 50%, 75%, 95% and 100%, as indicated in the figure. Fig. 20 shows that when the network outputs a prediction interval between 4.85 days and 10.87 days, on average, we can only ensure that we are covering 50% of

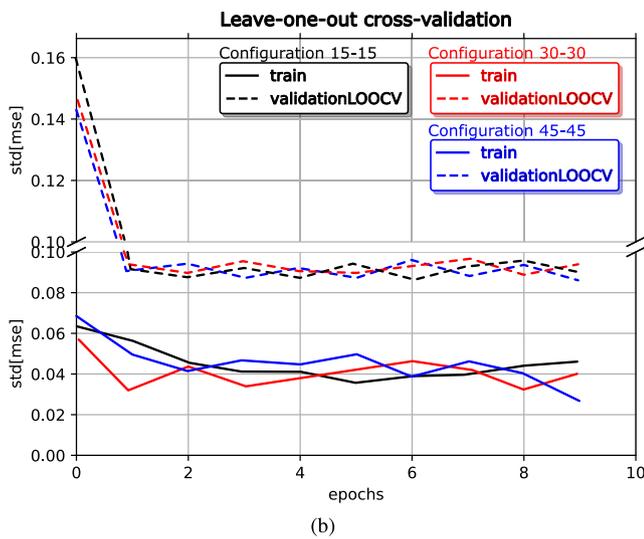
the patients. A prediction interval between 4.5 days and 12.3 days covered 75% of the patients. It can also be said that, on average, the obstruction of the patient will occur no earlier than 4.5 days and no later than 12.3 days with a 75% probability. For a percentage of 95%, the prediction interval should be between 1.95 days and 13.79 days. This same interval also covers 100% of the patients. As has been done for the analysis of trends (anticipation and lag) in Fig. 20, we can also analyse the appropriate prediction interval for the set of close predictions (between 3 days and 0 days). In this regard, we detail the configuration of intervals that we obtain when we are 48 h from the moment of obstruction in Fig. 21. We focus on the interval of the previous 48 to 49 h, and we see that the network predicts with 95% confidence that the obstructions did not arrive before 0.44 days (10.64 h).

3.3.2. Sensitivity analysis of the BigLSTM

The most common methods for dealing with missing observations in time signals (missing data imputation methods) generate synthetic observations based on the known observations, assuming that the statistical model (trends and statistical characteristics) of the existing observations is the same as the statistical model of the missing observations. This assumption implies that the missing data are ignorable



(a)



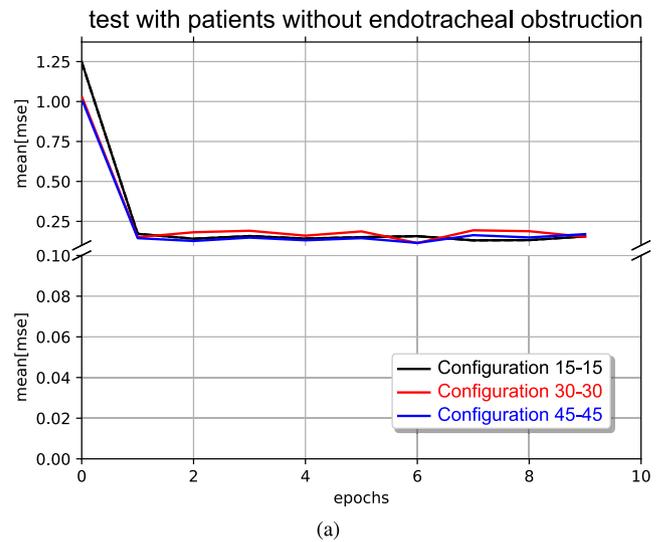
(b)

Fig. 16. Evolution of the (a) Mean MSE (training and validation) and the (b) standard deviation according to epochs when we work with alignment according to the ATO, in both types of patients (with and without obstruction).

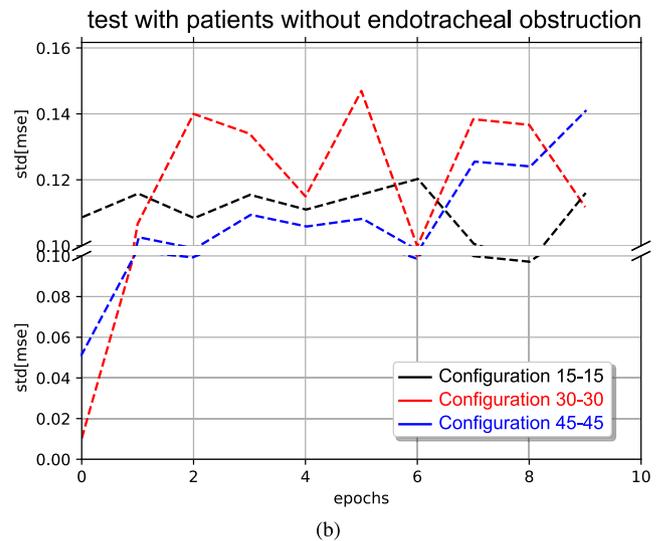
and that the available data are sufficient to correct for the missing data [47].

When the above does not happen, or that knowledge is not available, van Buuren 2018 [47] identifies two possible strategies: (1) extend the data in the imputation model (observations that we do not have) in the hope that its statistical scheme will be closer to the scenario where the probability of the missing data is completely random (MAR, Missing At Random), or (2) formulate and fit a non-ignorable imputation model and perform a sensitivity analysis of the critical parameters. The objective of this sensitivity analysis is to explore the results of our model against alternative missing data scenarios [47].

