PREDICTING THE EVOLUTION OF COVID-19 MORTALITY RISK: A RECURRENT NEURAL NETWORK APPROACH

Marta Villegas
Text Mining Unit
Barcelona Supercomputing Center
C/ Jordi Girona 1-3, 08034, Barcelona, Spain
marta.villegas@bsc.es

Aitor Gonzalez-Agirre
Text Mining Unit
Barcelona Supercomputing Center
C/ Jordi Girona 1-3, 08034, Barcelona, Spain
aitor.gonzalez@bsc.es

Asier Gutiérrez-Fandiño
Text Mining Unit
Barcelona Supercomputing Center
C/ Jordi Girona 1-3, 08034, Barcelona, Spain
asier.gutierrez@bsc.es

Jordi Armengol-Estapé
Text Mining Unit
Barcelona Supercomputing Center
C/ Jordi Girona 1-3, 08034, Barcelona, Spain
jordi.armengol@bsc.es

Casimiro Pio Carrino
Text Mining Unit
Barcelona Supercomputing Center
C/ Jordi Girona 1-3, 08034, Barcelona, Spain
casimiro.carrino@bsc.es

David Pérez Fernández
Ministry of Inclusion, Social Security and Migration
C/ Paseo de la Castellana 63, 28071, Madrid, Spain
david.perez@inv.uam.es

Felipe Soares
Universidade Federal do Rio Grande do Sul
Av. Osvaldo Aranha, 99, Porto Alegre, Brazil
fs@felipesoares.net

Pablo Serrano
Hospital Universitario 12 de Octubre
Av. de Córdoba, s/n, 28041 Madrid, Spain
pserranob@salud.madrid.org

Miguel Pedrera
Hospital Universitario 12 de Octubre
Av. de Córdoba, s/n, 28041 Madrid, Spain
miguel.pedrera@salud.madrid.org

Noelia García
Instituto de Investigación Sanitaria del
Hospital Universitario 12 de Octubre
Av. de Córdoba, s/n, 28041 Madrid, Spain
ngbarrio@salud.madrid.org

Alfonso Valencia
Life Sciences Department
Barcelona Supercomputing Center
C/ Jordi Girona 1-3, 08034, Barcelona, Spain
alfonso.valencia@bsc.es

December 11, 2020

ABSTRACT

Background: The propagation of COVID-19 in Spain prompted the declaration of the state of alarm on March 14, 2020. On 2 December 2020, the infection had been confirmed in 1,665,775 patients and caused 45,784 deaths. This unprecedented health crisis challenged the ingenuity of all professionals involved. Decision support systems in clinical care and health services management were identified as crucial in the fight against the pandemic.

Methods: This study applies Deep Learning techniques for mortality prediction in COVID-19 patients. Two datasets with clinical information (medication, laboratory tests, vital signs etc.) of 2,307 and 3,870 COVID-19 infected patients admitted to two Spanish hospital chains were used. Firstly, we built a sequence of temporal events gathering all the clinical information for each patient. Next, we used the temporal sequences to train a Recurrent Neural Network (RNN) model with an attention mechanism exploring interpretability. We conducted extensive experiments and trained the RNNs in different settings, performing hyperparameter search and cross-validation. We ensembled resulting RNNs to reduce variability and enhance sensitivity.

Results: We assessed the performance of our models using both global metrics, by averaging the performance across all the days in the sequences. We also measured day-by-day metrics starting from the day of hospital admission and the outcome day and evaluated the daily predictions. Regarding sensitivity, when compared to more traditional models, our best two RNN ensemble models outperform a Support Vector Classifier in 6 and 16 percentage points, and Random Forest in 23 and 18 points. For the day-by-day predictions from the outcome date, the models also achieved better results than baselines showing system's ability towards early predictions.

Conclusions: We have shown the feasibility of our approach to predict the clinical outcome (i.e. discharged alive or death) of patients infected with SARS-CoV-2. The result is a time series model that can support decision-making in healthcare systems and aims at interpretability. Despite the low-resource scenario, the results achieved are promising and suggests that more data will further increase the performance of the model.

***K*eywords** COVID-19 · mortality prediction · time series analysis · RNN

1 Introduction

According to the daily report of the Coordination Center for Health Alerts and Emergencies[1] of the Spanish Ministry of Health, as of 2 December 2020, a total of 1,665,775 confirmed cases of COVID-19 and 45,784 deaths had been reported in Spain. To effectively respond to a challenge of this magnitude and to optimise the hospitalization process of this emerging disease, decision support systems for clinical care and health services management are crucial. Artificial Intelligence and Deep Learning tools offer a range of possibilities for obtaining models that, trained with historical data, can anticipate future scenarios.

In this work, we apply Deep Learning techniques for predicting the clinical outcome of patients with COVID-19. The result is a model that predicts mortality risk to support decision making and logistical planning in healthcare systems. This study leverages two datasets with clinical information on 2,307 and 3,870 infected patients. The two datasets used to train and validate the predictive models include all patients admitted with COVID-19 in HM Hospitales (HM) and in the Hospital Universitario 12 de Octubre (H120). In the HM dataset, the cohort includes all COVID patients from the beginning of the pandemic in Spain until April 20th, 2020. In the H12O dataset, the cohort extends until October 14th, 2020. Having two different datasets allowed us to validate the results and to check that the architecture and design developed are equally suitable for the two cases. Although both datasets contain information from COVID patients, they have different characteristics. This corroborates the robustness and applicability of the proposal. The differences between the datasets encouraged us to explore different methods of feature representation which we fully report.

We propose a time series analysis system using a Recurrent Neural Networks (RNN) based ensemble model that responds to the reality of hospitals during the pandemic. Our main objective is to develop a model that generates daily predictions in a hospital environment to be used as early warning system. Thus, the system predicts the mortality risk for each inpatient on a daily basis considering all previous available data. In such a scenario, early prediction and unbiased population analysis is critical. We, therefore, expect to obtain encouraging sensitivity scores which is a highly relevant metric in the clinical domain.

In this sense, for the daily predictions, the RNN models achieve better results than competitive baselines obtained with traditional Machine Learning (ML) algorithms, such as Support Vector Classifiers (SVC) and Random Forest (RF) showing promising results when considering early predictions. Moreover, the RNN incorporates an attention mechanism that obtains useful insights about patients status over time.

Similarly, when analyzing global sensitivity performance, the RNN-based models achieve good scores and outperform baselines. In this case, the best HM RNN ensemble model outperforms SVC in more than 6 percentage points and RF

[1] `https://www.mscbs.gob.es/profesionales/saludPublica/ccayes/alertasActual/nCov/documentos/Actualizacion_263_COVID-19.pdf`

in nearly 23 percentage points, whereas the best H12O RNN ensemble model outperforms SVC and RF in 16 and 18 percentage points, respectively.

Another relevant characteristic of our proposal is the effort devoted to the interpretability of the model, which is based on an attention mechanism. It takes into account the RNN's outputs to the RNN's output history vectors to retrieve meaningful past dynamic information for the daily prediction, aiming at providing explanatory capabilities in the temporal dimension.

2 Related Work

Time series prediction problems are considered complex in predictive modeling. Recurrent Neural Networks (RNN) are well suited to capture sequential information from temporal data, since they learn long-term dependencies by incorporating a memory effect that is able to preserve state over time. RNN have been applied in different proposals addressing Intensive Care Unit (ICU) mortality prediction based on Electronic Health Records (EHR), as in [1, 2, 3].

Before that, Khosla et al., [4] developed models using SVCs for stroke prediction on the Cardiovascular Health Study (CHS) dataset and Letham et al. [5] used models based on interpretable decision trees, also called Bayesian rule lists. In our work, we use SVCs and RF as baselines.

Lipton et al. [6] produced the first empirical study using a Long Short Term Memory RNN [7] to classify diagnoses given multivariate time series. In this experiment, the diagnostic labels had no timestamps and they were used only as predictive labels. Similarly to our work, in the training phase, the authors used target replication (an output is generated at every sequence step). However, at prediction time, the system considers only the output during the final step so that, although the system considers the whole series of events, it does not give a prediction until the end and, therefore, early prediction is not considered.

Choi et al. [8] developed Doctor AI, a predictive model that uses diagnosis, medication or procedure codes as input for the RNN to predict the diagnosis and medication categories for a subsequent visit. The authors used a dataset of 263,706 patients who made on average 54.61 visits per person. Contrary to us, they had enough data to use Skip-gram [9][2] to capture the latent representation of medical codes from the EHR. By applying Skip-gram, they were able to project diagnosis, medication and procedure codes into a lower dimensional space. The authors reported better results when using Skip-gram approach compared to the traditional multi-hot label encoding approach.

Choi et al., in [10], suggest a Reverse Time Attention Mechanism with a RNN 1 to create an interpretable predictive model and to cope with long sequences of visits. They used a two-level neural attention model that detects influential past visits and relevant clinical variables within those visits. In our proposal, the attention mechanism also focuses on the time sequence trying to identify influential episodes but we do not reverse sequences as, in our case, these are much shorter.

Fenglong et al. [11] use RNNs with a simple attention mechanism to interpret the results and to forecast health information of patients from the historical sequences of visits over time. They employ bidirectional RNNs to remember all the information of both past and future visits, and they introduce three attention mechanisms to measure the relationships of different visits for the prediction. We also experiment with attention mechanism but we use unidirectional RNN as, in our setting, we want daily predictions and thus we unroll the RNN.

Douglas Teoh et al., in [12] worked on forecasting stroke probability using RNNs to enable proactive forms of healthcare measures. As in our experiment, stroke cases resulted in imbalanced class outputs which caused bias in trained Artificial Neural Network models towards negative predictions. To address this issue, they designed and incorporated regularization terms into the standard cross-entropy loss function. We also experimented with the vanilla cross-entropy and additionally we used the Focal Loss.

Xia et al. [13] used an ensemble algorithm of LSTMs to deal with heterogeneous ICU data for mortality prediction. The eventual ensemble models make their predictions by merging the results of multiple parallel LSTM classifiers. The base LSTM learners are trained on different subsets which are generated using bootstrapped samples and random feature subspace. The results show that, compared with traditional clinical scoring systems [3], RF, and the single LSTM classifier, the ensemble LSTM model achieved better performance. We also explore the use of ensemble methods to improve sensitivity. However, our approach is different as we did not use bootstrapping neither random feature subspace on training data but we rather designed a voting system and defined an algorithm that, starting with the best

[2] Skip-gram is an algorithm for learning word embeddings by predicting the surrounding words (i.e. the context) of a given word from the given word. It was designed for natural language processing, but it can be applied to other scenarios as long as data consist of discrete sequences.

[3] SAPS II, SOFA, and APACHE II.

model in terms of cross-validated F1 score, heuristically chooses the most different models (i.e., the ones that make the most different predictions, to encourage model diversity in the ensemble). Finally, the authors used a subset of 18,415 patients from MIMIC-III[4] with a length of stay ≥ 10 days to predict the 28-day post-admission mortality. Though the experiment is similar to ours, our setting is different as we do not filter sequences shorter than 10 and, therefore, we also work with time sequences that contain less information.

In 2020, Yan et al. [14] also addressed the COVID-19 crisis. They used blood samples from 485 infected patients to identify predictive bio-markers of mortality for COVID-19 patients using a supervised XGBoost [15] classifier. Although for most patients the data set contained multiple blood samples taken throughout their hospital stay, the model was just trained and tested with the last samples. However, according to the authors, the model can be applied to all other blood samples to estimate the predictive potential of the identified bio-markers in the horizon prediction.

More recently, a COVID-19 mortality risk study that used HM Hospitales data among others has been published by Bertsimas et al. [16], in the first multi-center COVID-19 mortality risk study that uses data from 3,062 patients across four different countries including 1,390 from HM Hospitales. They used the XGBoost algorithm to predict mortality using 22 features, including patient demographic information, comorbidities, vitals upon admission, and laboratory test results. Unlike us, and because XGBoost is a decision tree algorithm, the experiment does not consider the sequence of events but rather uses the data from the first event in its analysis. Surprisingly, the average mortality rate observed in this population is much higher than the one in our dataset (26.84% vs 14.43% HM vs 12.91% H12O). The authors reported an 85.02% accuracy for the test dataset and 74.92%, 86.76% and 61.3% for the three external validation datasets they used.

