Reservoir Computing for Short High-Dimensional Time Series: an Application to SARS-CoV-2 Hospitalization Forecast

Thomas Ferté, Dan Dutartre, Boris P. Hejblum, Romain Griffier, Vianney Jouhet, Rodolphe Thiébaut, Pierrick Legrand, Xavier Hinaut

Reservoir Computing for Short High-Dimensional Time Series: an Application to SARS-CoV-2 Hospitalization Forecast

Thomas Ferté [1 2 3] **Dan Dutartre** [3] **Boris Hejblum** [2 3] **Romain Griffier** [1 4] **Vianney Jouhet** [1 4]
Rodolphe Thiébaut [1 2 3] **Pierrick Legrand** [5 6] **Xavier Hinaut** [3 7 8]

Abstract

In this work, we aimed at forecasting the number of SARS-CoV-2 hospitalized patients at 14 days to help anticipate the bed requirements of a large scale hospital using public data and electronic health records data. Previous attempts led to mitigated performance in this high-dimension setting; we introduce a novel approach to time series forecasting by providing an alternative to conventional methods to deal with high number of potential features of interest (409 predictors). We integrate Reservoir Computing (RC) with feature selection using a genetic algorithm (GA) to gather optimal non-linear combinations of inputs to improve prediction in sample-efficient context. We illustrate that the RC-GA combination exhibits excellent performance in forecasting SARS-CoV-2 hospitalizations. This approach outperformed the use of RC alone and other conventional methods: LSTM, Transformers, Elastic-Net, XGBoost. Notably, this work marks the pioneering use of RC (along with GA) in the realm of short and high-dimensional time series, positioning it as a competitive and innovative approach in comparison to standard methods.

[1]Bordeaux Hospital University Center, Pôle de santé publique, Service d'information médicale, F-33000 Bordeaux, France [2]Inserm Bordeaux Population Health Research Center UMR 1219, Inria centre of Bordeaux University, team SISTM, F-33000 Bordeaux, France [3]Inria centre of Bordeaux University, F-33000 Bordeaux, France [4]Inserm Bordeaux Population Health Research Center UMR 1219, team AHeaD, F-33000 Bordeaux [5]ASTRAL, Centre Inria de l'université de Bordeaux [6]IMB, Institut de Mathématiques de Bordeaux, UMR CNRS 5251 [7]LaBRI, Univ. Bordeaux, Bordeaux INP, CNRS UMR 5800. [8]Univ. Bordeaux, CNRS, IMN, UMR 5293, Bordeaux, France. Correspondence to: Xavier Hinaut <xavier.hinaut@inria.fr>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

1. Introduction

Since late 2020, SARS-CoV-2 induced heavy stress on healthcare system leading to modification of organisation of care and to heavy measures such as lockdowns (WHO, 2020). Better anticipation of the need for care, especially hospitalization, might help decision makers. For that purpose, several short-term prediction models have been proposed relying on different approaches to forecast the SARS-CoV-2 pandemic including statistical approaches, compartmental models, machine learning and deep learning models (Cramer et al., 2022; Rahimi et al., 2021).

In this context, hospital datawarehouse provides up-to-date data from electronic health records (EHRs) which significantly improved forecasting performance (Ferté et al., 2022). However, numerous predictors can be extracted from EHRs data while the number of observations (i.e the number of days) is limited for emergent epidemics such as SARS-CoV-2 leading to a high-dimension problem for such shortage of data anteriority.

Most previous work focused on SARS-CoV-2 forecast using public data in a low dimension setting (i.e where the number of predictors is limited) (Cramer et al., 2022; Rahimi et al., 2021; Paireau et al., 2022; Pottier, 2021; Carvalho et al., 2021; Mohimont et al., 2021). In contrast, we previously integrated public data with EHRs data and found that elastic-net penalized linear regression was the best model in this high-dimension setting compared to random forest and Poisson regression (Ferté et al., 2022). However, graphical evaluation outlined difficulties to anticipate dynamic shifts and, given the heterogeneous evaluation period of the different work, the best approach to tackle the problem of forecasting the number of hospitalizations in a high-dimension setting remains unknown.

While elastic-net penalized linear regression was identified as the best predictor (Ferté et al., 2022), this model performs both feature selection and prediction linearly without temporal consideration. In this context, Reservoir Computing (RC) (Jaeger, 2001; Maass et al., 2002; Lukoševičius & Jaeger, 2009), which also has a regularized linear regression as output layer might outperform this predictor. Indeed,

RC forecasts rely on non-linear temporal computations performed inside a random recurrent network (i.e. a reservoir). However, RC is mostly used in a low-dimension forecast, even in the case of epidemic forecast (Ghosh et al., 2021; Ray et al., 2021; Liu et al., 2023).

To the best of our knowledge, the utilization of RC for forecasting with high-dimensional inputs is limited. Application of RC with substantial number of predictors was identified in the work by Arcomano et al. (2020), involving 528 input features for weather forecasting. However, it's noteworthy that this study employed training data consisting of over 10^5 observations. Another notable instance is the research conducted by Weddell et al. (2021), where 544 input features were utilized for microsleep forecasting based on more than 10^4 observations. Given that RC relies on the expansion of the feature space into a higher-dimensional reservoir, a persistent question remains regarding the viability of RC in high-dimensional scenarios, especially when the number of features approaches the number of neurons, and the quantity of observations is limited (i.e. small sample size). This issue gains significance in certain domains, such as epidemic forecasting, where numerous predictors may be derived from epidemic surveillance, public data, or EHRs, despite the availability of a restricted number of observations (Ferté et al., 2022).

Metaheuristic approaches have been suggested to address the challenges of hyperparameter optimization for RC and feature selection in machine learning (Agrawal et al., 2021; Bala et al., 2018). These optimization algorithms assess a specified objective function based on given optimization parameters, treating the task as a black box. In their review and comparison of different metaheuristic approaches, Ezugwu et al. (2020) found that best approaches included genetic algorithm (GA), particle swarm optimization and differential evolution.

