

1 A dynamic ensemble model for short-term 2 forecasting in pandemic situations

3

4 Authors

5 Jonas Botz*^a, Diego Valderrama^a, Jannis Guski^a, Holger Fröhlich*^{a,b}

6 ^aAff1

7 Department of Bioinformatics, Fraunhofer Institute for Algorithms and Scientific Computing
8 (SCAI), Schloss Birlinghoven, 53757, Sankt Augustin, Germany.

9

10 ^bAff2

11 Bonn-Aachen International Center for IT, University of Bonn, Friedrich Hirzebruch-Allee 6,
12 53115, Bonn, Germany.

13

14

15

16 * Corresponding authors

17 Email: jonas.botz@scai.fraunhofer.de

18 Email: holger.froehlich@scai.fraunhofer.de

19

20 **Running title:** Dynamic ensemble model for pandemics

21 **Authors report no conflicts of interest.**

22 **Keywords and abbreviations:** Pandemic, COVID-19, Machine Learning, Artificial Intelligence,
23 Ensemble Model, Time Series Forecasting

24 **Abstract**

25 During the COVID-19 pandemic, many hospitals reached their capacity limits and could no longer
26 guarantee treatment of all patients. At the same time, governments endeavored to take sensible measures to
27 stop the spread of the virus while at the same time trying to keep the economy afloat. Many models
28 extrapolating confirmed cases and hospitalization rate over short periods of time have been proposed,
29 including several ones coming from the field of machine learning. However, the highly dynamic nature of
30 the pandemic with rapidly introduced interventions and new circulating variants imposed non-trivial
31 challenges for the generalizability of such models.

32
33 In the context of this paper, we propose the use of ensemble models, which are allowed to change in their
34 composition or weighting of base models over time and can thus adapt to highly dynamic pandemic or
35 epidemic situations. In that regard, we also explored the use of secondary metadata - Google searches - to
36 inform the ensemble model. We tested our approach using surveillance data from COVID-19, Influenza,
37 and hospital syndromic surveillance of severe acute respiratory infections (SARI). In general, we found
38 ensembles to be more robust than the individual models. Altogether we see our work as a contribution to
39 enhance the preparedness for future pandemic situations.

40 **1. Introduction**

41 In late 2019 a novel coronavirus SARS-CoV-2 emerged [1]. This not only gave rise to the COVID-19
42 pandemic but also affected every aspect of human life, from an economic downturn, and disruption in
43 education and social interactions to severe health implications including millions of deaths [2–4]. Early on,
44 governments struggled to find a balance between containing the spread of the virus and maintaining as
45 much economy, social interactions, and educational services as possible. Important indicators for decision-
46 making were the number of confirmed cases and the hospitalization rate. During that time many models
47 were developed for short-term forecasting of the number of incident cases and hospitalizations, respectively
48 [5], modeling strategies in this field include mechanistic, machine learning, and hybrid modeling strategies
49 [5]. All these models learn patterns from historical data to make forecasts, i.e. there is the implicit
50 assumption of a stationary dynamical process. However, the highly dynamic nature of the pandemic with
51 the rapid introduction of non-pharmaceutical interventions, new vaccines, and new circulating virus
52 variants contradicted this assumption and thus imposed non-trivial challenges for the generalizability of all
53 forecasting models over longer periods of time, regardless of the chosen modeling strategy.

54 Since each modeling technique unavoidably comes along with its own assumptions and limitations,
55 ensemble models have been proposed for forecasting the spread of infectious diseases like Influenza [6–8]
56 or Ebola [9] and later for COVID-19 [10,11]. In principle, ensemble models can be understood as a
57 collection of rather simplistic base models, which all produce an output based on each model's assumption
58 plus an algorithm or meta-model that combines them into one ensemble output. The advantage of such an
59 ensemble approach is that the bias of the individual models is reduced, making the final output more robust
60 [12]. In the literature, such ensemble methods often use the mean (e.g., [11]) or median (e.g., [10]) of the
61 base model outputs. However, pandemics like COVID-19 are dynamic, there are times when the number
62 of cases barely changes, there is exponential growth and decay, and there are turning points of waves, which
63 can all depend on external factors like interventions [13,14], people's behavior [15], seasonality [16,17],
64 or variants of concern [18,19].

65 To capture these dynamics and to be better prepared for future pandemics we here propose an ensemble
66 modeling approach that is dynamically adjusted, to either select the right model at the right time or to weigh
67 the models' predictions according to the current situation by using a meta-model. As base models, we
68 implemented a linear regression, ARIMA [20], XGBoost [21], Random Forest [22], and an LSTM [23]
69 model. We then evaluated the performance of each base model and compared this to baseline ensemble
70 methods. In the next step, we implemented a multi-layer perceptron (MLP) with softmax heads as a meta-
71 model. The base models' forecasts and performances were used as input for the meta-model which was
72 trained in either one of two ways: 1. select one of the models (selection), 2. combine the model's predictions
73 into one prediction (stacking). In addition, we tested whether the inclusion of metadata coming from Google
74 Trends could inform the meta-model to make better decisions.

75 2. Materials and Methods

76 77 2.1 Surveillance Data

78
79 We incorporate six different datasets related to COVID-19: the daily number of incident cases, hospital
80 admissions (hospitalization), and deaths for Germany and France, respectively. Additionally, we evaluated
81 the models on weekly Influenza cases and weekly hospital admissions related to severe acute respiratory
82 infections (SARI) in Germany. While the weekly data is only available on a country level, the daily data is
83 on a regional level (16 Bundesländer in Germany and 13 Régions in France, excluding overseas regions).
84 Moreover, we also included the country-level data for the daily data. While in the SARI and Influenza
85 datasets, the hospitalized and incident cases are provided, normalized to 100 thousand people (incidence),
86 in the other datasets we worked with absolute numbers. An overview of the used surveillance data can be
87 found in Table 1. The German surveillance data were received from the Robert Koch Institute (RKI)
88 (<https://github.com/robert-koch-institut>) and the French surveillance data from Santé Publique France
89 (SPF) (<https://www.data.gouv.fr/fr/organizations/sante-publique-france>). For all models, the time series
90 were log-transformed, because the raw data is locally expected to demonstrate exponential growth behavior.
91 The daily data was smoothed using a centered moving average over seven days.

92

Name	Source	Period	Fitting Windows	Time Resolution	Spatial Resolution (# regions)
COVID Cases, Hosp., Deaths DE	RKI	2020-2023	140	daily	Regional (16+1)
COVID Cases, Hosp., Deaths FR	SPF	2020-2023	140	daily	regional (13+1)
Influenza Cases DE	RKI	2020-2024	30	weekly	country (1)
SARI Hosp DE	RKI	2014-2024	80	weekly	country (1)

93 **Table 1. Surveillance data.**

94 2.2 Metadata

95
96 As metadata, we incorporated data from Google Trends following Wang et al. [24]. First, we identified the
97 20 top symptoms of COVID-19 which were used as search terms in Google Trends. By accessing their API
98 (<https://github.com/googleapis/google-api-python-client>) we extracted the normalized daily number of
99 counts each term was searched for. For smoothing we applied a centered moving average over seven days.