Very recently, Klann et al. [17] developed a patient severity classification algorithm for COVID-19 patients. They used a ML algorithm fed with EHR. Their results in sensitivity were 73% and 83% in specificity combining both ICU and dead outcomes. Our work only focuses death events and is able to take ICU stay evolution as input (on the HM dataset).

Contributions Our work constitutes one of the first effective predictive models of time series data with COVID-19 patient records. Unlike [14] and [16], we apply full time series inference and make daily predictions, instead of using static data or collapsing the dynamic data into fixed-size vectors. To achieve this goal trained different RNN models, and assembled them prioritising a given metric. Using this technique, we can substantially increase the sensitivity of the system, at the same time as we produce considerably more stable predictions.

In contrast to most of the studies above[5], especially [9], which leverages data belonging to over 260,000 patients, we worked with a considerably low-resource settings (2,307 and 3,870 patients). For this reason, to create data representations, we relied on classic feature extraction methods.

Most importantly and in contrast with the above-mentioned studies, we were extremely conservative when filtering out patients. For instance, in [13], which is a study substantially similar to ours, they only considered patients with a time window of 10 days. On the contrary, in our work, the filtering is not very restrictive and, consequently, the lengths of the dynamic data exhibit high variance. In the same way, we are not restrictive when filtering features. Whereas [16] omit features whose values are more than 40% missing, we do not filter features based on the percentage of missing values, and deal with sparse data representations. Our system worked with real world data and it is more realistic in medical terms, so that it can be run out of the box with new patient records directly coming from hospitals. That being said, our model is a binary classifier, not a multi-classification task as in [8]. Finally, as [10] [18] [11], we experimented with attention to improve the interpretability of our model.

3 Datasets

The experiments reported in this article uses two different COVID datasets. The first one corresponds to the dataset made available by HM Hospitales from April 25th 2020, in their "COVID Data Save Lives" initiative[6] and contains information of patients diagnosed with COVID until that date. The second one comes from H12O and the COVID patient cohort extends to October 14th, available due to a Collaboration Agreement signed between H12O and BSC. In both cases, the data contain two different types of variables: **static variables** that are constant throughout admission, such as sex or age, and **dynamic variables** that are measured at different times during the hospital stay, such as medication, laboratory analysis and vital signs.

Although both datasets contain information from COVID patients, they have different characteristics which prevented to join the datasets in a single one. The disparity in coding (e.g. ICD-9 vs ICD-10) and the differences in the type of

[4] `https://mimic.physionet.org/`

[5] Except [14], in which authors dealt with around 400 patients.

[6] `https://www.hmhospitales.com/coronavirus/covid-data-save-lives`

information contained made it difficult to aggregate data and would produce a reduced dataset due to the low overlap. Note, however, that having two different datasets allowed us to validate the robustness of the proposal. Further details are given in the following lines.

3.1 HM Hospitals Dataset

The HM dataset contains the anonymized records of 2,307 patients admitted with a diagnosis of COVID POSITIVE or COVID PENDING, since the beginning of the epidemic to April 24th 2020. HM Hospitales opened this anonymous dataset freely available to the medical and scientific community with clinical information on patients treated in their hospital centers. The dataset includes demographic data and information about medication, vital signs, laboratory, diagnostics, procedures and ICU stay.

Of the 2,307 patients, 1,377 are males and 930 are females, around 60% and 40% respectively. The minimum and maximum ages are 0 and 106 years, with a mean of 67.6 (see Table 1 and Figure 1). The mortality rate is 14.43% (16% for males and 12% for females). Of the total of 333 deaths, 218 are males and 115 are females, being the 65% and the 35% of the death population and the 9% and 6% of the whole patients.

The observed mean duration from admission to outcome (death, discharge, or other reason) is 8.08 days with a standard deviation (SD) of 6.18 days and a median of 7 days.

160 patients (7% of the records) were admitted to the ICU (9% for males and 4% for females) with the shortest stay being one day and the longest being 35 days, with a mean of 8.97 days, SD of 8.66 and median of 6 days. Of the total of 160 patients admitted to the ICU, 120 are male and 40 are female, being the 75% and the 25% of the ICU population and the 5% and the 2% of all patients.

Table 1 and Figure 1 show the age and sex distribution of patients in relation to mortality and ICU stay.

Information about **medication** administered during admission includes the dates for the first and last administration of each drug. Medications are identified by their brand name and Anatomical Therapeutic Chemical (ATC) 5 and 7 classification codes. The data includes 60,460 rows with 275 and 464 ATC5 and ATC7 unique codes. During the hospital stay, the minimum number of medications per patient is 1 and the maximum is 183, with a mean of 26 and a median of 22. Table 1 summarizes the medication distributions.

The **vital signs** include heart rate, maximum and minimum blood pressure and temperature collected during admission, together with their date and time of registration. The minimum number of registrations per patient is 1 and the maximum is 190, with a mean of 24.5 registrations and a median of 21 as summarised in Table 1. There are 37,861 records other than 0s for heart frequency; 21,174 and 21,166 records for maximum and minimum blood pressure and 54,431 records for temperature.

The information about **laboratory** contains all the results of the requests made to each patient during admission and in the previous emergency, if any. This includes 398,884 results for 401 different determinations unevenly distributed. As shown in Figure 3, and in the Additional Material file, the 82% of determinations have less than 500 reported results and there is a long tail of really infrequent determinations. 2,080 patients have at least one lab result with a maximum of 4,395 and a mean of 190 and a median of 111 as summarized in Table 1.

Finally, the dataset also contains **diagnoses** present on admission using the International Classification of Diseases (ICD-10)[7]. Though the dataset also contains diagnoses for episodes of hospital admission, these were not considered in the experiment as they had no time information associated. Only the diagnoses marked as being present on admission were included, as part of the static variables.[8].

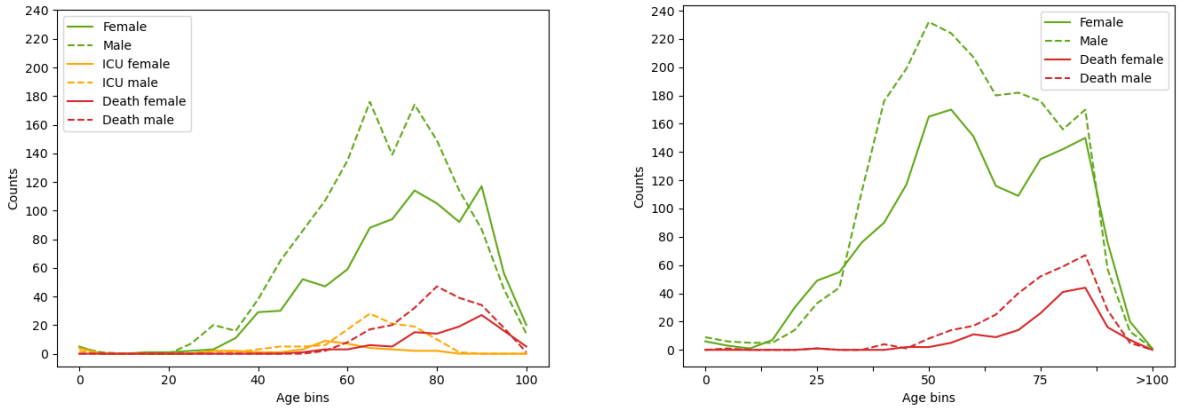
[7]Does not include COVID.

[8]Note that including diagnoses that lack temporal information would not be fair since we could be introducing information from future events.

HM					
Metric	Medications	Vital sign records	Determinations	Diagnoses	Procedures
Mean	26.28	24.51	189.66	7.44	6.69
Standard Deviation	0.40	0.37	6.56	4.28	2.69
Median	22	21	111	7	6
Range	182	189	4,394	18	19
Minimum	1	1	1	1	1
Maximum	183	190	4,395	19	20
Sum	60,460	55,515	394,490	13,213	11,878
Count (patients)	2,301	2,265	2,080	1,775	1,775
Distinct	275/464	4	401	1,363	251

H12O					
Metric	Medications	Vital sign records	Determinations	Diagnoses	Procedures
Mean	24.68	113.51	174.10	4.37	2.17
Standard Deviation	26.03	96.74	258.15	5.98	3.07
Median	16	87	108	2	1
Range	341	1,538	3,568	76	23
Minimum	1	1	1	1	1
Maximum	342	1,539	3,622	77	24
Sum	95,439	433,743	652,038	7,031	383
Count (patients)	3,867	3,821	3,745	1,607	176
Distinct	798	9	80	397	18

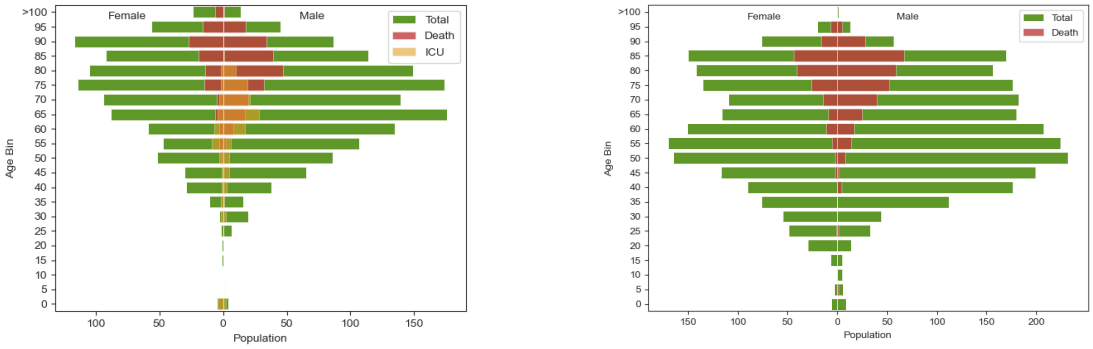
Table 1: Number of medications, vital sign records, determinations, diagnoses and procedures per patient in HM and H12O. Medications from HM are in ATC5 codes whereas in H12O they are in ATC7 (for the sake of comparison we show the number of distinct ATC7 codes in HM). Diagnoses and Procedures in HM include only those Present On Admission whereas in H12O their timestamp goes from 1st 2019 to discharge date. Diagnoses and Procedures codes from HM are in ICD-10 whereas in H12O they are ICD-9 encoded.



(a) HM

(b) H12O

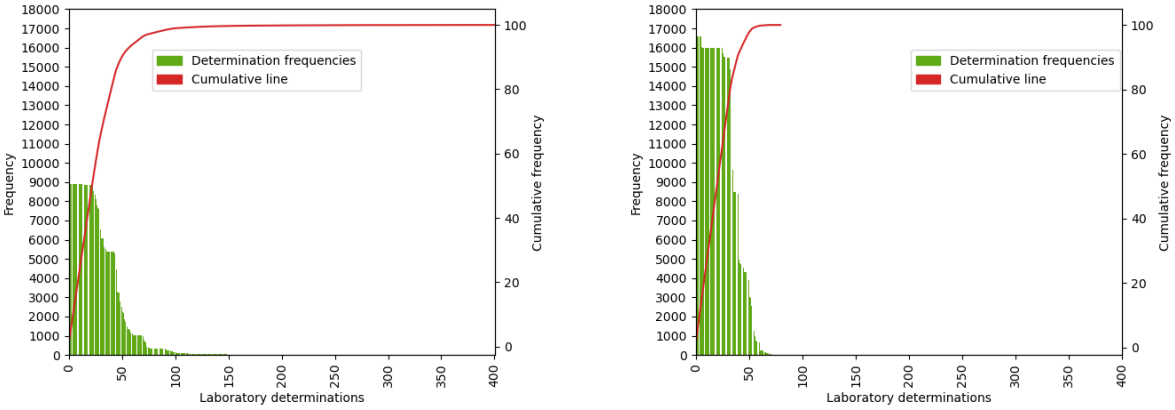
Figure 1: Age and sex distribution of patients in HM and H12O datasets. Solid lines are for females and dotted lines for males. Green lines show number of patients, read lines show the number of dead patients. For HM, orange lines show the number of 'in ICU' patients.



(a) HM

(b) H12O

Figure 2: Age and sex distribution of patients in HM and H12O dataset. Left females, right males. Bar fills are the number of male/female (green) and dead male/female patients (red). For HM, also 'in ICU' male/female patients (orange).