We propose a new approach which uses RC in conjunction with an evolutionary strategy algorithm known as Genetic Algorithm (GA) Goldberg D.E (1989); Schwefel H.P (1990); Jong & Alan (1975); Holland J.H (1975): we will call *RC-GA* our proposed conjunction method. This algorithm was implemented in a high-dimensional context, utilizing GA for dimension reduction through feature selection. We evaluate the performance of RC-GA for predicting at t + 14 days the number of SARS-CoV-2 hospitalizations at a large scale hospital in France. Furthermore, we present a comparative analysis of RC to other machine learning methods.

2. Methods

2.1. Data

The objective was to forecast the number of SARS-CoV-2 hospitalized patients at 14 days (h_{t+14}) at Bordeaux University Hospital (BUH). To perform this task we used an ensemble of data from different sources: department-level COVID-19 data coming from Santé Publique France (Etalab, 2020), department-level weather data from the National Oceanic and Atmospheric Administration (NOAA) (Smith et al., 2011), and hospital-level COVID-19 data from BUH Electronic Health Record (EHR) from Ferté et al. (2023). The study utilized aggregated data spanning from May 16, 2020, to January 17, 2022. This resulted in a high-dimensional problem with a total of 409 predictors over 586 days, where each observation represents one day.

Primary outcome was the mean absolute error (MAE), secondary outcomes included median relative error (MRE), mean absolute error to baseline (MAEB) and median relative error to baseline (MREB) which we defined as :

- $MAE = mean(|\widehat{h_{t+14}} - h_{t+14}|)$.
- $MRE = median(|\frac{\widehat{h_{t+14}} - h_{t+14}}{h_{t+14}}|)$.
- $MAEB = mean(|\widehat{h_{t+14}} - h_{t+14}| - |h_t - h_{t+14}|)$.
- $MREB = median(|\frac{\widehat{h_{t+14}} - h_{t+14}}{h_t - h_{t+14}}|))$

The median was preferred over the mean for both MRE and $MREB$ metrics due to the tendency of these metrics to exhibit exceptionally high values when the denominator approaches zero. This situation occurs when the number of hospitalized patients is close to zero for MRE or when the number of patients hospitalized at 14 days is close to the current number of hospitalized patients for $MREB$. As for $MAEB$ and $MREB$, these metrics compare the model's performance against a baseline model that predicts the current number of hospitalized patients at 14 days. The use of such metrics aids in assessing the additional information provided by the model. It's noteworthy that these metrics offer a valuable baseline, as indicated by Cramer et al. (2022), considering that COVID-19 forecast models do not consistently surpass the performance of this basic forecast.

Those different metrics emphasize different type of error. MAE and MAEB will output large error when the number of hospitalization at 14 days is high. MRE will output large error when the number of hospitalization at 14 days (i.e the denominator) is low. MREB will output large error when the number of hospitalization at 14 days is close to the current number of hospitalization. Consequently, each

metric underscores specific phenomena: MRE is sensitive to new increases in hospitalizations, MREB is sensitive to shifts from a plateau regime to an increase or decrease, and MAE/MAEB are sensitive to large prediction errors, which are more likely to occur when the number of hospitalizations is high. Given the particular importance of the impact on hospital organization when the number of hospitalizations is high, we selected MAE as the primary outcome.

Data preprocessing involved computing the first and second derivative over the last 7 days to enrich the model with additional information. Furthermore, the data were smoothed using a local polynomial regression with a span of 21 days to account for daily noise variation as it was suggested to improve performance by Ferté et al. (2022). Finally, features were scaled between -1 and 1 using maximum absolute scaling, (i.e by dividing the observed value by the maximum absolute value of the respective feature).

All data utilized in this study are publicly available. Weather data can be obtained from Smith et al. (2011) through the R package `worldmet` (Carslaw, 2023). The vaccine data can be downloaded from Etalab (2020). Electronic Health Records (EHRs) data are accessible on Dryad (Ferté et al., 2023). To address privacy concerns, EHRs data publicly available below 10 patients were obfuscated to 0. Therefore, we set the outcome, the forecast and the hospitalization to 10 when their value was 0 when evaluating the model performance. For convenience, all data were downloaded, merged, and are provided as part of the replication material in the Github repository https://github.com/thomasferte/-Ferte2024ICML_HighDimensionReservoir.

2.2. Evaluation framework

Period from May 16, 2020 to March 1, 2021 served to identify relevant hyperparameters only. Remaining data was used to evaluate the model performance while updating the hyperparameters each month. Model was trained to forecast the variation of hospitalization instead of the raw number of hospitalization as we found out it was easier for machine learning approaches to learn this task (i.e $outcome_{t+14} = h_{t+14} - h_t$). However, all metrics and graphical representations will be in the raw hospitalization scale. An additional analysis was performed to evaluate forecast at 7 and 21 days instead of 14 days.

Models were trained and evaluated one day over two to ensure leveraging on the most up to date data and to limit computation time. We used all data available from the previous year. Because COVID-19 times series is not stationary we found that limiting the learning to a sliding window of one year yield better performance in previous experiments.