100

101 2.3 Base Models

102

103 In the following, we introduce the used base models and explain why they are suitable for time series
104 forecasting. The exact training and tuning procedure is explained in section 2.5.

105

106 **Linear Regression**

107

108 Assuming that a pandemic follows exponential-like behavior - exponential growth and decay in waves, log-
109 transforming the data will locally yield linear slopes. Therefore, linear regression can be used to fit linear
110 models to the log-transformed data. Using the regression parameters the fit can then be extrapolated to
111 estimate short-term forecasts. We used the scikit-learn library (version 1.0.2) “linear-model”.

112

113 **ARIMA**

114

115 Autoregressive integrated moving average (ARIMA) models use the statistical characteristics of stationary
116 data. They are popular for time-series forecasting and have previously been applied to modeling of COVID-
117 19 surveillance data [25–28]. A stationary series has no trends and consistently varies around its mean.
118 That means short-term random time patterns can be extracted accordingly and used for forecasting. Here
119 we employed a non-seasonal ARIMA model fitted to short-term periods which are not expected to show
120 seasonal effects. This also applies to the Influenza and SARI data. Using seasonal ARIMA would only
121 become effective when including at least two seasons. In this case, the ARIMA models depended on three
122 parameters:

123

- 124 • p the number of autoregressive terms
- 125 • d the degree of differencing the data to make it stationary
- 126 • q the number of lagged forecast errors

127

128 With these parameters the general ARIMA forecasting equation is defined as:

129

$$130 \hat{y}_t = \mu + \varphi_1 y_{t-1} + \dots + \varphi_p y_{t-p} - \theta_1 e_{t-1} - \dots - \theta_q e_{t-q} \quad (1)$$

131

132 Here \hat{y}_t corresponds to the forecast which is computed as the deviation of the mean μ of a stationary time
133 series with φ , the slope parameters for each of the p previous values y , and q moving average parameters θ
134 with autocorrelation errors e . This means the model learns to predict future steps based on the mean of a
135 stationary time series with adjusted autocorrelation errors and a lagged period [20]. To ensure stationarity
136 we employed the differencing technique. Differencing refers to the process of computing the differences

137 between consecutive values in a time series. Doing so transforms the time series to the fluctuations of
138 consecutive values, which in first or second order often leads to stationarity [29]. To find the best
139 parameters (p, d, q) we implemented the auto-ARIMA functionality which is part of the pmdarima library
140 (version 2.0.3) [30]. This essentially corresponds to a hyperparameter tuning.

141

142 **Random Forest and XGBoost**

143

144 Both Random Forest and eXtreme Gradient Boosting (XGBoost) are based on decision trees. However,
145 they differ to a great extent in their training algorithm. Random Forest builds an unweighted ensemble of
146 decision trees, which are - by applying bagging - trained in parallel on different subsets of the data and then
147 averaged [22]. In contrast, XGBoost builds its decision trees one after the other and corrects the residual
148 errors made by the previously trained weighted decision tree ensemble using gradient descent [21]. Both
149 models are commonly applied to tabular data but have also been shown to be successful in time series
150 forecasting [31,32], also for COVID-19 [33–36]. Since they are based on decision trees, they can only
151 extrapolate based on previously seen training data. If the models are tasked to predict values outside of the
152 training data, they will predict an average of this. Therefore, we first log-transformed the data and then
153 applied the previously explained differencing technique [29] to ensure stationarity. We tested for
154 stationarity by applying the augmented dickey-fuller (ADF) test [37]. Using stationary data does not only
155 mean that the extrapolation problem is reduced, but also that it is possible to apply k -fold cross-validation
156 for hyperparameter tuning since stationarity breaks the time dependence [38]. For Random Forest we used
157 the scikit-learn library (version 1.0.2) “ensemble” and for XGBoost the xgboost library (version 1.7.3).

158

159 **LSTM**

160

161 Recurrent Neural Networks (RNNs) are commonly used for sequencing data. Their advantage compared to
162 standard neural networks is their internal memory, i.e., their ability to remember and learn the influence of
163 previous steps on current steps. Opposed to standard RNNs, a Long Short Term Memory (LSTM) can learn
164 longer-range time patterns of time series without suffering from the vanishing gradient problem [23].
165 LSTMs have also been applied for time series forecasting in COVID-19 [39,40] Here we implemented an
166 LSTM model in which the last hidden state - the state that contains the latent information about the time
167 series - is decoded in a fully connected layer with output dimension according to the prediction window (14
168 days / 2 weeks). The LSTM model and fully connected layer were implemented using pytorch (version
169 1.11.0).

170

171 **2.4 Sliding Window Approach for Model Training, Tuning, and** 172 **Evaluation**

173

174 We split the time series into N (140, 30, 80) training and testing windows. For this, we followed a sliding
175 window approach (see Figure 1) with a training window size of 70 days for the data with daily resolution
176 and 52 weeks for the data with weekly resolution, respectively. A testing window size of 14 days (daily
177 data) / 2 weeks (weekly data), and a step size of 7 days (daily data) / 1 week (weekly data) were used. The
178 objective was to forecast the value of the time series 14 days ahead of time, counted from the end of the

179 training window. ARIMA, Random Forest, as well as XGBoost, were trained on the whole training set, and
180 the log-linear regression was fitted on the last seven days (daily data) / five weeks (weekly data) of the
181 training data. These models were applied separately for each region. For the LSTM model, however, we
182 needed more data. Therefore, we applied another sliding window approach by creating fitting windows of
183 size 7 days (daily data) / 5 weeks (weekly data) and evaluation windows of size 14 days (daily data) / 2
184 weeks (weekly data), with a stride of 1 day (daily data) / 1 week (weekly data). We decided to not train one
185 LSTM model per region but to shuffle the regional windows, to increase the amount of training data.
186 Therefore, the LSTM models' training objective was to predict 14 days (daily data) / 2 weeks (weekly data)
187 ahead based on 7 days (daily data) / 5 weeks (weekly data) of training data. We then tuned the
188 hyperparameters of Random Forest, XGBoost, and LSTM models for each training window and region if
189 applicable using Optuna (version 2.10.1). For more information about hyperparameter tuning, we refer to
190 the supplementary materials (see S1). Since we were using the auto-ARIMA functionality, the
191 hyperparameter tuning was done via a grid search, where the maximum parameter values for (p, q) were set
192 to (14,14). Random Forest and XGBoost were tuned with an inner 5-fold cross-validation, while for the
193 LSTM we split the training windows into 80% training and 20% validation sets. Here k -fold cross-
194 validation was not possible, because we had to consider the time dependencies of the data. Also, we decided
195 not to follow the classical time series split for time series cross-validation due to the increased run time and
196 insufficient data. For more details we refer to [38]. After hyperparameter tuning we retrained the models -
197 using the best hyperparameters for each fitting window - on the whole training data and progressively
198 predicted and evaluated on the test windows.
199

