Tel-Aviv University

Raymond and Beverly Sackler Faculty of Exact Sciences

The Blavatnik School of Computer Science

A Machine Learning Model for Predicting Deterioration of COVID-19 Inpatients

Thesis submitted in partial fulfillment of graduate requirements for

The degree "Master of Sciences" in Tel-Aviv University

School of Computer Science

By

Omer Noy

Under the supervision of

Prof. Ron Shamir

April 2022

Acknowledgement

I would like to express my deep gratitude to the people who accompanied me during this period and helped to make it happen.

Foremost, I wish to deeply thank my outstanding supervisor Prof. Ron Shamir. I am honored to have been advised by an inspiring person like Ron, an exceptional researcher, a professional mentor, and a kind-hearted person. Under his guidance, I gained a broad perspective about research, from brainstorming to conducting excellent research, with uncompromising professionalism. Also, I thank Ron for pushing me to opportunities to present my work in domestic and international conferences, and in a scientific journal.

Secondly, a special thanks to my research partner and friend Dan Coster for his extensive contribution to this research. Dan has been an integral part of this study, from the very first initiatives to the recent analyses. Thank you for this fascinating journey.

Thirdly, I would like to thank my great collaborators– Prof. Ori Rogowski, Prof. Galia Rahav, Prof. Shlomo Berliner, Dr. Shani Shenhar-Tsarfaty, Itai Atar, and Maya Metzger. Thank you for your significant contribution to this work.

I wish to thank my lab mates – Lianrong, David, Tom, Nimrod, Hagai, Hadar, Yonatan, Dan F., Naama, Eran and Ron, for the helpful discussions, support, and for being great friends. Additionally, I owe special thanks to Gilit Zohar-Oren for her administrative help, always accompanied with smile and grace.

I deeply thank for the financial support I was granted during my studies: the Edmond J. Safra Center for Bioinformatics at Tel Aviv University, The Israel Science Foundation (grant 1339/18 and grant 3165/19 within the Israel Precision Medicine Partnership program) and German-Israeli Project Cooperation DFG-DIP RE 4193/1–1.

And last but not the least, I wish to deeply thank my dear family, my parents Yossi and Anat, my sister Nofar, and to my love Guy, for your endless understanding and support. This would not have happened without you.

Abstract

The COVID-19 pandemic has been spreading worldwide since December 2019, presenting an urgent threat to global health. Due to the limited understanding of disease progression and of the risk factors for the disease, it is a clinical challenge to predict which hospitalized patients will deteriorate. Moreover, several studies suggested that taking early measures for treating patients at risk of deterioration could prevent or lessen condition worsening and the need for mechanical ventilation. We developed a predictive model for the early identification of patients at risk for clinical deterioration by retrospective analysis of electronic health records of COVID-19 inpatients at the two largest medical centers in Israel. Our model employs machine learning methods and uses routine clinical features such as vital signs, lab measurements, demographics, and background disease. Deterioration was defined as a high NEWS2 score adjusted to COVID-19. In the prediction of deterioration within the next 7–30 h, the model achieved an area under the ROC curve of 0.84 and an area under the precision-recall curve of 0.74. The model achieved sensitivity of 44% with a positive predictive value of 87%. In external validation on data from a different hospital, it achieved values of 0.76 and 0.7, respectively.

Table of Content

Abstract

1. Introduction

2. Clinical Background

2.1. COVID-19

2.2. Electronic Health Records

2.3. Early Warning Scores

3. Computation Background

3.1. Machine Learning

3.1.1. Algorithms

3.1.2. Regularization

3.1.3. Model Evaluation and Tuning

3.1.3.1. Data Partition

3.1.3.2. Cross-Validation

3.1.3.3. Evaluation Metrics

3.1.3.4. Hyperparameters Tuning

3.2. Feature Selection

3.3. Data Imputation

3.4. Anomaly Detection

4. Methods

4.1. Cohort Description

4.2. Inclusion and Exclusion Criteria

4.3. Outcome Definition

4.4. Outlier Removal

4.5. Data Imputation

4.6. Feature Engineering

4.7. Model Development and Feature Selection

4.8. Evaluation Approach

5. Results

5.1. Cohort Description

5.2. COVID-19 Deterioration Model

5.3. External Validation

6. Discussion

7. References

8. Supplementary Material

1. Introduction

The coronavirus disease 2019 (COVID-19) emerged in China in December 2019, and since then has spread rapidly around the world. In March 2020, the World Health Organization declared the COVID-19 outbreak as a global pandemic [1]. As of March 2022, worldwide cases exceeded 450 million and more than six million died [2]. The extent of the disease varies from asymptomatic to severe, characterized by respiratory and/or multi-organ failure and death [3], [4]. Healthcare systems worldwide have faced an overwhelming burden of patients with COVID-19. At the same time, there is limited understanding of disease progression, risk factors for deterioration, and the long-term outcomes for those who deteriorate. Moreover, early treatments such as antiviral medications may prevent clinical deterioration in COVID-19 patients [5]. Therefore, early warning tools for COVID-19 deterioration are required. Tools that predict deterioration risk in individuals can also improve resource utilization in the clinical facility and its wards, by aggregating risk scores of patients for anticipating expected changes in patient load [6].

Prognostic scores for clinical deterioration of patients are widely used in medicine, particularly in critical care. The National Early Warning Score 2 (NEWS2), the quick Sequential Organ Function Assessment (qSOFA), and CURB-65 [7]–[9] are commonly used clinical risk scores for early recognition of patients with severe infection. The NEWS2 score incorporates pulse rate, respiratory rate, blood pressure, temperature, oxygen saturation, supplemental oxygen, and level of consciousness or new confusion. Liao et al. [10] suggested an early warning score for COVID-19 patients termed "modified-NEWS2" (mNEWS2). It adds to the NEWS2 formula the factor age≥65

years, reflecting the observation that older age is associated with elevated risk for severe illness (**Supplementary Table 1**).

Machine learning methods integrate statistical and mathematical algorithms that enable the analysis of complex signals in big-data environments [11], [12]. In recent years, such methods were shown to be highly effective for data-driven predictions in a multitude of fields, including healthcare [12]. They enable rapid analysis of large electronic health records (EHRs) and can generate tailored predictions for each patient. Consequently, machine learning methods have great potential to help improve COVID-19 care.

In this thesis, we developed a machine learning model for early prediction of deterioration of COVID-19 inpatients, defined as mNEWS2 score ≥ 7. The model was developed by analyzing longitudinal EHRs of COVID-19 inpatients in Sheba Medical Center (Sheba), the largest hospital in Israel. To validate the generalizability of its performance, we applied our model on EHRs of inpatients diagnosed with COVID-19 from the second largest hospital in Israel, the Tel-Aviv Sourasky Medical Center (TASMC).

The results of this study were recently published in the journal *Scientific Reports* [13].

The thesis is organized as follows: Chapter 2 provides basic clinical background, and Chapter 3 provides the main required computational background. Chapter 4 describes our methods, including cohort description, data processing, model development and evaluation approach. Chapter 5 describes the results and Chapter 6 contains our discussion. Additional information is provided in the supplement.

2. Clinical Background

2.1. COVID-19

The coronavirus disease 2019 (COVID-19) is an infectious disease caused by severe acute respiratory syndrome coronavirus 2 (SARS-CoV-2). The novel virus emerged in Wuhan, China, in December 2019, and since then has spread rapidly worldwide. The World Health Organization (WHO) declared the COVID-19 outbreak as a global pandemic in March 2020 [1]. The impact of the disease is immense, with worldwide cases exceeding 450 million and more than six million dead as of March 2022 [2].

The disease manifestations range from asymptomatic to a severe condition, characterized by respiratory distress, multi-organ failure, or death [3], [4]. The symptoms are highly variable between individuals and along time. In addition, the effect of the disease varies among individuals, with some continuing to experience a range of symptoms for months after recovery (long COVID) [14]. While most infected are asymptomatic or experience mild symptoms, others require hospitalization and medical monitoring and treatment. The hospitalization criteria depend on the clinical presentation and vary along time periods. The symptoms can deteriorate within hours during the hospitalization, leading to oxygen support requirement or to intensive care unit (ICU) administration [15], [16]. The extent and variability of the disease have resulted in an overwhelming burden of COVID-19 patients in healthcare systems around the world. Consequently, appropriate clinical decisions and efficient utilization of medical resources may be impaired. In addition, even though the understanding of COVID-19 is evolving, there is still limited understanding of the risk factors for deterioration and the long-term outcomes for those who deteriorate.

Early treatments such as antiviral medications may prevent clinical deterioration in COVID-19 patients [5]. Hence, tools for early identification of patients at high risk for COVID-19 deterioration are required. Such tools are helpful not only for individual medical decision-making, regarding follow-up or treatment strategies, but can also improve resource utilization at the ward level, by anticipating the expected changes in patient load [6].

2.2. Electronic Health Records

An electronic health record (EHR) is a digital collection of a patient's health information that is systematically gathered as part of the clinical setting, typically over time. It may contain various types of health data, including patient demographics, laboratory test results, vital signs, medical history, medication, clinical images, and nursing notes. The data is stored and maintained in the hospital EHR systems, enabling the examination of the patient's health over time. In recent years, the utilization of EHR has increased dramatically. The abundance of data and the high-dimensional clinical features can be leveraged for unique healthcare studies and applications at both individual and population levels. In recent years, a growing number of studies have applied machine learning methods on EHR data, for various healthcare applications [17]–[20]. The utilization of complex models for EHR data raises unique opportunities to improve healthcare, e.g., by improving disease diagnosis, predicting individual risks, or guiding treatment strategies.

2.3. Early Warning Scores

An Early Earning Score (EWS) is a tool used by medical services to assess the severity of a patient's condition. These tools typically assign numeric values to several physiological variables (e.g., heart rate, oxygen saturation, respiratory rate, etc.) to produce a score that can identify a patient at risk of deterioration. A range of early warning scores has been developed to meet different clinical needs. These include the National Early Warning Score 2 (NEWS2), the quick Sequential Organ Function Assessment (qSOFA), and CURB-65 [7]–[9], which are commonly used clinical risk scores for early recognition of patients with severe infection. In particular, the NEWS2 score was developed by the Royal College of Physicians to provide a standard for early warning scores in the United Kingdom, and it is widely used worldwide in healthcare settings. The NEWS2 score incorporates pulse rate, respiratory rate, blood pressure, temperature, oxygen saturation, supplemental oxygen, and level of consciousness or new confusion. The calculation is done by assigning points to each physiological parameter according to its condition (between 0 to 3), such that a high score indicates that the parameter is further from the normal range. Summing the points for each parameter results in a total score, with a higher score representing higher risk and vice versa. After the pandemic emerged, Liao et al. [10] adjusted the NEWS2 score for COVID-19 patients, by adding the factor age≥65 years to the formula (**Supplementary Table 1**). The score, termed “modified-NEWS2” (mNEWS2), now reflects the observation that older age is associated with elevated risk for severe illness.

3. Computational Background

3.1. Machine Learning

Machine learning is a subfield of Artificial Intelligence (AI), in which data is being leveraged to build models with learning capabilities in order to make predictions or decisions. Machine learning methods integrate statistical and mathematical algorithms that enable the analysis of complex signals in big-data environments [11], [12]. In recent years, many applications have been developed using machine learning in various fields, such as banking, social media, or healthcare. There are primarily three types of machine learning: Supervised, Unsupervised, and Reinforcement learning. In next the sections, we will mainly cover supervised algorithms, due to their relevance to this study.

Supervised Learning

Let us start with fundamental machine learning terminology. The input to the supervised problem is a collection of pairs $(\boldsymbol{x_i}, y_i)$ where $\boldsymbol{x_i} \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. $\mathcal{X}$ is called the *input space* and $\mathcal{Y}$ is the *output space*. A typical example is where $\mathcal{X} = R^n$ and $\mathcal{Y}$={-1, 1}. x_i is called an *example* and its coordinates are called the *feature* values. y_i is called the *label* of x_i. Given such labeled examples, the goal is to develop an algorithm to predict the labels of new unlabeled ones. The process of building the algorithm is called *training* and the process of applying it to new data is called *inference*.

Training means *learning* or *fitting* the model to the input examples. That is, the model gradually learns the relationships between features and labels. Formally, given the *training set*, a collection of N labeled examples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, the goal is to learn a function $f: \mathcal{X} \to \mathcal{Y}$ such that $f(x)$ is a "good" predictor of y. More precisely, we assume that each pair $(\boldsymbol{x_i}, y_i)$ is drawn i.i.d. from a joint probability distribution $P(x, y)$. The true distribution P is unknown and the goal is to try to estimate it by the empirical distribution of the observed data. A fundamental element of learning and optimization is the *loss function* $L(f(x), y)$, a non-negative real-valued function, which estimates the distance between the model's output and the true label. If the model's prediction is perfect, the loss is zero, and it is greater otherwise. Therefore, learning algorithms aim to minimize the *expected loss*, also known as the risk $R(f)$ of function f, given by:

$$R(f) = \mathbb{E}[L(f(x), y)] = \int L(f(x), y)\, dP(x, y)$$

However, since the true distribution P is unknown, $R(f)$ cannot be computed. If the training set is representative of P, the expected loss can be estimated by the *empirical loss*, as follows:

$$R_{emp}(f) = \frac{1}{N} \sum_{i} L(f(x_i), y_i)$$

Supervised machine learning algorithms attempt to find a model that minimizes the empirical loss. This process is known as *empirical risk minimization*.

In inference, we use the learned model and apply it to unlabeled examples for making predictions. Machine learning models aim to predict well when applied to new unseen data. This is the basis for the fundamental partition of a dataset into two subsets: a *training set*, the labeled data used to train the model, and a *testing set*, a disjoint set of

samples on which we test the learned model. Sections 3.1.3.1-3.1.3.2 discuss workflows for data partition and model training.

Supervised learning can be further divided into *classification* and *regression* problems, where the former is used when the output y is categorical variable, and the latter is used when y is continuous. In section 3.1.1, we mainly focus on classification, since this is the type of supervised learning problem that arises in our study.

Classification

The goal of classifications is to assign a set of labels $\{1, \dots, K\}$ to input $\boldsymbol{x}$. For instance, in clinical application, $\boldsymbol{x}$ could be patients' variables extracted from EHRs systems, and the labels could represent future disease conditions. In what follows, we focus on binary classification due to its simplicity and relevance to our goals. Specifically, we will consider N training samples $\{(\boldsymbol{x}_i, y_i)\}_{i=1}^{N}$, where $\boldsymbol{x}_i \in \mathbb{R}^d$ and $y_i \in \{0,1\}$. Extensions to multiple-class classification are possible, but the principles are similar to the binary case. We discuss various classification models in section 3.1.1.

Unsupervised learning

The goal of unsupervised learning is to learn the underlying patterns in unlabeled data. Learning in this context is called unsupervised as there are no ground truth labels to guide the learning process. Common unsupervised learning algorithms include clustering, anomaly detection, and approaches for learning latent variable models. We discuss anomaly detection in detail in section 3.4.

3.1.1. Models

In the machine learning area, the problem of our study can be formulated as a multivariate time-series classification problem. We have considered several supervised ML models to address this problem, including both linear and non-linear classifiers. This section briefly presents the algorithms of interest.