2.3. Models

Originally developed by Jaeger (2001) and Maass et al. (2002), Reservoir Computing (RC) emerged as a distinctive machine learning paradigm (Lukoševičius & Jaeger (2009)) tailored for the analysis of information derived from dynamic systems. RC represents a recurrent neural network in which the internal connections are randomly set (this is called the *reservoir*), while only the output layer (called *read-out*) undergoes training. The reservoir performs a non-linear projection of inputs into high-dimensional space, facilitating temporal non-linear interactions. Then, "useful" computations are extracted (*read-out*) from the reservoir by a linear output layer. This alternative approach to error-backpropagation has demonstrated performance comparable to traditional RNNs but with reduced computing time, as noted by Vlachas et al. (2020). Its applications extend across a diverse array of tasks, including bird songs (Trouvain & Hinaut, 2021), language processing (Hinaut & Dominey, 2013), epidemic (Ghosh et al., 2021; Ray et al., 2021; Liu et al., 2023), power plants, internet traffic, stock prices, and beyond (see (Lukoševičius & Jaeger, 2009; Tanaka et al., 2019; Zhang & Vargas, 2023) for reviews). RC relies on several hyperparameters which influence the internal dynamics of the reservoir. Selecting the most suitable hyperparameters can be challenging (some expertise is needed) and researchers often experiment different combinations to identify the most relevant set of hyperparameters. For the task tackled in this study, it is even more challenging because of the hundreds of inputs variables, which is uncommon for RC.

We conducted a comparative analysis involving RC combined with GA feature selection (RC-GA), elastic-net penalized regression (Enet), XGBoost, Transformers, Prophet, Informer, PatchTST and Long Short-Term Memory (LSTM). Additionally, we evaluated the performance of RC with GA optimization without feature selection (RC-GA (no FS)), as well as RC with random search feature selection (RC-RS). Furthermore, drawing inspiration from Weddell et al. (2021), we propose to perform dimension reduction by principal component analysis and then perform RC with GA (RC-PCA-GA). As LSTM and Transformers lack inherent feature selection and are computationally intensive, we also performed dimension reduction with PCA for those two methods.

We used RC where both the input layer and the reservoir are connected to the output layer. Number of recurrent units was set to 500. Additionally, we evaluated the performance of RC-GA with 2000 units, for which we used the same hyperparameters as those selected by RC-GA with 500 units, except for ridge penalization (see section 2.4); this model was denoted RC-(GA-500)-2k. For hyperparameter optimization, we assessed the MAE of the median forecast

across 3 RC instances with the same hyperparameter sets. We performed an additional analysis using 20 RC for hyperparameter optimization instead of 3 to evaluate its impact on performance and hyperparameter selection. Forecast evaluation was based on the MAE of the median forecast of 40 RC instances, with one instance for each of the top 40 hyperparameter sets. It is worth noting that aggregating forecasts from multiple RC instances is a good practice due to the random connections in RC, resulting in variations in forecasted values even with similar hyperparameter sets. To investigate whether the performance gain of aggregating several reservoir is similar when applied to other approaches, we also examined the impact of aggregating 20 LSTMs instead of using a single LSTM.

To emphasize the need of high-dimensional data in this context, we performed an additional analysis similar to one performed using Enet in our previous paper (Ferté et al., 2022). In this analysis, all the data are included except one data source among the department-level COVID-19 data, the weather data, the emergency units data or the first and second derivatives.

2.4. Hyperparameter optimization

For RC-GA, we employed a GA with barycentric crossover, tournament selection, and Gaussian mutation, optimizing the MAE as the objective function. The algorithm, detailed in Algorithm 1 in Appendix, is initialized by randomly drawing initial individuals similarly to random search. For each individual, two parents are selected through tournament selection from the previous generation, followed by a barycentric crossover (i.e weighted mean for numeric hyperparameters, and weighted random choice for categorical hyperparameters), and mutation (i.e Gaussian for numeric hyperparameters, random choice for categorical hyperparameters). We used consistent GA hyperparameters for all experiments: $NGeneration = 30$ for initial optimization and $NGeneration = 10$ for monthly updates, $Npop = 200$, $Nchildren = 100$, $Ntour = 2$, $pmutNum = 0.5$, $pmutCat = 0.25$, $sigma = 1$, except for the leaking rate optimization, which has a shorter range compared to other hyperparameters (i.e., from $1e-5$ to 1) with $sigma = 0.1$. No optimization of the GA metaparameters was performed. It's worth noting that some features were log-transformed before processing by GA, as indicated in Table 1.

For LSTM, we used a single LSTM (with forget gate) as originally presented in Gers et al. (2000). We set the maximal number of units inside the LSTM to 10, in order to make the comparison fair with RC in term of total number of trainable parameters, which is around 2 000 (compared to less than 800 parameters for RC). We add a fully connected layer of units with linear activation function on top of the LSTM to forecast. A L2 regularization is applied to the weights of the fully connected layer, with a regularization coefficient. We then trained the LSTM using Adam gradient descent algorithm. Loss was computed with the mean-squared error. To prevent overfitting, we designated the last available week as the validation dataset and implemented early stopping of 5 epochs. On average, LSTMs were trained on 314 ± 104 (std) epochs before achieving their best measured performance on validation data.

For Transformers we based our implementation on the work of Zeng et al. (2023). We selected the depth of the model in order to approximate the number of trainable parameters of RC. As for LSTM the last week was kept as validation set. We also add 3 time features defined as the Day of week, Day of month, and Age of series (i.e a feature value which is small for distant past timestamps and monotonically increases the more we approach the current timestamp). We then trained the Transformers using Adam gradient descent algorithm, with a learning rate of $1e{-}3$.

The hyperparameter range was defined according to table 1. GA was used to optimize hyperparameters of RC-GA, RC-GA (no FS) and RC-PCA-GA. For RC-PCA-GA, the number of component was selected through the proportion of explained variance optimized by GA which was passed as a hyperparameter to GA. For RC-(GA-500)-2k, we used the hyperparameters selected by RC-GA with 500 units except the ridge penalization which was adapted to the greater number of units based on 100 iterations of random search (RS). We used RS for other algorithms where new hyperparameters sets were randomly sampled in the search space defined in table 1. Both for GA and RS, 200 best hyperparameters sets were passed from one month to the next and reevaluated in order to leverage from the previous monthly hyperparameter search. We chose to optimize all hyperparameters of elastic-net, the main hyperparameters of XGBoost according to Yang & Shami (2020), the main hyperparameters of RC according to reservoirpy documentation Trouvain & Hinaut (2022).