In general terms, the objective of the sensitivity analysis is based on quantifying the effects of variations in the environmental variables to which the model is subjected on the calculated results [48]. Therefore, our objective in this analysis focuses on studying the influence on the BigLSTM results of the level of missing data and sampling irregularity, both of which are present at very high levels in our set of working observations. Such influence is measured based on the MSE values of the predictions made by our BigLSTM model. To perform the above, all patients who have presented obstruction (first and second wave) are used, which are grouped into different analysis zones according



(a)



(b)

Fig. 17. Evolution of the (a) Mean MSE and (b) Associated standard deviation according to the epochs when we work with alignment according to the ATO, with patients without obstruction only for validation.

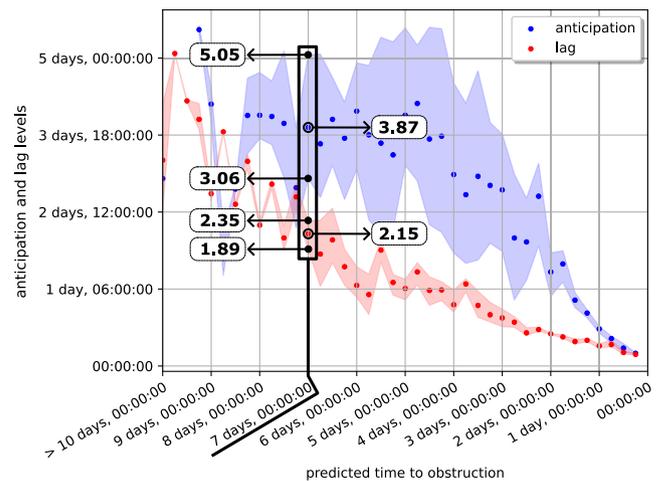


Fig. 18. Tendencies to lead and lag of the network when making distant predictions. The mean values of the errorAnticipation and errorLag variables are shown, as well as their maximum and minimum values.

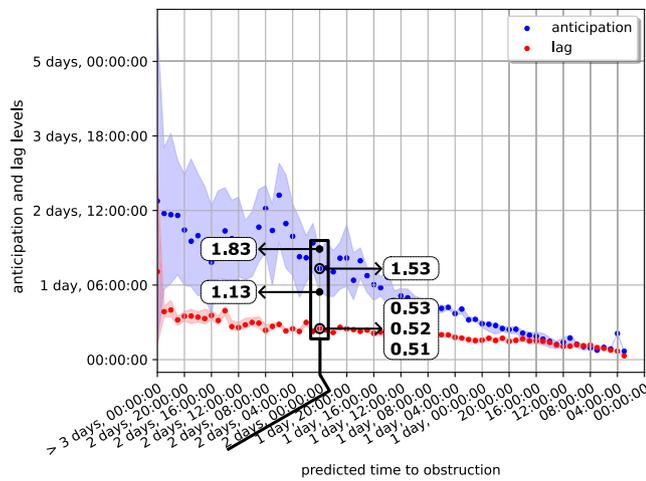


Fig. 19. Leading and lagging tendencies of the network when making distant predictions. The mean values of the errorAnticipation and errorLag variables are shown, as well as their maximum and minimum values.

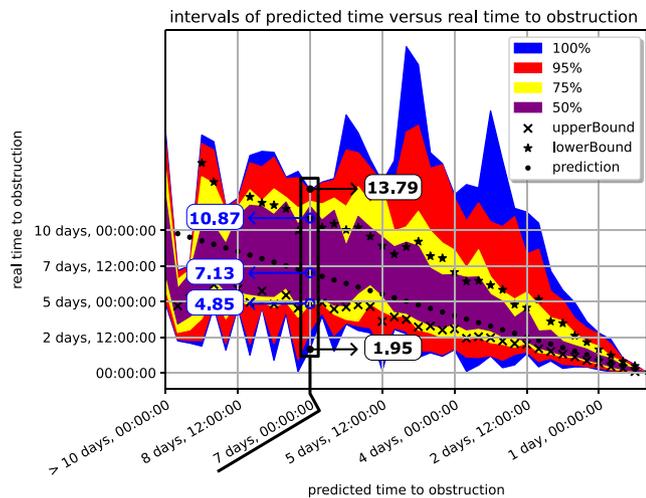


Fig. 20. Prediction intervals versus percentile distribution of the real time of obstruction in distant predictions.

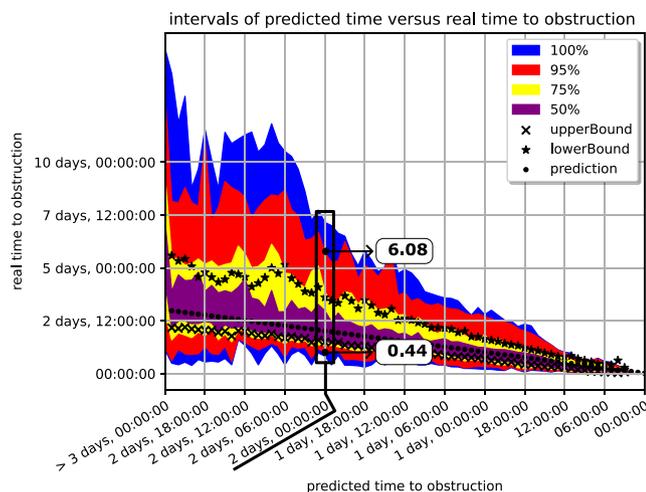


Fig. 21. Prediction intervals (close predictions) versus real-time percentile distribution of obstruction.