Naïve Bayes

Naïve Bayes is a statistical classifier based on Bayes theorem. It is a conditional probability model that calculates the conditional probability $p(y_i|x)$ for each instance x and for each of the K possible classes $y_i \in \{1, \dots, K\}$. The predicted class label is:

$$\hat{y} = argmax_{y_i \in \{1,\dots,K\}} p(y_i|x)$$

In other words, the goal of Naïve Bayes is to find the class y_i that maximizes the posterior probability for a given test sample x. In practice, this is estimated using Bayes' theorem with strong (naive) independence assumptions between the features.

Linear Regression

Linear regression [21], [22] is a model that assumes a linear relationship between the input x and the output y (see **Figure 1a**). Mathematically this is written as:

$$f(x) = w \cdot x + b$$

Where $w \in \mathbb{R}^d$ represents the *weights* and $b \in \mathbb{R}$ represents the *bias*. The objective of linear regression is to find the regression line that best fits the training dataset, with the goal of minimizing the total distance between the predicted and true values. It can be applied for binary classification by assigning a threshold for separating two classes. However, as the predicted value is continuous, it is usually less suitable for classification and logistic regression is preferred (see next section).

Logistic Regression

Logistic regression [23] is named after the function used at the core of the method, the *logistic* function, or the *sigmoid* function:

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

Which outputs values between 0 and 1. The goal of logistic regression is to find the model that best describes the relationship between the dependent and the independent variables. The dependent variable is dichotomous in nature, so it can be suitable for binary classification. See illustration in **Figure 1b**.

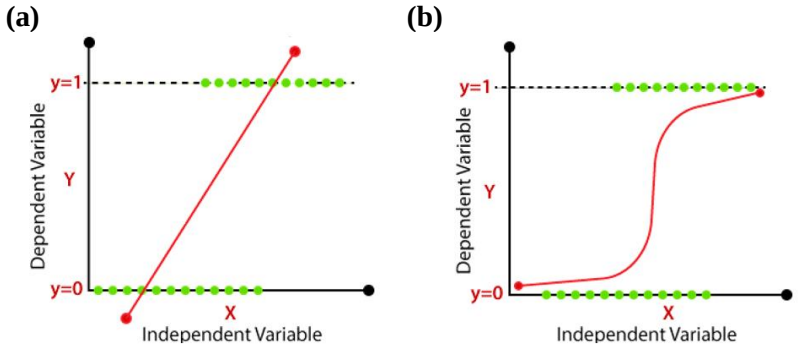


Figure 1: Linear regression (a) and logistic regression (b) fitted to a 2D dataset (Figure source: www.javatpoint.com/linear-regression-vs-logistic-regression-in-machine-learning).

Support Vector Machine (SVM)

Support-vector machines (SVMs) [24] are supervised machine learning models that can be used for classification, regression, or other tasks like anomaly detection. SVM performs binary classification by constructing a d-dimensional hyperplane (d is the number of features) that separates the space into two half-spaces and thereby the data into two categories. Many hyperplanes might classify the data. A good separation, in terms of generalization, is achieved by the hyperplane that has the maximal *margin* to the nearest training data points. By margin, we mean the minimal distance between the separator and any data point $\boldsymbol{x_i}$. The data points that are closest to the separating hyperplane are called *support vectors*, and they influence the position and orientation of the hyperplane. An illustration of SVM for binary classification in linearly separable data can be seen in **Figure 2a**.

In addition to linear classification, SVMs can efficiently perform non-linear classification using non-linear *kernel* functions. A kernel function implicitly maps the input $\boldsymbol{x}$ into a higher dimensional feature space $\phi(\boldsymbol{x})$, where the data is linearly separable. In the new high dimensional space, SVM can be easily applied. The function ϕ has the property that dot products can be computed efficiently without computing ϕ explicitly. This is called the *kernel trick* (see **Figure 2b**). Different types of kernels can be used for this mapping, including linear, polynomial, sigmoid and radial basis function (RBF).

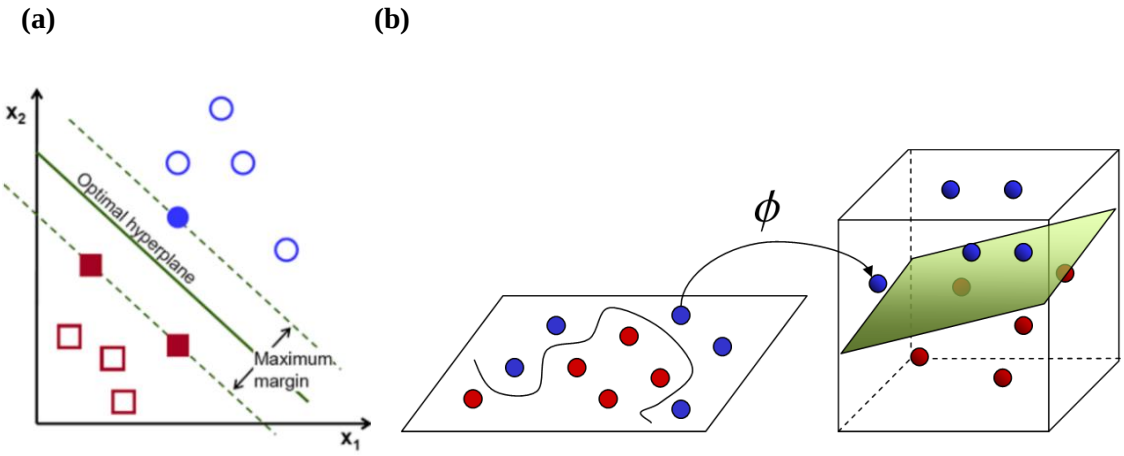


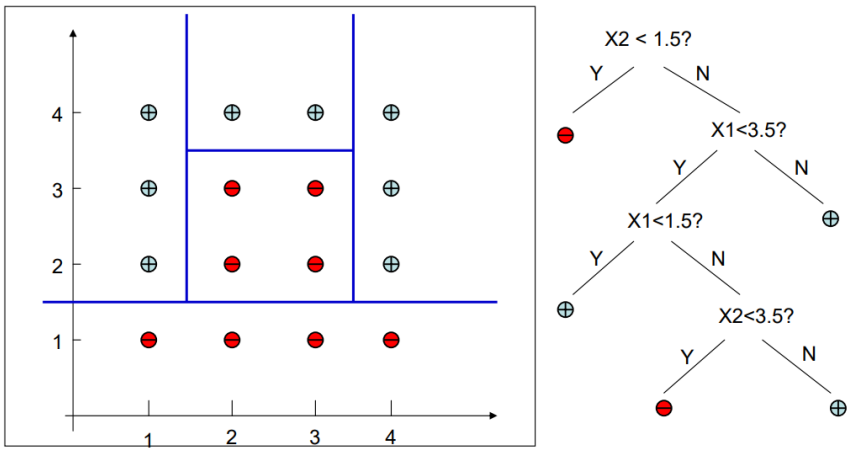
Figure 2: (a) Linear SVM in 2D data. Colors represent the ground truth labels (y), and the solid line represents the separating hyperplane. (b) Kernel SVM: mapping the complex 2D input data into higher dimension space (here 3D) enables linear separation (Figure source: towardsdatascience.com/svm-feature-selection-and-kernels-840781cc1a6c).

Random Forest (RF)

Decision trees (DTs) [25] are non-parametric supervised machine learning models used for both classification and regression. The key idea is that the examples are repeatedly partitioned according to simple decision rules inferred from the features. Each tree consists of a root note, several branches, and leaf nodes. An internal node represents a subset of examples and a test on a particular feature. The examples in the subset are assigned to the node's children according to the test result. The leaf nodes, or terminal nodes, represent classification or decision, where the node's class is that of the majority of the samples assigned to the leaf. Classification of a new sample is done by going down the tree according to the test values of the sample and the class is that of the leaf node reached. An illustration of a decision tree for binary classification of 2D input data can be seen in **Figure 3a**.

A Random Forest (RF) [26] is an ensemble of decision trees used for classification and regression. RF fits a number of decision trees on sub-samples of the training set with randomly selected subsets of features. For a test sample, the outcome is determined based on applying to it all the trees and aggregating the predictions, using a majority vote for classification or average for regression (see illustration in **Figure 3b**).

(a)



(b)

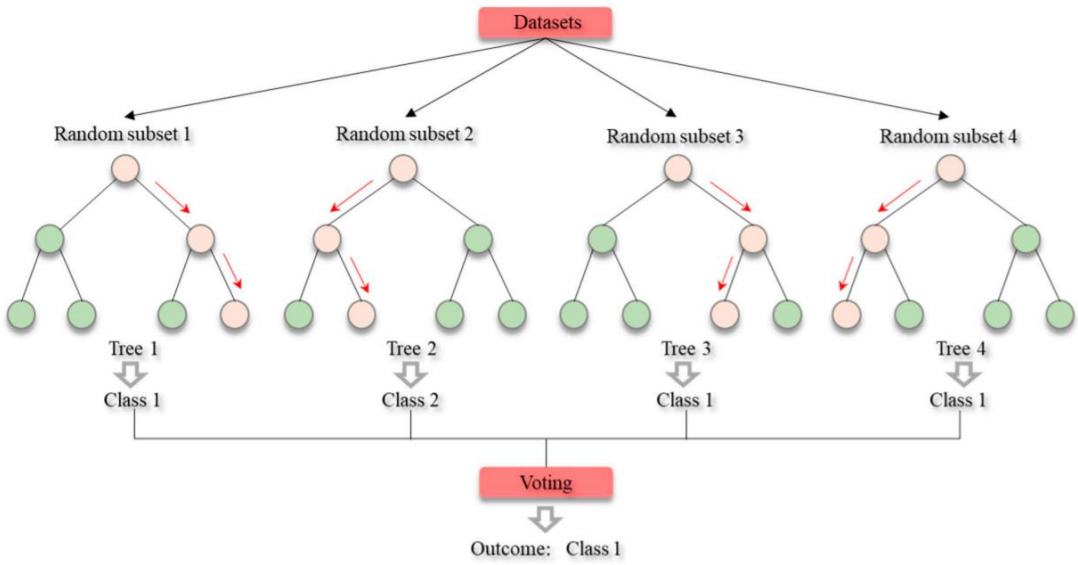


Figure 3: (a) Decision Tree classifier. The tree splits the input space according to its features (right) and generates decision boundaries (left), used to assign class labels to the examples. (b) Random Forest classifier (Figure source: www.mdpi.com/2071-1050/11/21/6159).

Gradient Boosted of Decision Trees

Gradient boosting is a machine learning technique that uses an ensemble of weak learners to improve the performance of a machine learning model. The term *weak learners* refers to simple models that perform only slightly better than random, and they are usually decision trees models. In gradient boosting, the weak learners are built sequentially, such that each model tries to improve on the error of the previous model (in contrast to *bagging*, where the models are fitted in parallel). It is a powerful ensemble algorithm that became highly popular in recent years. There are various implementations of gradient boosting decision trees, such as XGBoost [27] or CatBoost [28], each with slightly different extensions. The idea of gradient boosting is illustrated in **Figure 4**.

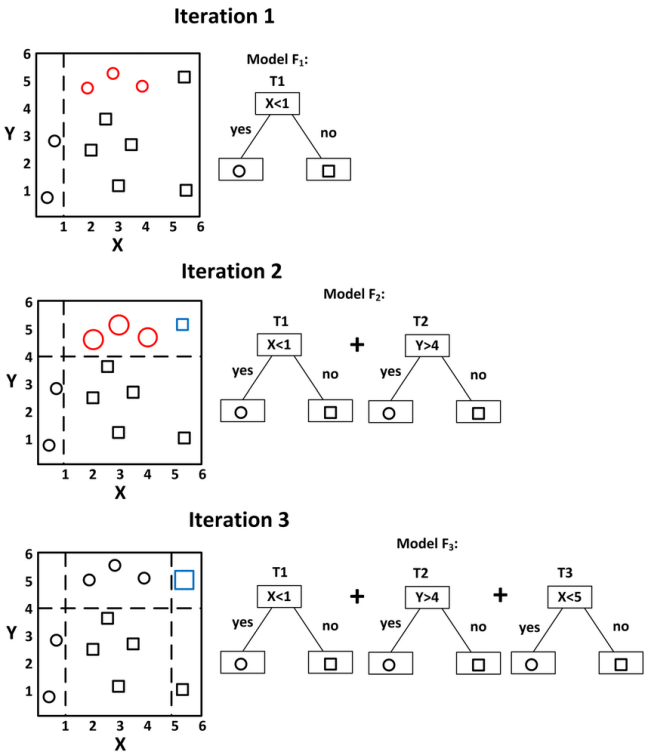


Figure 4: A simple example of gradient boosting decision trees on 2D data. The shapes represent the ground truth labels, and the goal is to perform binary classification. To improve

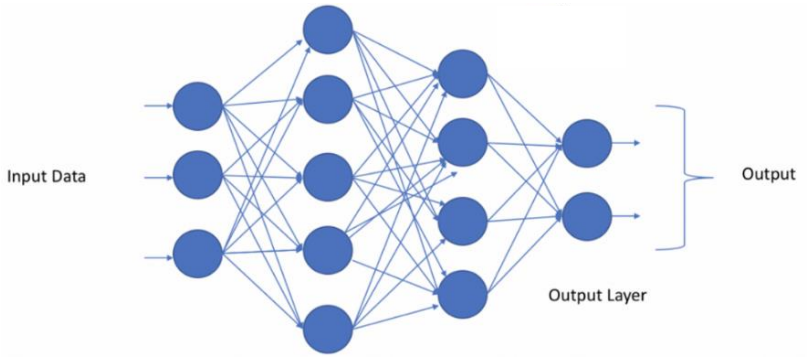
the predictions, in each iteration, higher weights are assigned to the errors of the previous step (that is, to the misclassified data points).

Artificial Neural Network (ANN)

Artificial Neural Networks (ANNs) or Neural Networks (NNs) are powerful computational models that are used extensively in recent years [29]. Their architecture is inspired by the biological neural networks, consisting of a collection of *neurons* (or *nodes*) organized in layers, such that each connection (directed edge) between neurons can transmit a signal to other neurons. Edges are weighted according to the connection strength. The basic architecture of a neural network consists of an *input* layer, one or more *hidden* layers, and *an output* layer (**Figure 5a**). The input to the network is often the feature values of a sample of the dataset. The output of each neuron is calculated by the weighted sum of its inputs, which is then passed through a nonlinear activation function (**Figure 5b**). The number of hidden layers and the number of neurons in each layer are hyperparameters of the network that should be pre-specified (see section 3.1.3.4 for more details about hyperparameter tuning).

There are various NN architectures used for different tasks, including feed-forward neural network, recurrent neural network (RNN) [30], and convolutional neural network (CNN) [31]. In general, NNs were shown to be capable of performing a broad spectrum of tasks, including classification, regression, data processing, reinforcement learning, etc. For their superior performance, NNs usually require abundant training data, which is not always feasible in real-world scenarios.

(a)



(b)

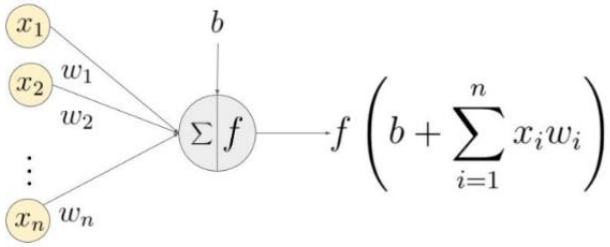


Figure 5: The building blocks of an ANN. (a) An illustration of a 3-layer feed-forward neural network. Circles represent nodes and arrows represent connections. Each layer contains a varying number of nodes. The connections are between nodes across layers, but not within a layer. (b) A Node in the network. The output of a node is computed as $f(b + \sum_{i=1}^{n} x_i w_i)$ where x_i are the inputs, w_i are the weights, b is the bias and f is non-linear activation function (Figure source: medium.com/swlh/activation-functions-in-artificial-neural-networks-8aa6a5ddf832).