Code and supplementary results are available at https://github.com/thomasferte/-Ferte2024ICML_HighDimensionReservoir.

3. Results

3.1. Performance

Performance of the different models are presented in table 2 and graphical predictions are outlined in figure 1. The best models were RC-GA optimized with 20 reservoirs and monthly update, RC-GA optimized with 20 reservoirs and without monthly update and RC-GA optimized with 3 reservoir and without monthly update, with MAE of 14.70, 14.84, 15.27 respectively. Additionally, when using 3 reservoir for

Table 1. HYPERPARAMETER RANGES

MODEL[1]	HYPERPARAMETER	RANGE	LOG[2]
XGB	N_ESTIMATORS	3;300	FALSE
	MAX_DEPTH	5;100	FALSE
	LEARNING_RATE	1E-5;1	TRUE
	SUBSAMPLE	0;1	FALSE
	COLSAMPLE_BYTREE	0;1	FALSE
ENET	RIDGE	1E-10;1E5	TRUE
	L1_RATIO	0;1	FALSE
LSTM	N_UNITS	1;10	FALSE
	NUM_EPOCHS	200;2000	FALSE
	LEARNING_RATE	1E-4;1E-2	TRUE
	L2_REGULARISATION	1E-7;1E-2	TRUE
	VARIANCE EXPLAINED	0;0.9999999	TRUE
RC	RIDGE	1E-10;1E5	TRUE
	SPECTRAL_RADIUS	1E-5;1E5	TRUE
	LEAKING_RATE	1E-5;1	TRUE
	INPUT_SCALING	1E-5;1E5	FALSE
	FEATURE_SELECTION[3]	Y, N	
PCA	VARIANCE EXPLAINED	0;0.9999999	TRUE
PROPHET	CHANGEPOINT	1E-3;0.5	TRUE
	SEASONALITY	1E-2;10	TRUE
	HOLIDAYS	1E-2;10	TRUE
	SEASONALITY	MULT.;ADD.	FALSE

[1] XGB : XGBoosting, Enet : Linear regression with elastic-net penalization, RC : Reservoir computing

[2] For features with $Log = True$, features are log-transformed before being processed by GA or RS

[3] One hyperparameter per feature

hyperparameter optimization, increasing the number of units in RC-GA to 2000 improved the MAE from 15.27 to 14.66. Lack of features selection or optimisation using RS instead of GA greatly hampered the performance of RC, with an MAE of 17.59 and 18.45 respectively compared to a MAE of 15.27. Most model did not improve when monthly updating hyperparameters with the exception of RC-RS and RC-GA (no FS). RC-PCA-GA was the best model when monthly updating hyperparameters. RC-GA consistently outperformed XGB at 7 and 21 days. However, both model performance at 21 days were worse than baseline. In addition, table 3 outlined the need for combining the different data source. While removing department-level data, weather data, emergency units data or derivatives, MAE increased by 0.4 (department), 0.66 (weather), 0.35 (ermergency) and 3.47 (derivatives) respectively.

Graphically, predictions of different models outlined different properties. While RC-GA with 2000 units was overall better, XGBoost seemed more sensitive to epidemic increase (i.e summer 2021 increase, winter 2021-2022 decrease) and LSTM anticipated better the spring 2021 initial decrease.

We evaluated the MAE of RC-GA using all data to compare when using only previous year data. MAE for both model were respectively 15.71 and 15.27. We also evaluated the performance of LSTM and Transformers when given the features selected by RC-GA instead of performing dimension reduction by PCA. Performance of LSTM improved with a MAE of 15.08 while Transformers performance remained low with a MAE of 18.1. More details is available at supplementary table 3.

Graphical representation of RC-GA using all data is available in figure 5.

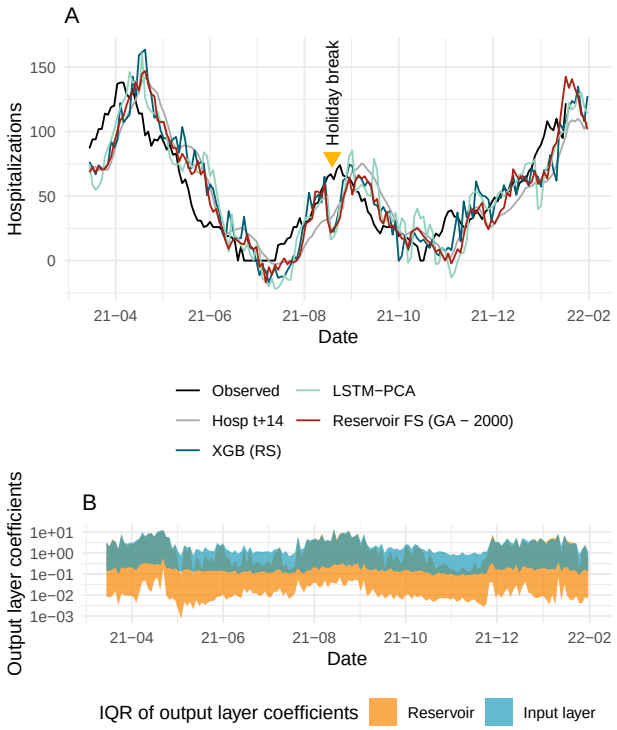

Figure 1. Panel A : Forecast of Reservoir computing with genetic algorithm (GA) & feature selection (FS), Elastic-net and XGBoost (XGB). For elastic-net, XGBoost and LSTM, hyperparameter optimization was performed by random search (RS). All algorithms are shown without monthly update of hyperparameters. Of note, the summer prediction error depicted by the orange triangle is concurrent to the summer holiday break characterized by a decrease of the number of RT-PCR test. Panel B : Reservoir computing with GA feature selection. The figure displays the output layer coefficient of the Reservoir and the input layer as both of them are connected to the output layer. The shaded area corresponds to the area between the 25th and 75th percentile of the output layer over time.