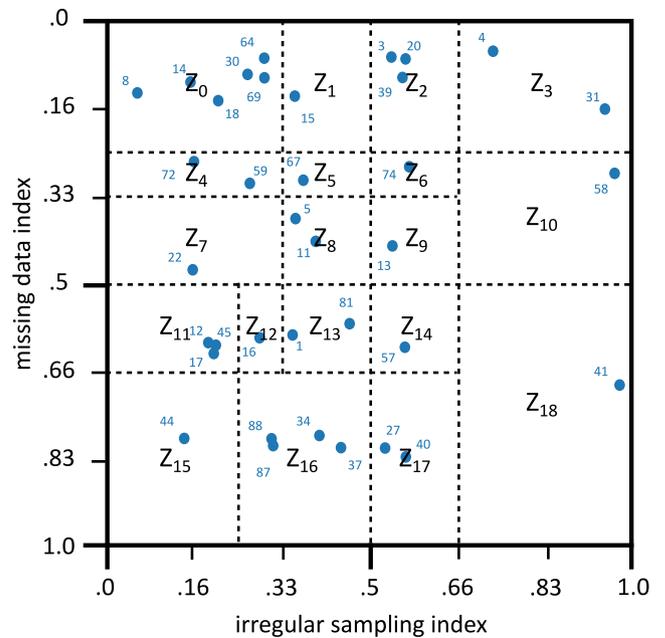


Fig. 22. Sensitivity analysis zones of the BigLSTM model as a function of the Missing Data Ratio (MDR) and Irregularity Sampling Ratio (ISR) values. The blue dots correspond to values of the missing data and irregular sampling rates presented by the patients (the identifier of each patient is shown).

to the value presented by the MDR and ISR indices. Both the MDR and ISR have a range of values between 0 (corresponding to 0% missing data and minimum irregular sampling in the observations) and 1 (corresponding to 100% missing data and the highest irregular sampling found in the data). Annex A4 details how to calculate the MDR and ISR values associated with patient observations.

The grouping of patients, based on the MDR and ISR values of their history of observations, is performed by zones (see Fig. 22), each of these corresponding to a validation process of our model, from which we obtain the validation and training MSE. The former is taken as a quantification of the influence of the level of missing data and sampling irregularity on the predictions of our model.

Table 2 shows the validation MSE values for the BigLSTM (working with a TRP based on ATO) for all study zones and all inputWidth-numberLSTMUnits configurations. The number of training epochs has been 9 and the validation procedure used has been of the LOOCV type where the patient is the main element on which the validation MSE is calculated when the model has not been trained with that patient. In this table we can see how the configurations with the highest number of minimums correspond to configurations 15–15, 30–30 and 45–45. Annex A4 shows the details of the training MSE values for each of the zones and configurations.

The 15–15, 30–30 and 45–45 configurations show a better performance, Table 2, so they are the ones used in our study. Fig. 23 shows the validation MSE value as a function of the values taken by the MDR and ISR. It allows us to analyse the sensitivity of the BigLSTM model as a function of the level of missing data and irregular sampling, and the following can be observed:

- The range of MSE values obtained is between 0.028 and 0.089 in all configurations, with the best value in the zone Z_0 (low level of missing data and minimum irregular sampling), and the worst value in zone Z_{15} (high level of missing data and minimum irregular sampling) of the 45–45 configuration.
- It is observed, when the three configurations are compared, that the sensitivity presented by the BigLSTM to the level of missing data is greater than to irregular sampling, since as the MDR value

Table 2

Validation MSE values for all analysis zones and BigLSTM model configurations. The best values obtained in each zone are highlighted in bold.

	inputWidth-numberLSTMUnits configurations								
	15-15	30-45	30-30	45-30	45-45	15-30	30-15	15-45	45-15
Z ₀	0,02871	0.03083	0.02901	0.03217	0.02859	0.03086	0.03069	0.03188	0.03190
Z ₁	0.03678	0.03920	0.03795	0.04079	0.03603	0.03918	0.03969	0.03938	0.04216
Z ₂	0.04235	0.04581	0.04348	0.04793	0.04162	0.04580	0.04546	0.04533	0.04878
Z ₃	0.05062	0.05398	0.05228	0.05600	0.05002	0.05347	0.05115	0.05575	0.05677
Z ₄	0.04314	0.04753	0.04358	0.05132	0.04309	0.04754	0.04751	0.05128	0.04989
Z ₅	0.05121	0.05590	0.05253	0.05995	0.05053	0.05586	0.05651	0.05877	0.06015
Z ₆	0.05865	0.06464	0.06002	0.06847	0.05784	0.06429	0.06395	0.06624	0.06818
Z ₇	0.04726	0.05077	0.04667	0.05534	0.04810	0.05161	0.05240	0.05520	0.05383
Z ₈	0.06051	0.06571	0.06136	0.06617	0.06076	0.06436	0.06140	0.06780	0.06611
Z ₉	0.06794	0.07446	0.06885	0.07469	0.06807	0.07280	0.06884	0.07526	0.07414
Z ₁₀	0.06691	0.07282	0.06882	0.07654	0.06625	0.07197	0.06964	0.07666	0.07617
Z ₁₁	0.07703	0.07614	0.06613	0.06887	0.06633	0.08707	0.06889	0.07667	0.07821
Z ₁₂	0.07274	0.08310	0.07243	0.07413	0.07323	0.07357	0.08505	0.08298	0.07311
Z ₁₃	0.07598	0.03803	0.05712	0.03496	0.07589	0.03632	0.03405	0.03558	0.03539
Z ₁₄	0.08134	0.08755	0.07745	0.08499	0.08308	0.08146	0.08083	0.08690	0.08179
Z ₁₅	0.06071	0.02041	0.06038	0.02190	0.08930	0.02061	0.02139	0.01906	0.02289
Z ₁₆	0.07642	0.01737	0.05668	0.01717	0.08620	0.01711	0.01755	0.01536	0.01779
Z ₁₇	0.06177	0.06689	0.05701	0.06719	0.06339	0.06225	0.06434	0.06669	0.06419
Z ₁₈	0.08031	0.08591	0.07742	0.08684	0.08126	0.08063	0.08163	0.08830	0.08381

