Future Generation Computer Systems 133 (2022) 68–83

Contents lists available at ScienceDirect

Future Generation Computer Systems

journal homepage: www.elsevier.com/locate/fgcs

Interpretable clinical time-series modeling with intelligent feature selection for early prediction of antimicrobial multidrug resistance

Sergio Martínez-Agüero [a], Cristina Soguero-Ruiz [a,*], Jose M. Alonso-Moral [b,c], Inmaculada Mora-Jiménez [a], Joaquín Álvarez-Rodríguez [d], Antonio G. Marques [a]

[a] Department of Signal Theory and Communications, Telematics and Computing Systems, Rey Juan Carlos University, 28942, Fuenlabrada, Spain
[b] Centro Singular de Investigación en Tecnoloxías Intelixentes (CiTIUS), Universidade de Santiago de Compostela, 15782, Santiago de Compostela, Spain
[c] Departamento de Electrónica e Computación, Universidade de Santiago de Compostela, 15782, Santiago de Compostela, Spain
[d] Intensive Care Department, University Hospital of Fuenlabrada, 28942, Fuenlabrada, Spain

ARTICLE INFO

Article history:
Received 31 July 2021
Received in revised form 17 December 2021
Accepted 23 February 2022
Available online 10 March 2022

MSC:
92C50
37M10
68T01
68T07
94D05

Keywords:
Explainable artificial intelligence
Multivariate Time Series
Recurrent neural network
Linguistic fuzzy models
Antimicrobial multidrug resistance
Intensive Care Unit

ABSTRACT

Electronic health records provide rich, heterogeneous data about the evolution of the patients' health status. However, such data need to be processed carefully, with the aim of extracting meaningful information for clinical decision support. In this paper, we leverage interpretable (deep) learning and signal processing tools to deal with multivariate time-series data collected from the Intensive Care Unit (ICU) of the University Hospital of Fuenlabrada (Madrid, Spain). The presence of antimicrobial multidrug-resistant (AMR) bacteria is one of the greatest threats to the health system in general and to the ICUs in particular due to the critical health status of the patients therein. Thus, early identification of bacteria at the ICU and early prediction of their antibiotic resistance are key for the patients' prognosis. While intelligent data-based processing and learning schemes can contribute to this early prediction, their acceptance and deployment in the ICUs require the automatic schemes to be not only accurate but also understandable by clinicians. Accordingly, we have designed trustworthy intelligent models for the early prediction of AMR based on the combination of meaningful feature selection with interpretable recurrent neural networks. These models were created using irregularly sampled clinical measurements, both considering the health status of the patient and the global ICU environment. We explored several strategies to cope with strongly imbalance data, since only a few ICU patients are infected by AMR bacteria. It is worth noting that our approach exhibits a good balance between performance and interpretability, especially when considering the difficulty of the classification task at hand. A multitude of factors are involved in the emergence of AMR (several of them not fully understood), and the records only contain a subset of them. In addition, the limited number of patients, the imbalance between classes, and the irregularity of the data render the problem harder to solve. Our models are also enriched with SHAP post-hoc interpretability and validated by clinicians who considered model understandability and trustworthiness of paramount concern for pragmatic purposes. Moreover, we use linguistic fuzzy systems to provide clinicians with explanations in natural language. Such explanations are automatically generated from a pool of interpretable rules that describe the interaction among the most relevant features identified by SHAP. Notice that clinicians were especially satisfied with new insights provided by our models. Such insights helped them to trust the automatic schemes and use them to make (better) decisions to mitigate AMR spreading in the ICU. All in all, this work paves the way towards more comprehensible time-series analysis in the context of early AMR prediction in ICUs and reduces the time of detection of infectious diseases, opening the door to better hospital care.

1. Introduction

In the last decade, there has been a growing interest in analyzing clinical data as time-series sequences, allowing clinical experts to assess better the patient's health evolution [1–3]. Multivariate Time Series (MTS) have a strong presence beyond clinical

* Corresponding author.
E-mail addresses: sergio.martinez@urjc.es (S. Martínez-Agüero), cristina.soguero@urjc.es (C. Soguero-Ruiz), josemaria.alonso.moral@usc.es (J.M. Alonso-Moral), inmaculada.mora@urjc.es (I. Mora-Jiménez), joaquin.alvarez@salud.madrid.org (J. Álvarez-Rodríguez), antonio.garcia.marques@urjc.es (A.G. Marques).

https://doi.org/10.1016/j.future.2022.02.021

applications, with relevant examples including finance, meteorology, or video processing, to name a few [4]. Focusing on healthcare applications, different data-driven approaches based on MTS have been developed [5,6].

Given the complexity and irregular patterns present in clinical datasets, deep neural networks (NNs) have emerged as a suitable alternative to model and handle MTS [7]. Lasko et al. pioneered the application of deep learning tools to healthcare, demonstrating the capacity of deep learning to generalize patterns from serum uric acid measurements [8]. Three of the most widely-used deep learning approaches for dealing with time-series sequences are the Gated Recurrent Unit (GRU) [9], the Long Short-Term Memory (LSTM) [10] and the Bidirectional LSTM (Bi-LSTM) [11]. The GRU, LSTM, and Bi-LSTM are different instances of Recurrent NNs (RNNs), which are widely used for prediction using MTS due to their ability to deal with time-varying observations and capture long-term temporal dependencies [10]. For example, Lipton et al. applied an LSTM network to classify diagnoses based on the temporal data recorded in the Electronic Health Record (EHR) of a pediatric Intensive Care Unit (ICU) [12]; Pham et al. used an LSTM to model the interaction between diagnosis and medication [13]; and Nguyen et al. developed a Bi-LSTM model to predict ICU mortality outcomes [14].

In this paper, we describe how different RNNs can predict antimicrobial multidrug resistance (AMR) in the ICU. AMR can be defined as the bacteria's ability to withstand the effects of a variety of harmful chemical agents designed to damage them [15]. The adaptation of the bacteria to different antimicrobials (to which they were previously susceptible) is a serious challenge due to the reduction of appropriate treatments and the scarcity of secondary antimicrobials [15,16]. As a result, situations such as cuts, care of premature babies, chemotherapy against cancer, or infections can cause debilitating or even lethal epidemics in the absence of effective treatments [15,17].

Understanding AMR factors (e.g., epidemiology, emergence, prevalence, or burden of infectious diseases) is crucial for early AMR prediction. It is also likely to improve decision-making processes in ICU management, e.g. by allowing early patient isolation and therefore reducing AMR rates. Even if RNNs are ready to achieve high performance, they behave in practice as black boxes, hindering their interpretation by humans. The lack of interpretability is even more severe for MTS-based models like GRUs and LSTMs due to their fairly opaque hidden states [18]. In particular, because the information stored in the hidden states is a mixture of all the MTS, it is impossible to discern the individual contribution of each time series. This lack of interpretability is one of the main reasons why data-driven machine learning (ML) models in general, and RNNs in particular, are not intensively used in healthcare applications yet [19]. Indeed, interpretability is of paramount importance to make intelligent systems ready to assist clinicians in high-risk decision-making processes [20]. Accordingly, intelligent clinical models need to be endowed with interpretability as a requirement to become explainable, trustworthy, and used worldwide [21].

The so-called Responsible and Trustworthy Artificial Intelligence (AI) pays attention to fairness, accountability, responsibility, and privacy, in scenarios where Explainable AI (XAI in short) plays a key role [22]. XAI is an endeavor to develop human-centric AI sensitive not only to technical but also legal and ethical issues. XAI is rooted in knowledge engineering, which transforms raw data into meaningful knowledge (through data collection, data pre-processing, feature engineering, interpretable modeling, validation, etc.) ready to be understood by humans while respecting the "chain of trust" [23]. All in all, the aim of XAI is twofold [24]: (i) building "white-box" AI models (e.g., decision trees, rule-based systems, expert systems, etc.) that are interpretable by design [25]; and (ii) developing novel techniques to endow opaque data-driven AI models (e.g., RNNs) with interpretability [26]. More precisely, approaches for explaining black-box models can be categorized into two main groups regarding the type of explanations: (ii.a) intrinsic explanations supported by interpretable surrogate models [27]; and (ii.b) extrinsic post-hoc explanations (e.g., SHAP [28]) that only pay attention to the model output while disregarding the model internal mechanisms.

In this work, we apply XAI for assisting clinicians in the discovery and understanding of how AMR develops and spreads in the ICU. This is the main novelty of this work compared to previous studies that have attempted to model ICU information as MTS to predict the AMR onset [29,30]. This paper extends the preliminary work published in [30]. Unlike the preceding study, we have further expanded the proposed model by introducing different time window lengths, new meaningful features such as the ICU occupation, the treatment provided in the ICU, and the application of XAI for assisting clinicians. Our main contributions are as follows:

- Analyzing and modeling MTS related to AMR in the challenging scenario of an ICU. The dataset compiles data with the evolution of 3,470 patients. Data have been carefully cleaned and pre-processed before modeling.
- In the modeling stage, we coped with missing values in MTS and class imbalance.
- Regarding XAI, we first studied the effect of Feature Selection (FS), finding out relevant and meaningful features for clinicians. Then, we built several predictors based on RNNs (endowed with post-hoc interpretability) to model the temporal relation among the previously selected features. Then, we built linguistic (interpretable by design) models to better understand the interaction among the most relevant features in the model that exhibited the best interpretability-performance trade off.
- Validating with clinicians the interpretability of the models in AMR prediction.

The rest of the paper is organized as follows. Section 2 presents the notation and methods used in this paper. Section 3 describes the dataset and the related pre-processing tasks. Experiments and results are shown in Section 4. Finally, the main conclusions and associated discussions are drawn in Section 5.

2. Methods

The experimental pipeline is sketched in Fig. 1 and discussed in the following sections. Data pre-processing is introduced in Section 2.1. Then, Section 2.2 describes the MTS classification stage. Finally, validation is addressed in Section 2.3, paying attention to performance and interpretability.

2.1. Data preparation

We start by introducing the notation adopted throughout the remainder of the manuscript. We consider I patients, indexed by $i = 1, 2, \ldots, I$. Each patient is modeled as a collection of D time series, all of them with the same length (duration) T_i. Therefore, data associated with the ith patient can be arranged in the matrix $\mathbf{X}_i = [\mathbf{x}_i^1, \mathbf{x}_i^2, \ldots, \mathbf{x}_i^{T_i}] \in \mathbb{R}^{D \times T_i}$. The column vector $\mathbf{x}_i^t$ contains the D variables associated with the tth time slot, i.e., $\mathbf{x}_i^t = [x_i^{(t,1)}, x_i^{(t,2)}, \ldots, x_i^{(t,D)}]^\top$. Thus, $x_i^{(t,d)}$ represents the value of the dth feature in the tth time slot for the ith patient. Note that, while the value of D is the same across patients, the value of T_i can be different, since the length of the patient's ICU stay depends on the condition and evolution of the particular patient.

The task that we address is cast as a binary classifier, with label '1' identifying AMR patients. We use y_i to represent the

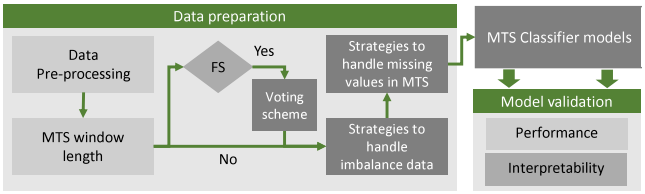


Fig. 1. Graphical illustration of the workflow implemented. First, the MTSs are preprocessed considering different time window lengths (see Section 3). An FS process is performed using three FS procedures and a voting strategy (see Section 2.1.2). Next, different strategies are applied to handle the imbalance data and deal with missing MTS values. At that point, we create different models using four NN architectures for MTS (see Section 2.2). Afterwards, we evaluate their performance and interpretability using several figures of merit (see Section 4).

label associated with the ith patient, and $\hat{y}_i$ to denote the output generated (predicted) by the ML model at hand.

Working with data collected from the EHR is challenging since the observations come from different sources, often have outliers and require homogenization, especially when working with MTS [31,32]. For this reason, a pre-processing stage is required to guarantee consistent and reliable results. Towards that end, we followed a process of normalization, database integration, outliers cleaning, and window modeling. Further details on the pre-processing stage will be given in Section 3.

2.1.1. MTS windowing

As already pointed out, the length of the MTS (i.e., the number of columns of $\mathbf{X}_i$) can change with the index i. Since most ML models require inputs to have the same size across samples, we use a windowing technique to render the MTS length homogeneous across patients. Windowing requires setting a window length (denoted as W) and then, for every patient i, setting a time interval $[t_i^{ini}, t_i^{end}]$ with $t_i^{end} = t_i^{ini} + W - 1$. Note that the values of t_i^{ini} and t_i^{end} depend on the particular patient, since the data is asynchronous and our database was collected throughout several years.

The windowed input data for the ith patient is given by

$$\bar{\mathbf{X}}_i = \left[\mathbf{x}_i^{t_i^{ini}}, \ldots, \mathbf{x}_i^{t_i^{end}}\right] \in \mathbb{R}^{D\times W},$$

which, as desired, has the same size across patients.

For notational convenience, we will use $\bar{\mathbf{x}}_i^t \in \mathbb{R}^D$ with $t = 1, 2, \ldots, W$ to denote the tth column of $\bar{\mathbf{X}}_i$ and $\underline{\mathbf{x}}_i^d \in \mathbb{R}^W$ with $d = 1, 2, \ldots, D$ to denote the dth row of $\bar{\mathbf{X}}_i$. This way, the vector $\bar{\mathbf{x}}_i^t$ collects the D values of the features of the ith patient in the tth instant. Analogously, $\underline{\mathbf{x}}_i^d$ represents the time series of length W associated with the dth feature of patient i.