Table 2. Model performance. Best model without monthly update of hyperparameters is outlined in bold. Best model with monthly update is outlined in bold and italic. RC-(GA-500)-2k was underlined when it outperformed best model without monthly update of hyperparameters. Table with std in Appendix.

Model	Update	MAE	MAEB	MRE	MREB
RC-GA[1]	No	15.27	-3.31	0.24	0.84
RC-(GA-500)-2k	No	<u>14.66</u>	<u>-3.92</u>	0.25	0.83
RC-GA (20 HP)	No	**14.84**	**-3.75**	0.25	0.81
RC-GA	Yes	15.79	-2.80	***0.27***	0.90
RC-GA (20 HP)	Yes	***14.70***	***-3.88***	0.27	***0.81***
RC-RS	No	18.45	-0.14	0.36	1.00
RC-RS	Yes	17.81	-0.77	0.31	0.98
RC-GA (no FS)	No	17.59	-0.99	0.31	0.96
RC-GA (no FS)	Yes	16.95	-1.64	0.29	0.97
RC-PCA-GA	No	15.59	-3.00	0.28	0.87
RC-PCA-GA	Yes	15.63	-2.95	0.29	0.85
Enet-RS	No	15.83	-2.76	0.29	0.86
Enet-RS	Yes	16.22	-2.36	0.30	0.88
XGB-RS	No	15.45	-3.14	0.28	0.81
XGB-RS	Yes	16.32	-2.27	0.29	0.89
Transformers-PCA	No	19.28	0.69	0.32	1.00
Informer-PCA	No	18.70	0.12	0.34	1.00
PatchTST-PCA	No	18.34	-0.25	0.31	1.00
LSTM-PCA[2]	No	15.74	-2.84	**0.24**	0.76
20 LSTM-PCA[3]	No	16.23	-2.35	0.25	**0.74**
Prophet	No	21.30	2.71	0.39	1.05
RC-GA - dep[4]	No	15.67	-2.92	0.27	0.90
RC-GA - weather	No	15.93	-2.66	0.27	0.88
RC-GA - emergency	No	15.62	-2.97	0.28	0.85
RC-GA - deriv	No	18.74	0.15	0.36	1.01
(7 days) RC-GA	No	8.59	-2.42	0.16	0.80
(7 days) XGB-RS	No	8.96	-2.04	0.14	0.80
(21 days) RC-GA	No	25.46	0.12	0.52	1.00
(21 days) XGB-RS	No	25.85	0.51	0.47	1.13

[1] RC-GA : Reservoir computing with genetic algorithm. RC-(GA-500)-2k, RC-GA (20 HP), RC-RS, RC-GA (no FS), RC-PCA-GA are similar to RC-GA but with 2000 neurons instead of 500, 20 reservoir for hyperparameter optimisation instead of 3, random search instead of genetic algorithm and PCA instead of feature selection respectively.
[2] LSTM-PCA with 1 LSTM.
[3] LSTM-PCA with median forecast of 20 LSTM.
[4] Similar to RC-GA but with department, weather, ermergency units or derivatives data removed.

3.2. Hyperparameters

Hyperparameters evolution of numeric hyperparameters of RC-GA are displayed in figure 2. We observe that best selected hyperparameters evolved over time. At first, only ridge exhibit a clear optimum whereas input scaling and spectral radius seemed working nicely on a large range of value and leaking rate exhibits a bimodal distribution where best leaking rate could be either close to 1 or close to 1e-4. After September 2021, hyperparameters are mostly stable with a precise optima close to 0.5 and 1e4 for leaking rate and ridge respectively and a large area of optimal values ranging from 1 to 1e4 and 1e-4 to 1e2 for input scaling and spectral radius respectively.

Of note, the superiority of the absence of monthly update of hyperparameters with RC-GA compared to monthly update of hyperparameters is likely due to the best individuals with a leaking rate below 1e-3. Indeed the performance of the aggregation of those 14 models showed a MAE of 15.07 on the test set compared to 15.52 for models with leaking rate above 1e-3. More details are available in Appendix in table 4 and figure 7. Similarly, as depicted in figure 8 in Appendix, the models with hyperparameters defined at 2021-03-01 outperformed models updated later in September, November and December which is after GA converged to a high leaking rate. When increasing the number of reservoir from 3 to 20 for hyperparameter optimization, MAE improved from 14.84 to 14.70 when monthly updating the hyperparameters. This improvement might be explained by the convergence to a leaking rate close to 1e-2 instead of 1, which was the case when using only 3 reservoirs for hyperparameter optimization, as shown in supplementary figure 9. As previously demonstrated in supplementary table 4, a leaking rate of 1 might be suboptimal compared to a lower leaking rate. Therefore, having 20 reservoirs instead of 3 may help the genetic algorithm converge to a more optimal leaking rate.

RC-GA (no FS) and RC-RS also exhibited the bimodal distribution of leaking rate. Other hyperparameters were close to the one observed for RC-GA. For RC-PCA-GA, best percentage of variance explained was mostly between 0.6 and 0.8. The leaking-rate of this algorithm was optimum at 1e-4 without bimodal distribution unlike RC-GA which might explain the less discrepancy between update and no update of hyperparameters.

For other algorithms we found that hyperparameters of XGBoost were stable over time except the proportion of subsample that present a bimodal distribution at 0.3 and 0.9. For Elastic-net regression, ridge penalisation remained around 1.0 but l1_ratio was more in favor of ridge penalty at the beginning of epidemic whereas it gave more importance to lasso penalisation after September 2021. More details are available in figure 6.