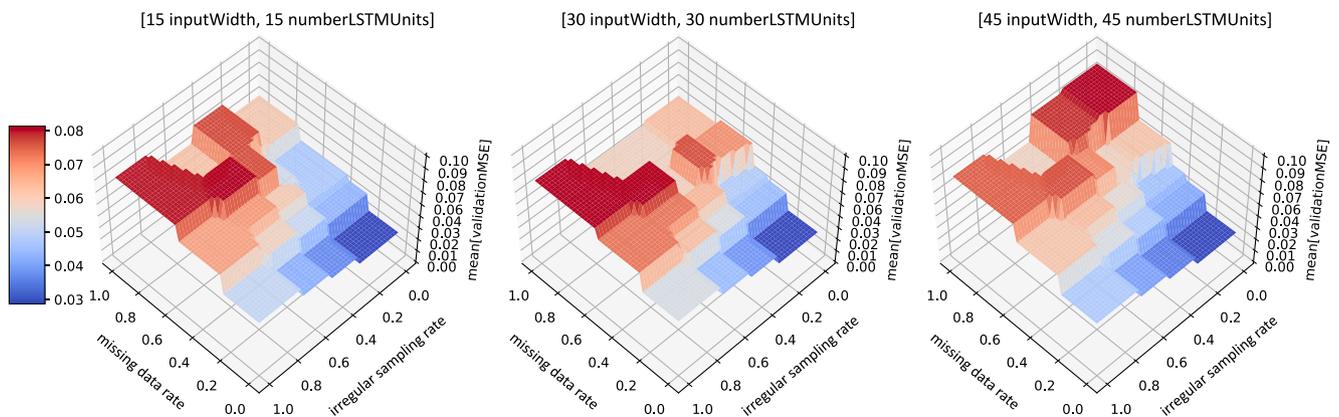


Fig. 23. Validation MSE as a function of MDR and ISR for the 15-15, 30-30 and 45-45 configurations of the BigLSTM model.

grows in the interval [0, 1], the validation MSE value increases in a larger range (from 0.02# to 0.08#) than the range of increase that occurs when the ISR index grows in the interval [0, 1].

- The largest modification of the validation MSE occurs at high values of the MDR (close to 1) and when the network size is modified. This influence of the network size on high values of the ISR index is due to the treatment of the time axis by the BigLSTM.

3.3.3. Comparative analysis of the BigLSTM with other existing approaches

In this section we present a comparative study of the computational solution proposed by us to predict endotracheal obstruction in COVID-19 patients admitted to the ICU, the BigLSTM, with another Deep Learning model, such as a nonlinear autoregressive neural architecture with external input, the Nonlinear Autoregressive with Exogenous Input (NARX) [49], along with various missing data imputation methods (Mean Imputation and Last Observation Carried Forward - LOCF). NARX models are a type of artificial neural network used for time-series prediction. They incorporate both autoregressive and exogenous inputs to forecast future values. NARX models are adept at capturing complex dependencies and nonlinear relationships within sequential data [50].

This comparative study performs the same grouping of patients by zones that we used in the previous section. Therefore, we compare the results obtained in the validation of our BigLSTM model (configurations 15-15, 30-30 and 45-45) for each of these zones, with the results obtained with the two NARX MeanImp and NARX

LOCFImp models configured as follows: 2 input delays, 2 feedback delays, Levenberg-Marquardt training algorithm.

The results obtained by the 5 models in relation to the validation MSE are shown in Table 3. In it we can see how there are two zones (Z₀ and Z₁₀) where the lowest value corresponds to the NARX MeanImp model, these zones being in direct correspondence with low values (less than 0.5) of MDR (see Fig. 24 for the location of the zones). The NARX LOCFImp model obtains the best values in zones Z₃, Z₅, Z₆ and Z₉ also corresponding to MDR values lower than 0.5 and medium-high values (higher than 0.33) of ISR.

In the rest of the zones (see Fig. 24), which are mainly associated with high MDR values (higher than 0.5), it is the BigLSTM model that obtains better mean MSE values, showing the superiority of our model for working with missing data, regardless of the value of the sampling irregularity in the input signals.

4. Conclusions

In this work, we detail BigLSTM, a recurrent neural network model for the treatment of anomalous temporal signals, such as those found in ISMTs. This new architecture has a modular structure and is composed of five interconnected modules. Four of these modules are information processing modules, and the fifth module is dedicated to controlling tasks in the internal functioning of the network. By virtue of the

Table 3

Validation MSE values for all comparison zones, and for various data imputation methods when using the NARX model, and various configurations when using the BigLSTM model.