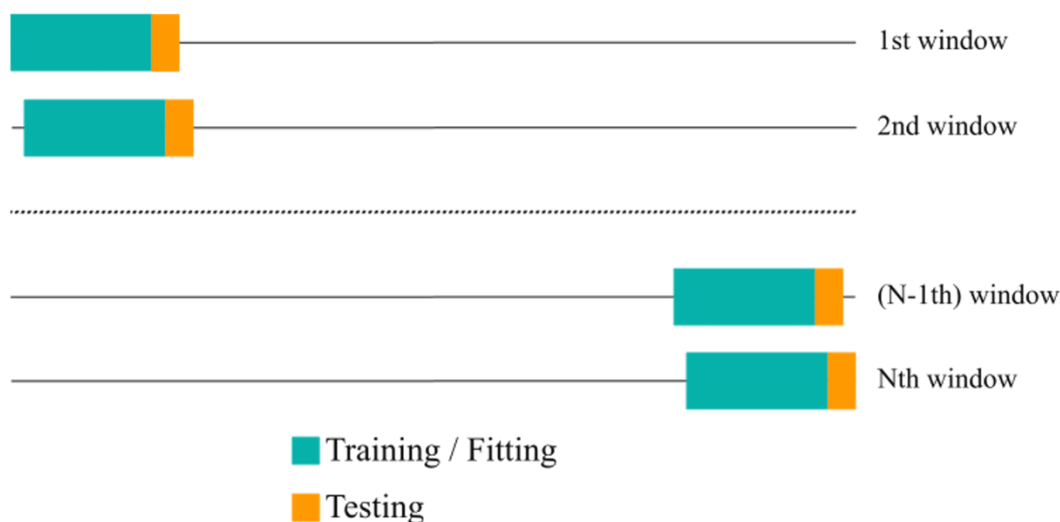


Fig 1: Sliding window approach. The time series was split into N training and testing windows. Models were tuned on the training windows using cross-validation (green), retrained with the best hyperparameters, and then forecasted 2 weeks ahead. Predictions were compared against real values observed in the test window (yellow).

209 2.5 Model Evaluation Metrics

210
211 To evaluate the performance of the base models and later the ensemble we used the mean absolute
212 percentage error (MAPE) as a metric:

$$213 \quad MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{Y_i - \hat{Y}_i}{Y_i} \right| * 100 \quad (2)$$

214
215 with Y as the real value, \hat{Y} as the predicted value, and n as the number of data points, in our case the
216 prediction window (14 days / 2 weeks). The MAPE represents the deviation of the prediction from the real
217 data in percent and is therefore a more tangible measure than the mean squared error. The MAPE alone
218 should not be used for determining the performance of a model, since it is scale-dependent [41]. However,
219 it is a good measure to quantitatively compare the performances of different models.

221 2.6 Baseline Ensemble Approaches

222
223 As a baseline, we implemented two basic ensemble algorithms, more specifically the mean and the median
224 of the model forecasts. Additionally, we built an ensemble algorithm that always chooses the model that
225 performed best in the previous testing period (Prev.-Best). This corresponds to a first step in accounting for
226 the dynamics of the pandemic and thus the dynamic performance of each base model.

227 2.7 Dynamic Model Selection and Stacking

228
229 We here propose two possible extensions of the baseline ensemble methods discussed before: i) dynamic
230 model selection and ii.) dynamic model stacking, an extension of a classical stacked regressor approach
231 [42]. In practice, we realize both approaches by training a meta-model, which we chose as a simple MLP
232 architecture with a tunable hidden layer size. The input for the meta-model constituted of the predicted
233 values as well as estimated prediction performances of all base models, by concatenating the MAPEs of the
234 previous testing period to the log-transformed forecasts of the current testing period. Therefore, the MLP
235 has five input vectors - one per base model. After each hidden layer a rectified linear unit (ReLU) activation
236 function is applied. The output layer is designed to hold one node per model including a softmax head at
237 the end. As mentioned above, there are two learning objectives:

- 238
239
- 240 1. Dynamic model selection: The meta-model is trained to always select the model with the highest
241 softmax output. This essentially corresponds to a classification task, where the model with the
242 highest probability score is selected.
 - 243 2. Dynamic model stacking: The meta-model is trained to multiply the base model forecasts by the
244 individual softmax inputs. These weighted outputs are then aggregated into one final ensemble
245 output. This essentially corresponds to a weighted mean, since the softmax outputs add up to 1.

246 Both learning objectives are trained with a weighted MSE (WMSE) loss function:

$$248 \quad WMSE = \frac{1}{n} \sum_{i=1}^n \frac{(Y_i - \hat{Y}_i)^2}{Y_i} \quad (3)$$

249

250 with Y as the real value, \hat{Y} as the predicted value, and n as the number of data points, in our case the
251 prediction window (14 days / 2 weeks). The WMSE has the advantage that it penalizes the relative deviation
252 rather than the absolute deviation. For example: say the real value was 100 and the predicted value was 101
253 the MSE would be 1. It would be the same for the real value being 1 and the predicted value being 2.
254 However, the relative deviation would be 1% vs. 100%. Weighting the MSE by the real values results in
255 normalizing this error to the real value scale. In this case, the WMSE would be 0.01 and 1, respectively -
256 the deviation of 100% is accordingly penalized much more than the deviation of 1%.

257

258 2.8 Inclusion of Metadata

259

260 Our above-described modeling approaches used only surveillance data, their forecasts, and estimated
261 prediction performances as input for the base models and meta-model, respectively. In our previous
262 publication, we showed that social media data is not only correlated with surveillance data but can also be
263 used to forecast up- and downtrends of pandemic waves [24]. Therefore, we wanted to test if the inclusion
264 of social media data or further metadata could improve the prediction performance of the meta-model. We
265 employed Google Trends data as described above and applied a sliding window approach, where we used
266 the past n ($n=2,3,4$) weeks (before the last date of our fitting period) as the training period for the metadata.
267 To extract time patterns, we used an LSTM model and concatenated the last hidden state to the input of the
268 meta-model, extending the input feature vector to include the forecasted value, the prediction performance
269 estimated from the previous testing period, and now the information coming from the metadata. The meta-
270 model was then trained in the same way as before, but now the weights of the LSTM were also updated
271 according to the weighted MSE loss between the output and the real values. Due to the high computational
272 burden and to be consistent with Wang et al. [24], we evaluated this approach on German surveillance data
273 only.