3.3. Feature selection and importance

Based on the genetic algorithm we can outline the most selected features. Figure 3 outlines the most selected features by RC-GA at 2021-03-01 and their evolution over time. In-

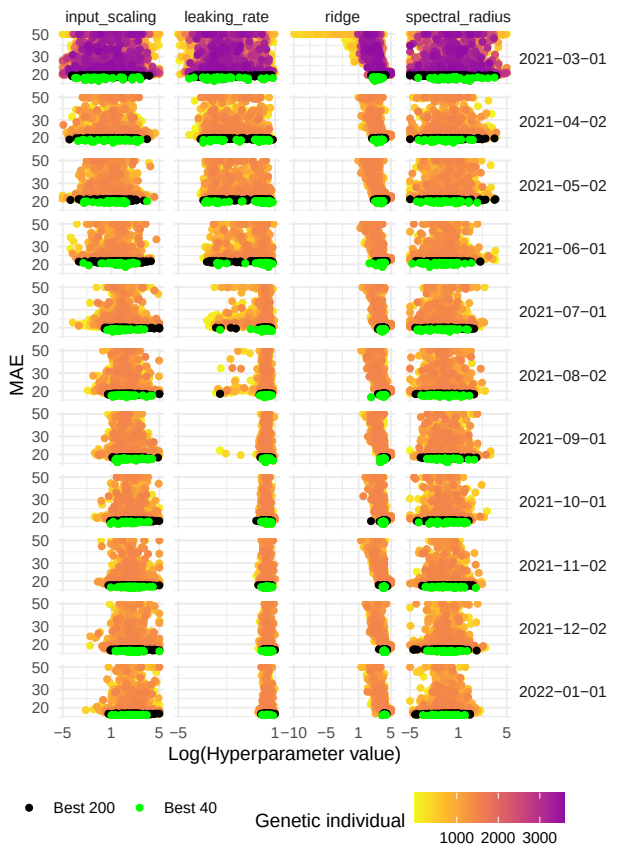

Figure 2. Hyperparameters selected by GA. Values above 50 were set to 50 for easier visualization. At each month, 200 best individuals (in black) according to mean absolute error (MAE) are passed to the next month for initialization of the GA except for the first month where the GA is initialized with 200 individuals draw from random search. The best 40 individuals (in green) for each month are used to forecast hospitalizations.

terestingly, we can also have a feature importance measure based on Elastic-net coefficient and XGBoost feature gain. RC-GA often selected feature related to RT-PCR in the elderly which was also observed for Elastic-net and XGBoost. Details about most important features for each algorithm is provided in Appendix in figure 10.

Overall, among the models used to forecast on the test set, RC-GA selected a median (interquartile range (IQR)) of 197 (191 ; 204) features, Elastic-net selected a median (IQR) of 233 (228, 236) features, RC-PCA-GA selected a median (IQR) of 14 (12 ; 18) components.

We examined the principal components of PCA as of January 17, 2022. PCA exhibited a tendency to group features thematically; for example, dimensions 1 and 5 were associated with SAMU (French ambulance service), dimensions 2, 9, and 10 were linked to emergency departments, and dimension 6 pertained to weather. Beyond dimension 14, the principal components became more diverse thematically. For instance, dimension 14 grouped dew point with features related to the emergency department, dimension 15 combined SAMU features with those related to the pediatric emergency department, and dimension 17 associated windspeed with fever at the emergency department. More details are available in figure 11 in Appendix.

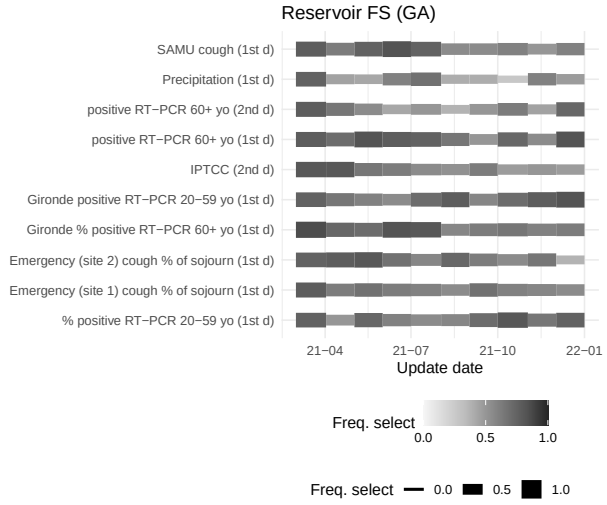

Figure 3. Feature selection by GA among the 40 best hyperparameter sets at each hyperparameter update. Only 10 most selected features at 2021-03-29 are shown.

3.4. Reservoir importance

Given the structure of RC, where both the input layer and reservoir layer are connected to the output layer, we can evaluate the importance of each layer by analyzing the coefficient of each connection to the output layer. In Panel B of Figure 1, we compare these coefficients. We observe that sudden changes in the respective importance of layers occur consecutively to shifts in slopes. For instance, the decrease in May is followed by a major decrease in the 75th quantile of output coefficients for the reservoir. This indicates that, following the decrease, RC-GA relies more on the input layer compared to the previous period. A similar inverse phenomenon is observed during the summer increase and winter 2021-2022 increase. Generally, the output layer tends to attribute more importance to the reservoir layer during epidemic increases. This suggest that better forecasting during increases depends more on non-linear combinations of inputs than other periods.

3.5. Computation time and number of parameters

We evaluated the computation time of the different algorithms by computing the forecast of December 2021 (i.e training and forecasting on 16 days as model evaluation is performed one day over two). The computation were performed on a 12th Gen Intel i7-12700H, 4.600GHz with 20 CPUs with Ubuntu 20.04.6 LTS x86_64 OS. We found a computing time of 0.2s, 32.2s, 74.5s, 145.5s, 290.7s, 868.7s for Elastic-net, LSTM, XGBoost, RC-GA with 500 units, RC-GA with 2000 units and Transformer respectively. Of note, as outlined in the methods section, the forecasting using RC-GA involved training 40 RC, and the displayed time reflects this cumulative training time[1].