(a) HM

(b) H12O

Figure 3: Distribution of laboratory determinations in HM and H12O datasets in descending order with a cumulative line on the left axis as a percentage of the total.

3.2 12th October Hospital Dataset

The H12O dataset contains the anonymized records of 3,870 COVID patients since the beginning of the epidemic to October 14th. The dataset includes demographic data and information about medication, vital signs, laboratory, diagnostics and procedures.

Of the 3,870 patients, 2,201 are males and 1,669 are females, around 57% and 43% respectively. The minimum and maximum ages are 0 and 103 years, with a mean of 61.4 (see Table 1 and Figure 1). The mortality rate is 13% (14.63% for males and 10.67% for females). Of the total of 500 deaths, 322 are males and 178 are females, being the 64% and the 36% of the death population and the 8% and 5% of the whole patients.

The observed mean duration from admission to outcome (death, discharge, or other reason) is 9.51 days with SD 4.99 days and a median of 8 days, which is slightly longer than HM.

Figure 2 show the age and sex distribution of patients in relation to mortality.

Medications in H12O dataset are represented in ATC7 coding, with 798 unique medications. The minimum number of medications per patient is 1 and the maximum is 342, with a mean of 24,68 per patient and SD of 26.03 and MD of 16.

Vital signs used in the H12O dataset include heart rate, breathing frequency, O2 saturation, temperature, mean blood pressure, diastolic tension, systolic tension, O2 saturation previous to intubation and minimum O2 saturation during intubation. There is a mean of 113.51 records per patient with an SD of 96.74 and a MD of 87. The minimum vital signs record for a patient is 1 and the maximum is 190. The total vital signs recorded are 433,743 which is much higher than in the case of HM because H12O has more vital signs (9 vs. 4) and more patients (3,821 vs. 2,265).

Records regarding **laboratory determinations** count 80 unique determinations, with a mean of 174.1 measurements per patient, a SD of 258.15 and a MD of 108. The minimum laboratory determinations for patients is 1 and the maximum is 3,622. As Table 1 and Figure 3 show, H12O has less unique determinations but higher amount of records than HM. HM has more distinct laboratory determinations but some of them have negligible appearances and, therefore, it shows a longer cumulative line.

Finally, for both **diagnoses** and **procedures**, we take the (ICD-9) codes from January 1st of 2019. There are 397 and 18 unique codes respectively distributed among patients with a mean of 4.37 diagnoses and 2.17 procedures, SD 5.98 and 3.07 and median of 2 and 1.

Table 1 summarizes the medications, vital signs and determination distributions.

4 Feature Representation

Information in EHR needs to be transformed into appropriate data representations to be used on clinical ML. Learning better representations is critical for performance improvement. The feature generation is a data processing used to encode variables into numerical values. The encoding operation depends on the values of the variables and our dataset includes

- Categorical variables: a variable that has multiple values with no order, each one associated to a given case (e.g. medications).
- Ordinal variables: a variable taking discrete values with a given order. The discrete values are defined by comparing the original values of the variables with a given interval of reference (e.g. vitals signs and some lab results).
- Continuous variables: a sub-type of ordinal variable that takes numerical values ordered along a continuous scale (e.g temperature).

In general, categorical variables are *one-hot encoded*, where each value of the variable becomes a binary variable itself, indicating absence or presence of the original variable's value. Ordinal variables can be either encoded using binary values or 0, 0.5, 1 values depending on the reference interval or normalized in the [0,1] interval. Continuous variables are normalized in the [0, 1] interval.

For many variables, we found a considerable amount of **missing values** that can be classified into 'zero values' and 'missing measurements'. Medications and diagnosis do typically have 'zero values' and, in these cases, they indicate that the patient was not prescribed a certain medication or was not diagnosed with a certain feature. On the other hand, laboratory results and vital signs do typically have 'missing measurements'. Thus, a missing value in blood pressure does not mean that the patient has no blood pressure but, simply, that it was not measured or recorded. To handle the missing values, we use the following criteria: (i) 'zero values' are assigned 0 whereas (ii) 'missing measurements' are replaced with their previous values for the same patient, if they exist, otherwise, they are filled with the median of the original value over the whole sample.

For **static variables**, the gender variable was binarized and the age variable was normalized between [0, 1]. Diagnosis (HM and H12O) and procedures on admission (H12O) were included as one-hot-encoded values, taking the three first digits of the ICD-10 and ICD-9 for HM and H12O respectively.

For **dynamic variables**, we applyed similar methods for both datasets, but with slight variations. For **HM** the information about **ICU stay** was used to create a binary variable indicating, whether the patient is in the ICU or not each day of the hospital stay; **in the case of H12O**, information about **ICU stay** could not be extracted and was not included. **Diagnostics and procedure** variables, encoded as ICD-10 (HM) or ICD-9 (H12O) codes, were one-hot-encoded taking only the first three digits. For **missing measurements**, we experimented with two different ways of imputation: (i) we used the previous day values of the same patient, when available, and the median value of of the dataset (excluding the test set) when not available and (ii) just used median value for all cases. Additionally, we also used the extra binary

missing values indicators and also generated extra datasets without this feature. For the **laboratory** determinations, we tried two methods: applying reference values and normalizing between [0,1].

As we can see for both HM and H12O, we experimented and generated different dataset representations identified as follows:

- *base*: when using mean for missing measurement imputation.
- *imputation*: when using previous value of a feature for that patient (if available), and mean of the sample (if not), for missing measurement imputation.
- *missing*: a new binary feature is added to indicate if the original value is missing following Che et al. [19].
- *imputation/missing*: previous two representations together.
- *reference*: when using reference values in determination results.

Finally, for a given patient, we joined all static features together to form the final static feature vector $\mathbf{x}_s$. Similarly, for a given patient and a given day, we joined all the dynamic features together to create a sequence of dynamic feature vectors $[\mathbf{x}_d^{t_1}, ..., \mathbf{x}_d^{t_n}]$ where t_i are the hospital days. Note that, as mentioned above, while creating the dynamic feature vectors, a few extra features may be generated to identify 'missing measurements'.

4.1 Data Cleaning

To guarantee the quality of the data we performed some data cleaning processes as described below:

- Temporal sequences of hospital stays are very varied in length. In order to reduce the length of the sequences, without removing the samples (i.e patients), sequences were cut to 20 events from outcome (last 20 events).
- Only patients that were sent home or resulted in death are taken into account. Other discharge reasons, like transfers to other hospitals, do not provide clear and explicit information about the outcome.
- For all variables we perform an IQR for each column; we define Q1 as a quantile of 0.01 and 0.03, and Q3 as a quantile of 0.99 and 0.97, respectively for HM and H12O, having $IQR = Q3 - Q1$, we set to missing values those values that were not between $[Q1 - 1.5 * IQR, Q3 + 1.5 * IQR]$.
- Binary columns, with values showing a frequency lower than 1% are removed, once the dataset is generated. Notice that this can include missing indicator columns.
- In HM the **medication** table, "XXXXX"[9] codes and codes starting with "D"[10] were removed, since they represent nested medications and dermatological medications, respectively, that do not provide additional meaningful information as stated by the medical doctor.

As a result of this process, the final dataset includes samples from 1,939 and 3,735 patients for HM and H12O respectively. For HM and H12O, the resulting vectors have different sizes as we generated different datasets combining different strategies[11], the default datasets have 54 and 55 features in $\mathbf{x}_s$, 177 and 246 in $\mathbf{x}_d$, label vector $\mathbf{y}$ contains 1 column. In the eventual datasets, the mortality rate increased in HM to 16.92% (328 out of 1,939 patients died) and in H12O maintained unchanged 13.38% (500 out of 3,735 died).

5 Model

Motivated by the temporal nature of our problem and the effectiveness of the RNN models in the medical domain [20, 21, 22], we propose a RNN-based model to monitor the mortality risk of the patient by producing a daily prediction during the patient's hospital stay. To this purpose, we feed each daily records of temporal and static data day-by-day to output a prediction. The resulting RNN model is a combination of several modules. In the next section, we describe the architecture in details.

5.1 Architecture

As depicted in Figure 4, we designed an Artificial Neural Network with four modules, namely, the embedding module (grey boxes), the recurrent module (dark green boxes), the classifier module (light green boxes) and, optionally, the attention modules (blue boxes).

[9] "XXXXX" code had 1,289 appearances.

[10] Codes starting with "D" had 6,904 appearances.

[11] See Section 6.4.

The embedding module only acts on the static vector through a fully-connected layer that encodes the high-dimensional sparse feature vector into a lower dimensional dense vector.

The recurrent module accounts for both the encoding and the memorization of the temporal information provided by the dynamic vectors (red boxes). It consists of an unidirectional[12] RNN, built with either an LSTM or a Gated Recurrent Unit (GRU) [23]. At each day, the RNN cells process the relevant information of the patient dynamic data and produce an output vector that stores the relevant clinical information until that day to perform the predictions.

The attention module of dynamic field finds the correlations of all previous RNN's outputs and merges all the global relevant information of the sequence until a given day. First, the attention module creates a context vector as a linear combination of the RNN's outputs using the dot-product of the attention scores as in [24]. Then, at every day, the context vector and the current RNN output are concatenated and fed to a fully-connected layer to which it was applied a hyperbolic tangent function to get the final attention vector[13]. The attention module on the static field finds correlations of unembedded static data with respect to the dynamic hidden layer. As we will discuss in the next sections, in our work, the attention module is the component responsible for the interpretability of the model.

Finally, the classifier module consists of two fully-connected layers followed by a sigmoid activation function to produce the binary mortality predictions. We placed the classifier module on top of the RNN cells and the static embeddings to predict the mortality risk at each day.

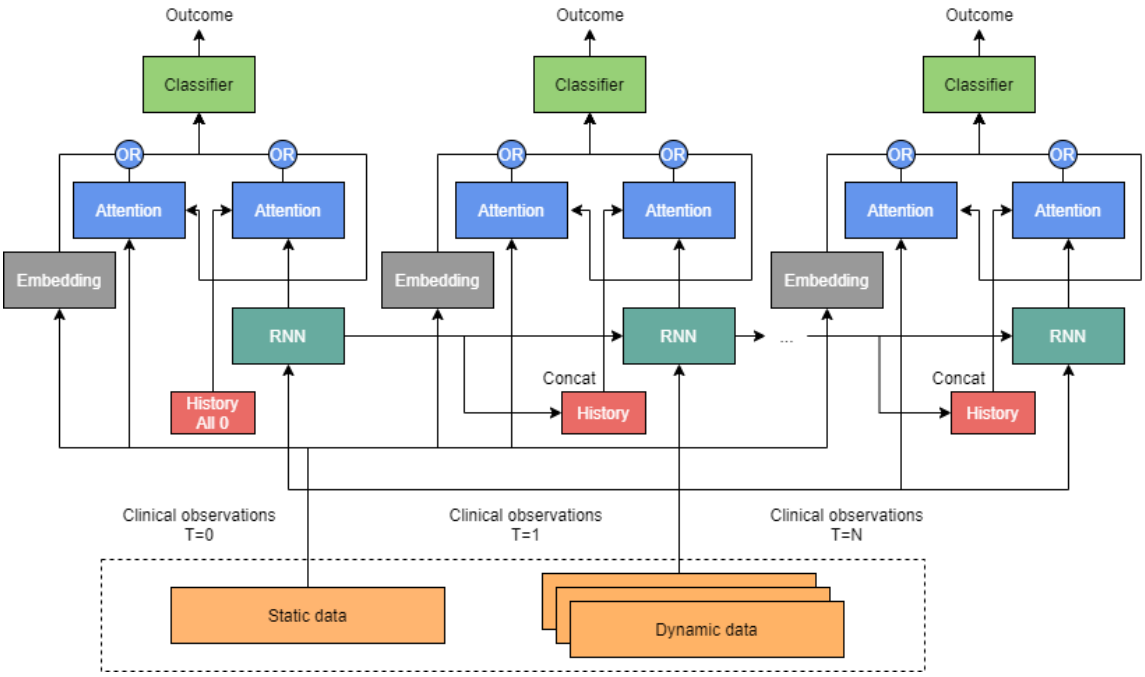


Figure 4: Architecture of the RNN described in Section 7.2.