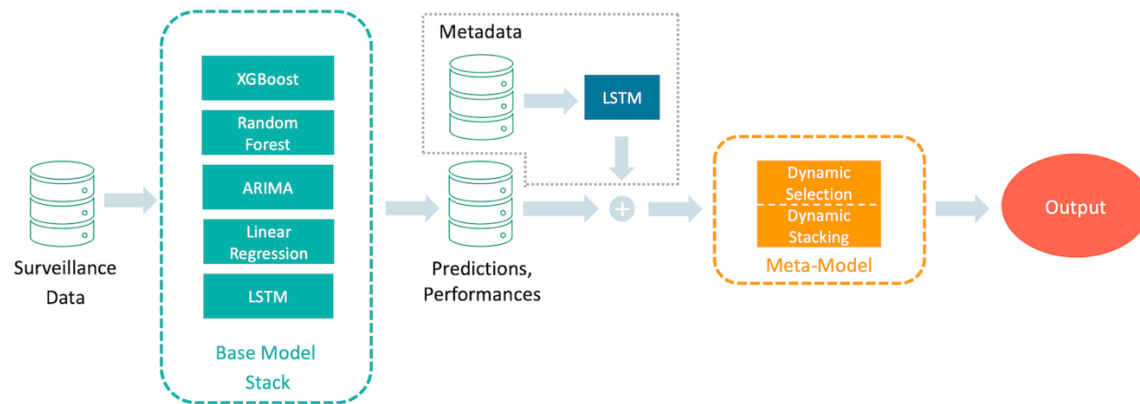
274

275 2.9 Overall Ensemble Model Pipeline

276

277 The final ensemble model pipeline can be seen in Figure 2. The surveillance data, which was previously
278 split into training and testing windows according to the sliding window approach explained above, is used
279 as input for the base models. The tuned base models are trained in parallel and create a rolling forecast
280 based on the testing windows. After each testing window, the baseline models (mean, median, Prev.-Best)
281 are created. The predictions are evaluated using the current test data and the MAPE as a metric. The base
282 models' forecasts together with their performance on the previous testing period are concatenated to form
283 the input vectors of the meta-model. If metadata is included the metadata is fed into an LSTM model, of
284 which the last hidden state - the latent representation of the metadata - is concatenated to the input vectors
285 of the meta-model. As described in Section 2.4 the overall data was split into 80% training and 20% test.
286 The training data was further split into 5 folds for an inner 5-fold cross-validation and hyperparameter
287 tuning (see S1). Finally, the models' performances over the test data were averaged and an output containing
288 these mean performances was returned. The code for the ensemble model can be accessed on Git Hub
289 (https://github.com/SCAI-BIO/Dynamic_Ensemble).

290



291
292 **Fig 2: Overall ensemble Model Pipeline.** The surveillance data is fed into the base models which produce forecasts. All forecasts
293 and their evaluation (plus the latent representation of the metadata) are used as input for the meta-model which outputs forecasts
294 either based on dynamic selection or dynamic stacking.
295

296 2.10 Model Ranking and Post-Hoc Analysis

297
298 To quantitatively compare the model performances across datasets we first ranked all models according to
299 a consensus ranking [43] - allowing for ties - based on Kemeny's axiomatic approach [44]. The algorithm
300 compares models pairwise and counts how often one model is ranked above the other. The total sum of
301 counts is then used to form the consensus ranking. This ranking alone, however, does not necessarily mean
302 that one model's performance is significantly different from another model. To test for statistical
303 significance across models we thus used a Kruskal-Wallis test [45]. To test which individual models
304 differed significantly from each other we then used a pairwise Wilcoxon test as post-hoc test [46]. All p-
305 values are adjusted for multiple testing based on the Holm-Bonferroni method [47]. Statistical tests were
306 implemented using R (version 4.3.0) and the libraries ConsRank (version 2.1.4) and stats (version 4.3.0).

307 3. Results

308 In the following, we display the results on the country level and the regional results aggregated (mean over
309 all regional results) at the country level. The complete set of results can be found in the supplementary
310 material. Model performances are displayed as the mean MAPE of all test windows in percent together with
311 its standard error in parentheses. We use the following abbreviations for the models: Linear Regression -
312 LR, XGBoost- XG, Random Forest - RF, and ensemble baseline by best model selection - Prev.-Best. Since
313 the Influenza dataset only contains 30 test windows (and just 6 test windows for the meta-model) the results
314 should be interpreted cautiously, as the small sample size leads to a reduced statistical meaningfulness.
315 Additionally, we provide the results from the consensus ranking and the Kruskal-Wallis as well as the
316 Wilcoxon test. For all results, the Kruskal-Wallis test returned a significant p-value of less than 5%.

317

318

319

3.1 Base Models versus Baseline Ensembles

320
321
322 First, we evaluated the base and baseline ensemble models by computing and testing a rolling forecast over
323 the full time series. This resulted in 140 test windows for the daily COVID-19 datasets, 30 test windows
324 for the weekly Influenza cases, and 80 test windows for the weekly SARI hospitalization. The results are
325 summarized in Table 2. For a better overview, we colored the three best models for each dataset / dataset
326 aggregation. First, looking at the base models' performances on the daily COVID-19 datasets, it can be
327 seen that mostly linear regression and ARIMA performed best and LSTM and XGBoost worst. On the
328 weekly dataset, Random Forest and XGBoost were able to perform similarly well as ARIMA. Here the
329 linear regression showed a reduced performance. Taking a look at the baseline ensemble methods shows
330 that mean and median baseline ensembles rarely performed as one of the three best models, but the Prev.-
331 Best method was able to outperform most of the base models in many instances; at least for the daily
332 COVID-19 datasets. Evaluated on the SARI hospitalization dataset, mean managed to be the best model.
333 Since Prev.-Best always selects the best model of the previous week, we integrated a counter to keep track
334 of this selection. The number of model selections per dataset can be seen in Figure 3. This agrees with the
335 results displayed in Table 2. The base models that performed best were also the ones being selected most
336 often. But, still, the other base models were selected a considerable amount of times. Finally, it can be
337 observed that the variance of the Prev.-Best method performances tended to be smaller than the variance of
338 the base models (perhaps excluding ARIMA) performances. Now looking at the consensus ranking, we can
339 see that ARIMA and Prev.-Best were both ranked first, followed by the mean and median baseline ensemble
340 methods. Table 3 shows the p-values computed by the pairwise Wilcoxon test. It can be seen that indeed
341 no significant difference between the Prev.-Best method and ARIMA could be found. However, they were
342 both found to be significantly different than all other methods.

Geography	LR	LSTM	XG	RF	ARIMA	Mean	Median	Prev.-Best
Daily COVID-19 Cases DE (N=140)								
DE	21.07 (2.27)	38.10 (3.76)	27.90 (1.71)	24.54 (1.36)	19.15 (1.14)	23.04 (1.30)	22.51 (1.15)	17.37 (1.04)
DE_reg	29.84 (2.61)	38.56 (7.44)	31.43 (1.40)	28.22 (1.08)	24.93 (1.08)	27.27 (1.88)	26.26 (1.05)	24.75 (1.05)
Daily COVID-19 Hospitalization DE (N=140)								
DE	12.75 (0.77)	28.95 (2.21)	20.51 (1.39)	19.17 (1.15)	8.88 (0.66)	14.97 (0.90)	15.40 (0.98)	11.94 (1.54)
DE_reg	27.28 (1.65)	27.06 (2.46)	28.22 (1.29)	25.79 (1.01)	17.64 (0.72)	21.39 (0.92)	21.30 (0.75)	20.19 (0.87)
Daily COVID-19 Deaths DE (N=140)								
DE	19.85 (1.27)	37.55 (2.04)	27.29 (1.72)	24.84 (1.39)	21.17 (1.02)	22.31 (1.15)	23.00 (1.19)	20.10 (1.28)
DE_reg	37.41 (0.95)	26.88 (2.20)	29.93 (1.36)	27.13 (0.96)	25.35 (0.87)	28.21 (0.73)	30.94 (0.92)	24.52 (1.18)
Daily COVID-19 Cases FR (N=140)								
FR	20.71 (1.33)	45.94 (4.46)	38.08 (3.44)	33.89 (2.71)	22.67 (1.89)	28.63 (2.28)	29.50 (2.42)	23.01 (2.08)