	Methods				
	NARX MeanImp	NARX LOCFImp	BigLSTM 15-15	BigLSTM 30-30	BigLSTM 45-45
Z ₀	0.02719	0,02757	0,02871	0.02901	0.02859
Z ₁	0,03761	0,04762	0.03678	0.03795	0.03603
Z ₂	0,05016	0,04860	0.04235	0.04348	0.04162
Z ₃	0,05024	0.04889	0.05062	0.05228	0.05002
Z ₄	0,04789	0,04824	0.04314	0.04358	0.04309
Z ₅	0,04831	0.04829	0.05121	0.05253	0.05053
Z ₆	0,04848	0.04783	0.05865	0.06002	0.05784
Z ₇	0,07249	0,07230	0.04726	0.04667	0.04810
Z ₈	0,06725	0,06765	0.06051	0.06136	0.06076
Z ₉	0,06743	0,0672	0.06794	0.06885	0.06807
Z ₁₀	0.04857	0,04992	0.06691	0.06882	0.06625
Z ₁₁	0,12534	0,12611	0.07703	0.06613	0.06633
Z ₁₂	0,13012	0,12876	0.07274	0.07243	0.07323
Z ₁₃	0,12489	0,12411	0.07598	0.05712	0.07589
Z ₁₄	0,13023	0,13059	0.08134	0.07745	0.08308
Z ₁₅	0,16318	0,16313	0.06071	0.06038	0.08930
Z ₁₆	0,16796	0,16579	0.07642	0.05668	0.08620
Z ₁₇	0,17330	0,17226	0.06177	0.05701	0.06339
Z ₁₈	0,11136	0,11152	0.08031	0.07742	0.08126

MeanImp: Mean Imputation, LOCFImp: Last Observation Carried Forward Imputation.

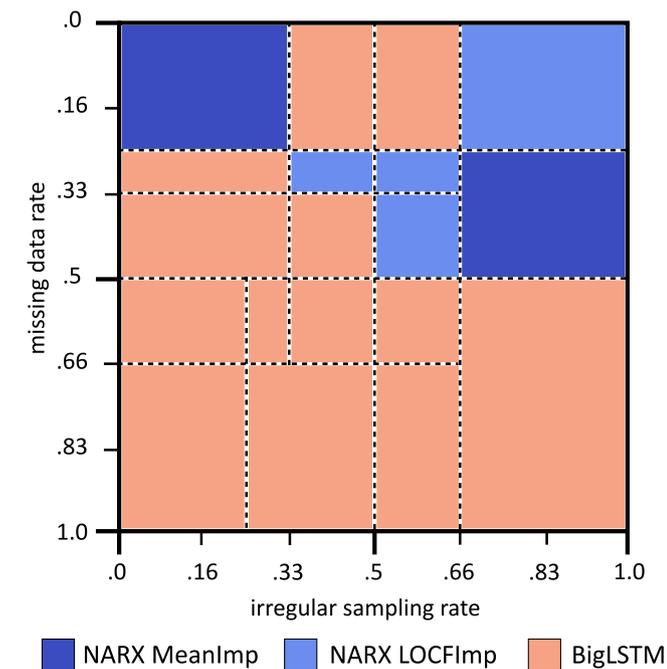


Fig. 24. Zones of analysis of the comparison between models according to the values of the MDR and ISR indexes. In each of these zones, the model with the best mean MSE value is indicated.

obtained results, we validate the explicit scheme of the function decomposition methodology, which has led us to use the structure of BigLSTM to construct computational prediction systems.

BigLSTM is a model that is based on raw data and allows direct operation with an ISMTS as input. Techniques of models based on irregular time intervals (modelling the ISTS as they are and incorporating an effective mechanism to manage the time axis of the observations) and models based on multiple sampling rates (through the injection of redundancy) are incorporated.

We analysed the performance of BigLSTM in developing a system that faces a real problem in the clinical field, such as the prediction of the time at which endotracheal obstruction can occur in COVID-19

patients who are in the ICU. We worked with signals from IMV that presented a very high level of missing data and irregular sampling. Our dataset consisted of 96 patients hospitalized for COVID-19 in the CHUIMI ICU between March 8, 2020, and February 4, 2021. This time period covers the first and second waves of COVID-19 in Spain.

To validate the previous system in the previous scenario from the computational perspective, a cross-validation procedure of the LOOCV type was followed, taking the patient as a basic element of validation.

Various work schemes were designed based on the alignment of the records when they were presented to the network, and we obtained different prediction models. The best results were obtained when we worked with an alignment of the observations centred on an ATO of 11.02 days, the latter being calculated based on the patients who presented with obstruction and who were involved in the training process and validation. The best validation mean MSE value reached 0.028. This corresponds to a configuration in which we had 45 units in each LSTM cell and a window size of 45 observations, and where the patients who presented with obstruction were included in the validation process. The system does not enter an overadjustment regime.

The previous result can be considered promising enough to build a system around the said registration alignment scheme (centred on an ATO of 11.02 days) for predicting a measurable obstruction. Furthermore, it can give us as a result not only the prediction but also the percentage of confidence. However, we must bear in mind that these results deteriorate if we also include in our validation process the records of the patients who never presented with an obstruction.

In the previous context, a first analysis was carried out where the system worked in a training and validation regime with the data of all patients (37 with and 39 without obstruction), obtaining a validation MSE value of 0.09 for a configuration of 30 units in each LSTM cell and for a window size of 30 observations. A second analysis was also carried out, where predictions were made for the records of observations associated with patients without obstruction, using a network trained on patients with obstruction, giving a mean MSE value that fluctuated around 0.20.

Using the scheme based on an alignment of the observations centred on an ATO of 11.02 days, the leading and lagging trends presented by the system were analysed. We found that in our system, when a long-term forecast was made (7 days before the time of the obstruction), the forecast showed (on average) a leading trend of 3.87 days and a lagging trend of 2.15 days. The above interval between days 4.85 and 10.87 from the time of the prediction was established.

If we work with short-term predictions made 48 h before the time at which the obstruction may occur, the trend towards advance is limited to values between 1.88 and 1.13 days, and the trend towards delay is between 0.53 and 0.51 days. Therefore, if our system predicted that the patient's obstruction would occur within 48 h, the actual obstruction would have occurred at the earliest in the next 32 h and not after the next 93.12 h (3.88 days).