5.2 Training and Inference

We designed a model to monitor the mortality risk of the patient by producing a daily prediction during the patient's hospital stay. Therefore, unlike standard training scheme for recurrent models, we designed an ad-hoc training scheme to produce daily predictions. Specifically, we do *not* feed the entire sequence of dynamic data into the RNN to output a single prediction. But, we feed each daily records of temporal and static data day-by-day and output a prediction for each day. Furthermore, to mitigate the impact of the class-imbalanced data on the learning process, we employed

[12]Given that the order of the causal relationship between the clinical event is strictly temporal, bidirectional RNN layers are not suitable for our use case.

[13]The implementation is available in PyTorch: `https://pytorchnlp.readthedocs.io/en/latest/_modules/torchnlp/nn/attention.html`

the Focal Loss introduced in [25] besides the classical Vanilla Binary Cross-Entropy. The Focal Loss was originally developed for computer vision as a strategy for countering class imbalance. Based on the assumption that the most frequent label will generate more confident predictions, the Focal Loss reshapes the Vanilla Cross-Entropy through two parameters (α and γ) loss such that well-classified samples have less weight than the incorrectly classified ones. During training, for the Focal Loss we fixed α to 1 and used γ as hyperparameter.

In addition, we experimented with other model's hyperparameters configurations, as detailed in Section 6.4. All the layers are trained end-to-end with stochastic gradient descent.

At inference, the model is fed with the static vector and then, at each time step, we input the corresponding dynamic vector and the model outputs the probability of patient mortality. To obtain the mortality predictions, the probability is discretized into a binary label using a threshold, as in logistic regression. We set the threshold to the default value of 0.5. Notice that, after training, other values of the model's threshold can be used, depending on how much we want to be confident in the prediction. For the metrics used for evaluation, see Section 6.5.

5.3 Interpretability

As noted in section 5.1, the attention layer is the building block of the interpretability in our model. The attention module of dynamic field relates the RNN's outputs to the RNN's output history vectors to retrieve meaningful past dynamic information for the daily prediction. The idea behind is to identify which days in the previous sequence are influencing most the prediction of the current day, since they are storing the most relevant clinical information for the outcome. The attention can be useful as a starting point to learn which clinical variables (e.g. medications) could have played a critical role. See the animated figure[14] to visualize how the attention scores dynamically change during the training process to focus on the most relevant days for a given patient.

6 Experimental Framework

We conducted extensive experiments within an experimental setting that includes: different dataset splits, a feature selection mechanism, two competitive baselines, a proper model's hyper-parameter tuning, an adequate model selection and exhaustive evaluation methods. In particular, we want to point out the role of model tuning as a necessary step considering our case. Since we designed our network from scratch and applied it on a dataset not yet studied[15], a suitable assessment of the model's capability through a functional exploration of hyperparameters was required. Analogously, a thorough evaluation helped in testing the robustness of the model given our small dataset size. In the next sections, we describe the experiments in details.

6.1 Dataset Splits

We created two dataset splits that were applied in different experimental settings, as follows:

- Initial split: We splitted the dataset in train (TRAIN-GRID), validation (VALID-GRID) and TEST sets, with proportions of 80% for training, 10% for validation, and 10% for test. The dataset is split randomly[16]. The split is stratified as in [26], that is, respecting the proportion of the labels in the different subsets [27], which is crucial in the case of class imbalanced datasets. We use the TRAIN-GRID and VALID-GRID sets to perform a grid search of the hyperparameters. The TEST set is reserved and only used for the final evaluation of the system.
- Cross-validation split: Using the train and validation sets of the initial split, we apply a 5-fold cross-validation, with the same proportions as before, and with stratification as well (the size of each train and valid sets is the same to the ones used in the grid search). To avoid repeating the split used in the grid search, we used a different seed this time, obtaining 5 different splits for train (TRAIN-CV) and validation (VALID-CV) sets. We train each fold on the TRAIN-CV set, stopping based on the score of the appropriate metric on the VALID-CV set. The validation average score is used for selecting the best model, and then the TEST set (unused until now) is used for evaluating it.

[14] Attention mechanism animation: `https://drive.google.com/file/d/14ivLc1IRql6V9dT-cdddbdpjTbgn0WCN/view?usp=sharing`

[15] Unlike e.g. textual data and standard Deep Learning models in NLP, which are well-known and have been studied in many works.

[16] Making sure to use a fixed random seed, for the sake of reproducibility.

HM	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Alive	161	161	157	151	133	117	94	81	63	49	42	38	36	29	24	21	17	12	7	6
Death	33	33	30	29	27	25	23	19	16	13	12	12	8	8	7	7	6	4	4	4
H12O	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20
Alive	324	324	324	317	293	257	221	177	138	117	102	86	78	63	56	49	45	37	32	29
Death	50	50	50	47	39	34	29	24	21	16	15	13	13	12	11	10	9	8	6	6

Table 2: Number of occurrences by days for each label.

All the models in our work are trained and evaluated with the same splits. The schema of the dataset split is shown in Figure 5.

Table 2 shows the number of occurrences by days for each label in the TEST set. This table evidences the differences in death ratios in the two datasets.

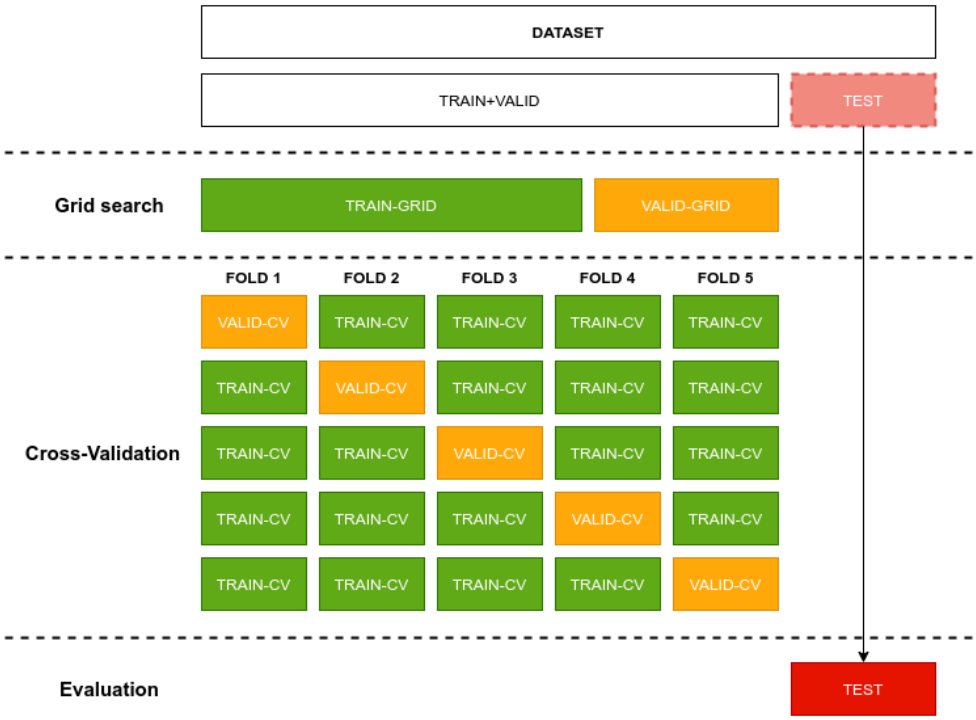


Figure 5: Dataset split of our experimental framework. The size of the train and validation sets is the same for each fold and for the grid search. None of the validation sets used in the CV is the one used in the grid search.

6.2 Feature Selection

Given the small number of examples and the resulting high-dimensionality of the feature vectors, we decided to use features selection techniques to reduce the dimensionality by removing irrelevant features and, thus, avoid the problems related to the curse of dimensionality and overfitting.

We started with the removal of binary columns with values less than $threshold = 1\%$ as explained in Section 4.1 for both static and dynamic vectors. This removes variables with insufficient variance.

Following [28], for dynamic variables, we used metrics of information theory, namely, entropy, information gain, Gini index and chi-square to rank features according to their deemed importance. Next, we select the final features by joining together the first top-k features for each metric. The result is a list of N features that we expect to be relevant for the

classification problem. The number of the first top-k is a threshold that determines how many features are selected and we considered it as another RNN model's hyperparameter. For computing the metrics we did not use the test set as it should be maintained unseen until evaluation.

6.3 Baseline Models

We trained two classical ML models: Random Forest[17] (RF) and Support Vector Machine for classification[18] (SVC). For each model, we designed two experiment types, EXP1 and EXP2, depending on how the data are fed to the model. In EXP1, for each day, we concatenated the static and the dynamic vector. As a consequence, we have independent data representations for each day, where each event of a patient generates a sample (i.e. a patient with a stay of 17 days generates 17 samples).

In EXP2, for each day, we merged the information from all the previous days into a fixed-size vector. We obtained the vector by performing min, max and average operation between the different feature vectors and concatenating them to the static and dynamic vectors[19]. Concretely, given a static vector $\mathbf{x}_s$ and the sequence of dynamic vectors $[\mathbf{x}_d^{t_1}, ..., \mathbf{x}_d^{t_n}]$, the merged vectors for each day $k \in [1, N]$ are computed as follows:

$$
\begin{aligned}
\mathbf{x'}^{t_1} &= \mathbf{x}_s \oplus \mathbf{x}_d^{t_1} \oplus min(\mathbf{x''}_d^{t_1}) \oplus max(\mathbf{x''}_d^{t_1}) \oplus avg(\mathbf{x''}_d^{t_1}) \\
&... \\
\mathbf{x'}^{t_k} &= \mathbf{x}_s \oplus \mathbf{x}_d^{t_k} \oplus min(\mathbf{x''}_d^{t_1}, ..., \mathbf{x''}_d^{t_k}) \oplus max(\mathbf{x''}_d^{t_1}, ..., \mathbf{x''}_d^{t_k}) \oplus avg(\mathbf{x''}_d^{t_1}, ..., \mathbf{x''}_d^{t_k}) \\
&... \\
\mathbf{x'}^{t_N} &= \mathbf{x}_s \oplus \mathbf{x}_d^{t_N} \oplus min(\mathbf{x''}_d^{t_1}, ..., \mathbf{x''}_d^{t_N}) \oplus max(\mathbf{x''}_d^{t_1}, ..., \mathbf{x''}_d^{t_N}) \oplus avg(\mathbf{x''}_d^{t_1}, ..., \mathbf{x''}_d^{t_N})
\end{aligned} \tag{1}
$$

where the min, max and avg functions are applied on the same vector's components in the sequence, and **x''** represents the dynamic vectors after removing the 'missing' features.

In this way, the resulting sequence of merged vectors $[\mathbf{x'}^{t_1}, ..., \mathbf{x'}^{t_n}]$ is supposed to propagate information across all the days of the patient's hospital stay. The purpose of this experiment was to train a model that was at some extent equivalent to temporal RNN-based model and to make a comparison with our proposed recurrent model.

Notice however, that since the **results with EXP2 were systematically worse we decided not to include them**. In Section 6.4, we explain how EXP1 is used in the baseline models. The results of the baselines are included in Section 7 for comparison with our model. The function included in *Scikit-learn* to perform the cross validation[20] does not return the predictions, only the scores; therefore, some day-by-day results may show minor inconsistencies.

6.4 Experiments

To test the capabilities of our model, we organized the experiments in two phases. First, we used the model's hyperparameter tuning to select the optimal values that improve the learning process. Then, we used the model selection to find out which are the best models based on the evaluation metrics and statistical tests.

We ran all the experiments in the CTE-Power[21] High-Performance Computing resource equipped with Nvidia V100 Graphical Processing Units and IBM Power9 8335-GTH Central Processing Units.

Model Selection and Tuning: We employed a grid search technique consisting of an exhaustive search over a manually specified configuration of hyperparameters representing a subset of the hyperparameter space and trained a model for each configuration. Table 3 shows the complete list of hyperparameters for RNN models.

For the RNN, we performed the grid search by training for 200 epochs with early stopping on validation accuracy and F1. We set the the early stop patience to 10 steps. For the baseline models, we executed a grid search using the EXP1 training strategy with the TRAIN+VALID split (see Sec. 6.1). Next, we selected the top-30 best models (for both baselines and RNN) with the highest scores on the VALID-GRID set. In the case of H12O, we selected the top-30 best models for each combination of the Feature Representation strategies explained in Section 4.