2.1.2. Mechanisms for FS

FS, oftentimes disregarded as a minor task, is essential in data-science pipelines. The elimination of input features that are extremely noisy or redundant is critical to enhance classification performance, avoid overfitting and boost generalization [33,34]. In addition, and equally important for clinical applications where a substantial amount of information is recorded (so that the value of D can be very high), FS provides a disciplined data-driven approach to identify the key features for the task at hand, providing insights on the problem, eliminating redundant features and contributing to enhancing the interpretability of models and results. Mathematically, FS for MTS amounts to designing a set $\mathcal{D}'$ with cardinality D' such that $\mathcal{D}' \subseteq \mathcal{D} = \{1, 2, \ldots, D\}$ and $D' < D$. The set $\mathcal{D}'$ contains the features to be kept and $\mathcal{D} \setminus \mathcal{D}'$ those to be eliminated; hence, the smaller the value of D', the more aggressive the selection mechanism is. We note that the value D' can be set beforehand or, alternatively, generated by the FS algorithm. Suppose now that the FS algorithm produces as output the set $\mathcal{D}' = \{d_1, d_2, \ldots, d_{D'}\}$ where, without loss of generality, we assume that $d_n < d_{n+1}$ so that the elements of $\mathcal{D}'$ are ordered. Leveraging $\mathcal{D}'$, we define the binary selection matrix $\mathbf{S}_{\mathcal{D}'} \in \{0, 1\}^{D'\times D}$ such that: (i) for every row, all the entries are zero except for a single one, and (ii) for the nth row, the entry whose value is one is that corresponding to the d_n-th column. That is, $[\mathbf{S}_{\mathcal{D}'}]_{n,d_n} = 1$ for $n = 1, 2, \ldots, D'$ and zero everywhere else. With this notation at hand, for each patient i, the original MTS input $\bar{\mathbf{X}}_i \in \mathbb{R}^{D\times W}$ is replaced with the reduced-dimensionality input $\mathbf{S}_{\mathcal{D}'}\bar{\mathbf{X}}_i \in \mathbb{R}^{D'\times W}$, where we emphasize that $\mathbf{S}_{\mathcal{D}'}$ is the same for all i.

Next, we discuss three sounded and widely-adopted FS methods and describe how those methods can deal with MTS. Our experiments will implement the three of them, analyzing their differences, and comparing their classification performance. Last but not least, rather than choosing the method with the best classification performance, the paper advocates a voting mechanism considering the three FS methods to enhance the classification results.

Confidence intervals with bootstrap

Bootstrap resampling is a non-parametric technique used to estimate the distribution of a statistic (e.g., the mean value) taking samples from a population without replacement [35]. Bootstrapping considers that the empirical and actual distributions are not too different, is appropriate when the actual distribution is unknown, and does not make any assumption related to the properties of the actual distribution function [36].

In our work, we use bootstrap resampling to assess if the value of a particular feature for the AMR population is significantly different from the value of the same feature in the non-AMR population performing a hypothesis test. More precisely, let $\mathcal{S}_{AMR}$ be the set (population) of AMR patients and $\mathcal{S}_{non-AMR}$ the set of non-AMR patients. The intermediate goal is to quantify the difference between μ_{AMR} (the mean value of a specific feature in the population $\mathcal{S}_{AMR}$) and $\mu_{non-AMR}$ (the mean of the same feature in the population $\mathcal{S}_{non-AMR}$) and assess if the difference $\Delta = \mu_{AMR} - \mu_{non-AMR}$ is relevant. Instead of computing Δ using all patients in $\mathcal{S}_{AMR}$ and $\mathcal{S}_{non-AMR}$ and comparing the (single and deterministic) number obtained to a pre-specified threshold, we implement a more statistically robust resampling bootstrap approach. Thus, we resample each of the populations R times, obtaining the sets $\{\mathcal{S}_{AMR}^{(r)}\}_{r=1}^R$ for AMR patients and $\{\mathcal{S}_{non-AMR}^{(r)}\}_{r=1}^R$ for non-AMR ones. As explained in the experimental sections, we use $R = 3,000$ and set the cardinality of the resampled sets to be the same (balancing the classes) and equal to 50% of the size of the minority class. Then, the mean statistic is computed across features and resamples, obtaining for each feature the values $\{\mu_{AMR}^{(r)}\}_{r=1}^R$ and $\{\mu_{non-AMR}^{(r)}\}_{r=1}^R$. Third, we obtain the difference between the statistic in both populations for each resample, generating $\Delta^{(r)} = \mu_{AMR}^{(r)} - \mu_{non-AMR}^{(r)}$ for $r = 1, 2, \ldots, R$. Fourth, we build the histogram of Δ and empirically calculate the 95% CI for Δ, denoted as CI_Δ. We consider that the null hypothesis H_0 (the feature being not relevant/informative) is true if $0 \in \text{CI}_\Delta$. In other words, if there is no statistically significant difference between the mean of the feature in the two considered populations, the feature is not selected. In contrast, the alternate hypothesis H_1 (the feature being relevant) is considered true if $0 \notin \text{CI}_\Delta$, indicating that a significant difference between the mean of both populations exists and, as a result, the feature is added to the set $\mathcal{D}'$.

The bootstrapping-based FS process described above assumes that the features are one-dimensional scalars. However, in an MTS environment, the problem to solve is, given the patient-data

matrices $\{\bar{\mathbf{X}}_i\}_{i=1}^I$ and focusing on a particular feature (say the dth one), whether to keep or remove the dth row of the data matrices for all the patients in the dataset. In other words, for each and every $d = 1, \ldots, D$, we need to decide if the W-dimensional vectors $\{\underline{\mathbf{x}}_i^d\}_{i=1}^I$ are selected to be part of the inputs provided to our ML architectures. In this work, we have computed $\Delta^{(r)} = \|\mu_{AMR}^{(r)} - \mu_{non-AMR}^{(r)}\|$ for each of the W entries of the vector $\underline{\mathbf{x}}_i^d$, assessing the relevance of each of the W entries separately and, then, implementing a majority-rule scheme where the dth feature is selected if more than half of the values within the window were deemed relevant.

Conditional mutual information

Mutual Information (MI) is directly related to the well-known and widely used Shannon entropy [37]. The Shannon entropy of a generic random variable X, which is denoted as $\mathbb{H}(X)$, is an information metric related to the probability of occurrence of the values of X [38]. A high value of entropy means that every event in X has the same probability of occurrence, while a low value means that the probability of occurrence of each event is different. With x denoting all the values the (discrete) random variable X can take, the entropy of X is defined as $\mathbb{H}(X) = -\sum_{x\in\mathcal{X}} p(x)log(p(x))$, where $p(x)$ is $Pr\{X = x\}$. If another variable Y is considered, the joint entropy can be computed as $\mathbb{H}(X, Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)log(p(x, y))$, with $p(x, y) = Pr\{X = x, Y = y\}$. We can define the conditional entropy as

$$\mathbb{H}(X|Y) = -\sum_{x\in\mathcal{X}}\sum_{y\in\mathcal{Y}} p(x, y)log(p(x|y)), \tag{1}$$

with $p(y|x) = Pr\{Y = y|X = x\} = Pr\{X = x, Y = y\}/Pr\{X = x\}$. The mutual information between X and Y measures the shared information between both variables, and is expressed as

$$\mathbb{I}(X, Y) = \mathbb{H}(X) - \mathbb{H}(X|Y) = \mathbb{H}(Y) - \mathbb{H}(Y|X) = \mathbb{I}(Y, X). \tag{2}$$

In other words, the MI is the amount of information that variable X has about variable Y. Lastly, the conditional MI is the expected value of the MI of two random variables given the value of a third [39,40]. The conditional MI can be defined as

$$\mathbb{I}(X, Y|Z) = \mathbb{H}(X, Z) - \mathbb{H}(Y|Z) - \mathbb{H}(X, Y, Z) - \mathbb{I}(Z). \tag{3}$$

When using MI for FS, the goal is to select the set $\mathcal{D}' \subseteq \{1, 2, \ldots, D\}$ of D' features that maximizes the MI between the reduced input $\mathbf{S}_{\mathcal{D}'}\bar{\mathbf{X}}$ and the associated label y. Such an optimization is NP-hard and, hence, suboptimal schemes must be adopted. The approach in this paper is to use a greedy-selection scheme that chooses the features in $\mathcal{D}'$ one-by-one using an iterative scalar optimization of the MI metric. From an algorithmic point of view, this entails initializing $\mathcal{D}_{sel}^{[0]} = \emptyset$ and $\mathcal{D}_{non-sel}^{[0]} = \mathcal{D}$, and running the following D' steps, with j denoting the iteration index and starting with $j = 0$:

1. Evaluate $\mathbb{I}\left(y, \underline{\mathbf{x}}^d \left| \{\underline{\mathbf{x}}^{d'}\}_{d'\in\mathcal{D}_{sel}^{[j]}}\right.\right)$ for all $d \in \mathcal{D}_{non-sel}^{[j]}$.
2. Select the feature $d_*^{[j]}$ with the highest MI and update the sets $\mathcal{D}_{sel}^{[j+1]} = \mathcal{D}_{sel}^{[j]} \bigcup \{d_*^{[j]}\}$ and $\mathcal{D}_{non-sel}^{[j+1]} = \mathcal{D}_{non-sel}^{[j]} \setminus \{d_*^{[j]}\}$, accordingly.
3. Set $j = j + 1$. If $j = D'$, stop and return $\mathcal{D}' = \mathcal{D}_{sel}^{[j]}$. If not, go to step 1.

The approach to estimating $\mathbb{I}\left(y, \underline{\mathbf{x}}^d \left| \{\underline{\mathbf{x}}^{d'}\}_{d'\in\mathcal{D}_{sel}^{[j]}}\right.\right)$ for our MTS dataset requires simply considering that the output is binary (so that $\mathcal{Y} = \{0, 1\}$), that the inputs are W-dimensional vectors (so that $\mathcal{X}$ is the Cartesian product of the value sets for each of the W entries), and that the probabilities need to be estimated using the population sets ($\mathcal{S}_{AMR}$ for $y = 1$) and ($\mathcal{S}_{non-AMR}$ for $y = 0$).

Group LASSO

LASSO stands for Least Absolute Shrinkage and Selection Operator and it is a popular regularization and FS selection method [41]. The three main advantages of LASSO are: (i) its ability to search for the best set of features jointly, without the need to resort to a greedy algorithm; (ii) a sound theoretical motivation; and (iii) the existence of computationally efficient algorithms [42]. The LASSO is formulated as an optimization problem, and it can be used both in regression and classification tasks. While the regression form is presented here for simplicity, the generalization to classification tasks is straightforward. Let us suppose that we have a set with I input–output pairs $\{(\mathbf{x}_i, y_i)\}_{i=1}^I$, with the output being a scalar and the input $\mathbf{x}_i$ having D dimensions. LASSO assumes that the predicted output $\hat{y}_i$ is estimated linearly as $\mathbf{x}_i^\top\boldsymbol{\alpha}$, where $\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_D]^\top$ is a vector with the D linear coefficients of the predictor. The optimal value of $\boldsymbol{\alpha}$ (denoted as $\boldsymbol{\alpha}_*$ is then obtained as

$$\min_{\boldsymbol{\alpha}\in\mathbb{R}^D} \frac{1}{2}\sum_{i=1}^I \left(y_i - \mathbf{x}_i^\top\boldsymbol{\alpha}\right)^2 + \lambda\sum_{d=1}^D |\alpha_d| \tag{4}$$

where $\|\boldsymbol{\alpha}\|_1 = \sum_{d=1}^D |\alpha_d|$ is the ℓ_1 norm of $\boldsymbol{\alpha}$, and $\lambda > 0$ is a regularization parameter. The objective combines a data fitting term with a regularizer that penalizes the coefficients of the regression variables, shrinking some of them to zero. The larger λ, the higher the number of coefficients α_{*d} that are set to zero. From an FS perspective, the approach is to solve the optimization for different values of λ, select the proper value of λ based on either the fitting error or the number of active coefficients and, then, construct the feature set $\mathcal{D}'$ with the indexes of the vector $\boldsymbol{\alpha}_*$ associated with the non-zero coefficients after the shrinking process.