3.1.2. Regularization

Regularization is a technique used to reduce errors and avoid overfitting in machine learning models. In most cases, it refers to modifying the loss function to penalize large values of the learned weights w and eliminate unimportant features from the final model. Formally, using the loss function $L(w)$, the new objective is to minimize:

$$L(w) + \lambda R(w)$$

Where $R(w)$ is the regularization term and λ controls the regularization strength. The commonly used regularization methods include L1 and L2 regularization.

L1 regularization. The regulizer is the L1 norm, formally:

$$R(w) = \|w\|_1$$

A linear regression model that uses L1 regularization is often called Lasso (Least Absolute Shrinkage and Selection Operator) regression [21].

L2 regularization. The regulizer is the squared L2 norm, that is:

$$R(w) = \|w\|_2^2$$

A linear regression model that uses L2 regularization is often called Ridge regression [22].

3.1.3. Model Evaluation and Tuning

3.1.3.1. Data Partition

In the introduction of this section (3.1) we introduced the basic partition of a dataset into training and testing sets. This enables to first train the model on the training set, and then test it on a distinct set, which was not used in the training phase, allowing the evaluation of the model on an unseen dataset. If the model's performance is not satisfying, we could repeatedly tweak the model hyperparameters, fit and evaluate it, and finally select the model that best performs on the testing set. However, this way the testing set is used when finalizing the model, introducing the risk of overfitting, where a model performs well on the training set, but does poorly on new data. This problem

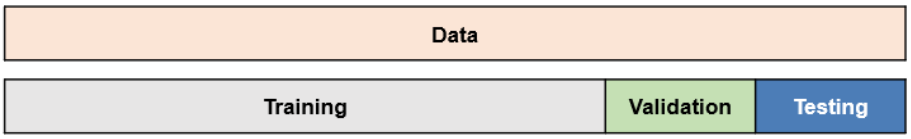
can be solved by splitting the data into three subsets: training, validation, and testing sets (**Figure 6a**). This way, we can use the *validation set* to evaluate and tune the model repeatedly. The best model is selected based on the performance on the validation set. Then, the testing set can be used to evaluate the model's performance on unseen data.

3.1.3.2. Cross-Validation

Cross-validation (CV) is a resampling technique used to evaluate ML models. The goal of cross-validation is to test to model's ability to generalize to unseen data, and to tackle problems such as overfitting or selection bias. It uses different data splits of the original training set, for training and testing the models on different subsets in an iterative fashion. The model performance on these splits allows tuning the model hyperparameters while keeping the testing set completely unseen for final evaluation. There are various types of cross-validation, including K-fold, stratified K-fold and Leave-one-out.

In K-fold cross-validation, the entire training set is partitioned into K folds of roughly equal size (**Figure 6b**). Then, we iteratively train the model on K-1 folds and use the remaining fold as the test fold. The process is repeated K times, where in each time, a different fold is left out. The performance measure of this process is then the average of the performance values computed (various metrics for performance evaluation are detailed in section 3.1.3.3). Leave-one-out cross validation is a special case of K-fold cross-validation with $K = N$, where N is the number of training examples.

(a)



(b)

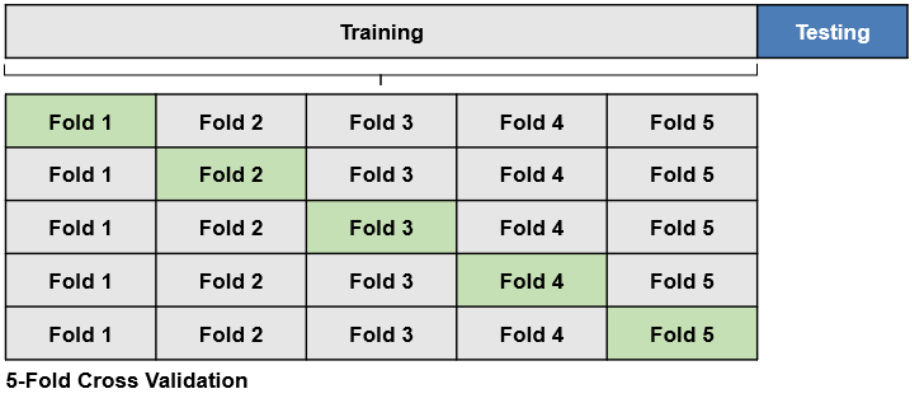


Figure 6: A schematic illustration of data partition and cross-validation. (a) The original dataset (orange) is partitioned into training (gray), validation (green), and testing (blue) sets. (b) K-fold cross-validation with subdivision of the training set into k=5 folds. In each iteration, a model is trained on 4 folds (gray) and evaluated on the remaining fold (green).

3.1.3.3. Performance Metrics

In this section, we discuss various performance metrics for ML models. We will focus on metrics for classification models, due to their relevance to this study. Recall that the goal of classification is to correctly assign a new data point to a particular class. First, let us define some notation. Let P and N be the number of positive and negative cases in the real data, respectively. In the binary classification, there are four cases to consider when estimating the total performance of a model (**Figure 7**):

- **True Positives (TP):** The number of cases in which the model correctly predicts the positive class.

- **True Negatives (TN):** The number of cases where the model correctly predicts the negative class.
- **False Positives (FP):** The number of cases in which the model incorrectly predicts the positive class.
- **False Negatives (TN):** The number of cases in which the model incorrectly predicts the negative class.

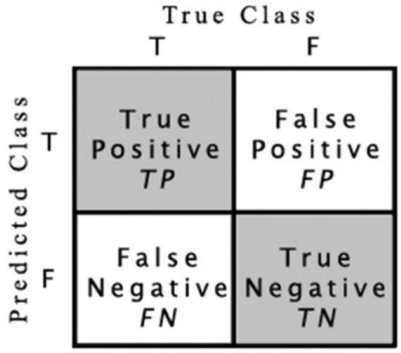


Figure 7: A confusion matrix of binary classification. It summarizes the performance of a binary classifier, accounting for its four outcomes: TP, FP, TN, and FN. (Figure source: academic.oup.com/bib/article/9/3/198/255891?login=true).

With these cases in mind, we now describe commonly used metrics:

Accuracy. *Accuracy (ACC)* describes how the model performs across all classes. It is the proportion of correct predictions (for both positive and negative cases) among the total number of cases:

$$ACC = \frac{TP + TN}{P + N}$$

Although widely used, classification accuracy can be inappropriate for imbalanced datasets (e.g., where P<<N). For example, high accuracy can be achieved by predicting the majority class for all samples.

Precision. *Precision*, or *positive predictive value (PPV)*, is the fraction of the true positive cases among the selected cases (i.e., predicted positive).

$$Precision = \frac{TP}{TP + FP}$$

Recall. *Recall*, *sensitivity*, or *True Positive Rate (TPR)* refer to the fraction of correctly identified positive examples (e.g., how many patients with the disease were correctly identified as having a disease). Formally:

$$Recall = TPR = \frac{TP}{P} = \frac{TP}{TP + FN}$$

Specificity. Specificity or True Negative Rate (TNR) refer to the fraction of correctly identified negative examples:

$$Specificity = \ TNR = \frac{TN}{N} = \frac{TN}{TN + FP} = 1 - FPR$$

Where FPR is the False Positive Rate.

Receiver Operating Characteristic (ROC) curve. ROC curve plots the true positive rate (or sensitivity) against the false positive rate (1-Specificity) for varying discrimination thresholds (see **Figure 8a**). It describes the ability of a binary classifier to discriminate between positive and negative cases. The Area Under Receiver Operating Characteristic curve (AUROC) is calculated as the area under the ROC curve and is often used to evaluate classification models. It can be shown that the AUROC equals to the probability that a randomly selected positive sample will have a higher predicted probability of being positive (e.g., a higher predicted risk score in clinical risk prediction model) than a randomly selected negative one. An AUROC of 0.5

corresponds to a random guess while an AUROC of 1 corresponds to a perfect classifier.

Precision-recall curve. Precision-recall curve plots the precision (or PPV) against the recall (or TPR) for different thresholds (See **Figure 8b**). Similarly to AUROC, the area under the precision-recall curve (AUPR) summarizes this information into a single performance value. In general, the higher AUPR is, the better the classifier performs.

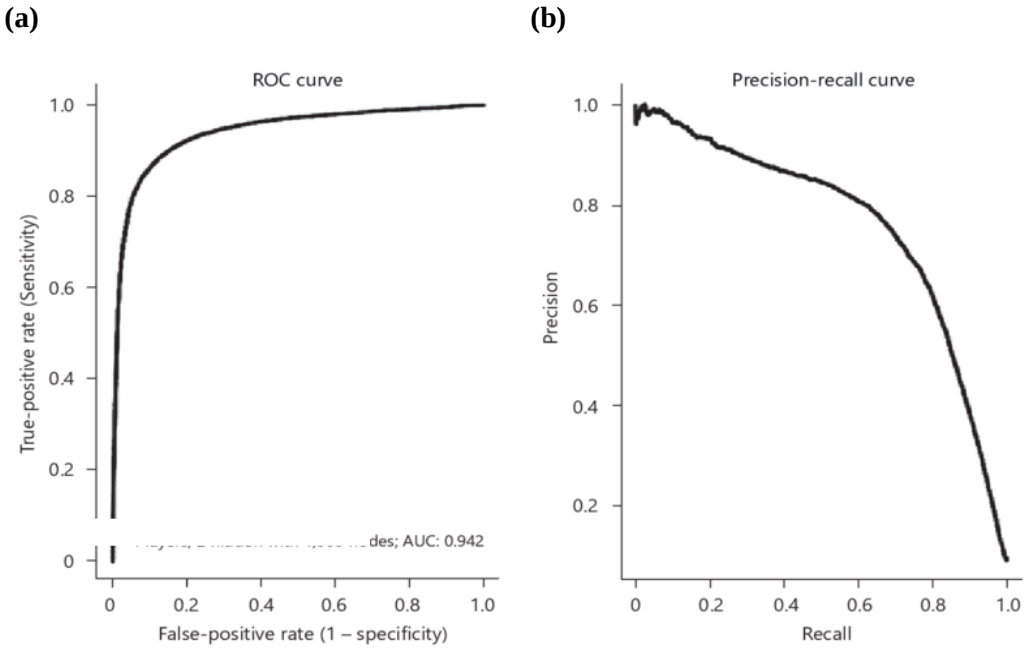


Figure 8: Examples of ROC (a) and (b) Precision-recall curves (Figure source: www.karger.com/Article/Abstract/492574).

3.1.3.4. Hyperparameter Tuning

Hyperparameters refer to parameters of a model or algorithm that are set before training and control the learning process, while learnable parameters are learned and derived

during the training process (e.g., weights and biases). They can be, for example, the number of trees in random forest, or the kernel function of SVM. The process of finding the optimal set of hyperparameters that maximizes the model performance is called *hyperparameter tuning* or *hyperparameter optimization.* The hyperparameters can be manually chosen according to background knowledge. However, there exist automated methods for optimizing this process, including *random search* and *grid search*. In both methods, we first create a grid of possible values for each hyperparameter. In grid search, all the hyperparameter combinations are examined, by training and testing the model with each combination. Alternatively, in random search, rather than exhaustively examining all the possible combinations, the combinations are randomly sampled. Then, the combination that yields the best result is selected. Cross-validation is used to estimate the models' performance in this process.

3.2. Feature Selection

Feature selection is the process of selecting a subset of features to be used in a ML model. Reducing the number of input features in high-dimensional space can improve model performance by removing irrelevant features and decrease computational cost. In addition, it could ease the interpretation of some model outputs. The final feature set used for training has a huge impact on the model performance. There are various common approaches for feature selection. Here we shortly describe two techniques.

Variance threshold. Variance threshold is a natural approach that can be used for both supervised and unsupervised learning. It removes all low-variance features, assuming that they are less informative for the model. By default, it removes all zero-variance features, as they have the same value in all observations.

Tree-based feature selection. Tree-based models can be used to calculate feature importance scores, based on how significant they are for the predictions. These scores, in turn, can be used to select the most important features for the model.

3.3. Data Imputation

Missing data is a major challenge in many practical domains, with a significant effect on data interpretation and analysis. The problem arises when values of one or more variables in the dataset are missing is some samples. The causes of missingness vary between and within fields. In healthcare, for instance, some patients undergo more comprehensive tests than others according to their medical conditions, patients might avoid particular tests or personal questions, or, information could be simply mis-recorded in the hospital EMR systems. This missingness can lead to biased estimates [32] and limit our ability to study and draw conclusions from the data. Furthermore, common machine learning models (e.g., decision trees, neural networks, support vector machines, etc.) can be applied only on complete datasets. This has driven the development of a variety of techniques for dealing with missing data.

Data imputation methods deal with the missing values problem, by filling in, or imputing, the missing data with artificial values. Such methods offer powerful tools for addressing missing data in large datasets with complex data patterns. Once the missing values are imputed, the completed dataset can be analyzed with standard algorithms and machine learning models.

Various imputation methods have been successfully applied to medical data. One of the powerful and commonly used methods is the Multivariate Imputation by Chained

Equations (*MICE*) [33]. MICE imputes the missing data through an iterative series of predictive models. In each iteration, each variable with missing values is imputed using the other variables in the dataset. This process is repeated until a desired level of convergence is met. MICE is a multiple imputation method, that is, it can be used to generate multiple imputed datasets for further analysis.

3.4. Anomaly Detection

Anomalies or outliers are datapoints that deviate from the expected behavior of the data. Anomaly detection is the task of successfully identifying these data points in a dataset. Various applications utilize anomaly detection, such as fraud detection and disease diagnosis. The presence of anomalous datapoints can be caused by measurement errors or data flaws, which in turn could lead to biased analysis and misleading results. Therefore, a common objective of anomaly detection is to eliminate the anomalies. For such cases, anomaly detection is often used in preprocessing to identify and remove the anomalous datapoints from the dataset. An alternative objective of anomaly detection is to focus on the anomalies, as they could carry significant information, such as fraud activity or an extreme health condition. In this study, we used anomaly detection techniques in feature engineering, to generate features that estimate how much each patient's observation is irregular. The unsupervised techniques used in this study are one-class SVM, Local Outlier Factor (LOF), and Isolation Forest (IF).

One-Class SVM

One-class SVM (OCSVM) [34] is a version of SVM that can be used in an unsupervised setting for anomaly detection. While standard SVM looks for a hyperplane that best separates the datapoints into two classes, one-class SVM maps the input data into a high dimensional space using a kernel function and tries to find a separating hyperplane of maximal margin between the samples and the origin. It learns a decision function that is either positive (+1) for the samples or negative (-1) for the origin. Only a small fraction of data points is allowed to be negative, namely, to lie on the origin's side. Those data points are considered as outliers. The model uses the hyperparameter ν, known as the outlier function, the expected proportion of outliers in the data. An illustration can be seen in **Figure 9**.