[17]RF: `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`

[18]SVC: `https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html`

[19]In order to reduce the number of features, we removed the 'missing' features before computing the min, max, and average vectors ('missing' features are still in the dynamic vector that represents the current day).

[20]Cross validation: `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.cross_validate.html`

[21]`https://www.bsc.es/user-support/power.php`

Finally, we employed cross-validation using the cross-validation split explained in 6.1. We used the average validation accuracy or F1 (depending on the early stopping metric) of the selected models to find the best one (i.e. the one that shows the higher generalization capabilities).

Attention Analysis: In order to understand how attention mechanism works for dynamic data in our model, we identified and characterized attention peaks. We hypothesize that the attention mechanism should target relevant days to perform correctly. If, for a given patient, there is a relevant day for the prediction, the attention mechanism should always focus on that day (during the whole sequence). Since the RNN runs with daily data, for each day, the model considers the attention of the dynamic embedding of the current day with respect to the previous days. This results in a 2-dimensional matrix.

We first start by finding peaks in the second dimension (i.e. grouping by the target day). Then, we group the days and we count the number of times they are peaks. Since first days appear more times than last days, a normalization is performed. For instance, the first day appears as many times as the length of the sequence; on the contrary, the second to last day only appears once. We normalize by doing $day_count_norm = day_count/(patient_length - day)$.

For this analysis, we use the best RNN with attention model and the test set of the TRAIN+VALID split.

RNN's params	SVC's params	RF's params
Batch size	Number of estimators	C value
Dropout	Criterion	Kernel
Static embedding size	Minimum sample split	Tolerance
Dynamic embedding size	Class weighting	Gamma
RNN hidden state size		Degree
RNN architecture (either LSTM or GRU)		Class weighting
Number of RNN layers		
Loss function (and γ value for the Focal Loss)		
Early stopping metric		
Attention mechanism		
Learning rate		
Number of features (top-k)		

Table 3: Hyperparameters used in the grid search.

6.4.1 Ensemble

With the motivation of increasing the performance, stability, and robustness to noise, we built two ensembles of RNNs. The first aims at maximizing F1 score (Ensemble-F1), and the second focuses on sensitivity (Ensemble-SEN). The latter had some constraints to avoid the trivial solution of declaring all patients as dead.

In order to build the ensembles, we aggregated the predictions of the models following a voting schema. Specifically, for a given patient and the corresponding votes of the models, the final prediction of the ensemble is defined as follows:

$$\texttt{daily_outcome_patient}(\texttt{votes}) = \begin{cases} \texttt{'Dead'}, & \text{if } \frac{\texttt{count(votes = 'Dead')}}{\texttt{count(votes)}} \geq \lambda \\ \texttt{'Alive'}, & \text{otherwise} \end{cases}$$

where λ is the threshold of the percentage of models required to predict whether a given patient will die. It follows that for a given patient and ensemble, the lower the λ, the better (or at least equal) the sensitivity.

To build the ensembles, we proceed with the following steps:

1. First, we pre-select the top 100 models, as per cross-validated accuracy or F1 score. This cut-off guarantees that the models integrating the ensemble will meet a reasonable performance requirement.
2. Then, we select 20 of them, using Algorithm 1. Instead of just taking the top 20 best models as per cross-validated F1 or accuracy, we define an algorithm that heuristically maximizes the differences in predictions in the cross-validation folds of the different models. Otherwise, it could be the case that all models made too similar predictions, making the ensemble unnecessarily grow in number of models.
3. Finally, we optimize the number of models and the aggregation threshold, λ. Specifically, we apply a grid search over the following values:

- `Number of models` $= \{2..20\}$: We need at least 2 models to build an ensemble, but having a large number of models might be infeasible in real-life settings, so, as we said, we restricted the ensemble to a maximum of 20 models. We experiment with adding the models in order from Algorithm 1, and discard the ones that do not improve.
- $\lambda = \{0.1, 0.2, 0.3, 0.4, 0.5\}$: One of the main goals was to improve the sensitivity, therefore it made no sense to try values larger than 0.5.

Thus, the final configuration maximizes the F1 score, in case of Ensemble-F1, or sensitivity, in case of Ensemble-SEN.

Algorithm 1: Selection of n candidate models for ensemble, with a cut-off of top models

```
Result: candidate_models
Input: n: integer
Input: top: integer
Input: metric: {'f1', 'accuracy'}
candidate_models ← List();
candidate_models_indices ← List();
all_predictions ← get_flattened_all_predictions_in_cv_sorted_by_model_score_in_cv(metric)[:top] ;
prediction_similarity_matrix ← distance_matrix(all_predictions, all_predictions) ;
all_models ← get_all_models_sorted_by_cv_score(metric)[:top];
best_model ← all_models[0];
candidate_models.insert(best_model);
candidate_models_indices.insert(0);
while candidate_models.length() < n do
    max_distance ← 0;
    max_index ← null;
    for i = 0; i++; i < all_models.length() do
        new_candidate ← all_models[i];
        if new_candidate in candidate_models then
            break;
        end
        for j = 0; j++; j < candidate_models.length() do
            distance ← distance + prediction_similarity_matrix[i, candidate_models_indices[j]];
        end
        if max_index is null or distance > max_distance then
            max_distance ← distance;
            max_index ← i;
        end
    end
    candidate_models_indices.insert(max_index);
    candidate_models.insert(all_models[max_index]);
end
```

6.5 Metrics and Evaluation

To evaluate the performance of the models and the consistency of the model's predictions, we considered the following metrics:

- Accuracy: measures the number of correct predictions over the total number of cases.
- F1 score: harmonic mean of the model's precision and recall. For this project, we use a macro F1 score, computing the metric for each label, and then finding the unweighted mean. Macro F1 does not take label imbalance into account, and therefore the score for each class counts up to half of the total score.
- Sensitivity: measures the capacity to correctly predict mortality, also known as recall.
- Specificity: proportion of the capacity to correctly predict survivals.
- Area Under the Curve (AUC): probability of a random example with true label of fatality to get a higher score than a random example with true label of survival.

When computing the **global performances**, for each of the metrics above, we used the average of metric computed at every day in the patient's sequence for all patients.

	Day from admission														
Patient	Day 1	Day 2	Day 3	Day 4	Day 5	Day 6	Day 7	Day 8	Day 9	Day 10	Day 11	Day 12	Day 13	Day 14	...
1	0	0	0	0	0	0	0	0	1	1	1	1	1	1	
2	0	0	0	1	0	0	0	0	0	0					
3	0	0	1	1	0	1	1	1							
4	0	0	0	1	1	1	1								
5	0	0	0	1	1	1									
...	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score

	Days before outcome														
Patient	...	Day 14	Day 13	Day 12	Day 11	Day 10	Day 9	Day 8	Day 7	Day 6	Day 5	Day 4	Day 3	Day 2	Day 1
1		0	0	0	0	0	0	0	0	1	1	1	1	1	1
2						0	0	0	1	0	0	0	0	0	0
3								0	0	1	1	0	1	1	1
4									0	0	0	1	1	1	1
5										0	0	0	1	1	1
...	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓	↓
	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score	Score

Figure 6: Example of system's predictions from admission and from outcome.

In addition, and more interestingly, we designed specific evaluation metrics to better test the model and reflect our intended use case providing daily predictions. In this case, we get the prediction vectors already returned by the system, and use them to calculate the daily performances as follows:

- **Daily performances from the admission** computes the performance every day from the admission day.
- **Daily performances from the outcome** computes the performance every day from the outcome day.

Notice that, in daily performance calculations, the system ignores whether the patients are long-term patients or not and there is no leakage of other past or future events; we just evaluate on the predictions already done by the system.

Figure 6 shows an example of how day by day predictions are calculated. To compute the daily performances from admission, we use the predictions of each patient, starting from the left. To compute the daily performances from the outcome, first we align all the predictions to the right, and then, we use the predictions of each patient, starting from the right. In this case, Day 1 is the outcome day.

7 Results

For **HM**, Table 4 shows the final global evaluation results (average in TEST) on cross-validation, comparing the best RNN models for each "data representation method" described in Section 4 and the best ensemble models. All models have two versions depending on whether accuracy or F1 are used for early stopping and model selection. We also experimented with feature selection using three different thresholds (100, 150 and 300) and, for each representation method, we give the best results. The best 'non-ensemble' model gets 90.72% in accuracy, 67.75% in sensitivity and 83.74% in F1. This winner model only uses imputation method, does not need feature selection and it is better in all metrics but specificity. In general, feature selection does not have a significant impact in the results. Ensemble models show a remarkable gain in sensitivity, reaching an impressive 84.50% with virtually no negative impact on F1. Ensemble models that use sensitivity as discard criteria, show much better results in sensitivity (84.50 and 84.25 vs. 74.94 and 76.56) and almost unnoticeable fall in F1. The fact that the ensemble mechanism applies on the top 100 models guarantees good results.

Table 5 shows the final global evaluation results on cross-validation, comparing the best SVC, RF, and RNN models (all of them selected as per results in VALID-CV set); complete results by fold in sensitivity are provided in Table 6.

Sections 2.1 to 2.3 in the additional material include the tables with the daily results of the selected RF, SVC and RNN, respectively, starting from the admission in the hospital and starting from the outcome. Here, Figure 7 summarizes the results.

Imp	Mis	Ref	FS	Met	Accuracy	AUC	Sensitivity	Specificity	F1
×	×	×	×	Acc	0.8933±0.0062	0.9166±0.0118	0.5319±0.0890	**0.9760±0.0153**	0.7917±0.0261
✔	×	×	×	Acc	0.8982±0.0056	0.9385±0.0063	0.6194±0.0806	0.9620±0.0129	0.8152±0.0203
✔	✔	×	×	Acc	0.8917±0.0092	0.9271±0.0082	0.6094±0.0844	0.9563±0.0183	0.8049±0.0206
✔	✔	✔	×	Acc	0.8892±0.0107	0.9317±0.0062	0.5569±0.0931	0.9653±0.0100	0.7912±0.0330
✔	×	✔	×	Acc	0.8916±0.0050	0.9216±0.0100	0.5569±0.0479	0.9681±0.0080	0.7956±0.0152
×	✔	×	×	Acc	0.8911±0.0102	0.9169±0.0065	0.5431±0.0888	0.9707±0.0183	0.7912±0.0284
×	✔	✔	×	Acc	0.8937±0.0087	0.9222±0.0112	0.5506±0.0467	0.9721±0.0105	0.7974±0.0193
✔	×	×	×	Acc	0.8962±0.0062	0.9225±0.0114	0.5731±0.0758	0.9701±0.0187	0.8046±0.0172
×	×	×	×	F1	0.8982±0.0036	0.9247±0.0088	0.5931±0.0594	0.9680±0.0103	0.8112±0.0147
✔	×	×	×	F1	**0.9072±0.0034**	**0.9408±0.0062**	0.6775±0.0228	0.9597±0.0041	0.8374±0.0073
✔	✔	×	×	F1	0.8949±0.0059	0.9269±0.0147	0.6738±0.0783	0.9455±0.0131	0.8195±0.0196
✔	✔	✔	×	F1	0.8824±0.0091	0.9259±0.0090	0.5713±0.0876	0.9535±0.0134	0.7853±0.0276
✔	×	✔	×	F1	0.8938±0.0056	0.9324±0.0071	0.5994±0.0737	0.9611±0.0114	0.8060±0.0194
×	✔	×	×	F1	0.8888±0.0048	0.9184±0.0138	0.5794±0.0758	0.9595±0.0143	0.7955±0.0199
×	✔	✔	×	F1	0.8967±0.0031	0.9287±0.0107	0.6150±0.0513	0.9611±0.0123	0.8132±0.0099
✔	×	×	×	F1	0.8925±0.0049	0.9196±0.0149	0.6031±0.0368	0.9587±0.0049	0.8057±0.0124
×	×	×	100	Acc	0.8909±0.0080	0.9135±0.0200	0.5556±0.0712	0.9675±0.0089	0.7938±0.0255
✔	×	×	300	Acc	0.8990±0.0044	0.9395±0.0089	0.6563±0.0423	0.9545±0.0075	0.8230±0.0114
✔	✔	×	100	Acc	0.8942±0.0149	0.9413±0.0123	0.6288±0.0726	0.9550±0.0078	0.8118±0.0323
✔	✔	✔	150	Acc	0.8839±0.0056	0.9236±0.0090	0.5238±0.1068	0.9663±0.0202	0.7765±0.0293
✔	×	✔	150	Acc	0.8911±0.0046	0.9282±0.0055	0.6200±0.0773	0.9531±0.0185	0.8060±0.0150
×	✔	×	100	Acc	0.8933±0.0090	0.9063±0.0181	0.5481±0.0630	0.9723±0.0121	0.7960±0.0225
×	✔	✔	×	Acc	0.8937±0.0087	0.9222±0.0112	0.5506±0.0467	0.9721±0.0105	0.7974±0.0193
✔	×	×	100	Acc	0.8894±0.0046	0.9106±0.0111	0.5419±0.0204	0.9688±0.0071	0.7901±0.0076
×	×	×	×	F1	0.8941±0.0076	0.9164±0.0118	0.5875±0.0492	0.9643±0.0140	0.8050±0.0143
✔	×	×	×	F1	**0.9072±0.0034**	**0.9408±0.0062**	0.6775±0.0228	0.9597±0.0041	0.8374±0.0073
✔	✔	×	×	F1	0.8949±0.0059	0.9269±0.0147	0.6738±0.0783	0.9455±0.0131	0.8195±0.0196
✔	✔	✔	150	F1	0.8917±0.0076	0.9281±0.0016	0.6169±0.0679	0.9545±0.0144	0.8064±0.0191
✔	×	✔	100	F1	0.8881±0.0034	0.9263±0.0055	0.5906±0.0612	0.9561±0.0177	0.7972±0.0085
×	✔	×	100	F1	0.8867±0.0078	0.9093±0.0158	0.5756±0.0649	0.9578±0.0080	0.7924±0.0214
×	✔	✔	100	F1	0.8949±0.0111	0.9062±0.0134	0.6250±0.0311	0.9567±0.0076	0.8129±0.0196
✔	×	×	100	F1	0.8951±0.0064	0.9186±0.0121	0.6256±0.0515	0.9567±0.0120	0.8128±0.0137
RNN-Ensemble-F1				Acc	0.9054±0.0045	-	0.7494±0.0244	0.9411±0.0106	**0.8443±0.0041**
RNN-Ensemble-F1				F1	0.8990±0.0075	-	0.7656±0.0189	0.9295±0.0129	0.8380±0.0081
RNN-Ensemble-SEN				Acc	0.8896±0.0054	-	**0.8450±0.0075**	0.8998±0.0073	0.8351±0.0063
RNN-Ensemble-SEN				F1	0.8821±0.0036	-	0.8425±0.0157	0.8912±0.0067	0.8258±0.0038