Since in this work we deal with MTS, the input data are not vectors but matrices, and this calls for using a generalization of the LASSO method referred to as *Group LASSO* [42]. Intuitively speaking, group LASSO splits the input features into different groups and then either activates or sets to zero all the variables within each group. Mathematically, we are given $\{(\bar{\mathbf{X}}_i, y_i)\}_{i=1}^I$ and define the vector $\boldsymbol{\alpha}^d = [\alpha_1^d, \alpha_2^d, \ldots, \alpha_W^d]$ whose entries are associated with the W samples recorded for feature d. Since we have D of those vectors, the total number of coefficients to learn is DW, which coincides with the number of entries in the input $\bar{\mathbf{X}}_i$. Recalling that $\underline{\mathbf{x}}_i^d$ is a vector collecting the entries of the dth row of $\bar{\mathbf{X}}_i$, the optimal regularized linear regressor for the MTS case can be obtained as the solution to

$$\min_{\{\boldsymbol{\alpha}^d\in\mathbb{R}^W\}_{d=1}^D} \frac{1}{2}\sum_{i=1}^I \left(y_i - \sum_{d=1}^D (\underline{\mathbf{x}}_i^d)^\top\boldsymbol{\alpha}^d\right)^2 + \lambda\sum_{d=1}^D \|\boldsymbol{\alpha}^d\|_2, \tag{5}$$

where $\|\boldsymbol{\alpha}^d\|_2 = ((\alpha_1^d)^2 + \cdots + (\alpha_W^d)^2)^{1/2} \geq 0$ is the ℓ_2 norm of $\boldsymbol{\alpha}^d$. The optimization resembles that in Eq. (4), but accounting for the multidimensional nature of the input and replacing $|\alpha_d|$ with $\|\boldsymbol{\alpha}^d\|_2$. This way, if the optimal solution sets $\boldsymbol{\alpha}_*^d = [0, 0, \ldots, 0]^\top$, then the dth row of matrices $\{\bar{\mathbf{X}}_i\}_{i=1}^I$ is not selected.

2.1.3. Strategies to handle imbalance data

Most (binary) classification architectures are trained assuming that the number of samples in each class is approximately the same [43]. However, there are many real-world applications, specifically in the healthcare domain, where that is not the case. Thus, in the task tackled in this paper, the number of AMR patients is lower than the number of non-AMR patients. When learning is performed with unbalanced classes, models can be biased to the majority class and led to poor generalization performance [44].

There are different strategies to deal with imbalance classes [45], including data-level approaches or cost-sensitive methods. In this work, we focus on two simple but effective methods: (i) undersampling the majority class, and (ii) defining asymmetric misclassification costs. When following an undersampling strategy, samples from the majority class are randomly discarded until the number of elements in the majority and minority populations is similar. The cost function used in this work to train models applying undersampling is the Binary Cross-Entropy (BCE) cost.

In the cost-sensitive approach, errors in a sample from the minority class are penalized more heavily than those from the majority class. A simple way to achieve this is to use the Balanced Binary Cross-Entropy (BBCE) function, a modification of the well-known binary cross-entropy cost function [46]. More specifically, upon setting the value of the weight $\beta \in (0, 1)$, the BBCE cost is defined as

$$-\frac{1}{I'}\sum_{i=1}^{I'}\left(\beta y_i \log\left(\hat{y}_i\right) + (1-\beta)\,(1-y_i)\log\left(1-\hat{y}_i\right)\right) \tag{6}$$

where I' is the number of patients in the training set. If the training set is balanced, then $\beta = 0.5$, and Eq. (6) is the BCE cost function. When $y_i = 1$ is associated with the minority class, then β must be chosen closer to one. On the other hand, if $y_i = 0$ is the minority class, then β must be chosen closer to zero. Following this approach, the value of β in this paper has been set as the number of samples of the majority class divided by the number of total samples.

2.1.4. Strategies to handle missing values in MTS

Missing values, which affect most real-world datasets, are pervasive when dealing with time series. In the clinical context, data is recorded irregularly, with measurement frequency varying between patients and even over time. Moreover, the values are typically not missing at random but reflect the patient's health status or decisions by caregivers [47]. Equally important, when working with windowed data, there may be cases where the window length is larger than the length of the patient's record and, hence, one has to decide how to fill the initial (or last) part of the record.

Common approaches to deal with missing values include filling missing values with zeros, (linear) interpolation, or statistical imputation approaches [6]. Given that most of our data features are binary and partially inspired by the methodology proposed by Lipton et al. [48] for RNN-based prediction using missing values in clinical data, we consider three strategies to handle missing values in $\bar{\mathbf{X}}_i$:

1. Remove from the populations the patients with missing data ("Removing"). This (filtering) approach bypasses the problem altogether, but reduces the number of training samples, hence impacting generalization. As a result, it is more suitable in setups with an abundant number of training examples.
2. Impute with zeros the missing values, including those at the beginning of the window ("Zero Padding"). This is an extremely common approach, especially dealing with binary data where the 0 value represents the "by-default" state (e.g., absence of a medical condition or a drug not being prescribed to a patient).
3. Use advanced ML architectures able to apply a masking scheme that accounts explicitly for missing values ("Masking"). This strategy is suitable for the three RNN-based architectures (GRU, LSTM, and Bi-LSTM). We implement a modified version that, for each input sample, uses as an additional input a mask indicating the positions of the input vector with missing values [49].

2.2. NN architectures for MTS classification

Due to their ability to deal with discrete data and unveil complex non-linear dependencies, artificial NNs are ML approaches widely used to deal with classification tasks [50]. Therefore, to address our binary classification task (i.e., predicting if a patient is infected by an AMR bacteria in the ICU), we consider different NN architectures. We start with a simple MLP, which will serve as a baseline, and then describe three more sophisticated RNN-based deep architectures that are able to account for the sequential (time) structure present in our MTS.

2.2.1. Multi-layer perceptron

The MLP is a feed-forward NN formed by 3 types of layers: an input layer, one (or more) hidden layers, and an output layer. Each neuron in the hidden layer computes the output of a scalar non-linear function whose input is a linear combination of the outputs of the previous layer and some linear weights. The weights are tunable during the learning process, which is performed by optimizing a non-convex (data-fitting error) cost using stochastic gradient-based approaches [50]. MLPs are fully connected architectures (meaning that there exists a weight between any pair of neurons) and, as the number of neurons grows, they are universal approximators capable of implementing any non-linear mapping [50]. In this paper, we have set the number of hidden layers of the baseline MLP to one, considered the Leaky ReLU [51] as the scalar non-linear activation function, and used the Adam algorithm to optimize the cost function [52]. We implemented an early-stopping technique to avoid overfitting, choosing the learning rate as a hyperparameter. At every epoch, the early-stopping procedure evaluates the evolution of the data-fitting cost in the validation set (20% of the training set in this work) and stops the training if the cost deteriorates or stagnates. Also, a dropout rate has been used as a regularization technique to reduce overfitting to the training set.

It is important to emphasize that both the training cost and the optimization algorithmic approach (Adam with early stopping) described here for the MLP are also used for the NN architectures described in Sections 2.2.2 and 2.2.3.

2.2.2. GRU networks

RNNs are a type of NNs where the layers (i.e., the connections between the neurons) form a directed path along a sequence, rendering them suitable to deal with time series. Similar to linear filters, RNNs use an internal state to preserve an 'artificial memory' of the previous inputs [53]. However, standard RNNs exhibit problems when dealing with long MTS, due to the successive application of gradient steps that either decay or blow exponentially (see, e.g., for more details on the so-called vanishing gradient problem [53]).

In this context, GRUs are a gating mechanism aimed at avoiding the gradient's problems of RNNs [54]. A "gate" is an NN located between two consecutive elements of the sequence chain of an RNN whose purpose is to regulate the flow of information going along the sequence chain. Taking this into account, a gate can be used, e.g., to amplify a gradient that is vanishing, guaranteeing that the error goes through all the elements of the sequence. The GRU has two mechanisms to regulate the information: (i) the reset gate eliminates the information of previous time steps that is not incorporated into the hidden state (i.e., the input of the gate is the output of the previous state and the input associated with the current time step); and (ii) the update gate is in charge of generating the output of the neuron; deciding what information to throw away and what new information to add. GRU networks require fewer parameters than other RNNs and this is a desirable feature in clinical applications, where the number of samples is typically limited.

2.2.3. LSTM and Bi-LSTM networks

The LSTM network, another RNN-based architecture, takes the definition of a GRU one step further by considering a new mechanism to transfer information from previous time steps (the cell state) and a new gate (the output gate) [10]. The cell state provides the model with a memory of past events that is longer than that of the hidden state. To handle the states, an LSTM cell implements the three different gates: the forget gate, the input gate, and the output gate. The forget gate decides the information from the previous cell state that has to be deleted. Once the non-relevant information from the previous cell state has been eliminated, the input gate chooses the new information to store in the current cell state. Finally, the output gate layer is in charge of computing the final output of the neuron, which is a combination of the current cell state and the current input time step. While more sophisticated than their GRU counterpart, LSTMs have more parameters to learn and, as a result, require larger training datasets.

The last NNs considered in this work are Bi-LSTMs, which are MTS-processing architectures consisting of two LSTMs [11]. The first LSTM processes the MTS in a forward direction while the second one carries out the processing in a backward direction. As in classical smoothing methods for time-varying stochastic processes, the main benefit of the Bi-LSTM is the ability to leverage information from both the past and the future. While this additional ability tends to boost estimation performance, the number of parameters in Bi-LSTM models is larger and, as a result, performance gains must be expected only if the number of training samples is sufficiently high.

2.3. Model validation

This section presents the figure of merit considered for measuring the goodness of the generated models, both in terms of performance and interpretability. Performance concerns the ability of a model to make correct predictions, while interpretability concerns to what degree the model allows for human understanding. Models exhibiting the former property are many times more complex and opaque, while interpretable models may lack the necessary accuracy. The trade-off between accuracy and interpretability for predictive models is considered of paramount concern for pragmatic purposes.

2.3.1. Performance

Performance metrics evaluate the ability of a model to make correct predictions. Accuracy measures the ratio between the correctly classified samples and all the samples under consideration, and it is defined as follows:

$$Accuracy = \frac{TP + TN}{TP + FP + TN + FN} \tag{7}$$

where TP (True Positives) are samples labeled as AMR and correctly classified; TN (True Negatives) are samples labeled as non-AMR and correctly classified; FP (False Positives) are samples labeled as non-AMR but wrongly classified as AMR; and FN (True Negatives) are samples labeled as AMR but wrongly classified as non-AMR.

In addition, we have considered two complementary metrics used worldwide in the context of binary classification problems: Specificity and Sensitivity.

On the one hand, Specificity, also known as TN rate, measures the ratio of non-AMR patients correctly classified by the model as non-AMR. On the other hand, Sensitivity, also known as TP rate, measures the ratio of AMR patients actually classified as AMR.

$$Specificity = \frac{TN}{TN + FP} \qquad Sensitivity = \frac{TP}{TP + FN} \tag{8}$$

Finally, the Receiver Operating Characteristics (ROC) curve, and the Area Under such a Curve (AUC), measures the ability of the model under study to deal properly with both classes (AMR and non-AMR). Thus, ROC AUC provides additional details about how Specificity and Sensitivity interact.

2.3.2. Interpretability

Interpretability metrics evaluate the ability of a model to be understood by humans [55]. It is worthy to note that measuring interpretability is a challenging task that depends on the inherent transparency and complexity of the model (what is usually referred to as structural interpretability) but also depends on the background and expertise of the person who is expected to interpret such a model. Accordingly, there are neither a universal definition nor interpretability metrics universally recognized and used worldwide.

In practice, in the case of models that are deemed as interpretable by design, structural interpretability is measured in terms of complexity. For example, the number of parameters in a linear model, the number of nodes and leaves in decision trees, or the number of premises and rules in rule-based systems.

On the other hand, in the case of black-box models, such as neural networks, there are two main trends: (i) extrinsic evaluation of post-hoc interpretability; and (ii) intrinsic evaluation of interpretability of surrogate models.

In this paper, we evaluate the post-hoc interpretability of LSTMs with SHAP [28], which is inspired by Game Theory.[1] The so-called Shapley values assign a contribution $\phi_j(\mathbf{x}_j^t)$ to each feature $\mathbf{x}_j^t$. SHAP is a model-agnostic approach for generating local explanations as linear combinations of binary variables. All features are ranked in terms of their relevance for each single classification. SHAP is distributed as open source.[2]

In addition, we have used ExpliClas [56] and GUAJE [57] for building explainable fuzzy systems [58] that are inherently interpretable by design and ready to generate local (and global) factual (and counterfactual) explanations in natural language. Among the algorithms provided by ExpliClas for generating interpretable models, we have selected the C4.5 Quinlan's algorithm [59] and the Fuzzy Unordered Rule Induction Algorithm (FURIA) [60]. ExpliClas provides us with a linguistic approximation of models that can be exported into an XML format complying with the IEEE Std 1855–2016 for fuzzy systems modeling [61]. These linguistic models are endowed with global semantics thanks to the use of meaningful and common-sense linguistic terms that are defined by strong fuzzy partitions and grounded on clinical expert knowledge. Accordingly, the models satisfy all required constraints to be deemed as interpretable [58]. In addition, each model includes a list of readable IF-THEN fuzzy rules (e.g., "IF Feature$_j$ is Small AND Feature$_k$ is Big THEN class is AMR"). Such rules and their interaction can be analyzed in depth by clinicians with the assistance of GUAJE, which provides them with visual and textual explanations. Moreover, GUAJE offers several metrics for measuring the interpretability of the given linguistic models. In this work, we will report the number of rules and the total rule length, i.e., the total number of premises and conclusions in the rule base.

3. Dataset and pre-processing

This section describes the clinical data in detail and elaborates on the pre-processing techniques adopted.

[1] SHAP stands for SHapley Additive exPlanations.

[2] Open source software at https://github.com/slundberg/shap.