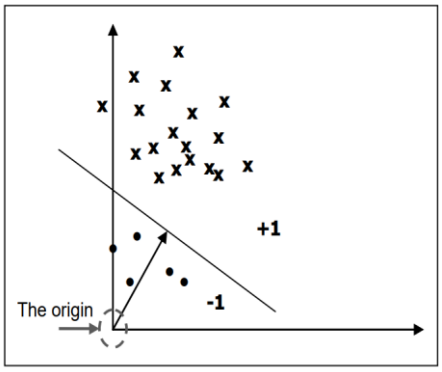


Figure 9: Illustration of One-class SVM in a 2D data set. The solid line represents the decision boundary, separating positive (training instances) and negative (anomalous) data points (Figure source: publications.waset.org/13827/one-class-support-vector-machines-for-protein-protein-interactions-prediction).

Isolation Forest

Isolation Forest [35] is an unsupervised method that is based on an ensemble of decision trees. The algorithm tries to split the data points such that each data point is isolated

from the others. To isolate a data point, the algorithm recursively generates partitions by randomly selecting a feature and then randomly selecting a split value for that feature within the feature values range. The key idea is that anomalous instances in a dataset tend to be easier to isolate (i.e., separate from the rest of the data), compared to normal points, in terms of the number of partitions required. This idea is illustrated in **Figure 10**. The number of partitions required is equivalent to the path length from the root node to the terminating node, in the representative tree structure. Accordingly, an anomaly score is assigned to each data point based on its average depth in the ensemble of isolation trees.

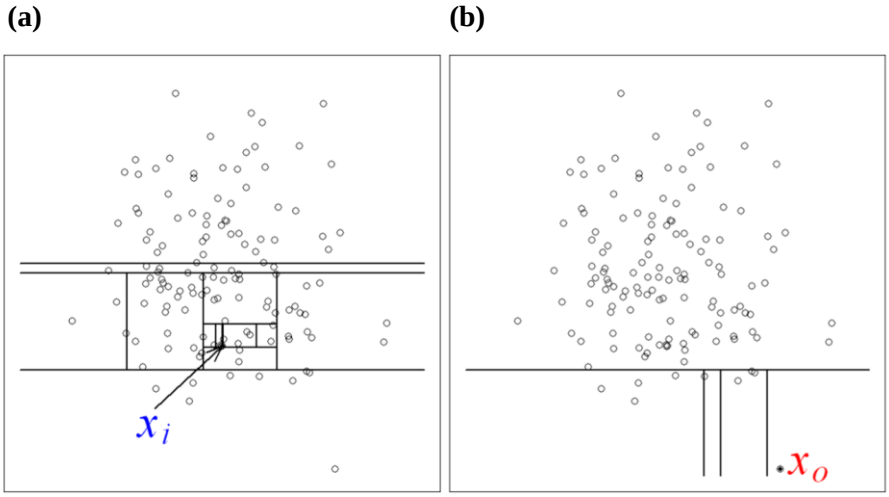


Figure 10: An example of isolating points in a 2D data set, using the Isolation Forest algorithm. (a) Isolating a normal point x_i requires a high number of random splits. (b) Isolating the anomalous point x_o requires less random splits (Figure source: Liu et al. [35]).

Local Outlier Factor (LOF)

The Local Outlier Factor (LOF) [36] algorithm is an unsupervised anomaly detection method that calculates the local density deviation of a given data point with respect to the local densities of its neighbors. The basic idea is that an outlier has a substantially

lower density than its neighbors (see illustration in **Figure 11**). The local density is estimated using additional measures, termed *k-distance* and *reachability distance*, which are used to calculate the $LOF_k(x)$ scores for each datapoint x and for any choice of k. $LOF_k \sim 1$ indicates similar density as neighbors. $LOF_k < 1$ indicates higher density than neighbors, and $LOF_k > 1$ indicates lower density than neighbors, that is, an outlier.

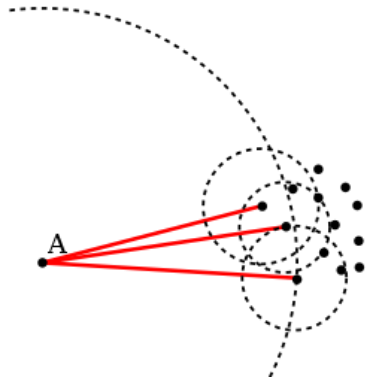


Figure 11: Illustration of the idea of LOF. The local density of point A is lower than the local densities of its neighbors. (Figure source: en.wikipedia.org/wiki/Local_outlier_factor).

4. Methods

4.1. Cohort Description

We conducted a retrospective study on two cohorts. The *development cohort* consisted of EHRs of all COVID-19 positive adults admitted to Sheba between March and December 2020. The *validation cohort* consisted of EHRs of all COVID-19 positive patients admitted to TASMC between March and September 2020. The study was reviewed and approved by the Sheba Medical Center Institutional Review Board (number 20-7064) and by the Tel Aviv Sourasky Medical Center Institutional Review Board (number 0491-17), and conformed to the principles outlined in the declaration of Helsinki. All methods were performed in accordance with the relevant guidelines and regulations. Patient data was anonymized. The IRBs approved the waiver of informed consent.

The data used was extracted from longitudinal EHRs and included both time-independent (static) and temporal (dynamic) features from the entire hospitalization period. The static features were age, sex, weight, BMI and background diseases. The background diseases included hypertension, diabetes, cardiovascular diseases, chronic obstructive pulmonary disease (COPD), chronic kidney disease (CKD), cancer, hepatitis B and human immunodeficiency virus (HIV). The dynamic features include measurement of vital signs (including oxygen saturation), complete blood count (CBC), basic metabolic panel (BMP), blood gases, coagulation panel and lipids panel, including kidney and liver function tests, and inflammatory markers (**Supplementary Table 2**). Features with more than 40% missing values or with zero variance were excluded. The temporal data was discretized to hourly intervals and multiple values

measured within the same hour were aggregated by mean. We use the term *observation* for the vector of hourly aggregated feature values of the patient. An observation was formed if at least one measurement was recorded in that hour.

While our goal was to predict individual positive observations, in order to provide early warning, a closely related question is the prevalence of continuously deteriorating patients. To answer this question, we defined continuously deteriorating patients as those who had a period of 12 consecutive hospitalization hours with at least two mNEWS2 measurements, the majority of which had scores ≥ 7. 25.2% and 21.1% of the patients in Sheba and TASMC, respectively, satisfied this criterion. Notably, the correlation between mortality and deterioration according to this criterion was ~0.5 in both datasets.

4.2. Inclusion and Exclusion Criteria

Inclusion criteria: Adult patients (age≥18) with at least one mNEWS2 score.

Exclusion criteria: Patients who were in a severe state upon their admission, defined as having mNEWS2 score≥7 in the first 12 hours after admission (n=156 patients). Observations from the six hours period prior to a deterioration event, as we wish to predict at least six hours in advance (n=28,069 observations), and observations from the eight hours after the deterioration event (n=5,157 observations). These two exclusions criteria defined the blocked prediction period during which no predictions are made (**Figure 12**). Observations where no mNEWS2 score was available in the next 30 hours, for which predictions could not be compared to the true outcome (n=9,812 observations). Patients with no laboratory results for BMP, CBC and coagulation

during their entire hospitalization, since our model is based mainly on laboratory features (n=15 patients). Patients' observations with ≥60% of the feature values missing (424 observations).

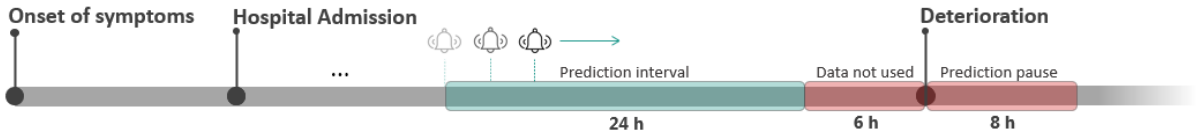


Figure 12: Patient timeline from symptoms to deterioration. Data from the entire hospitalization period was utilized for model prediction, starting from the hospital admission. The green interval is when the deterioration predictions are made. Red areas represent blocked prediction periods during which no predictions are made: the six hours period prior to the deterioration event, and the eight hours period afterward. The length of the prediction window and the blocked prediction periods were predefined with clinical experts and can be easily tailored to fit other clinical settings.

4.3. Outcome Definition

The mNEWS2 scores were routinely calculated and updated in the EHR systems, as part of clinical care (see calculation protocol in **Supplementary Table 1**). The mean time period between two consecutive mNEW2 records was ~2.7 hours in the development set before applying the inclusion and exclusion criteria, and ~2.5 hours afterward. Observations with mNEWS2 score ≥ 7 recorded in the next 7-30 hours were called positive, and the rest were called negative. Notably, observations where no mNEWS2 score was available in the next 30 hours were excluded (see section 4.2).

4.4. Outlier Removal

To remove grossly incorrect measurements due to manual typos or technical issues, we manually defined with clinicians a range of possible values (including pathological values) per each feature (**Supplementary Table 3**) and removed values outside this range. In total, 43,507 values were excluded.

4.5. Data Imputation

Missing values were observed mainly in lab tests and vital signs. We used linear interpolation for imputing missing data. The remaining missing data (e.g., missing values in observations that occurred before the first measurement of a feature, or features that were not measured for a patient at all) were imputed using the multivariate Iterative Imputer algorithm, implemented in the scikit-learn library in Python [37], which was inspired by *MICE* (Multivariate Imputation by Chained Equation) [33]. The Iterative Imputer uses regression to model each feature with missing values as a function of other features, in a round-robin fashion. In each round, each of the features is imputed in this way. The dataset obtained in the final round serves as the imputed dataset.

4.6. Feature Engineering

We created summary statistics over time windows of varying sizes to capture the temporal behavior of the data. The summary statistics were generated for 21 dynamic

features that were reported as risk factors for severe COVID-19 in previous studies [38]–[42] (**Supplementary Table 3**). We defined two time windows covering the last 24 and 72 hours. For each time window, the summary statistics extracted were the mean, difference between the current value and the mean, standard deviation, minimum and maximum values. In addition, we extracted the same summary statistics based on the entire hospitalization period so far, with the addition of the linear regression slope (the regression coefficient). To capture recent data patterns, the difference and trend of the last two observed values ($(v_2 - v_1)$ and $\frac{v_2 - v_1}{t_2 - t_1}$ for values v_1, v_2 recorded in times t_1, t_2 respectively) were generated as well. In addition, to capture interactions between pairs of variables, we generated features for the ratios of each pair of variables in the risk factors subset (for example, neutrophils to lymphocytes ratio).

As imputation masks the information about the measurement frequency, we added features that capture the time since the last non-imputed measurement. While these features indeed improved our performance, the intensity of monitoring of a patient may reflect her medical condition (a deteriorating patient will tend to have more frequent measurements). As we aimed to predict deterioration when it is not yet anticipated, we chose not to include these features in the developed model, since they can create bias due to measurement intensity.

We also added to the model features that aimed to estimate how much an observation is irregular. We applied three anomaly detection approaches, One-Class SVM [34], Isolation Forest [35], and local outlier factor (LOF) [36] to each hourly observation. Eventually, none of the anomaly features was included in the final model after the feature selection.

4.7. Model Development and Feature Selection

We performed a binary classification task for every hourly observation to predict deterioration in the next 7-30 hours. Deterioration was defined as mNEWS2≥7. As deterioration can usually be predicted by the physician several hours in advance, based on signs and symptoms, observations from the six hours prior to the deterioration event were excluded (**Figure 12**). Once deterioration has occurred, no predictions were made in the next 8 hours, and observations during that period were excluded. The length of the prediction window (30 hours) and the blocked prediction windows (six hours before and eight hours after the event) were predefined with our clinical experts. These lengths can be easily tuned to fit other clinical settings. The predictions start with data collection (namely, on hospital admission), as long as the available data so far meet the inclusion and exclusion criteria, in terms of missing rate, blocked prediction windows and additional considerations (see "Inclusion and Exclusion Criteria").

We evaluated ten supervised machine learning models for this prediction task: linear regression [21], [22], logistic regression [23], naïve Bayes, SVM [24], random forest [26] and several algorithms for gradient boosting of decision trees, including XGBoost [27] and CatBoost [28]. The hyperparameters of the models were determined using grid search over predefined ranges of possible values. The hyperparameter settings are listed in **Supplementary Table 4**. Data standardization was performed prior to model training when needed (for example, for SVM).

To handle the high dimension of our data after the feature engineering process, we examined two strategies or feature selection. The first selected the 100 features with the highest correlation with the target. The second used feature importance as calculated by XGBoost. Cross-validation of all algorithms was performed, where, in each

iteration, the top 100 features, according to each strategy, were used in training. Based on the cross-validation results, we chose the second strategy. We trained XGBoost on the full imputed training dataset and used the computed feature importance scores to select the top 100 features for training the final model (**Supplementary Table 5**).

4.8. Evaluation Approach

We partitioned the development dataset into 80% *training* and 20% *testing sets* (**Figure 13**). To avoid bias resulting from changes in clinical practice over time, the partition was done randomly across the hospitalization dates.

To estimate the robustness of the models on different patients and time periods, we used 20-fold cross-validation over the training set, and measured model performance using the area under the receiver-operator characteristics curve (AUROC) and the area under the precision-recall curve (AUPR). The testing set was used to evaluate the final model performance within the same cohort.

Finally, we used the validation dataset (TASMC) for external evaluation.

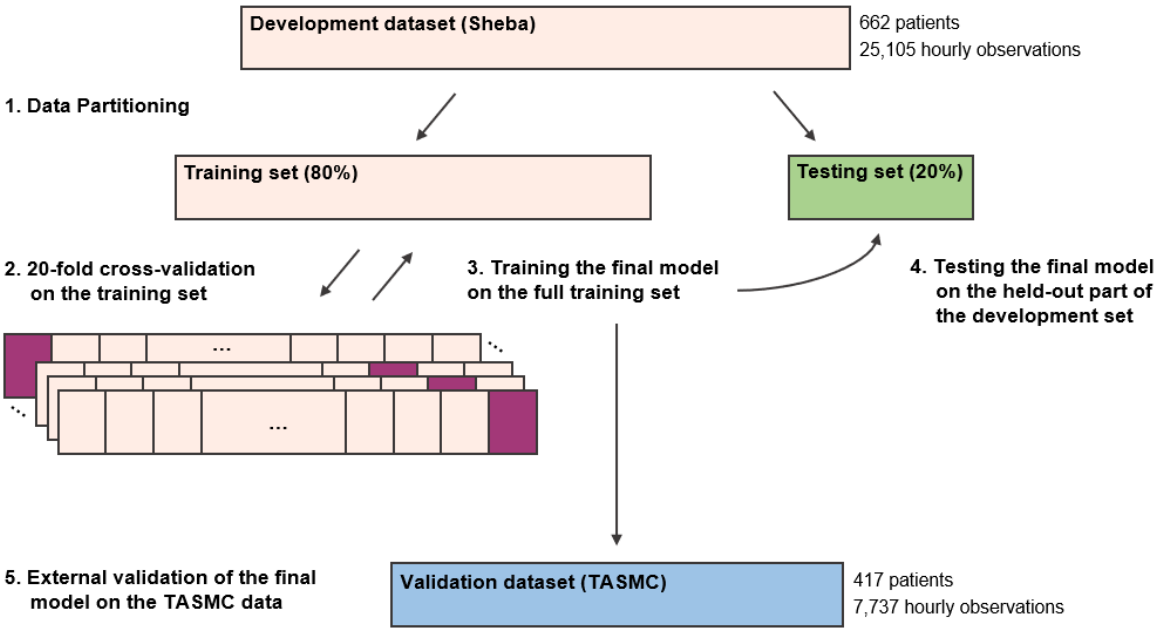


Figure 13: Outline of the data partition and model development. First, the development dataset (Sheba) was split into 80% training and 20% testing subsets (1). To estimate the performance of 14 machine learning models, 20-fold cross-validation over the training set was performed (2). Then, the final model was trained over the entire training set (3) and evaluated on the testing set (4). External validation was done on the validation set (TASMC) (5).