FR_reg	23.47 (1.27)	37.54 (4.32)	41.14 (3.97)	37.04 (3.05)	25.13 (2.05)	29.75 (2.21)	28.10 (1.84)	24.07 (1.43)
Daily COVID-19 Hospitalization FR (N=140)								
FR	16.45 (1.01)	33.61 (1.94)	25.95 (1.66)	24.26 (1.43)	16.84 (1.05)	20.43 (1.19)	21.02 (1.22)	17.57 (1.11)
FR_reg	26.90 (1.19)	29.39 (1.37)	28.87 (1.45)	26.52 (1.17)	23.08 (0.91)	23.83 (0.97)	24.70 (1.02)	23.90 (0.92)
Daily COVID-19 Deaths FR (N=140)								
FR	16.77 (1.35)	30.63 (1.87)	20.79 (1.27)	19.32 (1.19)	17.09 (1.12)	17.07 (1.03)	17.96 (1.10)	16.70 (1.42)
FR_reg	32.40 (1.17)	22.7 (1.23)	25.54 (0.87)	23.34 (0.67)	21.42 (0.62)	23.22 (0.70)	24.91 (0.77)	22.48 (0.79)
Weekly Influenza Cases DE (N=30)								
DE	47.57 (3.67)	23.53 (10.18)	14.85 (4.95)	15.44 (4.70)	11.28 (3.54)	38.67 (2.31)	44.34 (2.32)	28.46 (10.57)
Weekly SARI Hospitalization DE (N=80)								
DE	16.41 (1.43)	20.07 (1.96)	12.62 (1.27)	12.92 (1.28)	12.90 (1.34)	12.36 (1.14)	13.05 (1.16)	15.64 (1.60)
Consensus Ranking								
All	6	4	7	5	1	2	3	1

346
347
348
349
350
351
352
353

	Best Model		2nd Best Model		3rd Best Model
--	------------	--	----------------	--	----------------

Table 2: Base models versus baseline ensemble methods. The performances are given as the mean MAPE and its standard error in parentheses of the N test windows for each dataset / dataset aggregation. The best three models are colored according to the provided legend. DE (FR) stands for German (France) country level and DE_reg (FR_reg) for German (France) regional level aggregated to country level. The last two rows display the results from the consensus ranking over all datasets and models.

	ARIMA	LR	LSTM	Mean	Median	Prev. Best	RF
LR	<1E-16	-	-	-	-	-	-
LSTM	<1E-16	6.59E-02	-	-	-	-	-
Mean	<1E-16	7.55E-14	4.58E-06	-	-	-	-
Median	<1E-16	8.46E-03	8.89E-01	7.32E-06	-	-	-
Prev. Best	8.42E-01	<1E-16	<1E-16	<1E-16	<1E-16	-	-
RF	<1E-16	8.89E-01	2.01E-01	1.03E-11	9.41E-02	<1E-16	-
XG	<1E-16	5.90E-04	1.07E-09	<1E-16	1.03E-11	<1E-16	2.14E-05

354
355
356

Table 3: Base models versus baseline ensemble approaches: pairwise Wilcoxon Test (adjusted p-values).

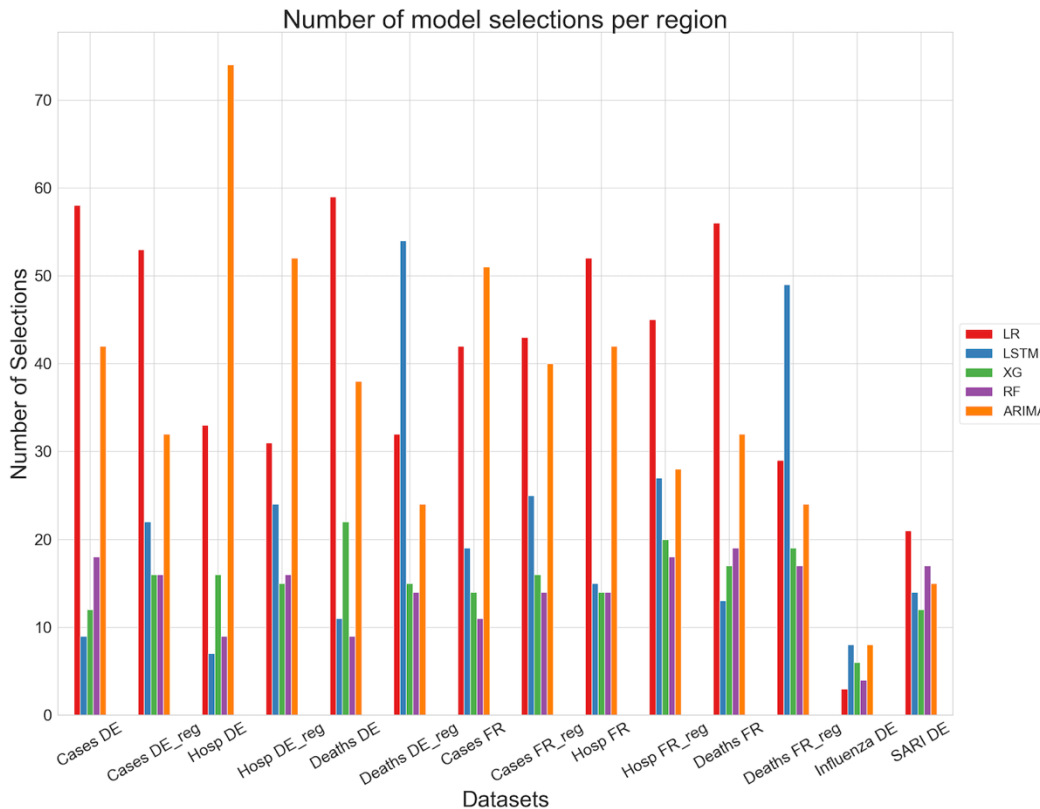


Fig 3: Number of model selections per region bar plot. DE (FR) stands for German (France) country level and DE_reg (FR_reg) for German (France) regional level aggregated to country level.

357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380

3.2 Baseline Ensembles versus Dynamic Model Stacking and Selection

Next, we evaluated our proposed Dynamic Model Stacking and Selection approaches against the previously tested Prev.-Best method. Since the meta-model was trained on 80% of the test windows, the number of test windows for the meta-model was reduced to 28 for the daily COVID-19 dataset, 6 for the weekly Influenza cases, and 16 for the weekly SARI hospitalization. According to the results presented in Table 3 Dynamic Selection was not able to outperform Prev.-Best. However, Dynamic Model Stacking outperformed Prev-Best and Dynamic Model Selection on the French and German COVID-19 deaths datasets and was the second-best model on the German COVID-19 hospitalization dataset. Moreover, it outperformed Dynamic Model Selection and Prev.-Best on the weekly datasets. Also, the variance of the Dynamic Model Stacking approach tended to be reduced compared to the other methods. The results of the consensus ranking are again in line with the findings above. The Dynamic Model Stacking method is ranked first being significantly better than Prev.-Best (see Table 5).

For the sake of completeness, we also list in Table S3.2.1 an additional comparison of Dynamic Model Stacking against all base models, which have been re-trained on the same dataset. Also, in this comparison, Dynamic Model Stacking could significantly outperform ARIMA as the best base model ($p = 5.74E-3$).