Table 4: HM, best RNN models for each "data representation method" as described in Section 4.

Mod	Imp	Mis	Ref	FS	Met	Accuracy	AUC	Sensitivity	Specificity	F1
RNN	✔	×	×	×	F1	**0.9072±0.0034**	0.9408±0.0062	0.6775±0.0228	0.9597±0.0041	0.8374±0.0073
RF	×	×	×	-	Acc	0.9032±0.0032	0.9361±0.0009	0.6169±0.0109	**0.9687±0.0018**	0.8228±0.0061
SVC	✔	✔	×	-	F1	0.8889±0.0035	**0.9458±0.0035**	0.7806±0.0121	0.9137±0.0028	0.8269±0.0055
RNN	Ensemble-F1				Acc	0.9054±0.0045	-	0.7494±0.0244	0.9411±0.0106	**0.8443±0.0041**
RNN	Ensemble-SEN				Acc	0.8896±0.0054	-	**0.8450±0.0075**	0.8998±0.0073	0.8351±0.0063

Table 5: HM, final evaluation results on cross-validations.

Mod	Imp	Mis	Ref	FS	Met	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
RNN	✔	×	×	×	F1	0.6938	0.6406	0.6969	0.6844	0.6719
RF	×	×	×	-	Acc	0.6312	0.6094	0.6281	0.6031	0.6125
SVC	✔	✔	×	-	F1	0.7781	0.7844	0.8000	0.7781	0.7625
RNN	Ensemble-F1				Acc	0.7813	0.7188	0.7438	0.7375	0.7656
RNN	Ensemble-SEN				F1	**0.8500**	**0.8156**	**0.8438**	**0.8563**	**0.8469**

Table 6: HM, sensitivity by folds of RNN, RNN & attention, SVC and RF.

We ran Student's t-test and Wilcoxon signed-rank tests on sensitivity scores of Table 6, Student's t-test shows that there is a significant statistical difference in the 5-fold cross-validation sensitivity scores for RNN (mean 0.8442, SD 0.0157) and RF (mean 0.6168, SD 0.0122), with $t(4) = 27.3674, p = 0.008$. Also for the RNN with respect to the SVC (mean 0.7806, SD 0.0135) there is a significant difference, with $t(4) = 5.9812, p = 0.01$. In the Wilcoxon signed-rank, test results are positive and the same as previous Wilcoxon test on the accuracy, in both RNN over RF and RNN over SVC (with $z = -2.0226, p < 0.05$).

For **H12O**, Table 7 shows the final global evaluation results (average in test) on cross-validation, comparing the best RNN models for different "data representation methods" described in Section 4 and the best ensemble models. Since *imputation & missing* models showed better performance, we also experimented with feature selection using three different thresholds (100, 150 and 300). As in the case of HM, all models have two versions depending on whether accuracy or F1 are used for early stopping and model selection. Models using feature selection show slightly better results than those not using it and there is no significant difference between accuracy and F1 models. Again, ensemble models are remarkable better in sensitivity, reaching 80.82% and still keeping satisfactory F1 scores. Ensemble models using sensitivity as discard criteria are better than those using F1.

Imp	Mis	Ref	FS	Met	Accuracy	AUC	Sensitivity	Specificity	F1
×	×	×	×	Acc	0.9023±0.0061	0.9174±0.0026	0.6073±0.0799	0.9468±0.0185	0.7811±0.0088
✔	×	×	×	Acc	0.9186±0.0062	0.9338±0.0055	0.6462±0.0357	0.9597±0.0103	0.8145±0.0100
✔	✔	×	×	Acc	0.9195±0.0028	0.9411±0.0040	0.6458±0.0357	0.9608±0.0075	0.8157±0.0056
×	✔	×	×	Acc	0.9140±0.0061	0.9316±0.0055	0.6683±0.0880	0.9511±0.0180	0.8099±0.0128
×	×	✔	×	Acc	0.9029±0.0051	0.9161±0.0055	0.6056±0.0512	0.9477±0.0115	0.7821±0.0099
×	×	×	×	F1	0.9029±0.0041	0.9138±0.0114	0.6013±0.0767	0.9485±0.0151	0.7809±0.0117
✔	×	×	×	F1	0.9164±0.0061	0.9349±0.0058	0.6765±0.0704	0.9526±0.0169	0.8154±0.0069
✔	✔	×	×	F1	0.9228±0.0057	0.9396±0.0064	0.6765±0.0528	0.9599±0.0138	0.8261±0.0061
×	✔	×	×	F1	0.9202±0.0025	0.9356±0.0033	0.6739±0.0433	0.9574±0.0071	0.8213±0.0080
×	×	✔	×	F1	0.9036±0.0042	0.9166±0.0046	0.5853±0.0423	0.9516±0.0082	0.7793±0.0105
✔	✔	×	100	Acc	0.9236±0.0033	0.9392±0.0019	0.6112±0.0316	0.9707±0.0062	0.8167±0.0078
✔	✔	×	150	Acc	0.9246±0.0031	0.9466±0.0025	0.6700±0.0497	0.9630±0.0101	0.8281±0.0063
✔	✔	×	300	Acc	0.9197±0.0048	0.9387±0.0038	0.6596±0.0532	0.9589±0.0107	0.8182±0.0103
✔	✔	×	100	F1	0.9238±0.0017	0.9444±0.0021	0.6505±0.0244	0.9650±0.0041	0.8238±0.0049
✔	✔	×	150	F1	**0.9264±0.0039**	0.9474±0.0034	0.6609±0.0448	0.9664±0.0071	0.8297±0.0111
✔	✔	×	300	F1	0.9178±0.0018	0.9389±0.0020	0.6488±0.0511	0.9584±0.0084	0.8133±0.0081
RNN-Ensemble-F1				Acc	0.9201±0.0047	-	0.7434±0.0294	0.9468±0.0092	**0.8315±0.0052**
RNN-Ensemble-F1				F1	0.9198±0.0058	-	0.7356±0.0332	0.9475±0.0111	0.8299±0.0060
RNN-Ensemble-SEN				Acc	0.9097±0.0099	-	0.7879±0.0228	0.9281±0.0144	0.8217±0.0117
RNN-Ensemble-SEN				F1	0.9011±0.0106	-	**0.8082±0.0327**	0.9152±0.0163	0.8119±0.0115

Table 7: H12O, best RNN models for each "data representation method".

Table 8 shows the final global evaluation results for **H12O** on cross-validation, comparing the best SVC, RF, and the top RNN models (all of them selected as per results in validation set); complete results by fold in test accuracy and sensitivity are provided in the Table 9.

Mod	Imp	Mis	Ref	FS	Met	Accuracy	AUC	Sensitivity	Specificity	F1
RNN	✔	✔	×	150	Acc	0.9246±0.0031	0.9466±0.0025	0.6700±0.0497	0.9630±0.0101	0.8281±0.0063
RF	✔	✔	×	-	Acc	**0.9341±0.0010**	**0.9504±0.0012**	0.6255±0.0102	**0.9806±0.0015**	**0.8380±0.0030**
SVC	✔	✔	✔	-	Acc	0.9203±0.0005	0.9413±0.0024	0.6397±0.0087	0.9626±0.0011	0.8161±0.0021
RNN	Ensemble-F1				Acc	0.9201±0.0047	-	0.7434±0.0294	0.9468±0.0092	0.8315±0.0052
RNN	Ensemble-SEN				F1	0.9011±0.0106	-	**0.8082±0.0327**	0.9152±0.0163	0.8119±0.0115

Table 8: H12O, final evaluation results on cross-validations.

Mod	Imp	Mis	Ref	FS	Met	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5
RNN	✔	✔	×	150	Acc	0.6241	0.6976	0.6220	0.7386	0.6673
RF	✔	✔	×	-	Acc	0.6242	0.6199	0.6307	0.6112	0.6415
SVC	✔	✔	✔	-	Acc	0.6523	0.6350	0.6415	0.6263	0.6436
RNN	Ensemble-F1				Acc	0.7192	0.7257	0.7322	0.7927	0.7473
RNN	Ensemble-SEN				F1	**0.7711**	**0.8035**	**0.7862**	**0.8531**	**0.8272**

Table 9: H12O, sensitivity by folds of RNN, RNN & attention, SVC and RF.

We run the same significance tests as in HM for H12O for sensitivity of Table 9, Student's t-test shows that there is a significant statistical difference in the 5-fold cross-validation sensitivity scores for RNN (mean 0.8082, SD 0.0326) and RF (mean 0.6255, SD 0.0114), with $t(4) = 10.9826, p = 0.017$ and also for the RNN with respect to SVC (mean 0.6397, SD 0.0097) there is a significant difference, with $t(4) = 9.2303, p = 0.018$. In the Wilcoxon signed-rank, test results are positive in both RNN over RF and RNN over SVC (with $z = -2.0226, p < 0.05$).

For **H12O**, Sections 3.1 to 3.3 in the additional material include the tables with the daily results of the selected RF, SVC and RNN, respectively, starting from the admission in the hospital and starting from the outcome. Here, we summarize the information in the Figure 8.

Global performance:

For **HM**, the best ensemble RNN model is significantly better than the other models when considering sensitivity, reaching an impressive **0.84 in sensitivity** with no penalty in F1 and a small impact in accuracy. These spectacular results are achieved in all the folds, demonstrating its consistency and robustness. All RNN models show slightly better scores in accuracy and F1 when compared to baselines, which is a positive result taking into account that F1 is specially useful when having an uneven class distribution. SVC results are surprisingly performant in sensitivity, clearly surpassing those of the RF, the non ensemble RNN and the Ensemble-F1 models.