3.1. Dataset description

The data analyzed in this work corresponds to clinical MTS recorded for ICU patients at the University Hospital of Fuenlabrada in Madrid, Spain. Data were registered for 16 years, from 2004 to 2020 (both included), counting a total of 3,470 patients (627 of them were identified as AMR). For determining the AMR acquisition, the result of a clinical procedure named antibiogram is considered together with the patient culture to test if a bacterium is resistant to one or more antibiotics. Since getting the antibiogram result can take more than 48 h, and a single patient may have several cultures with multiresistant bacteria throughout his/her stay, we limit our research to the first culture identified as multiresistant. Moreover, given that our focus is on early prediction of AMR using clinical time series, we discarded two kinds of patients: (i) non-AMR patients with an ICU length stay shorter than 24 h, and (ii) AMR patients whose multiresistance was detected in the first 24 h in the ICU. Taking into account the previous considerations, our dataset is made up of 3,178 patients (433 with AMR).

The families of the antibiotics taken by the patients during their ICU stay are: Aminoglycosides (AMG), Antifungals (ATF), Carbapenemes (CAR), 1st generation Cephalosporins (CF1), 2nd generation Cephalosporins (CF2), 3rd generation Cephalosporins (CF3), 4th generation Cephalosporins (CF4), unclassified antibiotics (Others), Glycyclines (GCC), Glycopeptides (GLI), Lincosamides (LIN), Lipopeptides (LIP), Macrolides (MAC), Monobactamas (MON), Nitroimidazolics (NTI), Miscellaneous (OTR), Oxazolidinones (OXA), Broad-Spectrum Penicillins (PAP), Penicillins (PEN), Polypeptides (POL), Quinolones (QUI), Sulfamides (SUL), and Tetracyclines (TTC). We also use the feature "Others" to identify any other antibiotic not belonging to the previous list. For any given patient (say the ith one), the feature associated with each family of antibiotics (say the dth one) is a sequence of binary variables $\mathbf{x}_i^d \in \{0, 1\}^{T_i}$ indicating whether the patient has taken (or not) that family of antibiotics during each of the T_i 24-hour periods that the patient spent in the ICU. Regarding mechanical ventilation, it is represented as a sequence of binary variables, each of them denoting whether the patient has been connected (or not) to a breathing machine at any time during the 24-hour period at hand.

Furthermore, we characterize the ICU occupation and a summary of the treatment provided to the rest of the ICU patients (neighbors) during the same time interval as the one considered for the patient who is being characterized. Thus, a total of 17 additional numeric features were created: the number of neighbors of the patient, the number of patients identified with AMR bacteria (# of AMR neighbors), and the number of neighbors taking each of the 15 antibiotic families listed before. To avoid any confusion between the time series describing if the i-th patient took a particular drug and that describing the number of the neighbors of i taking the same drug, we use the subscript n to denote features referring to neighbors (e.g., *AMG* is the feature indicating if the patient took the drug and AMG_n is the feature counting how many of his/her neighbors took the drug).

For completeness, we detail next the clinical criteria considered to identify multi-drug resistance for the most common bacteria in the ICU at HUF: *Pseudomonas*, *Stenotrophomonas*, *Acinetobacter*, *Enterobacter*, *Acinetobacter*, *Staphylococcus Aureus*, and *Enterococcus*. In general, *Pseudomonas* were considered multidrug resistant when they were resistant to three or more of the following families: CF4, CAR, QUI, AMG, POL or PAP. *Staphylococcus aureus* was resistant to OXA; *Enterococcus* was resistant to vancomycin (GLI Family); whereas *Stenotrophomonas* and *Acinetobacter* were considered resistant only upon appearance, regardless of the antibiogram result.

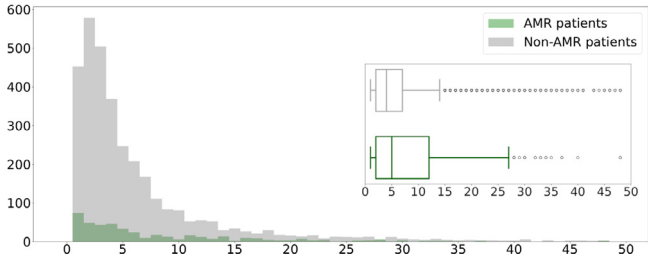


Fig. 2. Histogram and boxplots of the elapsed time (in days) from the ICU admission to the ICU discharge. Gray color is associated with non-AMR patients staying in the ICU for more than 24 h. The green color is used for AMR patients whose first culture flagged as positive occurs at least 24 h after their ICU admission.

3.2. Temporal windowing

We subsequently analyzed the temporal windowing for AMR and non-AMR patients. Towards that end, Fig. 2 shows the elapsed time from the ICU admission to: (i) the ICU discharge for non-AMR patients, and (ii) the first AMR culture for AMR patients. From these representations, we concluded that the identification of the first AMR usually occurs within the first few days of the AMR patients' stay. It can be observed that 50% of the AMR patients have the first culture flagged as positive before the fifth day after ICU admission. This value is very close to the median of the duration of the stay for non-AMR patients (4 days). Taking into consideration these values, when conducting the experiments we considered four different window lengths: $W = 3$, $W = 4$, $W = 5$, and $W = 6$ days.

To gain insights on the drugs administered during the duration of the windowing, we show in Fig. 3 the proportion (rate) of AMR patients taking each drug (green bars) and its counterpart for non-AMR patients (gray bars). Each rate has been computed over a different number of patients: 433 for AMR patients, and 2,745 for non-AMR ones.

Moreover, since four different windows are considered ($W \in \{3, 4, 5, 6\}$), four subplots are provided.

Note that these figures do not carry information about the temporal nature of each family of antibiotics, only their presence/absence ('1'/'0') during the window length. The results reveal that antibiotics like CAR, GLI, or ATF are administered in higher proportion to AMR patients, while PEN is more frequently administered to non-AMR patients. No significant differences are observed for QUI, LIP, and PAP.

For a more detailed explanation about the construction of the data-patient matrix in Eq. 2.1.1, Fig. 4 sketches the temporal windowing with $W = 5$ for two patients (patient i, who is AMR, and patient j, who is non-AMR). Each observation in the time series is defined by a 24-hour interval, with the starting time depending on the patient. More specifically, for the ith patient, we consider the last time instant of the window t_i^{end} to be associated with the day where the culture flagged as AMR was taken and, then, defining the $W - 1$ remaining days backwards.

For the jth patient and provided that his/her stay in the ICU was longer than W, the first slot of the window t_j^{ini} corresponds to the day the (non-AMR) patient was admitted to the ICU. In both cases, if the duration of the patient's stay is shorter than the window length (W), we set to zero the values associated with the first slots of the time series ("Zero Padding"). As already explained, temporal patient features contain information about the evolution of the patient during his/her stay in the ICU. Hence, the rows of the data matrix $\bar{\mathbf{X}}_i$ represent: a) the family of antibiotics taken by the patient during the time instants associated with the window, b) if the patient was under mechanical ventilation,

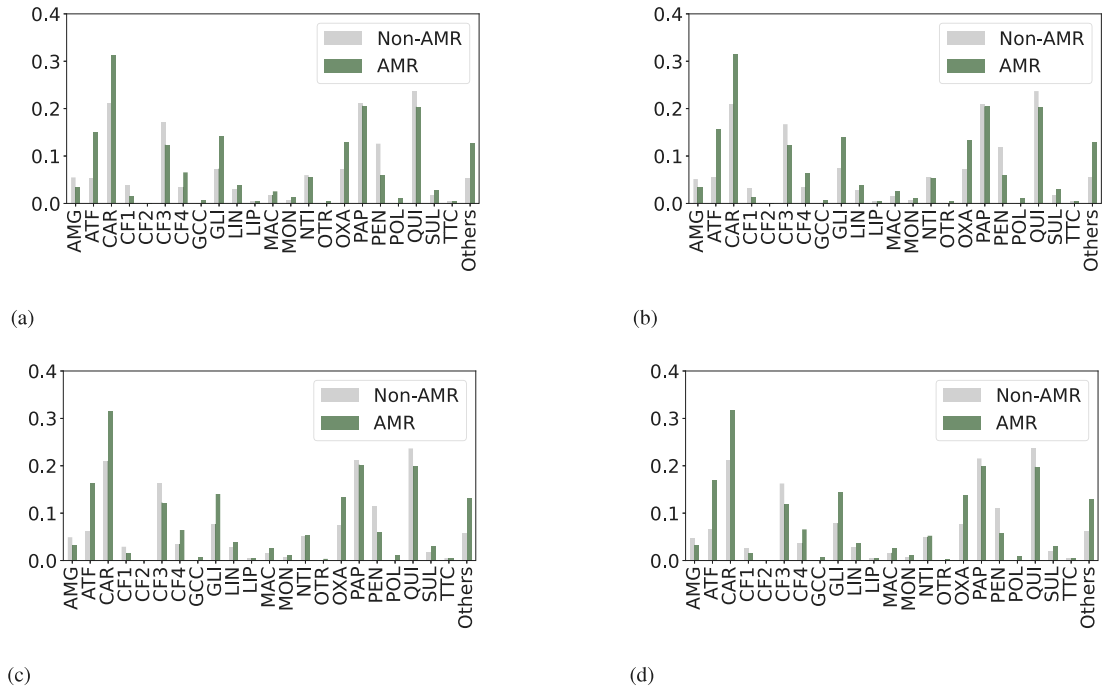


Fig. 3. Rate of AMR and non-AMR patients taking each family of antibiotics for different window lengths: (a) $W = 3$, (b) $W = 4$, (c) $W = 5$, (d) $W = 6$ days.

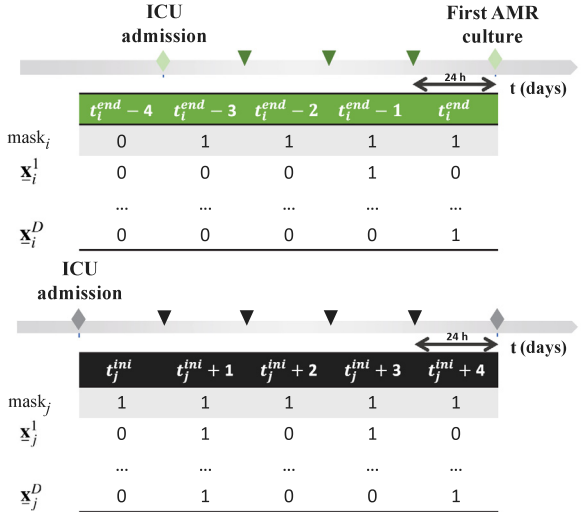


Fig. 4. Temporal feature matrix construction with a time window of 5 consecutive slots of 24 h. For the ith AMR patient (upper panel), t_i^{end} represents the time associated with the first AMR culture, whereas t_j^{ini} represents the admission time of the jth non-AMR patient (bottom panel).

c) the number of patients in the ICU during the time instant associated with the window, and d) the number of patients in the ICU taking each of the antibiotics. Finally, we also created a new W-dimensional binary variable called "mask" whose entries indicate if the patient was indeed present in the ICU during those W days. The default value for all the entries is "mask" = 1 and, as a result, if one of the entries of the "mask" vector (say the tth one) is zero, the meaning is that the patient was not present in the ICU that day. This readily implies that all the values in the corresponding tth column of $\bar{\mathbf{X}}_i$ will be zero, according to the (zero-padding) imputation procedure described before (cf. the left-most column of patient i in Fig. 4).

4. Experiments and results

This section starts by defining and discussing the experimental setup. We then present and discuss the FS results. After that, we analyze the prediction performance of the different ML models considered. Finally, we close the section by analyzing the interpretability of the generated models.

4.1. Experimental setup and parameter tuning

The dataset was randomly split into two independent subsets, the training set (70% of the samples) and the test set (30% of the samples). The training set was used to design the model, while its performance was evaluated with the test set. We have evaluated several strategies to deal with imbalance classes (undersampling and cost-sensitivity learning) and to handle missing values in MTS ("Removing", "Zero padding", "Masking").

Table 1 shows the total number of patients for each dataset. As expected, the number of patients changes in terms of W. In the "Removing" approach, we only consider patients who were in the ICU W days for non-AMR patients, or who stayed in the ICU at least W days before the first AMR. The number of patients in the training set decreases when W increases (354 for $W = 3$ and 234 for $W = 6$). With "Zero Padding" or "Masking" we discard those patients who did not take any drug and were not connected to a breathing machine during the window time under consideration. For this reason, the number of patients in the training set increases when the window length increases. For comparison purposes, the number of patients in the test set is the same for a specific window length regardless the procedure used for dealing with missing values. The size of the test set decreases as W increases (908 patients with $W = 3$; 773 with $W = 4$; 653 with $W = 5$; 531 with $W = 6$).

In the training set, a 5-fold cross-validation approach was followed to select the hyperparameters minimizing either the BCE or the BBCE cost function. The hyperparameters associated with the MLP, GRU, LSTM, and Bi-LSTM network architectures are the

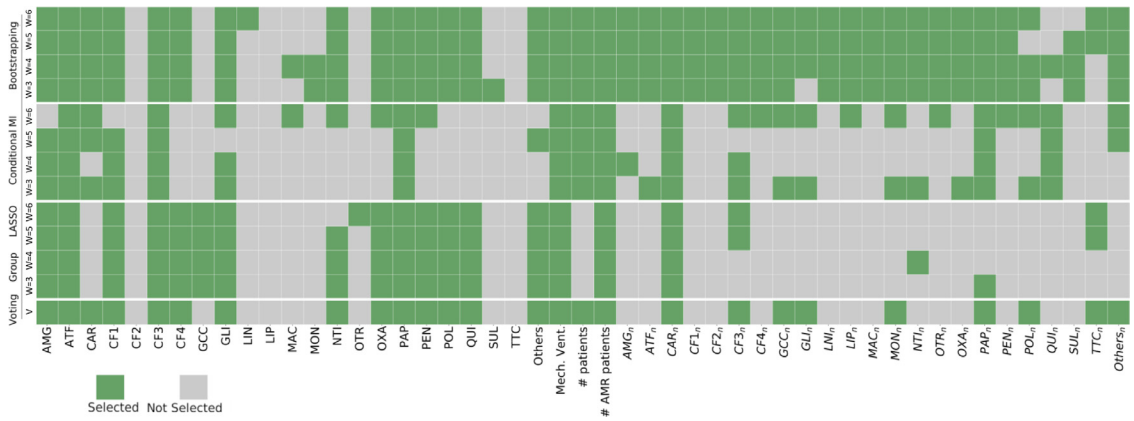


Fig. 5. Matrix of features (in columns) and FS approaches (Bootstrapping, Conditional MI and Group LASSO, detailed by window length W). The green cells represent the selected features whereas the gray cells represent the non-selected features.