381

Geography	Prev.-Best	Dynamic Selection	Dynamic Stacking
Daily COVID-19 Cases DE (N=28)			
DE	17.07 (2.38)	28.94 (7.10)	24.58 (2.41)
DE_reg	22.50 (2.13)	30.44 (3.59)	26.29 (2.37)
Daily COVID-19 Hospitalization DE (N=28)			
DE	12.25 (2.01)	24.51 (3.38)	14.32 (2.01)
DE_reg	17.47 (1.36)	24.39 (2.16)	19.68 (2.05)
Daily COVID-19 Deaths DE (N=28)			
DE	24.87 (3.83)	27.27 (4.57)	21.99 (2.40)
DE_reg	31.71 (1.78)	34.59 (8.26)	22.86 (2.70)
Daily COVID-19 Cases FR (N=28)			
FR	21.53 (3.08)	31.36 (8.19)	20.48 (2.58)
FR_reg	22.80 (2.90)	28.79 (5.23)	22.09 (2.46)
Daily COVID-19 Hospitalization FR (N=28)			
FR	21.35 (2.76)	29.80 (5.04)	17.60 (2.62)
FR_reg	28.88 (2.09)	32.53 (4.03)	20.73 (1.82)
Daily COVID-19 Deaths FR (N=28)			
FR	19.32 (3.83)	21.06 (4.57)	14.07 (2.40)
FR_reg	26.76 (1.78)	24.41 (8.26)	17.99 (2.70)
Weekly Influenza Cases DE (N=6)			
DE	10.87 (7.76)	35.80 (8.17)	7.82 (3.82)
Weekly SARI Hospitalization DE (N=16)			
DE	15.15 (3.66)	17.2 (2.64)	13.19 (3.12)
Consensus Ranking			
All	2	3	1

382

383 **Table 4: Comparison of ensemble modeling approaches.** The performances are given as the mean MAPE and its standard error
 384 in parentheses of the N test windows for each dataset / dataset aggregation. DE (FR) stands for German (France) country level and
 385 DE_reg (FR_reg) for German (France) regional level aggregated to country level.

386

387

388

389

390

	Prev.-Best	Dynamic Selection	Dynamic Stacking
Dynamic Selection	3.24E-01	-	-
Dynamic Stacking	2.57E-07	2.17E-10	-

391
392
393

Table 5: Prev.-Best versus Dynamic Model Stacking and Selection: pairwise Wilcoxon Test (adjusted p-values).

394 3.3 Potential Benefits of Including Metadata

395 Finally, we wanted to assess whether the inclusion of metadata - here Google Trends symptom counts -
396 could further enhance our proposed Dynamic Model Stacking approach. We could not find an improvement
397 by including metadata. Both Dynamic Model Stacking with and without metadata were ranked on position
398 1 after consensus ranking and the Wilcoxon-test showed no significant differences between both
399 approaches.
400

401 4. Discussion

402 The COVID-19 pandemic highlighted the need for robust models that can accurately forecast the spread of
403 the pandemic and can adjust dynamically to external factors such as newly imposed non-pharmaceutical
404 interventions, new virus variants, vaccination, seasonal effects, and others. In this work, we initially tested
405 and compared different base machine learning models against basic ensemble methods (mean, median,
406 Prev.-Best), demonstrating no statistically significant benefit of these simple techniques compared to a
407 state-of-the art ARIMA time series forecasting model. Only Prev.-Best was found to perform en par with
408 ARIMA. Even though the other base models did not perform as well as ARIMA we still found them to be
409 frequently selected as significantly often as being the best model in the previous test window. Therefore,
410 we decided to keep these other base models in the model ensemble.

411 We then developed a meta-model that can either dynamically select one of the base model predictions or
412 add the weighted base model predictions into one forecast. Interestingly, only Dynamic Model Stacking
413 turned out to outperform Prev.-Best while at the same time showing a reduced variance in prediction
414 performance. The inclusion of Google Trends symptom counts as metadata could not further improve
415 Dynamic Model Stacking significantly and thus cannot be recommended.

416 Comparing the performances on the country and regional level we found Dynamic Model Stacking on the
417 country level to be superior. We assume that this comes from the data quality of the regional data. Since
418 we are working with surveillance data that needs to be registered at local health departments, it can happen
419 that mistakes are made on a regional level. These mistakes would have reduced effects when the regional
420 data is aggregated to the country-level data.

421 In general, we saw considerable differences between model performances on the daily and the weekly
422 datasets. Specifically, the SARI hospitalization data suggests that decision tree-based models - especially
423 Random Forest - and the mean and median baseline ensemble methods work well here. In this regard, we
424 should point out that in the weekly data, the task is just to forecast the next two data points (2 weeks) which

425 seems to be handled well by decision tree models. Linear regression struggles here because the model is fit
426 to the past 5 data points (i.e. weeks), hence resulting in over-smoothing.

427 We should mention the limitations of the non-COVID datasets, specifically limited sample size and, in case
428 of the Influenza, also seasonal fluctuations (see Figure S2). Moreover, the SARI dataset contains
429 hospitalization due to different pathogens. These limitations lead to non-trivial challenges for learning a
430 good model.

431
432 A comparison of our findings with those in other studies is challenging because different datasets (perhaps
433 even just one wave rather than a whole pandemic), different forecasting horizons, and different metrics
434 have been used. Paireau et al. [11] developed an ensemble model (mean) and forecasted among other
435 indicators the COVID-19 hospitalization in France. On country-level data, they documented a mean MAPE
436 of 20% and on aggregated regional level of 30% for a 14-day forecast horizon. Our best model - evaluated
437 on the French COVID-19 hospitalization dataset - achieved a mean MAPE of around 17% and around 20%
438 to 23% (for the meta-model test windows or all test windows) on country-level and regional aggregated
439 country-level data, respectively. Heredia Cacha et al. [10] forecasted COVID-19 cases in Spain using
440 different ensemble methods (mean, median, weighted average) and documented a mean MAPE of around
441 30% for a 14-day forecasting horizon. We achieved a MAPE of around 17% to 25% for forecasting the
442 number of COVID-19 cases in Germany and France. Stating that our models are better than the ones of
443 Paireau et al. and Heredia Cacha et al. would not be fair, though, since we are not using the exact same
444 data. However, this comparison confirms that our models are generally competitive with others reported in
445 the literature.

446 **5. Conclusion**

447 A major challenge for the modeling of pandemic situations, specifically COVID-19, is their highly dynamic
448 character. Rapid introduction of non-pharmaceutical interventions, newly emerging virus variants,
449 vaccinations, and seasonal effects strongly violate the typical assumption of stationarity in time series
450 modeling and forecasting and thus negatively affect the generalization ability of models. In this regard, we
451 here proposed a novel ensemble learning strategy, in which a meta-model learns to dynamically weigh and
452 integrate a set of base models based on currently observed data and past performance indicators. Based on
453 results from 8 datasets, our Dynamic Model Stacking approach was able to outperform state-of-the art time
454 series forecasting techniques, such as ARIMA, and other ensemble learning approaches. Furthermore, we
455 could show that our method could not be further improved by adding further metadata, such as Google
456 searches.