We also evaluated the number of parameter of each algorithm. At the last day of forecast (i.e 2022-01-17), Elastic-net resulted in 394 parameters, RC-GA was between 683 and 727 depending on feature selection based on genetic algorithm, XGBoost in 13,705 (i.e number of splits and leaves). Of note, there were 409 predictors in the dataset but constant predictors were removed from the training set resulting in only 393 at the last day. As described in the methods section, the number of parameters of LSTM was chosen as close to 2,000 to be similar to the one used for RC-GA.

3.6. Reservoir aggregation

Due to the inherent randomness in reservoir comparison, employing the same set of hyperparameters can yield two reservoirs with distinct forecasts and performances. In order to address this variability, we explored the impact of aggregating multiple reservoirs, where the collective prediction is the mean of all individual reservoir predictions (see figure 4 in Appendix). In our post-hoc analysis, we observed that the forecast from a single reservoir exhibits considerable variability. However, this variability significantly diminishes when aggregating forecasts from 5 reservoirs and becomes notably stable after aggregating predictions from 10 reservoirs.

4. Discussion

In this work, we proposed to combine Reservoir Computing (RC) with Genetic Algorithms (GA) to forecast hospitalizations at Bordeaux University Hospital in a high-dimension setting. The proposed approach outperformed most state-of-the-art methods, including elastic-net (i.e the best method obtained by Ferté et al. (2022)), Transformers, PatchTST, Prophet, Informer and showed slight improvement over XGBoost and LSTM. Those results might indicate that the addition of non-linear temporal interaction are needed to accurately forecast SARS-CoV-2 epidemic which is lacking in a penalized linear regression model (elastic-net). In addition, we demonstrated that RC by itself (i.e without feature selection) was unable to perform well in this setting, but combined with GA for feature selection and hyperparameter optimisation it outperformed all other methods considered.

Limited research has explored the application of RC in high-dimensional settings. Weddell et al. (2021) used RC to forecast microsleep using 544 input features and over 10^4 observations. Here, the dimensionality problem was initially reduced to 60 predictors through principal component analysis before applying RC. Similar to Weddell et al. (2021), we utilize GA to optimize the variance explained threshold for selecting PCA components in RC-PCA-GA. This strategy yielded performance close to, albeit slightly lower than, RC-GA. Importantly, the reduced number of hyperparameters in this approach, where dimension reduction is governed by a single parameter as opposed to one per feature in our method, suggests that fewer generations of GA may be required. This makes it particularly appealing in settings with limited computing resources. Interestingly, as explained in Appendix in figure 11 the number of selected components by RC-PCA-GA corresponds to the number of components before they exhibit heterogeneity in terms of epidemic compartment representation (i.e the stage before each component can be identified as representing a distinct epidemic compartment). Thus, this confirms the validity of our approach.

In this work, we demonstrated that dimension reduction (either with PCA or with feature selection) was necessary to use RC in this context of high-dimension especially regarding the poor performance of RC when no feature selection was performed. Another approach in this setting could be to increase the number of neurons to facilitate dimension expansion. Indeed, we showed that augmenting the number of neurons to 2,000 instead of 500 improved the performance of the algorithm. This approach was explored by Arcomano et al. (2020) using RC for weather forecasting based on 528 input features and over 10^5 observations with 9,000 neurons. Furthermore, RC has been implemented for image recognition (Jalalvand et al., 2018) and video processing, presenting distinct high-dimensional challenges compared to time series forecasting Sun et al. (2020), which may justify further exploration.

The application of RC for epidemic forecasting is relatively recent, as evidenced by the works of Ghosh et al. (2021), Liu et al. (2023), and Ray et al. (2021). All focused on forecasting the COVID-19 epidemic and also dengue epidemic for Ray et al. (2021). Notably, Ray et al. (2021) demonstrated the superiority of RC over Feed-Forward Neural Network in three epidemic forecast tasks and its superiority over LSTM in one of the three tasks. However, a common limitation in

[1] As said in next subsection, aggregation of reservoirs could be reduced to 10 instead of 40, thus reducing the computation time.

these studies was the use of only one time series as input, overlooking the potential necessity for multiple predictors in predicting the spread of epidemics.

This work presents several strengths. It tackles an important and difficult challenge in high-dimension context with few data because it involves a high number of variables (409) with little data available (one sample per day for about a year for training). Previous attempts led to mitigated performance; in this work we improve forecasting significantly better than baseline compared to previous studies. Moreover, this work proposes an in-depth comparison of RC-GA with more classical RC approaches in order to outlines the advantage of combining RC with GA. We also compared RC-GA with other machine learning algorithm to better investigate the usefulness of this approach. In addition, we performed an in-depth analysis of hyperparameter and feature selection while also considering computation time.

This study has various limitations and areas for potential future research. The analysis indicates that monthly updates to RC hyperparameters did not enhance performance, suggesting potential issues such as hyperparameter overfitting or GA getting stuck in local optima. Indeed, the leaking rate exhibited a bimodal distribution until September 2021 and ultimately converged to a sub-optimal high leaking rate. Therefore, this implies there is an opportunity for improvement by refining the GA search strategy, allowing for more exploration, especially in the initial phases of each month.

In order to limit computation time, GA optimizes the hyperparameters for three reservoirs while the final prediction involves aggregating results from 40 reservoirs with the best hyperparameter sets. To address this gap, we conducted an additional analysis using 20 reservoirs instead of 3 for hyperparameter optimization. This improved the MAE, possibly by avoiding convergence to a local minimum in the leaking rate, especially when updating hyperparameters monthly. An alternative to gain similar performance was to increase the number of neurons to 2000 instead of 500. Future investigations are needed to understand which of these two approaches (or both maybe) is best for optimizing the hyperparameters regarding optimal solutions and computing time.