5. Results

5.1. Cohort Description

The development dataset consisted of all patients admitted to Sheba between March and December 2020 that tested positive for SARS-CoV-2. The validation dataset consisted of all patients admitted to TASMC between March and September 2020 who tested positive for SARS-CoV-2. The data used was extracted from structured longitudinal EHRs covering the entire hospitalization period, starting from the hospital admission. The data included both time-independent (static) and temporal (dynamic) features, such as demographics, background disease, vital signs and lab measurements (**Supplementary Table 2**). We use the term *observation* for the vector of hourly aggregated feature values of a patient. A new observation was formed whenever at least one measurement was recorded in that hour.

After applying the inclusion and exclusion criteria (see "Methods"), the development set contained 25,105 hourly observations derived from 662 patients; the validation set had 7,737 observations derived from 417 patients. The characteristics of the first measurements upon admission of the datasets are described in **Supplementary Table 2**.

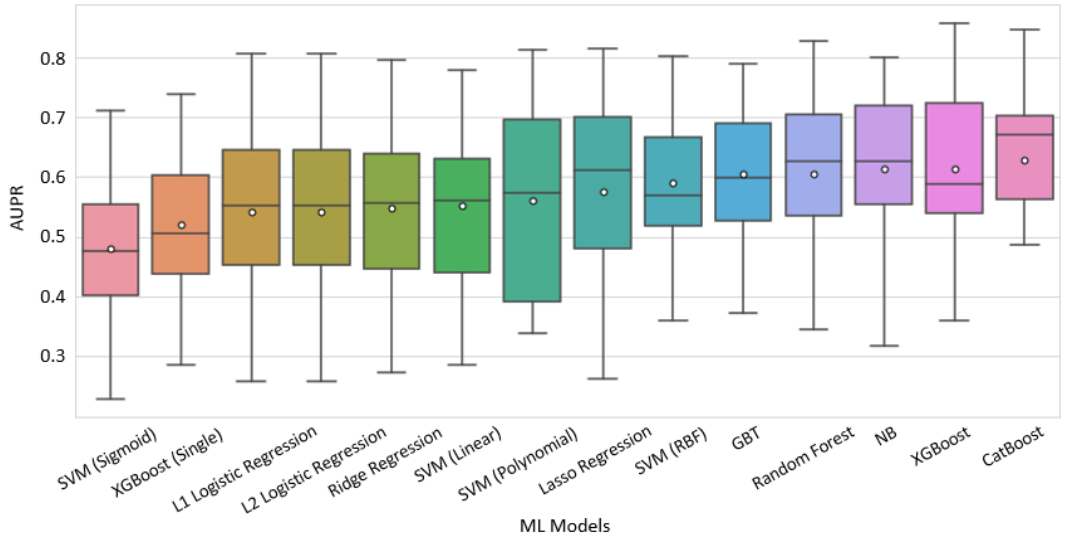
We defined the deterioration outcome as a recorded high mNEWS2 score (≥7), and aimed to predict such outcomes 7-30 hours in advance (**Figure 12**). Higher mNEWS2 scores were associated with higher mortality and ICU admissions rates in the development dataset (**Supplementary Figure 1**).

5.2. COVID-19 Deterioration Model

Our models predict the risk of deterioration for each hour that contains a new measurement. The development set was split into a training and testing subsets (**Figure 13**), where the training set consisted of 20,029 hourly observations derived from 530 patients, of which 6,349 (~31%) were labeled positive (mNEWS2≥7 in the next 7-30 hours). We trained 14 models on the training set.

Figure 14 summarizes the performance of 14 classifiers in cross-validation on the training set. Classifiers based on an ensemble of decision trees (CatBoost, XGBoost, Random Forest) performed best overall. We chose CatBoost as our final prediction model and trained it on the entire training set. Its results on the development testing set are shown in **Figure 15**. It had good discrimination and achieved AUROC of 0.84 and AUPR of 0.74. To estimate the robustness of the model, we performed a bootstrap procedure with 100 iterations, where, in each iteration, a sample with size half of the testing set was randomly selected from the testing set with replacement. The mean and standard deviation of the AUROC and the AUPR over these experiments achieved comparable results to those of the total testing set (Figure **15a-b**). **Figure 15c** presents a calibration curve of the model, showing good agreement between the predicted and observed probabilities for deterioration.

(a)



(b)

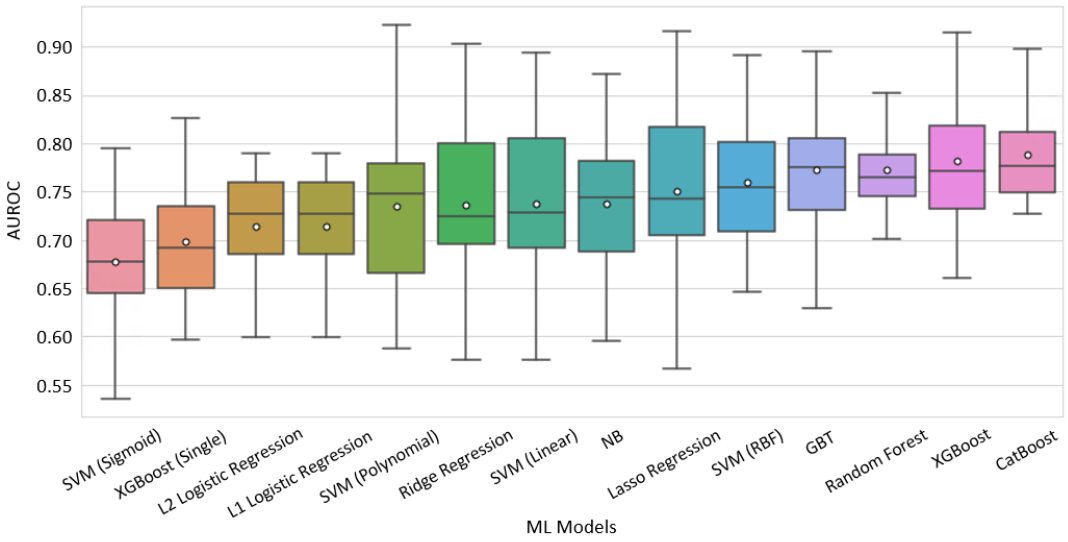


Figure 14: Performance of 14 machine learning models that predict mNEWS2≥7. Comparison of machine learning methods using 20-fold cross-validation over the training set within the development dataset. (a) AUPR. (b) AUROC. The horizontal line indicates the median, and the white circle indicates the mean. The models are sorted by the mean AUC.

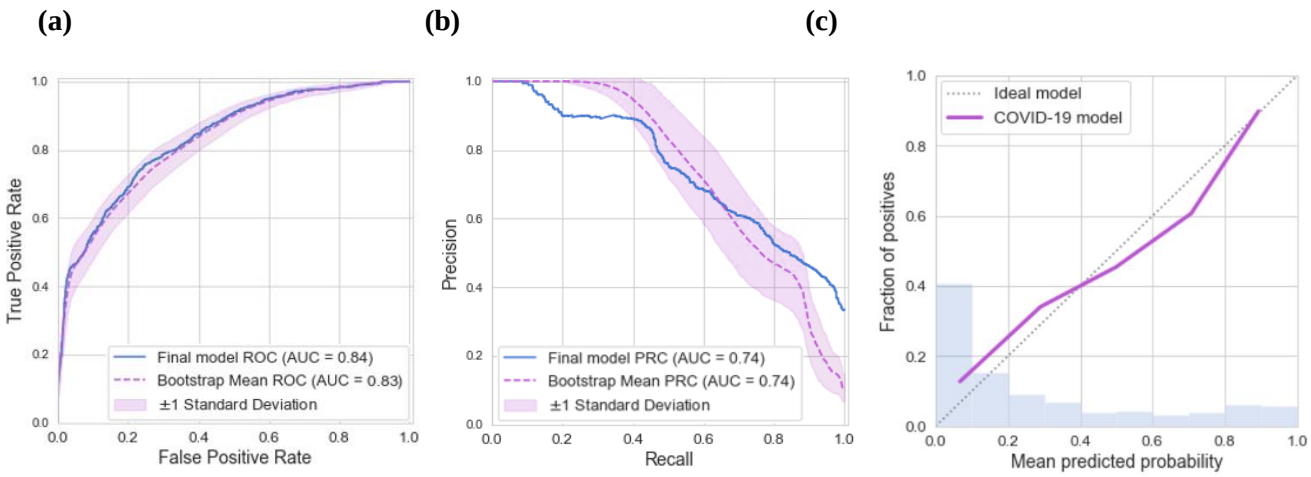


Figure 15: Performance of the final model on the testing set within the development set. (a) AUROC. (b) AUPR. Solid curves were computed on the total set. Dashed curves were computed with a bootstrap procedure with 100 iterations, where, in each iteration, a number of observations equals to half of the testing set was sampled from the testing set with replacement. (c) Calibration plot for the relationship between the predicted and observed probabilities for COVID-19 deterioration. The dashed diagonal line represents an ideal calibration. The purple line represents the actual model performance in five discretized bins. The blue histogram is the distribution of the risk predictions.

When using a classification threshold of 0.7 in the final model (namely, classifying as positive all observations with risk score > 0.7, and the rest as negative), it achieved an accuracy of 80% with a positive predictive value (PPV) of 87% on the testing set. Performance metrics for various classification thresholds are shown in **Table 1**.

To assess the contribution of each feature to the final model prediction, we used SHAP values [43]. The top 20 important features of the model are summarized in **Figure 16**. Age, arterial oxygen saturation, maximal LDH value and the standard deviation of body temperature were the most important features for predicting deterioration. An evaluation of feature importance as calculated by the CatBoost algorithm gave similar results (**Supplementary Figure 2**).

Threshold	Accuracy	Sensitivity	Specificity	PPV	NPV
0.1	0.66	0.88	0.56	0.48	0.91
0.2	0.74	0.78	0.71	0.56	0.87
0.3	0.77	0.69	0.80	0.62	0.84
0.4	0.79	0.60	0.87	0.69	0.82
0.5	0.79	0.55	0.91	0.73	0.81
0.6	0.79	0.48	0.94	0.79	0.80
0.7	0.80	0.44	0.97	0.87	0.79
0.8	0.78	0.34	0.98	0.90	0.76
0.9	0.73	0.17	0.99	0.93	0.72

Table 1: Performance metrics of the final model on the testing set for different thresholds. PPV: positive predictive value, NPV: negative predictive value.

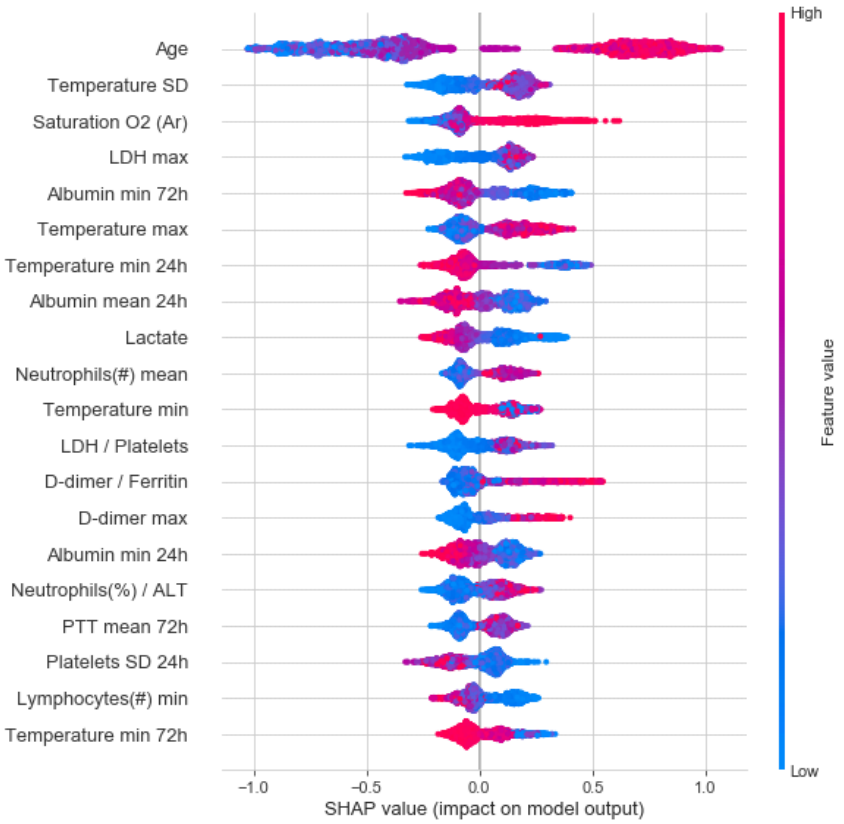


Figure 16: 20 features with highest mean absolute SHAP values. Features (rows) are ordered in decreasing overall importance to the prediction. The plot for each feature shows the SHAP value for each observation on the x-axis, with color representing the value of the feature from low (blue) to high (red). The absolute value indicates the extent of the contribution of the feature, while its sign indicates whether the contribution is positive or negative. SD: standard deviation; /: the ratio between two features. 24h,72h: time windows within the statistic was computed. If not mentioned, the statistics is calculated on the entire hospitalization period so far.

5.3. External validation

The dataset from TASMC was used for external validation of the final model. The results (**Figure 17**) show good performance with AUC 0.76 and AUPR 0.7, albeit less than in the development dataset. A certain reduction in performance is expected when validating a predictor on an independent data source. The slight decrease in performance here can be explained, in part, by the lower temporal resolution of the TASMC dataset, as well as by the higher rate of missing values.

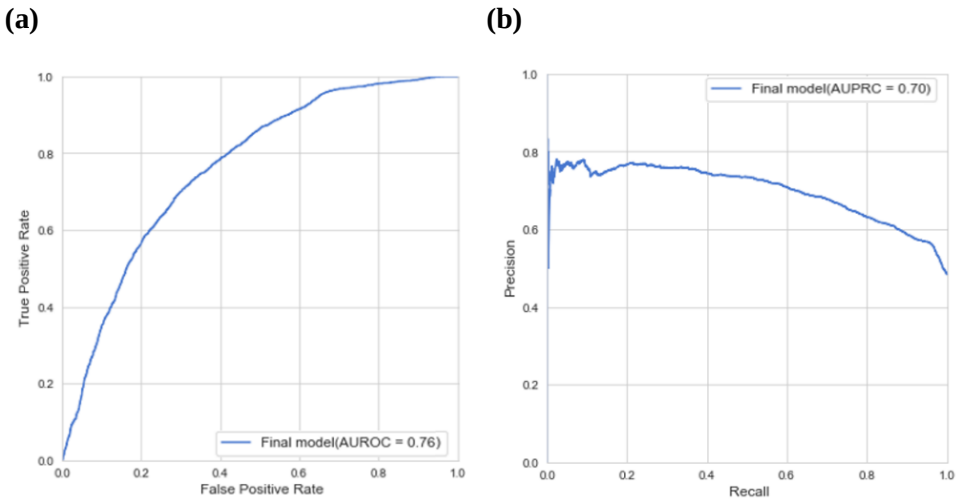


Figure 17: External validation of the final model on the TASMC data. (a) AUROC. (b) AUPRC.