457 Of course, our proposed Dynamic Model Stacking approach is not without limitations. Most importantly,
458 machine learning methods need a sufficient amount of training data, i.e. retrospective pandemic data, which
459 are not always available at the beginning of a pandemic. A potential strategy in future pandemics might
460 thus be to start building a collection of comparable simple base models, specifically including ARIMA, and
461 then to train Dynamic Model Stacking once sufficient historical data is available.

462 While we only evaluated Dynamic Model Stacking on surveillance data of COVID-19, SARI, and
463 Influenza, our method is not limited per se to these data. Dynamic Model Stacking could potentially be
464 applied also to other areas, where non-stationary time series forecasting plays a role, e.g. traffic, energy
465 consumption, air flights, and others. Moreover, future work could apply Dynamic Model Stacking to age-
466 distributed data, especially to the vulnerable, mostly elderly population.

467 **Acknowledgments**

468 This work has been supported by the AIOLOS (Artificial Intelligence Tools for Outbreak Detection and
469 Response) project. The project was supported by the French State and the German Federal Ministry for
470 Economic Affairs and Climate Action (grant number 01MJ22005A) and the French Ministry of Economy
471 and Finance in the context of the France 2030 initiative and the Franco-German call on Artificial
472 Intelligence technologies for risk prevention, crisis management, and resilience.
473

474 **Author Contribution**

475 Conceptualization, methodology, supervision, project administration, and funding acquisition: HF. Data
476 curation, formal analysis, visualization, investigation, validation, and writing—original draft: JB, DV, JG,
477 HF Writing—review and editing: JB, DV, JG, and HF. All authors contributed to the article and approved
478 the submitted version.
479

480 **Supporting Informations**

481
482 **S1 Table. Hyperparameters for base models and meta-models.**
483 (PDF)

484
485 **S2 Figure. Comparison SARI and Influenza.** SARI and Influenza incidence from May 2022 to May
486 2023.
487 (PNG)

488
489 **S3 Appendix. Complete results**
490 (PDF)

491

492

493

494

495

496

497

498

499 **References**

- 500 1. Coronavirus disease (COVID-19) pandemic [Internet]. [cited 2024 Jan 5]. Available from:
501 <https://www.who.int/europe/emergencies/situations/covid-19>
- 502 2. COVID - Coronavirus Statistics - Worldometer [Internet]. [cited 2024 Jan 5]. Available from:
503 <https://www.worldometers.info/coronavirus/>
- 504 3. Mofijur M, Fattah IMR, Alam MA, Islam ABMS, Ong HC, Rahman SMA, et al. Impact of
505 COVID-19 on the social, economic, environmental and energy domains: Lessons learnt from
506 a global pandemic. *Sustain Prod Consum*. 2021 Apr 1;26:343–59.
- 507 4. Naseer S, Khalid S, Parveen S, Abbass K, Song H, Achim MV. COVID-19 outbreak: Impact
508 on global economy. *Front Public Health* [Internet]. 2023 [cited 2024 Jan 4];10. Available
509 from: <https://www.frontiersin.org/articles/10.3389/fpubh.2022.1009393>
- 510 5. Botz J, Wang D, Lambert N, Wagner N, Génin M, Thommes E, et al. Modeling approaches
511 for early warning and monitoring of pandemic situations as well as decision support. *Front*
512 *Public Health*. 2022;10:994949.
- 513 6. Reich NG, McGowan CJ, Yamana TK, Tushar A, Ray EL, Osthus D, et al. Accuracy of real-
514 time multi-model ensemble forecasts for seasonal Influenza in the U.S. *PLOS Comput Biol*.
515 2019 Nov 22;15(11):e1007486.
- 516 7. Cheng HY, Wu YC, Lin MH, Liu YL, Tsai YY, Wu JH, et al. Applying Machine Learning Models
517 with An Ensemble Approach for Accurate Real-Time Influenza Forecasting in Taiwan:
518 Development and Validation Study. *J Med Internet Res*. 2020 Aug 5;22:e15394.
- 519 8. Kumar R, Maheshwari S, Sharma A, Linda S, Kumar S, Chatterjee I. Ensemble learning-
520 based early detection of Influenza disease. *Multimed Tools Appl*. 2024 Jan 1;83(2):5723–
521 43.
- 522 9. Viboud C, Sun K, Gaffey R, Ajelli M, Fumanelli L, Merler S, et al. The RAPIDD ebola
523 forecasting challenge: Synthesis and lessons learnt. *Epidemics*. 2018 Mar 1;22:13–21.
- 524 10. Heredia Cacha I, Sáinz-Pardo Díaz J, Castrillo M, López García Á. Forecasting COVID-19
525 spreading through an ensemble of classical and machine learning models: Spain's case
526 study. *Sci Rep*. 2023 Apr 25;13(1):6750.
- 527 11. Paireau J, Andronico A, Hozé N, Layan M, Crépey P, Roumagnac A, et al. An ensemble
528 model based on early predictors to forecast COVID-19 health care demand in France. *Proc*
529 *Natl Acad Sci*. 2022 May 3;119(18):e2103302119.
- 530 12. Re M, Valentini G. Ensemble Methods. *Adv Mach Learn Data Min Astron*. 2012 Mar 1;563–
531 93.
- 532 13. Lison A, Banholzer N, Sharma M, Mindermann S, Unwin HJT, Mishra S, et al. Effectiveness
533 assessment of non-pharmaceutical interventions: lessons learned from the COVID-19
534 pandemic. *Lancet Public Health*. 2023 Apr 1;8(4):e311–7.