Table 1
Total number of patients for specific training and test sets when considering different window lengths, several strategies to deal with imbalance classes (undersampling and cost-sensitivity learning) and to handle missing values in MTS ("Removing", "Zero Padding", "Masking").

Dataset	Strategies to handle imbalance data	Strategies to handle missing values in MTS	$W = 3$	$W = 4$	$W = 5$	$W = 6$
Training	Undersampling	Removing	354	319	269	234
		Zero Padding	428	448	453	447
		Masking	428	448	453	447
	BBCE	Removing	1470	1246	1008	836
		Zero Padding	1660	1687	1696	1704
		Masking	1660	1687	1696	1704
Test	-	-	908	773	653	531

learning rate {0.1, 0.01, 0.001, 0.0001}, the dropout rate {0.0, 0.1, 0.2, 0.3} and the number of neurons in the hidden layer. Since the dimension of the input is different for the MLP and RNN-based models, we explored the following number of neurons for the MLP {30, 35, 40, 45, 50, 55, 60, 70} and for the LSTM/GRU/Bi-LSTM {3, 5, 10, 15, 20, 25, 30, 35} architectures. Before training, each feature was normalized to have zero mean and unitary standard deviation [50].

4.2. FS results

We shift now our attention to Fig. 5, which indicates the features selected by each of the three methods presented in Section 2.1.2 for four different window lengths $W \in \{3, 4, 5, 6\}$. A green box means that feature was selected by the method and a gray box that it was not. A two-fold strategy was considered to obtain the final feature set $\mathcal{D}'$. Firstly, for each FS approach, we consider the dth feature as relevant if it was selected by two or more values of W. Secondly, we implemented another majority rule scheme where the dth feature was finally considered as relevant if it was selected by two or more FS methods.

Fig. 5 shows that the method using the CI obtained by bootstrap resampling selected a higher number of features (40 features out of 50) compared to Conditional MI or Group LASSO (19 features were selected for each approach). Note that Group LASSO selected a high number of variables related to the patient, whereas Conditional MI selected more features related to the ICU environment. It is also remarkable the stability of the Group LASSO results across the different time window lengths. After voting across methods, a total of 26 features were selected, being 14 of them associated with the antibiotics taken by the patients (AMG, ATF, CAR, CF1, CF3, CF4, GLI, NTI, OXA, PAP, PEN, POL, QUI, and Others), the MV, and 11 features associated with the ICU environment (# of patients, # of AMR patients, CAR_n, CF3_n, GCC_n, GLI_n, MON_n, PAP_n, POL_n, TTC_n and Others_n). Since feature importance is considered a way of endowing explainability to make an early prediction of AMR, we discuss in detail which ones are deemed clinically relevant to train the models.

All antibiotic families involved in the clinical criteria for the appearance of AMR germs (cf. the last paragraph in Section 3.1) were considered. Also, it is observed how some antibiotics were always identified as relevant, despite the window length and the FS method considered (among them, ATF, CF3, PAP, MV, # of AMR patients, and CAR_n). Clinicians have validated these results, concluding that they can be a suitable alternative for building appropriate data-driven models.

4.3. Early prediction of AMR using NNs

The purpose of this work is the early prediction of AMR with MTS recorded in the EHR before the actual complication occurs. Therefore, we pay attention first to MTS with $W = 5$, i.e., the median of the elapsed time from the ICU admission until the first AMR culture for AMR patients (see Fig. 2 for details). We compare the classification performance of conventional NNs (MLP) and RNN approaches (LSTM, GRU, and Bi-LSTM) using different methods for a) FS, b) handling class imbalance, and c) dealing with missing values.

Table 2 shows the mean and the standard deviation on 5 test partitions provided by different NNs models in terms of Accuracy, Specificity, Sensitivity, and ROC AUC. Note that, to keep the comparison fair, the same 5 test sets have been considered when evaluating all the methods. Several conclusions can be drawn from this table. In general, better performance is achieved when considering an FS strategy. For ease of comparison, the mean of the ROC AUC obtained for non-FS and FS results was computed (60.84 vs 62.09), verifying that FS offers better performance.

Table 2
Mean ± standard deviation of the performance (Accuracy, Specificity, Sensitivity, and ROC AUC) on 5 test partitions when training NNs considering a 5-days window with: non-FS and with FS (first row); undersampling and BBCE as strategies to handle class imbalance (second column); "Removing", "Zero Padding" and "Masking" strategies to handle irregular MTS (third column); and MLP, GRU, LSTM, and Bi-LSTM as classifiers (fourth column). The highest performance for each figure of merit is in bold.

Data Source	Strategies to handle imbalance	Strategies to handle missing values	Models	Accuracy	Specificity	Sensitivity	ROC AUC
Non-FS	Undersampling	Removing	MLP	64.15 ± 7.76	66.1 ± 11.08	53.1 ± 14.3	59.6 ± 3.52
			GRU	61.99 ± 3.99	62.91 ± 4.44	56.08 ± 4.21	59.50 ± 3.24
			LSTM	61.98 ± 4.32	62.92 ± 4.7	55.64 ± 11.28	59.28 ± 5.97
			Bi-LSTM	63.91 ± 6.28	65.65 ± 7.48	53.33 ± 6.86	59.49 ± 4.74
		Zero Padding	MLP	59.36 ± 2.26	59.15 ± 2.57	61.08 ± 4.5	60.11 ± 2.56
			GRU	59.74 ± 2.66	59.48 ± 3.71	61.1 ± 4.02	60.29 ± 0.75
			LSTM	59.14 ± 2.02	59.06 ± 3.18	59.84 ± 8.32	59.45 ± 3.14
			Bi-LSTM	57.99 ± 1.84	57.41 ± 2.19	61.73 ± 3.82	59.57 ± 2.16
		Masking	GRU	67.38 ± 2.59	68.91 ± 3.69	57.51 ± 7.01	63.21 ± 2.48
			LSTM	65.92 ± 1.79	67.38 ± 2.19	56.3 ± 1.98	61.84 ± 0.99
			Bi-LSTM	65.34 ± 2.74	66.92 ± 2.63	54.95 ± 3.06	60.94 ± 2.81
	BBCE	Removing	MLP	56.33 ± 6.22	54.0 ± 7.86	**71.52 ± 5.41**	62.76 ± 2.4
			GRU	57.78 ± 7.58	57.18 ± 10.33	62.42 ± 10.62	59.8 ± 2.07
			LSTM	55.54 ± 11.97	54.26 ± 14.95	65.12 ± 11.06	59.69 ± 3.27
			Bi-LSTM	55.38 ± 8.89	53.55 ± 11.45	68.75 ± 11.77	61.15 ± 1.95
		Zero Padding	MLP	55.47 ± 3.24	54.29 ± 3.15	63.57 ± 7.49	58.93 ± 4.66
			GRU	57.80 ± 4.58	56.12 ± 5.98	69.58 ± 6.4	62.85 ± 2.02
			LSTM	57.02 ± 3.58	55.71 ± 3.46	65.70 ± 4.74	60.70 ± 3.98
			Bi-LSTM	59.97 ± 7.31	59.57 ± 10.64	63.88 ± 14.82	61.73 ± 3.52
		Masking	GRU	67.03 ± 2.74	68.52 ± 3.92	57.51 ± 7.01	63.01 ± 2.35
			LSTM	60.86 ± 3.35	60.2 ± 4.12	65.67 ± 3.71	62.93 ± 1.60
			Bi-LSTM	59.52 ± 3.90	58.75 ± 5.33	65.01 ± 6.32	61.88 ± 1.49
FS	Undersampling	Removing	MLP	59.92 ± 2.97	60.19 ± 2.79	58.42 ± 5.79	59.31 ± 3.93
			GRU	60.32 ± 6.07	60.52 ± 6.66	59.16 ± 6.14	59.84 ± 5.03
			LSTM	64.24 ± 3.19	65.35 ± 3.71	57.12 ± 2.12	61.23 ± 1.95
			Bi-LSTM	60.9 ± 5.45	61.1 ± 6.65	59.17 ± 6.85	60.13 ± 3.91
		Zero Padding	MLP	63.11 ± 5.48	63.42 ± 6.9	62.14 ± 6.69	62.78 ± 2.55
			GRU	61.95 ± 2.88	62.26 ± 4.04	60.38 ± 6.59	61.32 ± 2.23
			LSTM	65.93 ± 1.71	66.64 ± 2.78	61.72 ± 7.32	64.18 ± 2.67
			Bi-LSTM	63.1 ± 5.38	63.36 ± 6.36	61.54 ± 4.89	62.45 ± 3.53
		Masking	GRU	64.08 ± 4.1	64.14 ± 5.85	64.16 ± 8.29	64.15 ± 1.65
			LSTM	**69.23 ± 2.28**	**70.79 ± 3.30**	59.41 ± 6.22	65.10 ± 2.18
			Bi-LSTM	68.62 ± 2.35	70.35 ± 2.69	57.18 ± 3.76	63.76 ± 1.99
	BBCE	Removing	MLP	57.91 ± 7.52	57.01 ± 9.54	65.34 ± 8.24	61.17 ± 2.85
			GRU	59.11 ± 4.37	57.58 ± 5.79	69.52 ± 5.81	63.55 ± 1.74
			LSTM	57.15 ± 6.06	55.75 ± 8.35	66.92 ± 9.91	61.34 ± 1.05
			Bi-LSTM	53.84 ± 11.01	51.33 ± 14.48	70.8 ± 12.48	61.07 ± 2.19
		Zero Padding	MLP	66.24 ± 2.32	66.89 ± 2.82	62.37 ± 5.27	64.63 ± 2.54
			GRU	58.01 ± 4.22	56.22 ± 5.13	69.68 ± 3.92	62.95 ± 2.56
			LSTM	60.81 ± 3.83	60.43 ± 5.01	63.45 ± 6.39	61.94 ± 2.29
			Bi-LSTM	55.59 ± 3.97	53.61 ± 4.93	69.19 ± 4.35	61.40 ± 1.27
		Masking	GRU	63.01 ± 2.93	61.95 ± 4.17	69.94 ± 5.75	65.95 ± 1.29
			LSTM	65.40 ± 3.94	64.88 ± 5.31	68.58 ± 6.43	**66.73 ± 1.80**
			Bi-LSTM	63.33 ± 2.47	62.98 ± 3.37	65.89 ± 4.04	64.44 ± 0.78

If we shift focus now to the strategies to handle imbalance data, we note that undersampling the training set works better than BBCE (the mean ROC AUC values are 63.95 and 62.30, respectively). However, in this work, the limited number of patients is a critical problem, and BBCE has the advantage of using a larger number of patients to train without neglecting the importance of the class imbalance problem. It can be seen also that "Masking" is the best approach of the three strategies to handle missing values in MTS, slightly outperforming the others approaches in all experiments in terms of ROC AUC (mean ROC AUC value for "Masking" is 63.66, whereas for "Removing" and "Zero Padding" is 60.56 and 61.58, respectively). The ML classifier with the best results (considering the ROC AUC as the most relevant figure of merit) is the LSTM scheme with BBCE and "Masking", achieving a ROC AUC level of 66.73%. It is also the best in terms of Sensitivity (68.58%), while for Accuracy (66.54%) and Specificity (64.88%) the best performance was achieved when BBCE was replaced with undersampling.

Since, in general, better results are obtained with LSTM, for completeness, we compare the obtained results with those provided with different windowed modelings for the GRU and Bi-LSTM models. Fig. 6 shows the boxplots of the performance on five test partitions in terms of Specificity, Sensitivity, and ROC AUC when considering the selected features after a voting strategy and training with a BBCE cost. Furthermore, we explore strategies to handle irregular MTS. As previously discussed, the number of patients changes with the length of the window W. Therefore, though the comparison of results among different windows does not allow us to conclude which the best window length is, it allows us to know which one works best for this particular problem. The four and five-day windows obtained better performance than three-day and six-day windows (see Fig. 6). Also, we can conclude that models based on GRU and LSTM perform slightly better than Bi-LSTM based models. The underperformance of the Bi-LSTM may be due either because the architecture is to complex or because joint consideration of past and future data is not relevant to our classification problem.