6. Discussion

We utilized machine learning models for predicting a deterioration event in the next 7-30 hours based on EHR data of adult COVID-19 inpatients. Deterioration was defined as a high COVID-19 early warning score (mNEWS2≥7). On held-out data, the model achieved AUROC of 0.84 and AUPR of 0.74. The model was tested on an independent patient cohort from a different hospital and demonstrated comparable performance, with only a modest decrease. The TASMC data had less frequent measurements than Sheba's. The slightly lower performance of the model on the TASMC cohort can be explained by its lower density and by the hourly discretization, which was chosen based on the Sheba data. Using our predictor, we could anticipate deterioration of patients 7-30 hours in advance. Such early warning can enable timely intervention, which was shown to be beneficial [5].

Several previous studies have assessed the utility of machine learning for predicting deterioration in COVID-19 patients [38], [44]–[47]; see also [48] for a review. Most studies used strict criteria as their primary outcomes, such as mechanical ventilation, ICU admission, and death. However, the mNEWS2 score provides a more dynamic measure for clinical deterioration, allowing to trace patient conditions throughout the hospitalization. Since the mNEWS2 score is broadly adopted as a yardstick of COVID-19 inpatient status in medical centers around the world, we believe that demonstrating early prediction of high scores could provide valuable insights to physicians and bring to their attention particular patients that are predicted to be at high risk to deteriorate in the near future. Notably, our model can be readily adapted to other criteria for deterioration, e.g., mechanical ventilation or other mNEWS2 cutoffs.

Consistently with previous studies [38], [39], [44]–[47] , we confirmed the importance of known medical and inflammatory markers for severe COVID-19, such as age, body temperature, oxygen saturation, LDH and albumin. While most previous studies used only raw variables as features, our work emphasizes the importance of including summary statistics, such as the standard deviation of body temperature, for predicting the risk of COVID-19 deterioration. We note that, despite its previously reported importance [38]–[40], [47], C-reactive protein was excluded from our analysis since it was not consistently available in our data.

Most previous works that predicted deterioration utilized only baseline data, obtained on admission or a few hours thereafter [38], [44]–[47]. Thus, they sought to predict the risk of a single deterioration event, possibly several days before its occurrence. Razavian et al. used data from the entire hospitalization period, but for prediction of favorable outcomes [49]. The novelty of our methodology lies in the fact that our model generates repeatedly updated predictions for each patient during the hospitalization, using both baseline and longitudinal data. This enables the identification of patients at risk throughout the hospitalization, while accounting for the temporal dynamics of the disease, allowing adjusted patient therapy and management. All predictions refer to events at least seven hours in advance, enabling early detection of patients at risk. Moreover, unlike many other prediction models, (see [48]), our method was validated on data from a different center.

The final model used in this work was CatBoost, an algorithm for gradient boosting on decision trees. Such models have been successfully applied to various clinical applications [20], [50]–[52]. They are often best performers for relatively small datasets, and have the additional advantage of being easily interpretable, an important factor in using machine learning models in the clinical setting [53]. Deep learning

approaches often do better when powered by massive amounts of data [54]-[56]. With a larger sample size, we intend to take advantage of deep architectures in future work, including variants of recurrent neural network (RNN).

Our study has several limitations. First, it is retrospective, and model development was done based on data from a single center, which may limit its generalizability to external cohorts, especially considering the high variability of COVID-19 outcomes. Second, the mNEWS2 scores present a noisy signal, with frequent changes in the severity condition during the hospitalization. This impairs the score's ability to be used as a robust predictor, compared to other approaches for predicting deterioration [45], [57] , which use other signals, such as initiation of mechanical ventilation or death.

A potential concern is that a deteriorating patient will tend to have more frequent mNEWS2 measurements. This may bias our model and impair its adaptability to a general population of patients. To mitigate bias due to measurement intensity, we chose to exclude features that capture measurement frequency, although including them can improve performance. In addition, the training data had a majority of negative observations (~69%), showing that mild and modest conditions are well represented in the data. Furthermore, by summarizing measurements per hour we mask the measurement intensity within the same hour. Future work could examine time discretization over longer time windows and utilization of balancing techniques.

It is a major challenge to develop a model that is robust to the changes in clinical care or in the characteristics of the disease over time. Our current model was developed on data collected over ten months. Its partition into training and testing subsets was done randomly across the hospitalization dates, to avoid the bias resulted from clinical changes over time. In order to evaluate the model robustness and generalizability across

the different pandemic periods, we are currently evaluating a cross-validation procedure adjusted for waves, termed *wave-fold* cross-validation. The wave-fold cross validation can be used to evaluate machine learning models for longitudinal time-series data over different periods, in addition to the traditional cross validation (see illustration in **Supplementary Figure 3**). Future work might utilize data collected from additional pandemic periods and evaluate the model using wave-fold cross-validation.

Future work should examine additional data imputation approaches for handling missing data. Such methods could have a large effect on the performance of a predictive model [58]. Various imputation methods are commonly used today, but when it comes to individualized time-series clinical data, some of the popular approaches are limited. Instead of incorporating data imputation as an integral pre-processing step, it can be treated as a hyperparameter and tuned in cross-validation.

To date, only a few prognostic COVID-19 models have been prospectively validated or implemented in clinical practice [49], [59]. The adoption of a model into clinical workflows requires the completion of several steps. First, to avoid site-specific learning, the model should be validated across several healthcare centers. Second, the model should be integrated into the institution's EHR system, so that each variable is extracted from the database and fed into the pipeline in real time. Third, prospective validation should be performed to assess the performance of the deployed model. Our study was done with future deployment in mind on several levels. It spanned two centers, with one used for validation only, and we plan to extend the study to additional centers. Collaborating with our clinical experts, we incorporated clinical standards into model development, for example when defining the inclusion and exclusion criteria and by addressing potential biases. In addition, by using SHAP values, we provided a decision support tool that could be interpretable to clinicians. Furthermore, the

deterioration threshold (mNEWS2 cutoffs) and the prediction window (the time interval in the future for which the predictions are made), can be easily tuned, enabling tailored alarm policy for clinical setting (e.g., how often the alarm is raised). Future prospective validation is needed to assess the impact of the deployed model on patient outcomes.

In conclusion, machine learning-based prognostic tools have great potential for both care decisions and resource utilization in hospital wards. We described the development and validation of a model for the prediction of deterioration of COVID-19 inpatients within the next 7-30 hours. In spite of the fact that the disease is novel and of high complexity, our model provides useful predictions for risk of deterioration, with good discrimination. Early detection and treatment of COVID-19 patients at high risk of deterioration can lead to improved treatment and to reduction in mortality. Further validation of this vision is needed.

7. References

[1] D. Cucinotta and M. Vanelli, "WHO declares COVID-19 a pandemic," *Acta Biomedica*, vol. 91, no. 1. Mattioli 1885, pp. 157–160, Mar. 19, 2020. doi: 10.23750/abm.v91i1.9397.

[2] "COVID-19 Map - Johns Hopkins Coronavirus Resource Center." https://coronavirus.jhu.edu/map.html (accessed Jun. 04, 2021).

[3] F. Lapostolle *et al.*, "Clinical features of 1487 COVID-19 patients with outpatient management in the Greater Paris: the COVID-call study," *Internal and Emergency Medicine*, vol. 15, no. 5, pp. 813–817, Aug. 2020, doi: 10.1007/s11739-020-02379-z.

[4] C. Huang *et al.*, "Clinical features of patients infected with 2019 novel coronavirus in Wuhan, China," *The Lancet*, vol. 395, no. 10223, pp. 497–506, Feb. 2020, doi: 10.1016/S0140-6736(20)30183-5.

[5] D. Mathies *et al.*, "A case of SARS-CoV-2 pneumonia with successful antiviral therapy in a 77-year-old man with a heart transplant," *American Journal of Transplantation*, vol. 20, no. 7, pp. 1925–1929, Jul. 2020, doi: 10.1111/ajt.15932.

[6] D. M. Bravata *et al.*, "Association of intensive care unit patient load and demand with mortality rates in US department of veterans affairs hospitals during the COVID-19 pandemic," *JAMA Network Open*, vol. 4, no. 1, p. e2034266, Jan. 2021, doi: 10.1001/jamanetworkopen.2020.34266.

[7] "National Early Warning Score (NEWS) 2 | RCP London." https://www.rcplondon.ac.uk/projects/outputs/national-early-warning-score-news-2 (accessed Jan. 28, 2021).

[8] N. Asai *et al.*, "Efficacy and accuracy of qSOFA and SOFA scores as prognostic tools for community-acquired and healthcare-associated pneumonia," *International Journal of Infectious Diseases*, vol. 84, pp. 89–96, Jul. 2019, doi: 10.1016/j.ijid.2019.04.020.

[9] J. D. Chalmers *et al.*, "Severity assessment tools to guide ICU admission in community-acquired pneumonia: Systematic review and meta-analysis," *Intensive Care Medicine*, vol. 37, no. 9. Springer, pp. 1409–1420, Sep. 10, 2011. doi: 10.1007/s00134-011-2261-x.

[10] X. Liao, B. Wang, and Y. Kang, "Novel coronavirus infection during the 2019–2020 epidemic: preparing intensive care units—the experience in Sichuan Province, China," *Intensive Care Medicine*, vol. 46, no. 2, pp. 357–360, Feb. 2020, doi: 10.1007/s00134-020-05954-2.

[11] A. Fred, T. M. Caelli, R. P. W. Duin, A. C. Campilho, and D. de Ridder, Eds., *Structural, Syntactic, and Statistical Pattern Recognition*, vol. 3138. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. doi: 10.1007/b98738.

[12] H. M. Krumholz, "Big data and new knowledge in medicine: The thinking, training, and tools needed for a learning health system," *Health Affairs*, vol. 33, no. 7, pp. 1163–1170, Aug. 2014, doi: 10.1377/hlthaff.2014.0053.

[13] O. Noy *et al.*, "A machine learning model for predicting deterioration of COVID-19 inpatients," *Scientific Reports 2022 12:1*, vol. 12, no. 1, pp. 1–9, Feb. 2022, doi: 10.1038/s41598-022-05822-7.

[14] N. Nabavi, "Long covid: How to define it and how to manage it," *BMJ*, vol. 370, p. m3489, Sep. 2020, doi: 10.1136/BMJ.M3489.

[15] W. Guan *et al.*, “Clinical characteristics of coronavirus disease 2019 in China,” *New England Journal of Medicine*, vol. 382, no. 18, pp. 1708–1720, Apr. 2020, doi: 10.1056/NEJMOA2002032/SUPPL_FILE/NEJMOA2002032_DISCLOSURES.PDF.

[16] F. Zhou *et al.*, “Clinical course and risk factors for mortality of adult inpatients with COVID-19 in Wuhan, China: a retrospective cohort study,” *The Lancet*, vol. 395, no. 10229, pp. 1054–1062, Mar. 2020, doi: 10.1016/S0140-6736(20)30566-3.

[17] J. R. Ayala Solares *et al.*, “Deep learning for electronic health records: A comparative review of multiple deep neural architectures,” *Journal of Biomedical Informatics*, vol. 101, p. 103337, Jan. 2020, doi: 10.1016/J.JBI.2019.103337.

[18] A. E. W. Johnson, M. M. Ghassemi, S. Nemati, K. E. Niehaus, D. Clifton, and G. D. Clifford, “Machine learning and decision support in critical care,” *Proceedings of the IEEE*, vol. 104, no. 2, pp. 444–466, Feb. 2016, doi: 10.1109/JPROC.2015.2501978.

[19] B. Shickel, P. J. Tighe, A. Bihorac, and P. Rashidi, “Deep EHR: A survey of recent advances in deep learning techniques for Electronic Health Record (EHR) analysis,” *IEEE Journal of Biomedical and Health Informatics*, vol. 22, no. 5, pp. 1589–1604, Sep. 2018, doi: 10.1109/JBHI.2017.2767063.

[20] S. L. Hyland *et al.*, “Early prediction of circulatory failure in the intensive care unit using machine learning,” *Nature Medicine 2020 26:3*, vol. 26, no. 3, pp. 364–373, Mar. 2020, doi: 10.1038/s41591-020-0789-4.

[21] R. Tibshirani, “Regression shrinkage and selection via the Lasso,” *Journal of the Royal Statistical Society.* Series B Vol. 58, No. 1 (1996), pp. 267-288.

[22] A. E. Hoerl and R. W. Kennard, “Ridge regression: Biased estimation for nonorthogonal problems,” *Technometrics*, vol. 12, no. 1, pp. 55–67, 1970, doi: 10.1080/00401706.1970.10488634.

[23] R. E. Wright, “Logistic regression,” *Reading and understanding multivariate statistics.* American Psychological Association, pp. 217–244, 1995, Accessed: Apr. 04, 2022. [Online]. Available: https://psycnet.apa.org/record/1995-97110-007.

[24] C. Cortes and V. Vapnik, “Support-vector networks,” *Machine Learning*, vol. 20, no. 3, pp. 273–297, Sep. 1995, doi: 10.1007/bf00994018.

[25] J. R. Quinlan, “Induction of decision trees,” *Machine Learning 1986 1:1*, vol. 1, no. 1, pp. 81–106, Mar. 1986, doi: 10.1007/BF00116251.

[26] L. Breiman, “Random forests,” *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct. 2001, doi: 10.1023/A:1010933404324.

[27] T. Chen and C. Guestrin, “XGBoost: A scalable tree boosting system,” in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, Aug. 2016, vol. 13-17-August-2016, pp. 785–794. doi: 10.1145/2939672.2939785.

[28] L. Prokhorenkova, G. Gusev, A. Vorobev, A. V. Dorogush, and A. Gulin, “CatBoost: unbiased boosting with categorical features,” *Advances in Neural Information Processing Systems*, vol. 31, 2018, Accessed: Feb. 04, 2022. [Online]. Available: https://github.com/catboost/catboost

[29] J. Schmidhuber, “Deep learning in neural networks: An overview,” *Neural Networks*, vol. 61, pp. 85–117, Jan. 2015, doi: 10.1016/J.NEUNET.2014.09.003.

[30] J. L. Elman, “Finding structure in time,” *Cognitive Science*, vol. 14, no. 2, pp. 179–211, Mar. 1990, doi: 10.1207/S15516709COG1402_1.

[31] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[32] O. F. Ayilara, L. Zhang, T. T. Sajobi, R. Sawatzky, E. Bohm, and L. M. Lix, “Impact of missing data on bias and precision when estimating change in patient-reported outcomes from a clinical registry,” *Health Qual Life Outcomes*, vol. 17, no. 1, Jun. 2019, doi: 10.1186/S12955-019-1181-2.

[33] S. van Buuren and K. Groothuis-Oudshoorn, “mice: Multivariate imputation by chained equations in R,” *Journal of Statistical Software*, vol. 45, no. 3, pp. 1–67, Dec. 2011, doi: 10.18637/jss.v045.i03.