All HM models (60) use **attention** and, as described in the attention peak analysis in Section 6.4, the results show that 56.59% of the patients had relevant peak days.

For **H12O**, the RNN ensemble model outperforms all models in sensitivity. In particular, the Ensemble-SEN model outperforms RF in 18 points and SVC in 16 points reaching a spectacular **0.80 in sensitivity**. Again, for sensitivity, the results are the same in all folds. Although RF is slightly better in accuracy and F1, the difference is minimal. As for **attention**, it is worth mentioning that 53 out of the top 60 RNN models use attention and the best non-ensemble RNN model had 66.36% patients with peak days.

Figure 9 compares the results in the two datasets. We can observe that for accuracy, specificity and F1 the models show similar results in both datasets. However, for sensitivity they show appreciable differences: RF shows a similar low performance in both datasets; RNN is better in HM but still above 0.80 in H12O; but SVC shows a dramatic drop in H12O, with a performance that is practically on par with the RF. We have to be cautious when comparing two different datasets, but it seems that with more features and more examples, the SVC fails to generalize.

Day-by-day performance:

When evaluating day by day **from the admission** to hospital, we have longer sequences as we go on. Thus, at each new step the model uses more information and gets closer to the outcome. As Figure 7 shows, in **HM**, this has small positive

effect for RF and SVC: accuracy, specificity and F1 curves are high and rather flat with no noticeable improvement. For sensitivity, the RF shows a moderate improvement until day 7, a slight fall from day 12 onward and a sudden peak on day 18. SVC shows a long and strong performance curve, starting at 0.70 with a sudden rise on day 17.

The RNN-model shows a similar behaviour, with high, rather flat curves for accuracy, specificity and F1. However, the model performs better with respect to sensitivity, with a curve starting at 0.60, a peak of 1.00 on day 10, and then stabilizing between 0.85 and 1.00 until the last day.

When evaluating **from the outcome**, we have shorter sequences as we evaluate backwards. In this case, at each new step, the model has less information, and it is farther away from the outcome. Notice also that, the farther we are from the outcome, the fewer examples we have. As shown in Table 2, this considerably reduces the number of patients labeled with *death*, which may cause fluctuations, especially in sensitivity.

For **HM**, all models show similar descending curves for accuracy, specificity and F1. For sensitivity, SVC has a sharper decrease. The RNN model shows a higher and flatter curve in sensitivity, starting at 0.97, ending at 0.55 and maintaining a performance of 0.79 or better untill day 14. This shows that the system is efficient at making early predictions, since it provides reliable mortality predictions up to 14 days before the outcome.

For **H12O**, the day-bay-day performance **from the admission** is similar to that of HM. Thus, in terms of sensitivity, the RNN shows a substantial improvement with respect to the baselines. RNN mainly leads the sensitivity curve and it is very performant on both F1 and accuracy which confirms the enhancement.

When evaluating **H12O from the outcome**, the RF and the SVC show a similar behaviour: the specificity curve keeps high and flat, accuracy shows a moderate decrease, F1 suffers a sharper drop and sensitivity plummeted. Again RNN show a higher curve and a better performance in sensitivity, maintaining a score of 0.64 or better until 7 days before the outcome,

As in the case of HM, sensitivity curves from the outcome have greater fluctuations. As mentioned before (see Table 2) the gradual reduction of data and death labels (in absolute terms) as we move forward makes the sensitivity curves more unstable.

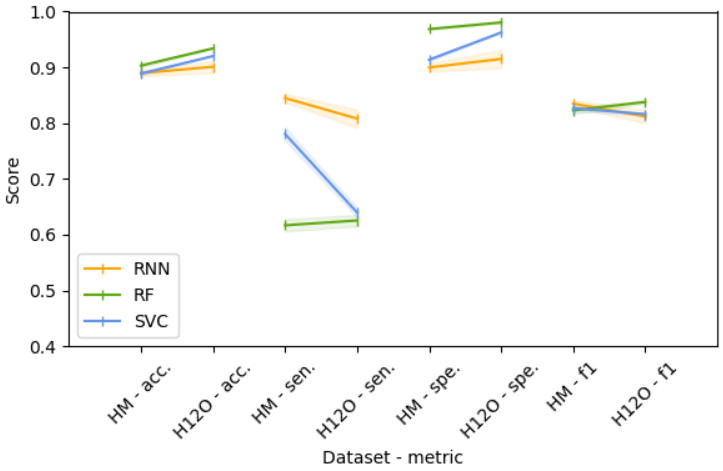


Figure 9: HM-H12O comparison in accuracy, sensitivity, specificity, and F1.

7.1 Hyperparameters

Table 10 show the best hyperparameters for RF and SVC baselines.

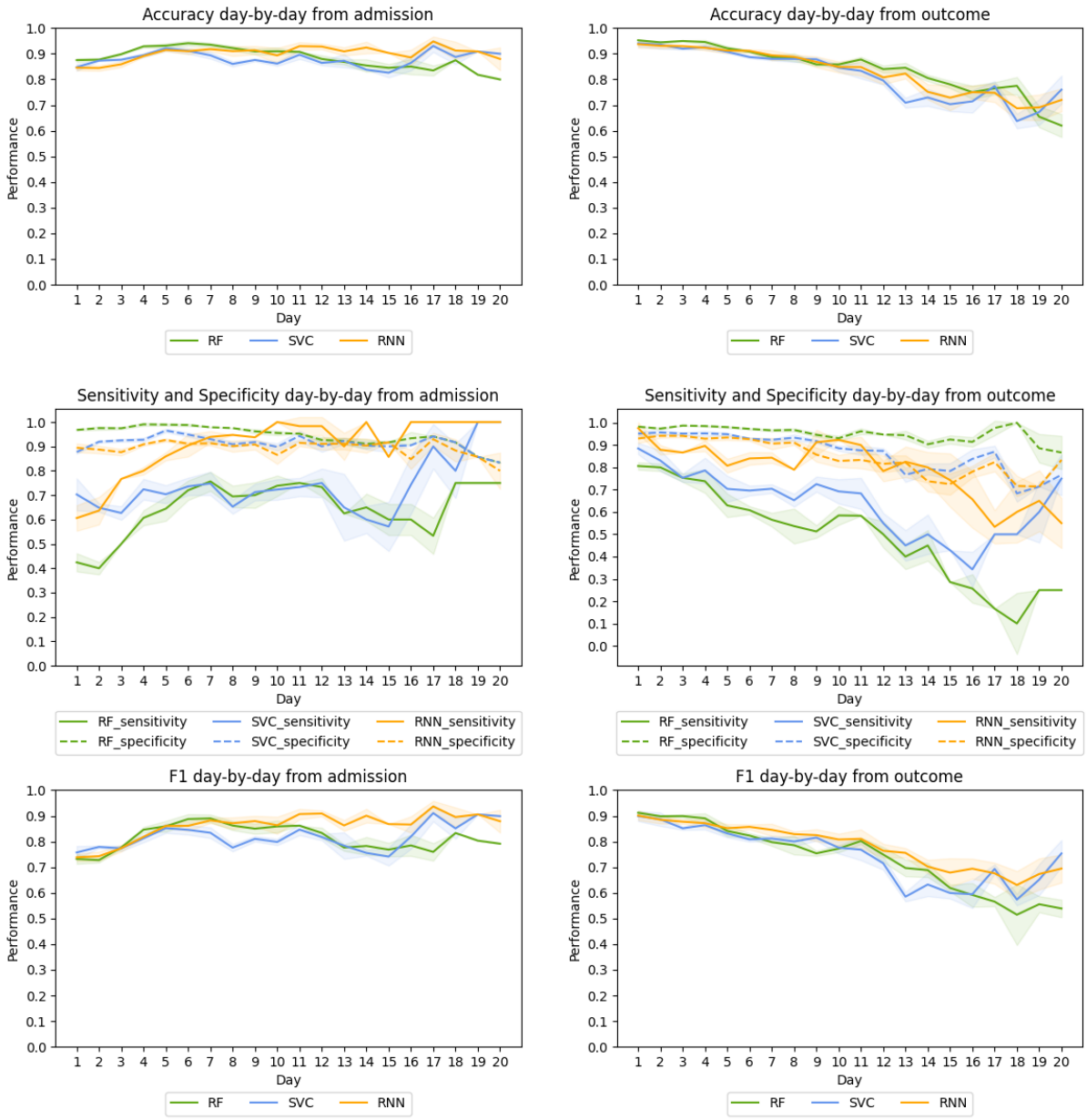


Figure 7: HM, Average day by day test performance from the admission (left) and from the outcome (right) in the cross-validation for the best RF, SVC, and RNN-based models.

RF Parameter	HM	H12O	SVC Parameter	HM	H12O
Class weighting	Balanced	None	C	3	1.2
Criterion	Gini	Gini	Class weighting	None	None
Minimum sample splits	10	9	Gamma	Scale	Scale
Nº estimators	125	175	Kernel	RBF	Poly
			Tolerance	0.01	0.01

Table 10: HM and H12O hyperparameters for RF and SVC baselines.

Figure 8: H12O, Average day by day test performance from the admission (left) and from the outcome (right) in the cross-validation for the best RF, SVC, and RNN-based models.

In the case of the RNN-based models, in the top 60 configurations (top 30 by accuracy plus top 30 by F1 combined, in the validation set) for each dataset, we observed that:

- Most configurations used attention: 60 models in HM, 53 in H12O.
- Most configurations used LSTM instead of GRU in H12O (56 vs. 4). In HM, most configuration used GRU (36 vs 24), but is more equitably distributed.
- Most configurations used the focal loss instead of the vanilla binary cross-entropy (45 vs. 15 in HM, 46 vs. 14 in H12O), so the former seems a better choice for handling the class imbalance found in our data.
- Most configurations used a batch size of 16 instead of 32 (58 vs. 2 in HM, 38 vs. 22 in H12O).
- Most configurations used a learning rate of 0.001 (46 in HM and 53 in H12O).

- Dropout values are significantly different for both dataset. In HM, a dropout of 0.1 (37) was more frequent than a dropout of 0.2 (23 in both cases); a dropout of 0.4 was never used. In H12O, the dropouts of 0.1 and 0.2 practically tied (29 vs. 28, respectively), and a dropout of 0.4 was the less frequent (3 configurations).
- In the case of HM, the static embedding is considerably more effective when using an embedding size of 16 instead of 32 (37 vs. 23). As in the case of the dynamic embedding size, there are no big differences in the case of H12O (29 used an embedding size of 16, and 31 and embedding size of 32).
- For both HM and H12O, slightly greater number of models used dynamic embedding size of 64 instead of 32 (32 vs. 28 in HM and 35 vs. 25 in H12O).
- Finally, regarding feature selection, in HM, using the top 100 features proved to be considerably more effective (29 models) than 150 features (6 models) or 300 (10 models). In H120, using the top 100 features was the most used configuration (20 models), followed by 150 features (16 models) and 300 features (8 models). Discarding too many features induced a loss of task-relevant information, while keeping too many of them made more difficult the learning process. In the case of HM, 15 models did not use feature selection, and in the case of H120, 16 models did not use feature selection.

7.2 Best RNN Configuration

For **HM**, best RNN-based model in terms of F1 is an ensemble that maximizes F1. This ensemble chooses among the 100 models with the highest accuracy in the VALID-CV split, combining 10 models with a λ of 0.3. The ensemble that maximizes sensitivity is obtained analogously, but using sensitivity instead of F1 for discarding combinations that lower the score of the ensemble. This model combines 10 models, less than the maximum of 20 models, using a λ of 0.1.

For **H12O**, the best RNN-based model in terms of F1 is an ensemble that maximizes F1. It combines 20 models with a λ of 0.2. The ensemble that maximizes sensitivity combines 10 models, also using a λ of 0.1.

These results demonstrate that we can build RNN ensemble models with a very high performance combining only 10 configurations.

8 Conclusions and Future Work

This work is a first approximation of a RNN based model to predict the outcome of patients with COVID-19. The models were trained with two relatively small COVID-19 datasets and will be the basis of a future system in a larger setting. The models consume dynamic information to generate daily predictions. Unlike standard training schemes for recurrent models, we do not feed the entire sequence of dynamic data into the RNN to output a single prediction but, we feed each daily record of temporal and static data day-by-day and output a prediction for each day.