4.4. LSTM post-hoc interpretability

In the previous section, we concluded that the LSTM architecture with 26 input features, $W = 5$ and "Masking" provided good performance for both undersampling and BBCE. However,

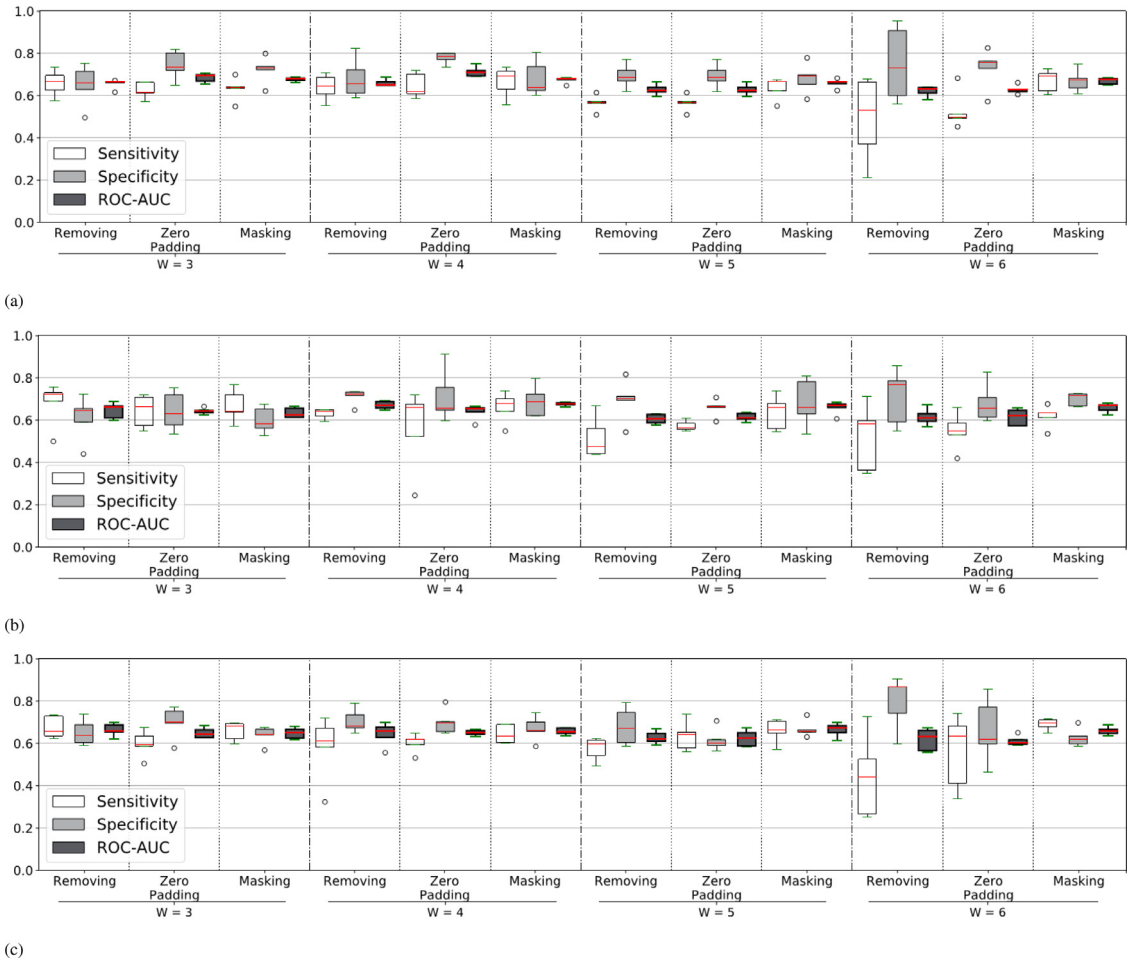


Fig. 6. Boxplot of the performance on 5 test partitions in terms of Specificity, Sensitivity and ROC AUC when considering FS, BBCE as strategy to handle imbalance data, different window lengths ($W = 3$, $W = 4$, $W = 5$, and $W = 6$), and different MTS classifiers: (a) GRU; (b) LSTM; and (c) Bi-LSTM.

LSTM networks are not easy to interpret. Therefore, we present here the results of an LSTM post-hoc interpretability analysis based on SHAP (see Section 2). We have explored the potential of SHAP to characterize the entire population (all given patients) according to the model prediction. The first SHAP analysis was carried out for the LSTM built with $W = 5$, undersampling and "Masking". Then, we paid attention to the behavior of individual patients when considering LSTM models with different window lengths ($W = 3$, $W = 4$, $W = 5$, $W = 6$), undersampling and "Masking". We calculated the Shapley values related to the contributions of all time steps for each single patient separately and then computed their average.

Fig. 7 shows a SHAP graph with the distribution of the Shapley values generated from the LSTM model trained with undersampling and "Masking" (considering all the 26 previously selected features). Features are depicted in order of relevance. Each dot represents a patient, the dot color indicates the real value of the feature, and the position of the dot in the x-axis represents the contribution this feature has to the model output (sum of the Shapley values). The further a point deviates from the mean of predictions (which is 0 in this case), the more impact this particular feature has on the model output for that patient. For example, the Shapley values associated with *Mech.Vent*. are positive when the *Mech.Vent*. value is high. That is, for our prediction model, by SHAP interpretation, we get the result that AMR patients are more likely to appear when ICU patients are receiving *Mech.Vent*. Accordingly, in short, the five most important features are *Mech.Vent*., ATF, CF1, # of AMR patients and GLI_n. These results are in agreement with clinicians' intuition and fit well with the literature. Notice that, controlling the isolation of patients with AMR germs and invasive devices are crucial tasks in tackling multi-resistance. Moreover, it is well known that the use of drugs such as CF1 is likely to reduce the chance of patients to become AMR. Interestingly, this fact is pointed out by our SHAP analysis.

Fig. 8 shows the model output values and the Shapley values for four different patients. The selected patients represent the four main types of patients we have in our database: AMR and non-AMR patients with stays longer than 5 days ("full data") and with stays shorter than 5 days ("no full data"). Once again, our SHAP analysis pays attention to LSTM models that were trained with undersampling and "Masking" (when considering different time windows and all the 26 features previously selected). Features are ranked and depicted in terms of relevance, with the one in the top being the most relevant feature. Gray vertical lines represent the base values associated with the underlying SHAP models. Each colored line corresponds to one specific patient. It is easy to observe how the contribution of each single feature to the model prediction varies from one patient to another. All contributions together with the baseline values form the final model outputs (see the top bar in each panel).

When jointly analyzing the four panels (each one associated with a different window length), even though slight differences exist, the following features emerge as the most relevant ones: # of AMR patients, *Mech.Vent*. and some drugs such as ATF, AMG and OXA. We observe that all models, except for $W = 4$, correctly classify (model output greater than 0.5) the AMR patient full data, whereas the AMR patient with no full data is correctly classified

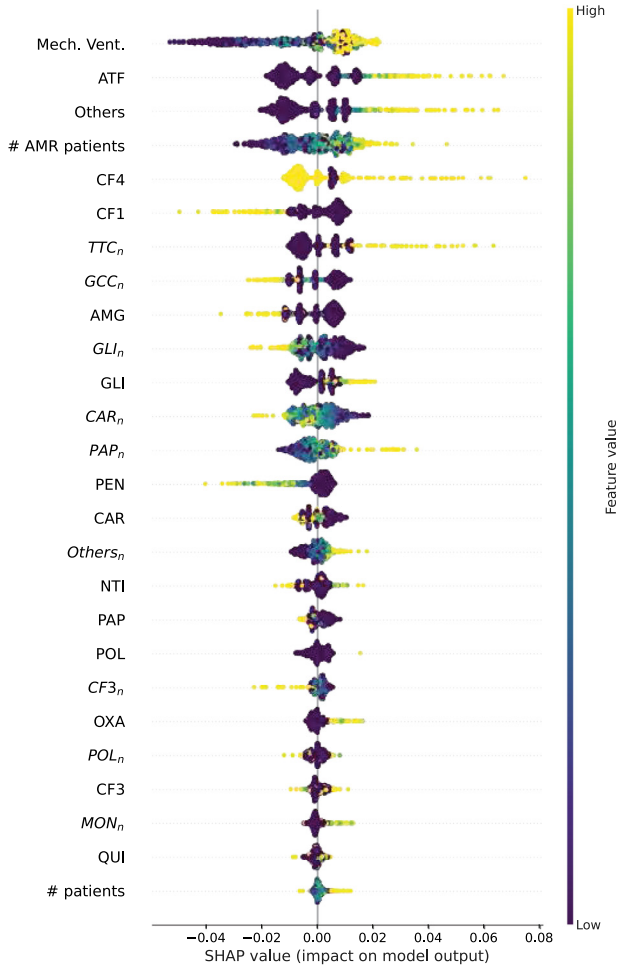


Fig. 7. Distribution of Shapley values generated from the LSTM model with undersampling and "Masking" with the 26 features selected by FS.

only by models with $W = 5$ and $W = 6$. A similar effect occurs with the non-AMR patients: the patient non-full data is correctly classified (model output lower than 0.5) by all models except for the model with $W = 6$, while the patient with full data is only correctly classified by models with $W = 3$ and $W = 4$. These findings illustrate that selecting the right window length is a very challenging task.

4.5. Linguistic interpretability

With the aim of providing readers with a quantitative assessment of the balance between interpretability and performance in our proposal, we built two interpretable by design models (a decision tree generated with the C4.5 algorithm [59] and a fuzzy rule-based classifier with the FURIA algorithm [60]) for the case of FS, undersampling and "Zero Padding" with W=5, that according to clinicians was the simplest to explain among all previously reported (see Table 2). Both models are enriched with linguistic interpretability, i.e., the original tree (C4.5) and rule list (FURIA) are translated into two explainable fuzzy systems as described in Section 2.3.2. Both linguistic models share the same global semantics, which is expressed by the same linguistic terms (*Very small*, *Small*, *Medium*, *Big*, *Very big*) associated with the same underlying strong fuzzy partitions (which were carefully defined in agreement with a clinician for each single feature to be meaningful). As a result, the knowledge embedded in this kind of model is described by a list of linguistic IF-THEN rules easy to understand by humans.

Table 3
Mean ± standard deviation of the performance (Accuracy and ROC AUC) and interpretability metrics (NR and TRL) on 5 test partitions when training the linguistic models with FS, undersampling and "Zero Padding" with W = 5. NR is the number of leaves (in decision trees) and the number of rules (in FURIA). TRL stands for total rule length. TRL counts all nodes in a tree and all premises and conclusions in FURIA. Values reported for MLP come from Table 2 and are included here only for facilitating comparison.

Model	Accuracy	ROC AUC	NR	TRL
C4.5	56.18 ± 2.96	55.38 ± 0.99	76.8 ± 5.9	772 ± 57
FURIA	52.77 ± 3.37	56.76 ± 2.31	8.8 ± 3.03	32.8 ± 18.3
MLP	63.11 ± 5.48	62.78 ± 2.55	–	–

Table 3 quantifies the structural interpretability of the generated models in terms of their number of rules (NR) and total rule length (TRL). These models were generated and validated with the same training and test datasets that were considered in Table 2. Nevertheless, for the sake of explainability, the temporal information was first aggregated to produce meaningful features. For example, "# of AMR patients_std" is the standard deviation (std) of the number of patients (# of AMR patients) in the considered time window; GLI_n_mean is the average of all temporal values associated with GLI_n corresponding to antibiotics of the family Glycopeptides (GLI) taken by the neighbors of the patient; or GLI_cons is '1' if the patient took GLI at any time instant and '0' otherwise.

For comparison purposes, we also report performance values (Accuracy and ROC AUC) that can be compared to those reported previously in Table 2. The results reveal that the highest interpretability of FURIA (NR=8.8 and TRL=32.8) comes with a reduction of performance. However, it is worth noting that comparing FURIA to C4.5, we observe how the interpretability gain is much larger than the reduction of performance. We carried out the Friedman Aligned Ranks non-parametric statistical test [62] (all versus all with significance level $\alpha = 0.05$) in order to detect significant differences among reported results for Accuracy in Table 3. The hypothesis H_0 (the means of the results of two or more algorithms are the same) is rejected when comparing FURIA versus MLP, and accepted in the rest of comparisons. Moreover, as illustrated below, FURIA rules are fairly simple and easy to interpret by clinicians:

R_1: **IF** ATF_cons is Small AND # of AMR patients_std is Very small **THEN** output is non-AMR
R_2: **IF** ATF_cons is Small AND # of AMR patients_std is Small **THEN** output is non-AMR
R_3: **IF** # of AMR patients_std is Big **THEN** output is AMR
R_4: **IF** ATF_cons is Big **THEN** output is AMR
R_5: **IF** # of AMR patients_std is Big AND GLI_cons is Big **THEN** output is AMR
R_6: **IF** # of AMR patients_std is Big AND Others_cons is Big **THEN** output is AMR
R_7: **IF** CF4_cons is Big AND $CF3_n$_mean is Small AND GLI_n_mean is Small **THEN** output is AMR

These rules provide clinicians with useful information that is complementary to that observed in Fig. 7. Interestingly, the 7 rules describe the interaction among only 7 out of the 26 features in Fig. 7. In addition, four of these features are in the top-5 ranking given by SHAP. Notice that, even if *Mech.Vent.* was identified by SHAP as the most relevant feature when considering single contributions, our linguistic analysis reveals that the interaction of the next four top features (ATF, Others, # of AMR patients and CF4) is also very relevant. Moreover, remind that ATF and # of