[34] B. Schölkopf, A. J. Smola, R. C. Williamson, and P. L. Bartlett, “New support vector algorithms,” *Neural Computation*, vol. 12, no. 5, pp. 1207–1245, May 2000, doi: 10.1162/089976600300015565.

[35] F. T. Liu, K. M. Ting, and Z. H. Zhou, “Isolation forest,” in *Proceedings - IEEE International Conference on Data Mining, ICDM*, 2008, pp. 413–422. doi: 10.1109/ICDM.2008.17.

[36] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, “LOF: Identifying density-based local outliers,” *ACM SIGMOD Record*, vol. 29, pp. 93-104, 2000, doi: 10.1145/335191.

[37] F. Pedregosa *et al.*, “Scikit-learn: machine learning in Python,” *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011, Accessed: Dec. 16, 2021. [Online]. Available: http://scikit-learn.sourceforge.net.

[38] A. D. Haimovich *et al.*, “Development and validation of the quick COVID-19 severity index: A prognostic tool for early clinical decompensation,” *Annals of Emergency Medicine*, vol. 76, no. 4, p. 442, Oct. 2020, doi: 10.1016/J.ANNEMERGMED.2020.07.022.

[39] J. Gong *et al.*, “A tool for early prediction of severe coronavirus disease 2019 (COVID-19): A multicenter study using the risk nomogram in Wuhan and Guangdong, China,” *Clinical Infectious Diseases*, vol. 71, no. 15, pp. 833–840, Jul. 2020, doi: 10.1093/cid/ciaa443.

[40] Y. Guo *et al.*, “Development and validation of an early warning score (EWAS) for predicting clinical deterioration in patients with coronavirus disease 2019,” *medRxiv*, p. 2020.04.17.20064691, Apr. 21, 2020. doi: 10.1101/2020.04.17.20064691.

[41] D. Ji *et al.*, “Prediction for progression risk in patients with COVID-19 pneumonia: The CALL score,” *Clinical Infectious Diseases*, vol. 71, no. 6, pp. 1393–1399, Sep. 2020, doi: 10.1093/cid/ciaa414.

[42] X. Liu *et al.*, “Prediction of the severity of the coronavirus disease and its adverse clinical outcomes,” *Japanese Journal of Infectious Diseases*, vol. 73, no. 6, pp. 404–410, Nov. 2020, doi: 10.7883/yoken.JJID.2020.194.

[43] S. M. Lundberg, P. G. Allen, and S.-I. Lee, “A unified approach to interpreting model predictions,” 2017. Accessed: Feb. 04, 2021. [Online]. Available: https://github.com/slundberg/shap

[44] D. Assaf *et al.*, “Utilization of machine-learning models to accurately predict the risk for critical COVID-19,” *Internal and Emergency Medicine*, vol. 15, no. 8, pp. 1435–1443, Nov. 2020, doi: 10.1007/s11739-020-02475-0.

[45] Y. Gao *et al.*, “Machine learning based early warning system enables accurate mortality risk prediction for COVID-19,” *Nature Communications*, vol. 11, no. 1, pp. 1–10, Dec. 2020, doi: 10.1038/s41467-020-18684-2.

[46] F. S. Heldt *et al.*, “Early risk assessment for COVID-19 patients from emergency department data using machine learning,” *Scientific Reports*, vol. 11, no. 1, p. 4200, Dec. 2021, doi: 10.1038/s41598-021-83784-y.

[47] Y. Zheng *et al.*, “A learning-based model to evaluate hospitalization priority in COVID-19 pandemics,” *Patterns*, vol. 1, no. 6, p. 100092, Sep. 2020, doi: 10.1016/j.patter.2020.100092.

[48] L. Wynants *et al.*, “Prediction models for diagnosis and prognosis of covid-19: Systematic review and critical appraisal,” *The BMJ*, vol. 369, p. 26, Apr. 2020, doi: 10.1136/bmj.m1328.

[49] N. Razavian *et al.*, “A validated, real-time prediction model for favorable outcomes in hospitalized COVID-19 patients,” *npj Digital Medicine*, vol. 3, no. 1, pp. 1–13, Dec. 2020, doi: 10.1038/s41746-020-00343-x.

[50] R. J. Delahanty, J. A. Alvarez, L. M. Flynn, R. L. Sherwin, and S. S. Jones, “Development and evaluation of a machine learning model for the early identification of patients at risk for sepsis,” *Annals of Emergency Medicine*, vol. 73, no. 4, pp. 334–344, Apr. 2019, doi: 10.1016/J.ANNEMERGMED.2018.11.036.

[51] J. Zhao *et al.*, “Learning from longitudinal data in electronic health record and genetic data to improve cardiovascular event prediction,” *Scientific Reports 2019 9:1*, vol. 9, no. 1, pp. 1–10, Jan. 2019, doi: 10.1038/s41598-018-36745-x.

[52] R. Wang *et al.*, “Integration of the extreme gradient boosting model with electronic health records to enable the early diagnosis of multiple sclerosis,” *Multiple Sclerosis and Related Disorders*, vol. 47, p. 102632, Jan. 2021, doi: 10.1016/J.MSARD.2020.102632.

[53] J. Amann, A. Blasimme, E. Vayena, D. Frey, and V. I. Madai, “Explainability for artificial intelligence in healthcare: a multidisciplinary perspective,” *BMC Medical Informatics and Decision Making*, vol. 20, no. 1, pp. 1–9, Dec. 2020, doi: 10.1186/S12911-020-01332-6/PEER-REVIEW.

[54] J. Jiang, R. Wang, M. Wang, K. Gao, D. D. Nguyen, and G. W. Wei, “Boosting tree-assisted multitask deep learning for small scientific datasets,” *Journal of Chemical Information and Modeling*, vol. 60, no. 3, pp. 1235–1244, Mar. 2020, doi: 10.1021/ACS.JCIM.9B01184/SUPPL_FILE/CI9B01184_SI_001.PDF.

[55] Z. Che, S. Purushotham, K. Cho, D. Sontag, and Y. Liu, “Recurrent neural networks for multivariate time series with missing values,” *Scientific Reports*, vol. 8, no. 1, pp. 1–12, Dec. 2018, doi: 10.1038/s41598-018-24271-9.

[56] D. Chen et al., “Deep learning and alternative learning strategies for retrospective real-world clinical data,” *npj Digital Medicine* 2019 2:1, vol. 2, no. 1, pp. 1–5, May 2019, doi: 10.1038/s41746-019-0122-0.

[57] N. J. Douville *et al.*, “Clinically applicable approach for predicting mechanical ventilation in patients with COVID-19,” *British Journal of Anaesthesia*, vol. 0, no. 0, 2021, doi: 10.1016/j.bja.2020.11.034.

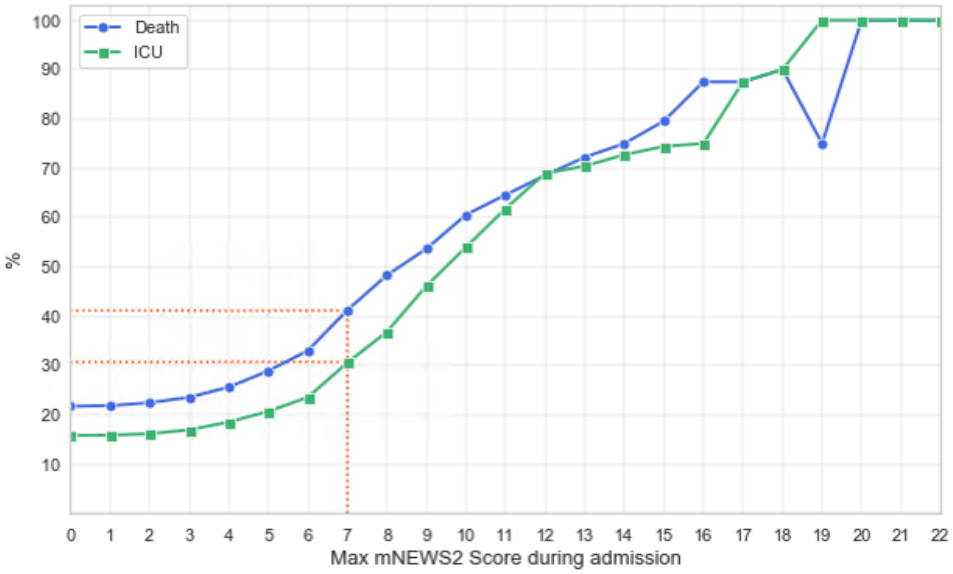
[58] M. K. Hasan, M. A. Alam, S. Roy, A. Dutta, M. T. Jawad, and S. Das, “Missing value imputation affects the performance of machine learning: A review and analysis of the

literature (2010–2021)," *Informatics in Medicine Unlocked*, vol. 27, p. 100799, Jan. 2021, doi: 10.1016/J.IMU.2021.100799.

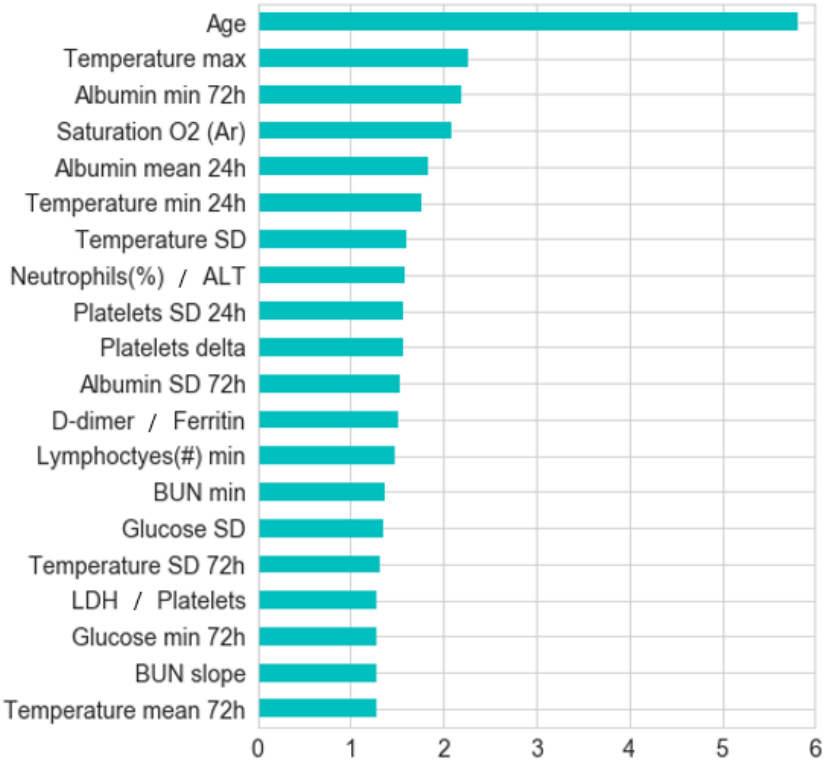
[59] Q. Li *et al.*, "A simple algorithm helps early identification of SARS-CoV-2 infection patients with severe progression tendency," *Infection 2020 48:4*, vol. 48, no. 4, pp. 577–584, May 2020, doi: 10.1007/S15010-020-01446-Z.

8. Supplementary Material

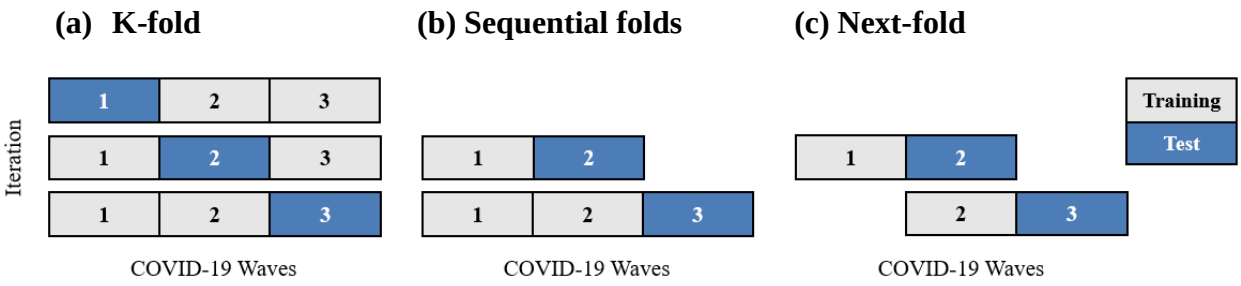
Supplementary Figure 1: Death and ICU admission rates as a function of the maximal mNEWS2 score during hospitalization in the development dataset.



Supplementary Figure 2: Feature importance calculated by CatBoost ("PredictionValuesChange" metric). Features (rows) are ordered in decreasing overall importance. The importance score of a feature (x axis) is determined by how much on average the prediction changes when the feature value changes. A bigger change in the prediction value implies a higher importance. SD: standard deviation; /: the ratio between two features. 24h,72h: time windows within the statistic was computed. When the time window is not mentioned, the measure refers to the entire hospitalization period up to the prediction time.



Supplementary Figure 3: Cross-validation procedures for model evaluation across different COVID-19 waves. The original data is partitioned into k distinct folds representing k waves (here k=3), each fold contains all patients admitted during the corresponding wave. Gray boxes represent training folds and blue boxes represent validation folds. (a) Each fold is used once for validation while the k-1 remaining folds are used as the training set. (b) In the k-th iteration, the first k folds are used as training set, and the (k+1)-th fold is used as test set. (c) In the k-th iteration, the k fold is used as training set, and the (k+1)-th fold as test set.



Supplementary Table 1: The mNEWS2 score. Scores are computed by summing the points for each category. This is an adapted version of the NEWS2 score, with the addition of 3 points for patients with age≥65 [10].

Points	**3**	**2**	**1**	**0**	**1**	**2**	**3**
Age				<65			≥65
Respiratory Rate	≤8		9-11	12-20		21-24	≥25
Oxygen Saturation	≤91	92-93	94-95	≥96			
Supplemental Oxygen		Yes		No			
Systolic BP	≤90	91-100	101-110	111-219			≥220
Heart Rate	≤40		41-50	51-90	91-110	111-130	≥131
Consciousness				Alert			Drowsiness Lethargy Coma Confusion
Temperature	≤35.0		35.1-36.0	36.1-38.0	38.1-39.0	≥39.1	

Score	**Risk Grading**
0	-
1-4	Low
5-6 or 3 in one parameter	Medium
≥7	High

Supplementary Table 2: Table of Characteristics. Population characteristics for the two datasets used to develop and test the model. Characteristics of both static features and first measurements of dynamic features are presented. P-values were calculated using Fisher's exact test and T-test for categorical and numerical values, respectively, and Bonferroni corrected for multiple comparisons. 'AR' and 'V' refer to arterial and venous blood, respectively.