Based on the previous results, we have been able to convert our point predictions into a prediction of an interval with a confidence percentage associated with it. Thus, to achieve a 50% confidence level in the long-term predictions, the network must provide an interval that places the obstruction no earlier than the next 4.85 days and no later than the next 10.87 days. To achieve a 75% confidence level, the network should provide an interval that places the obstruction no earlier than the next 4.5 days and no later than the next 12.3 days. For short term predictions (48 h before possible occurrence) and to achieve a 95% confidence level, the network must provide an interval that places the obstruction no earlier than the next 10.64 h and no later than the next 6.8 days.

A sensitivity analysis of the BigLSTM model has been carried out as a function of the MDR and ISR indices. This analysis has allowed us to identify that the best prediction values are obtained when the level of missing data is low and the observations have been made with a minimum irregular sampling, and that when the designed intelligent

system is faced with very high levels of missing data and highly irregular sampling, its performance is more limited, as it has a validation MSE value of up to 0.89. The BigLSTM has a higher sensitivity to the level of missing data than to irregular sampling.

Finally, a comparative study was carried out with another deep architecture, the nonlinear autoregressive neural architecture with external input (NARX). The results obtained show the superiority of our deep neural architecture model, the BigLSTM, for working with missing data, regardless of the value of the sampling irregularity in the input signals. This is determined by the fact that the BigLSTM model achieves better average MSE values than the NARX when working in environments with high levels of missing data.

All the above conclusions demonstrate the suitability and goodness of the new deep neural architecture, BigLSTM, presented in this work, for dealing with irregular time series and with a high level of missing data, as well as its absolute applicability in high-risk clinical prediction problems.

CRedit authorship contribution statement

Pablo Fernández-López: Writing – original draft, Validation, Software, Methodology, Investigation, Formal analysis, Conceptualization. **Patricio García Báez:** Writing – review & editing, Validation, Methodology, Investigation. **Ylermi Cabrera-León:** Writing – review & editing, Visualization, Data curation. **Juan L. Navarro-Mesa:** Writing – review & editing, Visualization, Validation, Formal analysis. **Guillermo Pérez-Acosta:** Writing – review & editing, Visualization, Data curation. **José Blanco-López:** Writing – review & editing, Visualization, Data curation. **Carmen Paz Suárez-Araujo:** Writing – review & editing, Validation, Supervision, Resources, Project administration, Methodology, Funding acquisition, Conceptualization.

Declaration of competing interest

None Declared. The authors have no conflict of interest to report.

Appendix A. Supplementary data

Supplementary material related to this article can be found online at <https://doi.org/10.1016/j.combiomed.2025.110146>.