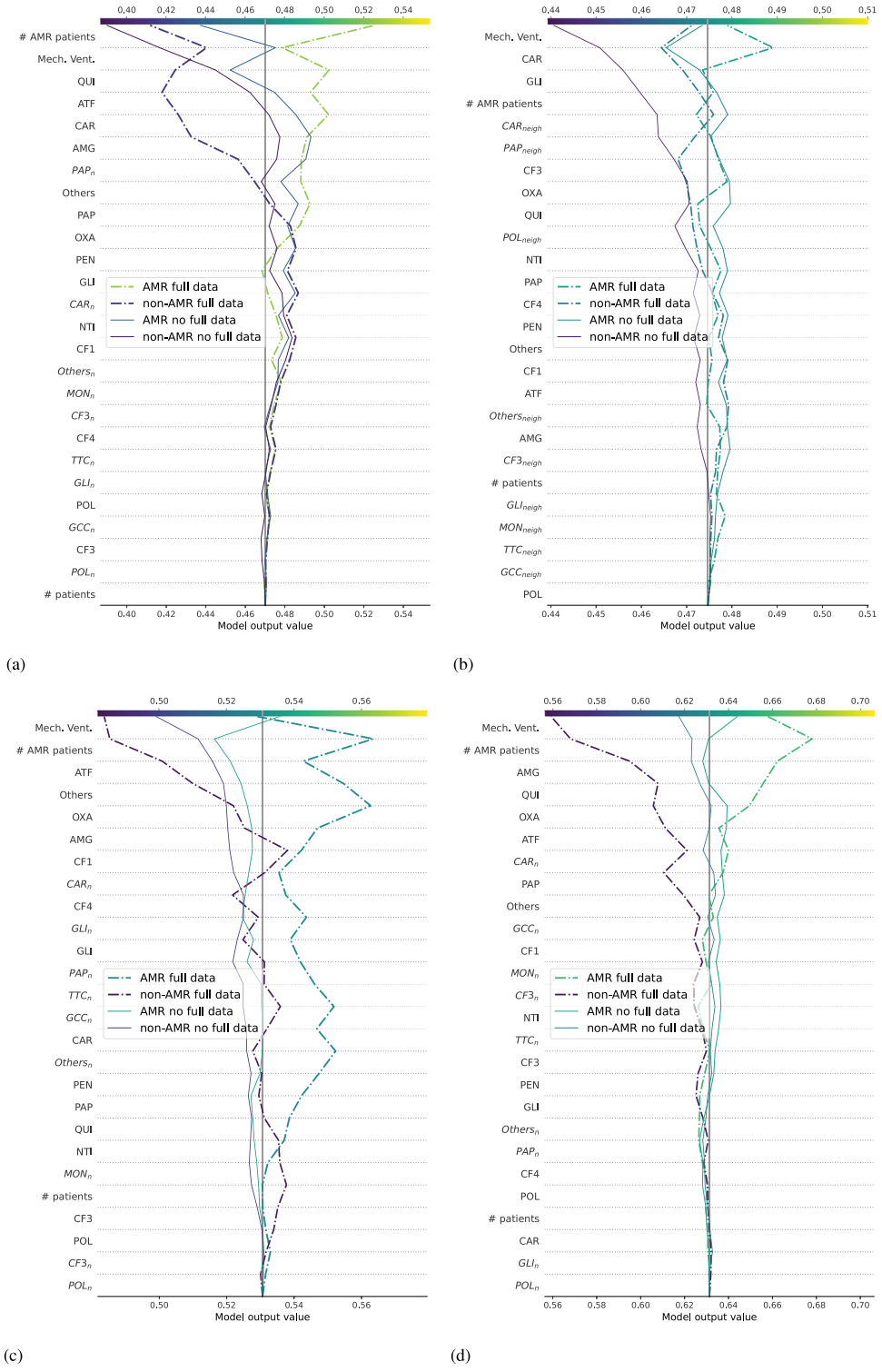


Fig. 8. Model output values and Shapley values associated with the LSTM model trained with undersampling and "Masking", with the 26 selected features, and different window length: (a) $W = 3$; (b) $W = 4$; (c) $W = 5$; (d) $W = 6$. The gray vertical line represents the base value of the SHAP models, and each colored line corresponds to one patient. Features are ranked and depicted in terms of relevance, being the top one the most relevant one. "Full data" represents patients' stays longer than the corresponding window (i.e., $T_i > W$), whereas "no full data" represents cases where $T_i < W$.

AMR patients were also previously pointed out as two of the most relevant features by all FS methods (see Section 4.2).

In addition, when considering single cases like those illustrated in Fig. 8, only specific rules are fired. For example, in the case of the patient with "AMR no full data" in Fig. 8(c), rule R_3 is fired and we obtain the following explanation, which is a mixture of factual and counterfactual pieces of information: *The patient is classified as AMR. In accordance with the third rule, a patient is AMR in case that the standard deviation of the number of AMR patients is big. It would be non-AMR if such standard deviation were smaller (0.345).* On the other hand, in the case of the patient with "non-AMR full data", R_1 is fired and the related textual explanation is as follows: *The patient is classified as non-AMR. It is very likely non-AMR, because in accordance with the first rule, a patient is*

non-AMR when the standard deviation of the number of AMR patients is very small and the consumption of ATF is small. It would be AMR if such consumption were bigger (1.430). It is worth noting that this kind of explanations highlights the interaction among different features, being a useful insight that complements the ranking of relevance given by SHAP.

5. Discussion and conclusions

The high rate of infections occurring in the ICU (20%–30% of all ICU admissions) [63] makes this unit one of the epicenters of the development of AMR. Previous clinical studies have analyzed the risk factors for getting AMR bacteria [64]. They concluded that the treatment with invasive devices (particularly the intensity and duration of the treatment) and the exposure to antibiotics are the principal causes. Minimizing the occurrence of AMR bacteria could be beneficial to reduce the duration of invasive devices treatment as well as to optimize its dosage [65].

Thus, AMR is nowadays a growing problem due to the inappropriate use of antimicrobials. Indeed, some bacteria that were previously treatable have become now a challenge to deal with, especially in the ICUs. In these units, AMR has created a high impact on morbidity, hospital costs, and sometimes patient survival. It is necessary to be aware of the growing problem caused by AMR, for which new research, efforts, and approaches are needed to prevent the further spread of AMR.

AI models can contribute to solving problems related to the clinical environment. These models reduce the time of detection of infectious diseases, resulting in a reduction in the number of deaths as well as health economic costs. In the healthcare domain, there has been a developing interest in breaking down clinical information as time-series sequences since it allows clinical specialists to better evaluate the progression of the patients. However, the complexity and irregular patterns present in clinical data render modeling MTS a hard and challenging task. Fortunately, RNNs have arisen as an appropriate choice to model and deal with MTS. In this work, we have explored the use of well-known RNNs such as GRU, LSTM, and Bi-LSTM. Although RNNs have demonstrated to achieve high classification performance, their lack of interpretability is a bottleneck for developing and deploying clinical MTS-based decision support systems where interpretability is of foremost concern. Note that, for the sake of interpretability, RNNs were trained using just data within short time windows. This was carried out in agreement with clinicians, who considered that the use of longer windows was harder to justify from a clinical point of view.

This work has paved the way towards comprehensible MTS analysis in the context of early AMR prediction in ICUs. We have applied different FS approaches, in combination with interpretable ML techniques, with the aim of extracting valuable insights about AMR. Our proposal has been validated with real-world data. Namely, we considered 3,178 patients, with 433 of them confirmed as AMR from 2004 to 2020 at University Hospital of Fuenlabrada in Madrid, Spain. With our study, we identified relevant clinical features for the onset of AMR which were afterwards confirmed by clinicians. For example, our proposal revealed how the family of antibiotics taken by patients as well as the time each patient has been assisted with mechanical ventilation, turn up as vital indicators to isolate a patient in advance and thus controlling the spread of the bacteria among other ICU patients.

It is worth noting that we have shown how findings provided by post-hoc interpretability analysis of data-driven models may be supportive to clinical decisions before the antibiogram result. More precisely, we used SHAP to assist clinicians in understanding the outputs given by black-box models. The SHAP results have shown the importance of mechanical ventilation for the predictions, which is in accordance with the literature. The importance of ATF is also noteworthy, with results showing that AMR patients took ATF more frequently than non-AMR patients. The # of AMR neighbors is relevant in our results: the higher the # of AMR neighbors, the higher the AMR probability returned by the models. In addition, we built explainable fuzzy systems to better understand how relevant and meaningful features previously identified with SHAP actually interact. As a result, we extracted linguistic IF-THEN rules that described how early AMR prediction can be explained in terms of the interaction among several features. Moreover, such rules were automatically interpreted and translated into narrative explanations in natural language to facilitate understanding by clinicians. All in all, clinicians were satisfied with the reported results and expressed how their trust in MTS-based AMR results was higher once they understood the model output with the assistance of both visual and textual explanations.

This novel methodology can save valuable time to start the adequate treatment for an ICU patient. This study was conducted using only MTS related to the antibiotics taken by the patients and the mechanical ventilation. To generalize the conclusions, different MTS should be considered, as well as demographic and clinical data such as age, gender, or diagnoses and procedures. As future research, we plan to endow with other interpretable NNs that take into account the importance of each time step, such as attentional NNs [66] or GRU-D [49]. Interpretability is expected to make those models more trustful and acceptable by clinicians. Finally, we plan to develop individual models for predicting each type of AMR bacteria emergence, since previous studies [67] have reported promising results when using this line of work.

CRediT authorship contribution statement

Sergio Martínez-Agüero: Data curation, software, Writing – original draft. **Cristina Soguero-Ruiz:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Jose M. Alonso-Moral:** Methodology, Writing – original draft, Writing – review & editing. **Inmaculada Mora-Jiménez:** Conceptualization, Methodology, Supervision, Writing – original draft, Writing – review & editing. **Joaquín Álvarez-Rodríguez:** Investigation, Validation, Writing – original draft. **Antonio G. Marques:** Conceptualization, Methodology, Supervision, Writing – original draft , Writing – review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Availability of data and materials

Access to the data can be provided upon official request if approved by the Committee of Ethics of the University Hospital of Fuenlabrada.

Acknowledgments

This work is supported by the Spanish NSF grants PID2019-106623RB-C41 (BigTheory), PID2019-105032GB-I00 (SPGraph), PID2019-107768RA-I00 (AAVis-BMR), RTI2018-099646-B-I00 (ADHERE-U); the Galician Ministry of Education, University and Professional Training grants ED431F 2018/02 (eXplica-IA) and ED431G2019/04; the *Instituto de Salud Carlos III*, Spain grant

DTS17/00158; as well as the Community of Madrid in the framework of the Multiannual Agreement with Rey Juan Carlos University in line of action 1, "Encouragement of Young Phd students investigation" Project Ref. F661 (Mapping-UCI). Sergio M. Aguero is a recipient of the Predoctoral Contracts for Trainees URJC Grant (PREDOC21-036). Jose M. Alonso-Moral is a *Ramon y Cajal* Researcher (RYC-2016-19802).