Variable	Development (Sheba Hospital)		Validation (TASMC)		P-Value
	N (%)	Mean ± SD	N (%)	Mean ± SD	
Overall	**662**		**417**		
Age	662 (100.0%)	65.91 ± 16.86	417 (100.0%)	67.16 ± 18.01	1
Gender			1		
Male	391 (59.06%)		241 (57.79%)		
Female	271 (40.94%)		176 (42.21%)		
Morality	144 (21.75%)		73 (17.51%)		1
BMI	497 (75.08%)	28.21 ± 7.51	234 (56.12%)	27.37 ± 5.88	1
Weight	507 (76.59%)	81.11 ± 19.76	245 (58.75%)	77.25 ± 18.48	0.5385
Hypertension			<0.0001		
Yes	287 (43.35%)		60 (14.39%)		
No	375 (59.65%)		357 (85.61%)		
Diabetes			0.0046		
Yes	195 (29.46%)		83 (19.9%)		
No	467 (70.54%)		334 (80.1%)		
Cancer			0.0008		
Yes	91 (13.75%)		26 (6.24%)		
No	571 (86.25%)		391 (93.76%)		
Hepatitis B			1		
Yes	5 (0.76%)		3 (0.72%)		
No	657 (99.24%)		414 (99.28%)		
CKD			1		
Yes	43 (6.5%)		27 (6.47%)		
No	619 (93.5%)		390 (93.53%)		
HIV			1		
Yes	1 (0.15%)		1 (0.24%)		
No	661 (99.85%)		416 (99.76%)		
CVD			0.0046		
Yes	149 (22.51%)		58 (13.91%)		
No	513 (77.49%)		359 (86.09%)		
COPD			1		
Yes	26 (3.93%)		22 (5.28%)		
No	636 (96.07%)		395 (94.72%)		
HBA1C (#)	108 (16.31%)	45.34 ± 15.92	39 (9.35%)	54.11 ± 20.1	0.3497
HBA1C (%)	108 (16.31%)	6.3 ± 1.46	39 (9.35%)	7.1 ± 1.84	0.3629
Albumin	658 (99.4%)	35.81 ± 5.44	377 (90.41%)	38.35 ± 4.85	<0.0001
ALT	660 (99.7%)	33.02 ± 29.06	412 (98.8%)	37.47 ± 72.29	1
AST	661 (99.85%)	46.63 ± 43.84	378 (90.65%)	43.44 ± 35.46	1
BUN	662 (100.0%)	23.42 ± 16.66	417 (100.0%)	22.8 ± 19.15	1
Calcium	662 (100.0%)	8.82 ± 0.64	379 (90.89%)	8.67 ± 0.59	0.0098
CPK	644 (97.28%)	219.57 ± 595.46	380 (91.13%)	232.15 ± 451.02	1
Creatinine	662 (100.0%)	1.18 ± 0.98	417 (100.0%)	1.2 ± 1.36	1
Direct bilirubin	359 (54.23%)	0.28 ± 0.54	385 (92.33%)	0.26 ± 0.34	1
D-dimer	627 (94.71%)	3.22 ± 4.81	371 (88.97%)	2.19 ± 3.88	0.0244
Ferritin	471 (71.15%)	641.34 ± 1094.3	348 (83.45%)	799.82 ± 1280.3	1
Fibrinogen	577 (87.16%)	500.56 ± 174.18	320 (76.74%)	524.01 ± 145.45	1

Glucose	662 (100.0%)	135.34 ± 59.23	417 (100.0%)	128.16 ± 66.65	1
HCO3	597 (90.18%)	24.62 ± 3.78	341 (81.77%)	24.35 ± 4.33	1
HGB	662 (100.0%)	12.67 ± 2.14	417 (100.0%)	13.2 ± 1.94	0.0022
INR	656 (99.09%)	1.13 ± 0.21	416 (99.76%)	1.08 ± 0.21	0.0078
Lactate	600 (90.63%)	2.07 ± 1.07	178 (42.69%)	2.07 ± 1.33	1
LDH	656 (99.55%)	353.77 ± 215.99	367 (88.01%)	565.31 ± 269.78	<0.0001
Lymphocytes (#)	660 (99.7%)	1.18 ± 0.83	417 (100.0%)	1.18 ± 0.95	1
Lymphocytes (%)	662 (100.0%)	18.3 ± 11.77	417 (100.0%)	17.6 ± 12.02	1
Neutrophils (#)	662 (100.0%)	5.59 ± 3.86	417 (100.0%)	6.09 ± 4.27	1
Neutrophils (%)	662 (100.0%)	71.7 ± 14.19	417 (100.0%)	73.12 ± 13.49	1
NRBC	98 (14.8%)	1.24 ± 0.97	416 (99.76%)	0.2 ± 0.38	<0.0001
Osmolality (urine)	25 (3.78%)	368.0 ± 158.68	47 (11.27%)	451.17 ± 186.09	1
PO2 (AR)	56 (8.46%)	74.69 ± 27.7	341 (81.77%)	40.04 ± 36.73	<0.0001
PO2 (V)	596 (90.03%)	36.44 ± 27.54	81 (19.42%)	34.89 ± 35.42	1
PCO2 (AR)	56 (8.46%)	49.83 ± 11.69	342 (82.01%)	42.15 ± 9.08	<0.0001
PCO2 (V)	598 (90.33%)	43.12 ± 8.95	76 (18.23%)	42.67 ± 11.0	1
PH	541 (81.72%)	7.37 ± 0.08	341 (81.77%)	7.38 ± 0.07	1
Platelet	662 (100.0%)	208.46 ± 99.93	417 (100.0%)	202.05 ± 82.31	1
Potassium	661 (99.85%)	4.16 ± 0.61	417 (100.0%)	4.04 ± 0.59	0.0758
PTT	649 (98.04%)	30.88 ± 9.79	416 (99.76%)	32.32 ± 6.42	0.4093
RBC	662 (100.0%)	4.48 ± 0.75	417 (100.0%)	4.48 ± 0.69	1
RDW	662 (100.0%)	14.83 ± 2.23	417 (100.0%)	14.43 ± 1.68	0.0874
Sodium	662 (100.0%)	136.05 ± 5.69	417 (100.0%)	136.0 ± 5.74	1
Saturation O2 (AR)	592 (89.43%)	57.34 ± 24.21	247 (59.23%)	64.32 ± 26.61	0.0119
Total bilirubin	661 (99.85%)	0.67 ± 0.68	412 (98.8%)	0.6 ± 0.48	1
Triglycerides	367 (55.44%)	162.83 ± 118.33	309 (74.1%)	135.95 ± 67.14	0.0217
Troponin	575 (86.86%)	98.5 ± 950.41	412 (98.8%)	51.92 ± 319.07	1
VB12	312 (47.13%)	610.9 ± 421.27	292 (70.02%)	852.07 ± 511.04	<0.0001
WBC	662 (100.0%)	7.55 ± 4.48	417 (100.0%)	8.85 ± 14.28	1
Temperature	662 (100.0%)	37.016 ± 1.91	417 (100.0%)	37.63 ± 0.92	<0.0001
Pulse	662 (100.0%)	86.23 ± 16.66	417 (100.0%)	87.59 ± 17.23	1
Respiratory Rate	513 (77.49%)	19.84 ± 9.0	113 (27.1%)	20.98 ± 9.74	1
SBP	662 (100.0%)	131.7 ± 24.65	417 (100.0%)	136.94 ± 23.69	0.0295
DBP	662 (100.0%)	75.53 ± 13.23	417 (100.0%)	75.96 ± 15.34	1
Saturation	110 (16.62%)	94.7 ± 5.92	415 (99.52%)	92.75 ± 7.31	0.5151

Supplementary Table 3: Minimum and maximum accepted values of the dynamic features. Feature engineering was applied for the bolded features. 'AR' and 'V' refer to arterial and venous blood, respectively.

Feature	Min	Max	Units	Feature	Min	Max	Units
HBA1C (#)	0	240	mmol/mol	**Lymphocytes (%)**	0.2	100	%
HBA1C (%)	0	24	%	**Neutrophils (#)**	0.1	60	10e3/µL
Albumin	0	100	g/L	**Neutrophils (%)**	0.2	100	%
ALT	0	20000	U/L	NRBC	0	100	%
AST	0	20000	U/L	Osmolality (urine)	50	2000	mosmo/kg
Indirect bilirubin	0	20	mg/dL	**PO2 (AR)**	0	1000	mmHg
Direct bilirubin	0	20	mg/dL	PO2 (V)	0	1000	mmHg
BNP	0	10000	PG/ML	PCO2 (AR)	0	150	mmHg
Respiratory rate	1	100	BPM	PCO2 (V)	0	150	mmHg
BUN	2	200	mg/dL	PH	6.6	7.8	
Calcium	0	20	mg/dL	**Platelet**	0	1000	10e3/µL
CKMB	0	10000	U/L	Potassium	1	10	mmol/L
CPK	0	10000	U/L	**PTT**	5	200	Sec
CRP	0	1000	mg/L	Pulse	10	300	BPM
Creatinine	0	20	mg/dL	RBC	1	8	10e6/µL
DBP	20	240	mmHG	RDW	5	40	%
D-dimer	0	50	FEU mg/L	SBP	40	250	mmHG
Ferritin	0	20000	ng/ml	Sodium	110	200	mmol/L
Fibrinogen	0	1500	mg/dL	Saturation O2 (AR)	5	100	%
Glucose	0	2000	mg/dL	**Saturation**	5	100	%
HCO3	0	100	mmol/L	Total bilirubin	0	20	mg/dL
HGB	2	25	g/dL	**Temperature**	20	43	C°
INR	0.5	5		Triglycerides	10	2000	mg/dL
Lactate	0.2	15	mmol/L	**Troponin**	1	40000	ng/L
LDH	0	50000	U/L	Vitamin B12	100	2500	pg/ml
Lymphocytes (#)	0	20	10e3/µL	**WBC**	0.2	100	10e3/µL

Supplementary Table 4: Hyperparameters used in the models. Hyperparameter search grid and fixed hyperparameters used for the predictive models. The hyperparameter combinations were evaluated on each fold in the cross-validation and the average performance was computed. The optimal values used for the final model (CatBoost) appear in bold type. 'poly': polynomial kernel function; 'rbf': Radial basis function.

Model	Hyperparameter	Grid / Fixed value
CatBoost	Maximum number of trees	1,000
	Maximum depth	[6, **8**, 10]
	Learning rate	[0.001, 0.01, **0.03**, 0.1, 0.3]
	L2 Regularization coefficient	[1, 3, **5**]
XGBoost	Number of trees	100
	Maximum depth	[6, 8, 10]
	Learning rate	[0.001, 0.01, 0.03, 0.1, 0.3]
	colsample_bytree	1
GBT	Number of trees	100
	Maximum depth	[6, 8, 10]
	Learning rate	[0.001, 0.01, 0.03, 0.1, 0.3]
Random Forest	Number of trees	100
	Maximum depth	[6, 8, 10]
Logistic Regression	Regularization	[L1, L2]
Linear Regression	Regularization	[L1, L2]
SVM	Kernel	['linear', 'poly', 'rbf', 'sigmoid']
NB	N/A	N/A

Supplementary Table 5: Top 100 features in importance as calculated by XGBoost. SD: standard deviation; /: ratio between two features. 24h,72h: time windows in which the statistic was computed. If the time window is not mentioned, the statistics is calculated on the entire hospitalization period up to the prediction time.

Age	Fibrinogen delta mean	Neutrophils (#) min
BMI	Fibrinogen delta mean 24h	Neutrophils (#) min 72h
Lactate	Fibrinogen max 72h	Neutrophils (#) / Glucose
Neutrophils (%)	Fibrinogen mean	Neutrophils (#) / Platelet
Sodium	Glucose mean	Neutrophils (#) trend
Saturation O2 - arterial blood	Glucose min 72h	Neutrophils (%) max
Albumin mean 24h	Glucose / Troponin	Neutrophils (%) max 24h
Albumin min 24h	Glucose SD	Neutrophils (%) max 72h
Albumin min 72h	LDH max	Neutrophils (%) min
Albumin / PTT	LDH max 72h	Neutrophils (%) min 72h
Albumin SD	LDH mean	Neutrophils (%) / ALT
Albumin SD 72h	LDH mean 72h	Neutrophils (%) / AST
ALT / Fibrinogen	LDH min 72h	Neutrophils (%) / D-dimer
AST min 72h	LDH / Albumin	Platelet delta
AST / Platelet	LDH / ALT	Platelet SD 24h
AST SD 72h	LDH / Platelet	Platelet SD 72h
BUN lr slope	LDH SD 72h	PTT lr slope
BUN delta mean 72h	Lymphocytes (#) min	PTT max 24h
BUN min	Lymphocytes (#) / D-dimer	PTT mean 72h
BUN / ALT	Lymphocytes (#) / Ferritin	PTT min 24h
BUN / Ferritin	Lymphocytes (#) / PTT	Temperature max
BUN / Troponin	Lymphocytes (#) SD 72h	Temperature mean 72h
BUN SD	Lymphocytes (%) max	Temperature min
D-dimer max	Lymphocytes (%) max 24h	Temperature min 24h
D-dimer max 72h	Lymphocytes (%) mean 72h	Temperature min 72h
D-dimer min	Lymphocytes (%) / AST	Temperature SD
D-dimer min 72h	Lymphocytes (%) / D-dimer	Temperature SD 72h
D-dimer / Albumin	Lymphocytes (%) / Fibrinogen	Troponin delta mean
D-dimer / AST	Lymphocytes (%) / PTT	Troponin SD 72h
D-dimer / Ferritin	Lymphocytes (%) SD	WBC min 24h
D-dimer / Fibrinogen	Lymphocytes (%) SD 24h	WBC / D-dimer
D-dimer / Platelet	Neutrophils (#) max	WBC SD
D-dimer SD	Neutrophils (#) max 72h	
Ferritin / Troponin	Neutrophils (#) mean	

תקציר

מגיפת הקורונה מתפשטת ברחבי העולם מאז דצמבר 2019, ומהווה איום בהול על בריאות הציבור. בשל ההבנה המוגבלת של התקדמות המחלה ושל גורמי הסיכון שלה, זהו אתגר קליני לחזות אילו חולים מאושפזים ידרדרו. יתרה מכך, מחקרים קודמים העלו כי נקיטת אמצעים מוקדמים לטיפול בחולים בסיכון להידרדרות עשויה למנוע או להפחית את החמרת המצב ואת הצורך בהנשמה מכנית.

בעבודה זו פיתחנו מודל חיזוי לזיהוי מוקדם של חולים בסיכון להידרדרות קלינית על ידי ניתוח רטרוספקטיבי של רשומות רפואיות דיגיטליות של חולי COVID-19 שהיו מאושפזים בשני המרכזים הרפואיים הגדולים ביותר בישראל. המודל שלנו מתבסס על שיטות למידת מכונה תוך שימוש במאפיינים קליניים שגרתיים כמו סימנים חיוניים, בדיקות מעבדה, נתוני דמוגרפיה ומחלות רקע. הידרדרות הוגדרה כציון גבוה של NEWS2 המותאם ל-COVID-19.

בניבוי הידרדרות עבור 7-30 השעות הבאות, המודל השיג שטח מתחת לעקומת ROC של 0.84 ושטח מתחת לעקומת precision-recall של 0.74. בפרט, המודל השיג רגישות של 44% עבור PPV של 87%. באימות חיצוני (external validation) על נתונים מבית חולים אחר, המודל השיג ערכים של 0.76 ו-0.7, בהתאמה.

אוניברסיטת תל אביב

הפקולטה למדעים מדויקים ע"ש ריימונד ובברלי סאקלר

בית הספר למדעי המחשב ע"י בלווטניק

מודל למידת מכונה לניבוי מוקדם של הידרדרות בחולי קורונה מאושפזים

חיבור זה הוגש כעבודת גמר לתואר 'מוסמך אוניברסיטה' באוניברסיטת תל אביב

בבית הספר למדעי המחשב

על ידי

עומר נוי

בהנחיית

פרופ' רון שמיר

אפריל 2022