References

- [1] C. Sun, S. Hong, M. Song, H. Li, A review of deep learning methods for irregularly sampled medical time series data, 2020, <http://dx.doi.org/10.48550/arxiv.2010.12493>, ArXiv.
- [2] S. Aydin, Time series analysis and some applications in medical research, *J. Math. Stat. Stud.* 3 (2) (2022) 31–36, URL: <https://doi.org/10.32996/jmss.2022.3.2.3>.
- [3] Mohammad Amin Morid, Olivia R. Liu Sheng, Joseph Dunbar, Time series prediction using deep learning methods in healthcare, *ACM Trans. Manag. Inf. Syst.* 14 (1) (2023) 2, URL: <https://doi.org/10.1145/3531326>.
- [4] S. Kaushik, A. Choudhury, P.K. Sheron, N. Dasgupta, S. Natarajan, L.A. Pickett, V. Dutt, AI in healthcare: Time-series forecasting using statistical, neural, and ensemble architectures, *Front. Big Data* 3 (2020) 4, <http://dx.doi.org/10.3389/fdata.2020.00004>.
- [5] Youseop Shin, *Time Series Analysis in the Social Sciences: The Fundamentals*, first ed., University of California Press, 2017, URL: <https://www.jstor.org/stable/10.1525/j.ctv1xxvhf>.
- [6] John M. Gottman, *Time-Series Analysis, A comprehensive Introduction for Social Scientists*, Cambridge University Press, 2009.
- [7] Pham Dinh Tuan, *Time series analysis and biology*, in: M. Cosnard, et al. (Eds.), *Rhythms in Biology and Other Fields of Application*, Springer-Verlag Berlin Heidelberg, 1983, pp. 368–387.
- [8] A. Chakraborti, M. Patriarca, M.S. Santhanam, Financial time-series analysis: a brief overview, in: A. Chatterjee, B.K. Chakraborti (Eds.), *Econophysics of Markets and Business Networks*, in: *New Economic Windows*, Springer, Milano, 2007, pp. 51–67, URL: https://doi.org/10.1007/978-88-470-0665-2_4.
- [9] Ngai Hang Chan, *Time Series: Applications to Finance*, Wiley, New York, 2002, p. 224.
- [10] D.M. Hanssens, Market response, competitive behavior, and time series analysis, *J. Mark. Res.* 17 (4) (1980) 470–485, URL: <https://doi.org/10.1177/002224378001700406>.
- [11] R.H. Shumway, D.S. Stoffer, Time series regression and exploratory data analysis, in: *Time Series Analysis and Its Applications*, in: *Springer Texts in Statistics*, Springer, Cham, 2017, pp. 45–74, URL: https://doi.org/10.1007/978-3-319-52452-8_2.
- [12] E.J. Keogh, K. Chakrabarti, M.J. Pazzani, S. Mehrotra, Dimensionality reduction for fast similarity search in large time series databases, *Knowl. Inf. Syst.* 3 (3) (2001) 263–286.
- [13] M. Marlin, D.C. Kale, R.G. Khemani, R.C. Wetzel, Unsupervised pattern discovery in electronic health care data using probabilistic clustering models, in: *Proceedings of the 2nd ACM SIGMINT International Health Informatics Symposium*, 2012, pp. 389–398.
- [14] Paul D. Allison, *Missing Data (Quantitative Applications in the Social Sciences)*, first ed., SAGE Publications, Inc, 2001.
- [15] A.N. Baraldi, C.K. Enders, An introduction to modern missing data analyses, *J. Sch. Psychol.* 48 (1) (2010) 5–37, URL: <https://doi.org/10.1016/j.jsp.2009.10.001>.
- [16] Shin-Fu Wu, Chia-Yung Chang, Shie-Jue Lee, Time series forecasting with missing values, *Endorsed Trans. Cogn. Commun.* (2015) URL: <https://doi.org/10.4108/icst.iniscom.2015.258269>.
- [17] Z.C. Lipton, D. Kale, R. Wetzel, Directly modeling missing data in sequences with rns: Improved classification of clinical time series, in: *Proceedings of the 1st Machine Learning for Healthcare Conference*, 2016, pp. 253–270.
- [18] H. Harutyunyan, H. Khachatrian, D.C. Kale, A. Galstyan, Multitask learning and benchmarking with clinical time series data, *Sci. Data* 6 (2019).
- [19] J. Yoon, J. Jordon, M. van der Schaar, GAIN: Missing data imputation using generative adversarial nets, in: *Proceedings of the 35th International Conference on Machine Learning*, 2018, pp. 5689–5698.
- [20] S.C.-X. Li, B. Jiang, B. Marlin, Learning from incomplete data with generative adversarial networks, in: *International Conference on Learning Representations*, 2019, URL: <https://doi.org/10.48550/arXiv.1902.09599>.
- [21] D. Pathak, P. Krahenbuhl, J. Donahue, T. Darrell, A.A. Efros, Context encoders: Feature learning by inpainting, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 2536–2544, <http://dx.doi.org/10.48550/arXiv.1604.07379>.
- [22] Eric Kandel, In search of memory: The emergence of a new science of mind, *FASEB J.* 20 (2006) <http://dx.doi.org/10.1096/fj.06-0604ufm>.
- [23] L. Chariker, R. Shapley, Lai-Sang Young, Rhythm and synchrony in a cortical network model, *J. Neurosci.* 38 (40) (2018) 8621–8634, <http://dx.doi.org/10.1523/JNEUROSCI.0675-18.2018>.
- [24] Amanda J.C. Sharkey (Ed.), *Combining Artificial Neural Nets. Ensemble and Modular Multi-Net Systems*, Springer, 1999, <http://dx.doi.org/10.1007/978-1-4471-0793-4>.
- [25] S. Hochreiter, J. Schmidhuber, LSTM can solve hard long time lag problems, in: M.C. Mozer, M.I. Jordan, T. Petsche (Eds.), *Advances in Neural Information Processing Systems 9, NIPS 9*, MIT Press, 1997, pp. 473–479, URL: <https://proceedings.neurips.cc/paper/1996/file/a4d2f0d23dccc84ce983ff9157f8b7f88-Paper.pdf>.
- [26] John F. Kolen, Stefan C. Kremer, Gradient flow in recurrent nets: The difficulty of learning LongTerm dependencies, in: *A Field Guide to Dynamical Recurrent Networks*, 2001, pp. 237–243, <http://dx.doi.org/10.1109/9780470544037.ch14>.
- [27] Sepp Hochreiter, Jürgen Schmidhuber, Long short-term memory, *Neural Comput.* 9 (1997) 1735–1780, <http://dx.doi.org/10.1162/neco.1997.9.8.1735>.
- [28] F.A. Gers, J. Schmidhuber, F. Cummins, Learning to forget: Continual prediction with LSTM, *Neural Comput.* 12 (10) (2000) 2451–2471, <http://dx.doi.org/10.1162/089976600300015015>.
- [29] F.A. Gers, J. Schmidhuber, Recurrent nets that time and count, in: *Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium*, vol. 3, 2000, pp. 189–194, <http://dx.doi.org/10.1109/IJCNN.2000.861302>.
- [30] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, *Neural Netw.* 18 (5–6) (2005) 602–610.
- [31] K. Greff, R.K. Srivastava, J. Koutnik, B.R. Steunebrink, J. Schmidhuber, LSTM: A search space odyssey, *IEEE Trans. Neural Netw. Learn. Syst.* (2016) 1–11, <http://dx.doi.org/10.1109/tnnls.2016.2582924>.
- [32] Jan Andreas Kloka, Lea Valeska Blum, Oliver Old, Kai Zacharowski, Benjamin Friedrichson, Characteristics and mortality of 561,379 hospitalized COVID-19 patients in Germany until december 2021 based on real-life data, *Sci. Rep.* 12 (2022) 11116, <http://dx.doi.org/10.1038/s41598-022-15287-3>.
- [33] C. Ferrando, R. Mellado-Artigas, A. Gea, E. Arruti, C. Aldecoa, A. Bordell, et al., Patient characteristics, clinical course and factors associated to icu mortality in critically ill patients infected with SARS-CoV-2 in Spain: A prospective, cohort, multicentre study, *Rev. Esp. Anesthesiol. Reanim. (Engl. Ed.)* 67 (8) (2020) 425–437, <http://dx.doi.org/10.1016/j.redar.2020.07.003>.
- [34] Li Yan, Hai-Tao Zhang, Jorge Goncalves, Yang Xiao, Maolin Wang, Yuqi Guo, Chuan Sun, Xiuchuan Tang, Liang Jing, Mingyang Zhang, Xiang Huang, Ying Xiao, Haosen Cao, Yanyan Chen, Tongxin Ren, Fang Wang, Yaru Xiao, Sufang Huang, Xi Tan, Niannian Huang, Bo Jiao, Cheng Cheng, Yong Zhang, Ailin Luo, Laurent Mombaerts, Junyang Jin, Zhiguo Cao, Shusheng Li, Hui Xu, Ye Yuan, An interpretable mortality prediction model for COVID-19 patients, *Nat. Mach. Intell.* 2 (2020) 283–288, <http://dx.doi.org/10.1038/s42256-020-0180-7>.