References

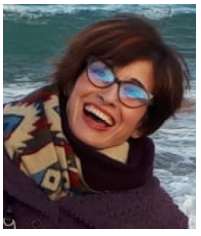
[1] A.A. Funkner, A.N. Yakovlev, S.V. Kovalchuk, Data-driven modeling of clinical pathways using electronic health records, Procedia Comput. Sci. 121 (2017) 835–842.
[2] M. Ghassemi, et al., A multivariate timeseries modeling approach to severity of illness assessment and forecasting in ICU with sparse, heterogeneous clinical data, in: 29th AAAI Conference on Artificial Intelligence, 2015, pp. 446–453.
[3] N.P. Tatonetti, P.Y. Patrick, R. Daneshjou, R.B. Altman, Data-driven prediction of drug effects and interactions, Sci. Transl. Med. 4 (2012) 125ra31.
[4] S.J. Taylor, Modelling Financial Time Series, World scientific, 2008.
[5] C. Soguero-Ruiz, et al., Predicting colorectal surgical complications using heterogeneous clinical data and kernel methods, J. Biomed. Inform. 61 (2016) 87–96.
[6] C. Soguero-Ruiz, et al., Data-driven temporal prediction of surgical site infection in: AMIA Annual Symposium Proceedings, 2015, p. 1164.
[7] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.
[8] T.A. Lasko, J. Denny, M.A. Levy, Computational phenotype discovery using unsupervised feature learning over noisy, sparse, and irregular clinical data, PLoS One 8 (6) (2013) e66341.
[9] K. Cho, et al., Learning phrase representations using RNN encoder–decoder for statistical machine translation, in: Conference on Empirical Methods in Natural Language Processing, 2014.
[10] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput. 9 (8) (1997) 1735–1780.
[11] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Trans. Signal Process. 45 (11) (1997) 2673–2681.
[12] Z. Lipton, D. Kale, C. Elkan, R. Wetzel, Learning to diagnose with LSTM recurrent neural networks, in: Proc. International Conference on Learning Representations, 2015.
[13] T. Pham, T. Tran, D. Phung, S. Venkatesh, Deepcare: A deep dynamic memory model for predictive medicine, in: Pacific-Asia Conference on Knowledge Discovery and Data Mining, 2016, pp. 30–41.
[14] P. Nguyen, T. Tran, S. Venkatesh, Deep learning to attend to risk in ICU, in: KHD@ IJCAI, 2017.
[15] C.A. Michael, D. Dominey-Howes, M. Labbate, The antimicrobial resistance crisis: causes, consequences, and management, Front. Publ. Health 2 (2014) 145.
[16] A.P. Magiorakos, et al., Multidrug-resistant, extensively drug-resistant and pandrug-resistant bacteria: an international expert proposal for interim standard definitions for acquired resistance, Clin. Microbiol. Infect. 18 (3) (2012) 268–281.
[17] I. D. S. of America (IDSA), Combating antimicrobial resistance: policy recommendations to save lives, Clin. Infect. Dis. 52 (5) (2011) 397–428.
[18] L. Zhang, C. Aggarwal, G.J. Qi, Stock price prediction via discovering multi-frequency trading patterns, in: Proc. 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, 2017, pp. 2141–2149.
[19] W. Jenna, et al., Do no harm: a roadmap for responsible machine learning for health care, Nature Med. 25 (2019) 1337–1340.
[20] S. El-Sappagh, J.M. Alonso, S.M.R. Islam, A.M. Sultan, K.S. Kwak, A multilayer multimodal detection and prediction model based on explainable artificial intelligence for Alzheimer's disease, Sci. Rep. 11 (2021).
[21] J. He, S.L. Baxter, J. Xu, J. Xu, X. Zhou, K. Zhang, The practical implementation of artificial intelligence technologies in medicine, Nature Med. 25 (2019) 30–36.
[22] A.B. Arrieta, et al., Explainable artificial intelligence (XAI): concepts, taxonomies, opportunities and challenges toward responsible ai, Inf. Fusion 58 (2020) 82–115.
[23] E. Toreini, M. Aitken, K. Coopamootoo, K. Elliott, C.G. Zelaya, A. van Moorsel, The relationship between trust in AI and trustworthy machine learning technologies, in: Proc. of the International Conference on Fairness, Accountability, and Transparency, ACM, 2020, pp. 272–283.
[24] D. Gunning, D. Aha, DARPA's explainable artificial intelligence (XAI) program, AI Mag. 40 (2019) 44–58.
[25] C. Rudin, Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead, Nat. Mach. Intell. 1 (2019) 206–215.
[26] R. Guidotti, A. Monreale, S. Ruggieri, F. Turini, F. Giannotti, D. Pedreschi, A survey of methods for explaining black box models, ACM Comput. Surv. 51 (2018) 1–42.
[27] J.M. Alonso, J. Toja-Alamancos, A. Bugarin, Experimental study on generating multi-modal explanations of black-box classifiers in terms of graybox classifiers, in: IEEE World Congress on Computational Intelligence, 2020, http://dx.doi.org/10.1109/FUZZ48607.2020.9177770.
[28] S.M. Lundberg, S.-I. Lee, A unified approach to interpreting model predictions, in: Advances in Neural Information Processing Systems, 2017, pp. 1–10.
[29] Ó. Escudero-Arnanz, I. Mora-Jiménez, S. Martínez-Agüero, J. Álvarez-Rodríguez, C. Soguero-Ruíz, Temporal feature selection for characterizing antimicrobial multidrug resistance in the intensive care unit. in: 24th European Conference on Artificial Intelligence, 2020, pp. 54–59.
[30] S. Martínez-Agüero, I. Mora-Jiménez, J. Álvarez-Rodríguez, A.G. Marques, C. Soguero-Ruíz, Aplying LSTM networks to predict multi-drug resistance using binary multivariate clinical sequences. in: 24th European Conference on Artificial Intelligence, 2020.
[31] C. Catley, H. Stratti, C. McGregor, Multi-dimensional temporal abstraction and data mining of medical time series data: Trends and challenges, in: 30th Annual International Conference of the IEEE Engineering in Medicine and Biology Society, 2008, pp. 4322–4325.
[32] H. Khazaei, C. McGregor, M. Eklund, K. El-Khatib, A. Thommandram, Toward a big data healthcare analytics system: a mathematical modeling perspective, in: World Congress on Services, IEEE, 2014, pp. 208–215.
[33] J. Tang, S. Alelyani, H. Liu, Feature selection for classification: A review, in: Data Classification: Algorithms and Applications, 2014, p. 37.
[34] S. Muñoz Romero, A. Gorostiaga, C. Soguero-Ruiz, I. Mora-Jiménez, J.L. Rojo-Alvarez, Informative variable identifier: Expanding interpretability in feature selection, Pattern Recognit. 98 (2020) 107077.
[35] B. Efron, The Jackknife, the Bootstrap and Other Resampling Plans, SIAM, 1982.
[36] B. Efron, R.J. Tibshirani, An Introduction to the Bootstrap, CRC Press, 1994.
[37] W. Li, Mutual information functions versus correlation functions, J. Stat. Phys. 60 (5) (1990) 823–837.
[38] C.E. Shannon, A mathematical theory of communication, Bell Syst. Tech. J. 27 (3) (1948) 379–423.
[39] S. Gao, G. Ver Steeg, A. Galstyan, Efficient estimation of mutual information for strongly dependent variables, in: Artificial Intelligence and Statistics, PMLR, 2015, pp. 277–286.
[40] F. Fleuret, Fast binary feature selection with conditional mutual information, J. Mach. Learn. Res. 5 (9) (2004).
[41] V. Fonti, E. Belitser, Feature selection using LASSO, in: VU Amsterdam Research Paper in Business Analytics, vol. 30, 2017, pp. 1–25.
[42] C. Chesneau, M. Hebiri, Some theoretical results on the grouped variables LASSO, Math. Methods Statist. 17 (4) (2008) 317–326.
[43] J.F. Díez-Pastor, J.J. Rodríguez, C. García-Osorio, L.I. Kuncheva, Random balance: ensembles of variable priors classifiers for imbalanced data, Knowl.-Based Syst. 85 (2015) 96–111.
[44] S. Wang, W. Liu, J. Wu, L. Cao, Q. Meng, P.J. Kennedy, Training deep neural networks on imbalanced data sets, in: 2016 International Joint Conference on Neural Networks, 2016, pp. 4368–4374.
[45] H. He, E.A. Garcia, Learning from imbalanced data, IEEE Trans. Knowl. Data Eng. 21 (9) (2009) 1263–1284.
[46] Y.S. Aurelio, G.M. de Almeida, C.L. de Castro, A.P. Braga, Learning from imbalanced data sets with weighted cross-entropy function, Neural Process. Lett. 50 (2) (2019) 1937–1949.
[47] K.Ø. Mikalsen, C. Soguero-Ruiz, F.M. Bianchi, A. Revhaug, R. Jenssen, Time series cluster kernels to exploit informative missingness and incomplete label information, Pattern Recognit. 115 (2021) 107896.
[48] Z.C. Lipton, et al., Modeling missing data in clinical time series with RNNs, Mach. Learn. Healthc. 56 (2016).
[49] Z. Che, S. Purushotham, K. Cho, D. Sontag, Y. Liu, Recurrent neural networks for multivariate time series with missing values, Sci. Rep. 8 (1) (2018) 1–12.
[50] O. Duda, P. Hart, D. Stork, Pattern Classification, John Wiley & Sons, 2001.
[51] V. Nair, G.E. Hinton, Rectified linear units improve restricted boltzmann machines, in: International Conference on Machine Learning, Omnipress, 2010, pp. 807–814.
[52] D.P. Kingma, J.L. Ba, Adam: A method for stochastic optimization, in: Proc. International Conference on Learning Representations, 2015, p. 13.
[53] A. Graves, Supervised Sequence Labelling with Recurrent Neural Networks, Springer, 2012.
[54] K. Cho, B. Merrienboer, D. Bahdanau, Y. Bengio, On the properties of neural machine translation: Encoder–decoder approaches, in: Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation, 2014.

[55] J.M. Alonso, L. Magdalena, G. Gonzalez-Rodríguez, Looking for a good fuzzy system interpretability index: An experimental approach, Internat. J. Approx. Reason. (2009) 115–134.
[56] J.M. Alonso, A. Bugarín, Expliclas: Automatic generation of explanations in natural language for weka classifiers, in: 2019 IEEE International Conferences on Fuzzy Systems, 2019, pp. 1–6.
[57] D. Pancho, J.M. Alonso, L. Magdalena, Quest for interpretability-accuracy trade-off supported by fingrams into the fuzzy modeling tool GUAJE, Int. J. Comput. Intell. Syst. (2013) 46–60.
[58] J.M. Alonso, C. Castiello, L. Magdalena, C. Mencar, Explainable Fuzzy Systems - Paving the Way from Interpretable Fuzzy Systems to Explainable AI Systems, 970, Springer International Publishing, 2021.
[59] J.R. Quinlan, C4.5: Programs for Machine Learning, Morgan Kaufmann Publishers, San Mateo, CA, 1993.
[60] J. Hühn, E. Hüllermeier, FURIA: an algorithm for unordered fuzzy rule induction, Data Min. Knowl. Discov. 19 (3) (2009) 293–319.
[61] J.M. Soto-Hidalgo, J.M. Alonso, G. Acampora, J. Alcala-Fdez, JFML: A java library to design fuzzy logic systems according to the IEEE std 1855-2016, IEEE Access 6 (2018) 56952–56964.
[62] J.L. Hodges, E.L. Lehmann, Ranks methods for combination of independent experiments in analysis of variance, Ann. Math. Stat. 33 (1962) 482–497.
[63] N. Brusselaers, D. Vogelaers, S. Blot, The rising problem of antimicrobial resistance in the intensive care unit, Ann. Intensive Care 1 (2011) 47.
[64] J.J. De Waele, et al., Antimicrobial resistance and antibiotic stewardship programs in the ICU: insistence and persistence in the fight against resistance. a position statement from ESICM/ESCMID/WAAAR round table on multi-drug resistance, Intensive Care Med. 44 (2) (2018) 189–196.
[65] S.H. Zinner, Antibiotic use: present and future, Microbiol.-Bol. 30 (3) (2007) 321.
[66] A. Vaswani, et al., Attention is all you need, in: Advances in Neural Information Processing Systems, 2017, pp. 5998–6008.
[67] A. Hernandez Carnerero, M. Sanchèz-Marrè, I. Mora Jiménez, C. Soguero Ruiz, S. Martínez Agüero, J. Álvarez Rodríguez, Antimicrobial resistance prediction in intensive care unit for pseudomonas aeruginosa using temporal data-driven models, Int. J. Interact. Multimedia Artif. Intell. 6 (5) (2021) 119–133.

Sergio Martínez-Agüero (M.Sc. in Telecommunication Engineering, Rey Juan Carlos University, Spain, 2020) is a Research Assistant at Rey Juan Carlos University currently working on his Ph.D. entitled "Deep Learning and Network Analytics for extracting knowledge from infectious diseases in the ICU". He has made several contributions to national and international congresses and published two papers in JCR journals. He is currently part of two competitive projects funded by the Spanish Government related to healthcare data-driven machine learning models. He is interested in data science, machine learning, data visualization and network analytics.

Cristina Soguero-Ruiz (Ph.D. in Machine Learning with Applications in Healthcare, Rey Juan Carlos University and University Carlos III of Madrid, Spain, 2015) is an Assistant Professor and the Coordinator of the Biomedical Engineering Degree at Rey Juan Carlos University. She won the Orange Foundation Best Ph.D. Thesis Award by the Spanish Official College of Telecommunication Engineering. She has published more than 30 JRC-indexed papers and 50 international conference communications. She has participated in several research projects related to healthcare data-driven machine learning systems (being the principal investigator in 5). Her current research interests include machine learning and data science.

Jose M. Alonso-Moral (Ph.D. in Telecommunication Engineering, Technical University of Madrid, Spain, 2007) is a "Ramón y Cajal" Distinguished Researcher (RYC-2016-19802) at CiTIUS-USC, board member of the ACL Special Interest Group on Natural Language Generation, Associate Editor of the IEEE Computational Intelligence Magazine, Chair of the IEEE-CIS Task Force on Explainable Fuzzy Systems, member of the IEEE-CIS Task Force on Fuzzy Systems Software, and President of the Executive Board of the H2020-MSCA-ITN-2019 (Grant Agreement No 860621) project "Interactive Natural Language Technology for Explainable Artificial Intelligence" (NL4XAI).



Inmaculada Mora-Jiménez (Ph.D. in Telecommunication Engineering, Carlos III University of Madrid, Spain, 2004) is a Full Professor at Rey Juan Carlos University, Spain. She has conducted her research mainly in data analytic and biomedical engineering. She is a co-author of more than 40 JCR-indexed papers and 50 contributions to international conferences. She has participated in 18 competitive research projects (principal investigator of 5) and collaborated in more than 20 projects with private funding entities. Her main research interests include data science and machine learning with application to image processing, bioengineering, and wireless communications.

Joaquín Álvarez-Rodríguez (Ph.D. in Medicine, Complutense University of Madrid, Spain, 1996) has been, since 2003, the head of the Intensive Care Medicine Department at the Hospital Universitario de Fuenlabrada. His lines of work have been the quality and safety of patients, medical information systems and infections in the ICU. He has actively participated in the national coordination of Zero Projects, which aim to reduce the main infections acquired in ICU and the emergence of AMR bacteria in the ICU. His main research area is the collection of data recorded in the electronic medical record.

Antonio G. Marques (Ph.D. in ECE, Carlos III University of Madrid, Spain, 2007) is a Full Professor at Rey Juan Carlos University, Spain and held different visiting positions with the Universities of Minnesota and Pennsylvania, USA. His current research focuses on signal processing, machine learning and optimization over graphs and networks. He has served as an Associate Editor and Technical/General Chair for different journals and conferences. His work has been awarded in several venues and he was the recipient of the 2020 EURASIP Early Career Award. He is a Member of the IEEE, EURASIP and the ELLIS society.