We defined an ensemble method as a voting system and developed an algorithm that, starting with the best model in terms of cross-validated accuracy or F1 score, heuristically chooses the most different models (i.e. the ones that make the most different predictions) to encourage model diversity in the ensemble but without penalizing the results. Results obtained with this ensemble method are not only superior, but they also reduce the standard deviation, producing less variation in all metrics among folds.

The system is robust enough to deal with real world data and it is able to overcome the problems derived from the sparsity and heterogeneity of the data. In addition, the approach was validated using two datasets showing substantial differences. Although both datasets contain information from COVID patients, they have different characteristics which made it impossible to join the datasets in a single one. The disparity in coding conventions and the differences in the type of information contained precluded aggregating data, as this would produce a reduced dataset due to the low overlap. Instead, we generated a model for each data set, following the same method and using the same architecture. This not only validates the robustness of the proposal but also meets the requirements of a real scenario where the interoperability between hospitals' datasets is difficult to achieve.

This ended up in an exhaustive experimentation maximizing the potential of real world data. We explored and evaluated different data representation methods and this information is reported in detail in the article and in the additional material.

Regarding interpretability, when using the attention mechanism on the dynamic data vector, we show that there are relevant days that the model puts the focus on. In other words, the model keeps track of relevant days to enhance its predictions. As we saw in Section 7, visualizing attention weights is useful for understanding the evolution of the patient over time. In addition, for both HM and H12O, we analyzed the attention on dynamic data in the TEST set to see whether the attention is evenly distributed or, on the contrary, it has peak days. The results show that 56.59% and 66.36% of the patients have peak days. This corroborates our hypothesis that the attention mechanism targets relevant

days to perform correctly. Ensembling models can reduce interpretability if some of the models selected that do not use attention. One should notice, however, that the vast majority of the top 60 models in both datasets use attention (all 60 for HM and 53 for H12O). When all the models selected use attention, it is recommended to group them by their predicted outcome and then analyze the aggregated attention to check why some of the models predict one outcome or another. We are planning new experiments to get more insights from attention data.

Regarding the global results, for **HM**, the best ensemble RNN model is significantly better than the other models when considering sensitivity, reaching an impressive **0.84 in sensitivity** with no penalty in F1 and a small impact in accuracy.

Similarly, for **H12O**, the RNN ensemble model outperforms all models in sensitivity. In particular, the Ensemble-SEN model outperforms RF in 18 percentage points and SVC in 16 percentage points reaching an **0.80 in sensitivity**.

When evaluating day-by-day performance from the outcome, the **HM** RNN model shows a higher and flatter curve in sensitivity compared to the baselines, starting at 0.97, ending at 0.55 and maintaining a performance of 0.79 or better until day 14. This shows that the system is capable of making early predictions. Similarly, the **H12O** RNN model shows a better performance in sensitivity, maintaining a score of 0.64 or better until 7 days before the outcome.

Even in our data constrained scenario, the models outperform the strong baselines of tuned RF and SVC in terms of sensitivity. We hypothesize that the model will further improve its performance when more data are available.

Currently, we are working to scale up the experiment with the collaboration of two big public hospitals in Spain to include more data and previous clinical history of patients. Having more data will allow better data representations, for example: pre-training embeddings for the different features (laboratory, medications, etc) using COVID and non-COVID data. In addition, apart from mortality, we suggest predicting and timing other events such as admission to ICU and the need for respiratory assistance.

With significantly more data, we could understand which medications and procedures are better as we could make simulations with previous patients. For new patients it also would be possible to make real-time simulations to maximize the positive outcome.

9 Acknowledgements

We want to thank,

José Antonio López-Martin from Hospital Universitario 12 de Octubre for his support and invaluable comments;

Xavier Pastor and M. Jesús Betran from the Hospital Clínic de Barcelona for their interest and support to the project;

Carlos Luis Parra and Sara González from Hospital Virgen del Rocio of Seville.

HM Hospitales[22] for such an interesting and unprecedented initiative sharing the data for the scientific community.

And finally, our colleague Eulàlia Farré i Maduell, a medical doctor and researcher at the Text Mining Unit in BSC for her support on medical issues during the research.

Hospital 12 de Octubre is supported by PI18/00981 "*Arquitectura normalizada de datos clínicos para la generación de infobancos y su uso secundario en investigación: caso de uso cáncer de mama, cérvix y útero, y evaluación*" funded by the Carlos III Health Institute from the Spanish National plan for Scientific and Technical Research and Innovation 2017-2020 and the European Regional Development Funds (FEDER).

This work has been funded by the State Secretariat for Digitalization and Artificial Intelligence (SEDIA) to carry out specialised technical support activities in supercomputing within the framework of the Plan TL[23] signed on 14 December 2018.

References

[1] Wendong Ge, Jin-Won Huh, Yu Rang Park, Jae-Ho Lee, Young-Hak Kim, and Alexander Turchin. An interpretable icu mortality prediction model based on logistic regression and recurrent neural networks with lstm units. *AMIA Annual Symposium proceedings.*, 2018:460–469, Dec 2018.

[22] `https://www.hmhospitales.com/coronavirus/covid-data-save-lives`

[23] `https://www.plantl.gob.es/`

[2] D. Chang, D. Chang, and M. Pourhomayoun. Risk prediction of critical vital signs for icu patients using recurrent neural network. In *2019 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1003–1006, 2019.

[3] Ke Yu, Mingda Zhang, Tianyi Cui, and Milos Hauskrecht. Monitoring icu mortality risk with a long short-term memory recurrent neural network. *Pacific Symposium on Biocomputing. Pacific Symposium on Biocomputing*, 25:103–114, 2020.

[4] Aditya Khosla, Yu Cao, Cliff Chiung-Yu Lin, Hsu-Kuang Chiu, Junling Hu, and Honglak Lee. An integrated machine learning approach to stroke prediction. In *KDD '10*, 2010.

[5] Benjamin Letham, Cynthia Rudin, Tyler H. McCormick, and David Madigan. Interpretable classifiers using rules and bayesian analysis: Building a better stroke prediction model. *ArXiv*, abs/1511.01644, 2015.

[6] Zachary C. Lipton, David C. Kale, Charles Elkan, and Randall Wetzel. Learning to diagnose with lstm recurrent neural networks, 2015.

[7] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural Comput.*, 9(8):1735–1780, November 1997.

[8] Edward Choi, Mohammad Taha Bahadori, and Jimeng Sun. Doctor AI: predicting clinical events via recurrent neural networks. *CoRR*, abs/1511.05942, 2015.

[9] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean. Distributed representations of words and phrases and their compositionality, 2013.

[10] Edward Choi, Mohammad Taha Bahadori, Jimeng Sun, Joshua A Kulas, Andy Schuetz, and Walter F. Stewart. Retain: An interpretable predictive model for healthcare using reverse time attention mechanism. In *NIPS*, 2016.

[11] Fenglong Ma, Radha Chitta, Jing Zhou, Quanzeng You, Tong Sun, and Jing Gao. Dipole: Diagnosis prediction in healthcare via attention-based bidirectional recurrent neural networks. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[12] Douglas Teoh. Towards stroke prediction using electronic health records. *BMC Medical Informatics and Decision Making*, 18, 2018.

[13] Jing Xia, Su Pan, Min Zhu, Guolong Cai, Molei Yan, Qun Su, Jing Yan, and Gangmin Ning. A long short-term memory ensemble approach for improving the outcome prediction in intensive care unit. *Computational and Mathematical Methods in Medicine*, 2019:8152713, Nov 2019.

[14] Li Yan, Hai-Tao Zhang, Jorge Goncalves, Yang Xiao, Maolin Wang, Yuqi Guo, Chuan Sun, Xiuchuan Tang, Liang Jing, Mingyang Zhang, Xiang Huang, Ying Xiao, Haosen Cao, Yanyan Chen, Tongxin Ren, Fang Wang, Yaru Xiao, Sufang Huang, Xi Tan, Niannian Huang, Bo Jiao, Cheng Cheng, Yong Zhang, Ailin Luo, Laurent Mombaerts, Junyang Jin, Zhiguo Cao, Shusheng Li, Hui Xu, and Ye Yuan. An interpretable mortality prediction model for covid-19 patients. *Nature Machine Intelligence*, 2(5):283–288, May 2020.

[15] Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[16] Dimitris Bertsimas, Galit Lukin, Luca Mingardi, Omid Nohadani, Agni Orfanoudaki, Bartolomeo Stellato, Holly Wiberg, Sara Gonzalez-Garcia, Carlos Luis Parra-Calderon, Kenneth Robinson, Michelle Schneider, Barry Stein, Alberto Estirado, Lia a Beccara, Rosario Canino, Martina Dal Bello, Federica Pezzetti, and Angelo Pan. Covid-19 mortality risk assessment: An international multi-center study. *medRxiv*, 2020.

[17] Klann Jeffrey G, Weber Griffin M, Estiri Hossein, Moal Bertrand, Avillach Paul, Hong Chuan, Castro Victor, Maulhardt Thomas, Tan Amelia LM, Geva Alon, Beaulieu-Jones Brett K, Malovini Alberto, South Andrew M, Visweswaran Shyam, Omenn Gilbert S, Ngiam Kee Yuan, Mandl Kenneth D, Boeker Martin, Olson Karen L, Mowery Danielle L, Morris Michele, Follett Robert W, Hanauer David A, Bellazzi Riccardo, Moore Jason H, Loh Ne-Hooi Will, Bell Douglas S., Wagholikar Kavishwar B, Chiovato Luca, Tibollo Valentina, Rieg Siegbert, Li Anthony L.L.J., Jouhet Vianney, Schriver Emily, Samayamuthu Malarkodi J, Xia Zongqi, Kohane Isaac S, Brat Gabriel A, and Murphy Shawn N. Validation of a derived international patient severity algorithm to support covid-19 analytics from electronic health record data. *medRxiv*, 2020.

[18] Edward Choi, Mohammad Taha Bahadori, Le Song, Walter F. Stewart, and Jimeng Sun. Gram: Graph-based attention model for healthcare representation learning. *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2017.

[19] Zhengping Che, Sanjay Purushotham, Kyunghyun Cho, David Sontag, and Yan Liu. Recurrent neural networks for multivariate time series with missing values. *Scientific Reports*, 8(1):6085, Apr 2018.

[20] Edward Choi, Andy Schuetz, Walter F. Stewart, and Jimeng Sun. Using recurrent neural network models for early detection of heart failure onset. *Journal of the American Medical Informatics Association : JAMIA*, 24(2):361–370, Mar 2017.

[21] Tingyan Wang, Robin G. Qiu, and Ming Yu. Predictive modeling of the progression of alzheimer's disease with recurrent neural networks. *Scientific Reports*, 8(1):9161, Jun 2018.

[22] Merijn Beeksma, Suzan Verberne, Antal van den Bosch, Enny Das, Iris Hendrickx, and Stef Groenewoud. Predicting life expectancy with a long short-term memory recurrent neural network using electronic medical records. *BMC Medical Informatics and Decision Making*, 19(1):36, Feb 2019.

[23] Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *CoRR*, abs/1406.1078, 2014.

[24] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate, 2014.

[25] Tsung-Yi Lin, Priya Goyal, Ross B. Girshick, Kaiming He, and Piotr Dollár. Focal loss for dense object detection. *CoRR*, abs/1708.02002, 2017.

[26] Cibele C. César and Marilia S. Carvalho. Stratified sampling design and loss to follow-up in survival models: evaluation of efficiency and bias. *BMC Medical Research Methodology*, 11(1):99, Jun 2011.

[27] Mohamed Elfil and Ahmed Negida. Sampling methods in clinical research; an educational review. *Emergency (Tehran, Iran)*, 5(1):e52–e52, 2017.

[28] Xiangao Jiang, Megan Coffee, Anasse Bari, Junzhang Wang, Xinyue Jiang, Jianping Huang, Jichan Shi, Jianyi Dai, Jing Cai, Tianxiao Zhang, Zhengxing Wu, Guiqing He, and Yitong Huang. Towards an artificial intelligence framework for data-driven prediction of coronavirus clinical severity. *Computers, Materials & Continua*, 63(1):537–551, 2020.