- 535 14. Ge Y, Zhang WB, Liu H, Ruktanonchai CW, Hu M, Wu X, et al. Impacts of worldwide individual
536 non-pharmaceutical interventions on COVID-19 transmission across waves and space. *Int*
537 *J Appl Earth Obs Geoinformation*. 2022 Feb;106:102649.
- 538 15. Lyu H, Imtiaz A, Zhao Y, Luo J. Human behavior in the time of COVID-19: Learning from big
539 data. *Front Big Data* [Internet]. 2023 [cited 2024 Feb 6];6. Available from:
540 <https://www.frontiersin.org/articles/10.3389/fdata.2023.1099182>
- 541 16. Wiemken TL, Khan F, Puzniak L, Yang W, Simmering J, Polgreen P, et al. Seasonal trends
542 in COVID-19 cases, hospitalizations, and mortality in the United States and Europe. *Sci Rep*.
543 2023 Mar 8;13:3886.
- 544 17. Liu X, Huang J, Li C, Zhao Y, Wang D, Huang Z, et al. The role of seasonality in the spread
545 of COVID-19 pandemic. *Environ Res*. 2021 Apr 1;195:110874.
- 546 18. Otto SP, Day T, Arino J, Colijn C, Dushoff J, Li M, et al. The origins and potential future of
547 SARS-CoV-2 variants of concern in the evolving COVID-19 pandemic. *Curr Biol*. 2021 Jul
548 26;31(14):R918–29.
- 549 19. Kim Y, Gaudreault NN, Meekins DA, Perera KD, Bold D, Trujillo JD, et al. Effects of Spike
550 Mutations in SARS-CoV-2 Variants of Concern on Human or Animal ACE2-Mediated Virus
551 Entry and Neutralization. *Microbiol Spectr*. 2022 Jun 29;10(3):e0178921.
- 552 20. Introduction to ARIMA models [Internet]. [cited 2024 Feb 6]. Available from:
553 <https://people.duke.edu/~rnau/411arim.htm#arima010>
- 554 21. Chen T, Guestrin C. XGBoost: A Scalable Tree Boosting System. In: Proceedings of the 22nd
555 ACM SIGKDD International Conference on Knowledge Discovery and Data Mining [Internet].
556 New York, NY, USA: Association for Computing Machinery; 2016 [cited 2024 Feb 6]. p. 785–
557 94. (KDD '16). Available from: <https://dl.acm.org/doi/10.1145/2939672.2939785>
- 558 22. Biau G, Scornet E. A random forest guided tour. *TEST*. 2016 Jun 1;25(2):197–227.
- 559 23. Hochreiter S, Schmidhuber J. Long Short-Term Memory. *Neural Comput*. 1997 Nov
560 15;9(8):1735–80.
- 561 24. Wang D, Lentzen M, Botz J, Valderrama D, Deplante L, Perrio J, et al. Development of an
562 early alert model for pandemic situations in Germany. *Sci Rep*. 2023 Nov 27;13(1):20780.
- 563 25. Chaurasia V, Pal S. COVID-19 Pandemic: ARIMA and Regression Model-Based Worldwide
564 Death Cases Predictions. *SN Comput Sci*. 2020;1(5):288.
- 565 26. Claris S, Peter N. ARIMA MODEL IN PREDICTING OF COVID-19 EPIDEMIC FOR THE
566 SOUTHERN AFRICA REGION. *Afr J Infect Dis*. 2022 Dec 22;17(1):1–9.
- 567 27. Satrio C, Darmawan W, Nadia B, Hanafiah N. Time series analysis and forecasting of
568 coronavirus disease in Indonesia using ARIMA model and PROPHET. *Procedia Comput*
569 *Sci*. 2021 Jan 1;179:524–32.

- 570 28. Somyanonthanakul R, Warin K, Amasiri W, Mairiang K, Mingmalairak C, Panichkitkosolkul W,
571 et al. Forecasting COVID-19 cases using time series modeling and association rule mining.
572 BMC Med Res Methodol. 2022 Nov 1;22(1):281.
- 573 29. Time Series Differencing: A Complete Guide | InfluxData [Internet]. [cited 2024 Feb 27].
574 Available from: [https://www.influxdata.com/blog/time-series-differencing-complete-guide-](https://www.influxdata.com/blog/time-series-differencing-complete-guide-influxdb/)
575 [influxdb/](https://www.influxdata.com/blog/time-series-differencing-complete-guide-influxdb/)
- 576 30. pmdarima: ARIMA estimators for Python — pmdarima 2.0.4 documentation [Internet]. [cited
577 2024 Feb 6]. Available from: <http://alkaline-ml.com/pmdarima/>
- 578 31. Masini RP, Medeiros MC, Mendes EF. Machine learning advances for time series forecasting.
579 J Econ Surv. 2023;37(1):76–111.
- 580 32. Lv CX, An SY, Qiao BJ, Wu W. Time series analysis of hemorrhagic fever with renal syndrome
581 in mainland China by using an XGBoost forecasting model. BMC Infect Dis. 2021 Aug
582 19;21(1):839.
- 583 33. Luo J, Zhang Z, Fu Y, Rao F. Time series prediction of COVID-19 transmission in America
584 using LSTM and XGBoost algorithms. Results Phys. 2021 Aug 1;27:104462.
- 585 34. Fang Z gang, Yang S qin, Lv C xia, An S yi, Wu W. Original research: Application of a data-
586 driven XGBoost model for the prediction of COVID-19 in the USA: a time-series study. BMJ
587 Open [Internet]. 2022 [cited 2024 Mar 4];12(7). Available from:
588 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9251895/>
- 589 35. Galasso J, Cao DM, Hochberg R. A random forest model for forecasting regional COVID-19
590 cases utilizing reproduction number estimates and demographic data. Chaos Solitons
591 Fractals. 2022 Mar;156:111779.
- 592 36. Özen F. Random forest regression for prediction of Covid-19 daily cases and deaths in
593 Turkey. Heliyon. 2024 Feb 29;10(4):e25746.
- 594 37. Dickey-Fuller Test - an overview | ScienceDirect Topics [Internet]. [cited 2024 Feb 6].
595 Available from: [https://www.sciencedirect.com/topics/economics-econometrics-and-](https://www.sciencedirect.com/topics/economics-econometrics-and-finance/dickey-fuller-test)
596 [finance/dickey-fuller-test](https://www.sciencedirect.com/topics/economics-econometrics-and-finance/dickey-fuller-test)
- 597 38. Bergmeir C, Benítez JM. On the use of cross-validation for time series predictor evaluation.
598 Inf Sci. 2012 May;191:192–213.
- 599 39. Aung NN, Pang J, Chua MCH, Tan HX. A novel bidirectional LSTM deep learning approach
600 for COVID-19 forecasting. Sci Rep. 2023 Oct 20;13(1):17953.
- 601 40. Chandra R, Jain A, Chauhan DS. Deep learning via LSTM models for COVID-19 infection
602 forecasting in India. PLOS ONE. 2022 Jan 28;17(1):e0262708.
- 603 41. Mean Absolute Percentage Error (MAPE): What You Need To Know [Internet]. Arize AI. [cited
604 2024 Feb 15]. Available from: [https://arize.com/blog-course/mean-absolute-percentage-](https://arize.com/blog-course/mean-absolute-percentage-error-mape-what-you-need-to-know/)
605 [error-mape-what-you-need-to-know/](https://arize.com/blog-course/mean-absolute-percentage-error-mape-what-you-need-to-know/)
- 606 42. Wolpert DH. Stacked generalization. Neural Netw. 1992 Jan 1;5(2):241–59.

- 607 43. A new rank correlation coefficient with application to the consensus ranking problem - Emond
608 - 2002 - Journal of Multi-Criteria Decision Analysis - Wiley Online Library [Internet]. [cited
609 2024 Feb 29]. Available from: <https://onlinelibrary.wiley.com/doi/abs/10.1002/mcda.313>
- 610 44. Kemeny JG, Laurie Snell J. Mathematical Models in the Social Sciences [Internet]. MIT Press.
611 [cited 2024 Feb 29]. Available from: [https://mitpress.mit.edu/9780262610308/mathematical-](https://mitpress.mit.edu/9780262610308/mathematical-models-in-the-social-sciences/)
612 [models-in-the-social-sciences/](https://mitpress.mit.edu/9780262610308/mathematical-models-in-the-social-sciences/)
- 613 45. Kruskal WH, Wallis WA. Use of Ranks in One-Criterion Variance Analysis. J Am Stat Assoc.
614 1952 Dec 1;47(260):583–621.
- 615 46. Wilcoxonon F. Individual Comparisons by Ranking Methods. Biom Bull. 1945;1(6):80–3.
- 616 47. Holm S. A Simple Sequentially Rejective Multiple Test Procedure. Scand J Stat.
617 1979;6(2):65–70.
- 618
619
620