- [35] A. Chen, Z. Zhao, W. Hou, A.J. Singer, H. Li, T.Q. Duong, Time-to-death longitudinal characterization of clinical variables and longitudinal prediction of mortality in COVID-19 patients: A two-center study, *Front. Med.* 8 (2021) 661940, <http://dx.doi.org/10.3389/fmed.2021.661940>.
- [36] Kai Zhou, Yaoting Sun, Lu Li, Zelin Zang, Jing Wang, Jun Li, Junbo Liang, Fangfei Zhang, Qiushi Zhang, Weigang Ge, Hao Chen, Xindong Sun, Liang Yue, Xiaomai Wua, Bo Shen, Jiaqin Xu, Hongguo Zhu, Shiyong Chen, Hai Yang, Shigao Huang, Minfei Peng, Dongqing Lv, Chao Zhang, Haihong Zhao, Luxiao Hong, Zhehan Zhou, Haixiao Chen, Xuejun Dong, Chunyu Tu, Minghui Li, Yi Zhu, Baofu Chen, Stan Z. Li, Tiannan Guo, Eleven routine clinical features predict COVID-19 severity uncovered by machine learning of longitudinal measurements, *Comput. Struct. Biotechnol. J.* 19 (2021) 3640–3649, <http://dx.doi.org/10.1016/j.csbj.2021.06.022>.
- [37] M. Seyedtabib, R. Najafi-Vosough, N. Kamyari, The predictive power of data: machine learning analysis for Covid-19 mortality based on personal, clinical, preclinical, and laboratory variables in a case–control study, *BMC Infect. Dis.* 24 (2024) 411, <http://dx.doi.org/10.1186/s12879-024-09298-w>.
- [38] Krishnaraj Chadaga, Srikanth Prabhu, Niranjana Sampathila, Rajagopala Chadaga, Shashikiran Umakanth, Devadas Bhat, Shashi Kumar G. S., Explainable artificial intelligence approaches for COVID-19 prognosis prediction using clinical markers, *Sci. Rep.* 14 (2024) 1783, <http://dx.doi.org/10.1038/s41598-024-52428>.
- [39] Zhijiang Yu, Negin Ashrafi, Hexin Li, Kamiar Alaei, Maryam Pishgar, Prediction of 30-day mortality for ICU patients with sepsis-3, *BMC Med. Inform. Decis. Mak.* 24 (2024) 223, <http://dx.doi.org/10.1186/s12911-024-02629-6>.
- [40] AE Johnson, TJ Pollard, L Shen, LW Lehman, M Feng, M Ghassemi, B Moody, P Szolovits, LA Celi, RG Mark, MIMIC-III, a freely accessible critical care database, *Sci. Data* 3 (2016) 160035, <http://dx.doi.org/10.1038/sdata.2016.35>.
- [41] T. Pham, L.J. Brochard, A.S. Slutsky, Mechanical ventilation: State of the art, *Mayo Clin. Proc.* 92 (9) (2017) 1382–1400, <http://dx.doi.org/10.1016/j.mayocp.2017.05.004>.
- [42] N.R. MacIntyre, Evidence-based guidelines for weaning and discontinuing ventilatory support: A collective task force facilitated by the American college of chest physicians; the American association for respiratory care; and the American college of critical medicine, *Chest* 120 (6 SUPPL) (2001) http://dx.doi.org/10.1378/chest.120.6_suppl.375s.
- [43] P.L. Silva, P. Pelosi, P.R.M. Rocco, Optimal mechanical ventilation strategies to minimize ventilator-induced lung injury in non-injured and injured lungs, *Expert. Rev. Respir. Med.* 10 (12) (2016) 1243–1245, <http://dx.doi.org/10.1080/17476348.2016.1251842>.
- [44] Picis, ICU care manager software, patient care unit system, 2021, <https://www.picis.com/en/solution/clinical-information-system-suite/critical-care-manager/>. (Accessed 25 November 2024).
- [45] C. Sammut, G.I. Webb, Leave-one-out cross-validation, in: *Encyclopedia of Machine Learning*, Springer, Boston, MA, 2011, pp. 600–601, http://dx.doi.org/10.1007/978-0-387-30164-8_469.
- [46] Pablo Fernández-López, Carmen Paz Suárez-Araujo, Patricio García-Báez, Francisco Suárez-Díaz, Juan L. Navarro-Mesa, Guillermo Pérez-Acosta, José Blanco-López, Toward an intelligent computing solution for endotracheal obstruction prediction in COVID-19 patients in ICU, in: *Lecture Notes in Computer Science*, vol. 12861, Springer, 2021, pp. 61–73, http://dx.doi.org/10.1007/978-3-030-85030-2_6.
- [47] S. van Buuren, *Flexible Imputation of Missing Data*, second ed., CRC Press, Boca Raton, FL, 2018.
- [48] Dan G. Cacuci, *Sensitivity and Uncertainty Analysis: Theory*, Chapman and Hall, 2003.
- [49] Stephen A. Billings, *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio-Temporal Domains*, first ed., John Wiley and Sons, Ltd., 2013.
- [50] José María P. Menezes, Guilherme A. Barreto, Long-term time series prediction with the NARX network: An empirical evaluation, *Neurocomputing* 71 (16–18) (2008) 3335–3343, <http://dx.doi.org/10.1016/j.neucom.2008.01.030>.