While hyperparameter optimization for other algorithms was conducted through random search, alternative optimization approaches might have yielded improvements. However, given the small number of hyperparameters, the high number of iterations, and the stability of hyperparameters across different months, it seems unlikely that we could obtain significantly better results for other approaches. Additionally, using more parameters in alternatives methods (e.g. LSTM, Transformers) could have led to improved performance, although this is not guaranteed due to the limited amount of data available. Similarly, we could have done the same for RC, and one could expect that the improvement (compared to other methods) would be even better given the decrease of error obtained by going from 500 to 2000 units inside the reservoir. More improvement could be expected also by parallelizing a population of reservoirs all connected to the output layer (instead of averaging across independent RC): this would enable various set of hyperparameters to coexist, thus creating various non-linear dynamics of input recombinations. This would be particularly interesting to handle various periods of the epidemic without having to extend much the processing time. For this study, we opted to limit exploration based on comparable computing time and a similar number of estimated parameters across different algorithms.

5. Conclusion

In this study, our objective was to predict the number of SARS-CoV-2 hospitalizations at a major hospital in France for t + 14 days, within the challenging context of high-dimensional data characterized by a scarcity of observations and an abundance of predictors. Our findings indicate that the integration of RC with GA surpasses state-of-the-art methods, even those highlighted as superior in prior papers. These results hold promise for enhancing hospital management during epidemics considering that we could easily increase the number of trainable parameters that we limited for the purpose of the study. Of course, the generalizability of these outcomes to diverse settings requires further investigation. In this high-dimension context (409 variables), because we do not have much parameters to train with RC we can afford to train on short timeseries (586 time points) and thus potentially manage to capture new variants or new trends quickly. Further work remains to show that we could update the hyperparameters in terms of few days in order to capture even quicker new trends.

Moreover, considering the growing importance of addressing environmental impact amid climate change, incorporating algorithmic energy consumption into a multi-objective genetic algorithm (GA), as suggested by Yokoyama et al. (2023) could significantly contribute to reducing environmental consequences.

Impact Statement

This paper aims to contribute to the advancement of time series forecasting in high-dimensional contexts with limited available data, with potential applications in forecasting future epidemics. Importantly, the paper aims to forecast the number of SARS-CoV-2 hospitalized patients at 14 days to help anticipate the bed requirements of a large scale hospital using public data and electronic health records data. In practice, this means that improvements in forecasting

can help hospital managers to be better prepared to host more people. Said crudely, better forecasting means to have the potential to save more lives. Moreover, it has an increasingly relevant concern given the context of global warming. The adoption of reservoir computing in this work is particularly noteworthy, as it requires fewer computing resources compared to deep learning approaches, because of the fewer parameters that are trained. This feature mitigates potential environmental impacts associated with resource-intensive methods.

Acknowledgements

We want to thank the reviewers for their thoughtful comments. Specifically, Reviewer KmgK challenged our approaches and suggested additional comparisons with Prophet, Informer, and PatchTST. Reviewer 3kD1 asked for more details about the computation time. Reviewer TYsA suggested exploring the discrepancy between the number of reservoirs used for hyperparameter optimization and forecasting, which greatly improved the approach.

Experiments presented in this paper were carried out using the PlaFRIM experimental testbed, supported by Inria, CNRS (LABRI and IMB), Université de Bordeaux, Bordeaux INP and Conseil Régional d'Aquitaine (see https://www.plafrim.fr) and by the MCIA (Mésocentre de Calcul Intensif Aquitain).

This study was carried out in the framework of the University of Bordeaux's France 2030 program / RRI PHDS.

References

Agrawal, P., Abutarboush, H. F., Ganesh, T., and Mohamed, A. W. Metaheuristic Algorithms on Feature Selection: A Survey of One Decade of Research (2009-2019). *IEEE Access*, 9:26766–26791, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3056407. URL https://ieeexplore.ieee.org/abstract/document/9344597. Conference Name: IEEE Access.

Arcomano, T., Szunyogh, I., Pathak, J., Wikner, A., Hunt, B. R., and Ott, E. A Machine Learning-Based Global Atmospheric Forecast Model. *Geophysical Research Letters*, 47(9):e2020GL087776, 2020. ISSN 1944-8007. doi: 10.1029/2020GL087776. URL https://onlinelibrary.wiley.com/doi/abs/10.1029/2020GL087776. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1029/2020GL087776.

Bala, A., Ismail, I., Ibrahim, R., and Sait, S. M. Applications of Metaheuristics in Reservoir Computing Techniques: A Review. *IEEE Access*, 6:58012–58029, 2018. ISSN 2169-3536. doi: 10.1109/ACCESS.2018.2873770. URL https://ieeexplore.ieee.org/abstract/document/8481347. Conference Name: IEEE Access.

Carslaw, D. worldmet: Import Surface Meteorological Data from NOAA Integrated Surface Database (ISD), June 2023. URL https://cran.r-project.org/web/packages/worldmet/index.html.

Carvalho, K., Vicente, J. P., Jakovljevic, M., and Teixeira, J. P. R. Analysis and Forecasting Incidence, Intensive Care Unit Admissions, and Projected Mortality Attributable to COVID-19 in Portugal, the UK, Germany, Italy, and France: Predictions for 4 Weeks Ahead. *Bioengineering*, 8(6):84, June 2021. doi: 10.3390/bioengineering8060084. URL https://www.mdpi.com/2306-5354/8/6/84. Number: 6 Publisher: Multidisciplinary Digital Publishing Institute.

Cramer, E. Y., Ray, E. L., Lopez, V. K., Bracher, J., Brennen, A., and et al. Evaluation of individual and ensemble probabilistic forecasts of COVID-19 mortality in the United States. *Proceedings of the National Academy of Sciences*, 119(15):e2113561119, April 2022. doi: 10.1073/pnas.2113561119. URL https://www.pnas.org/doi/10.1073/pnas.2113561119. Publisher: Proceedings of the National Academy of Sciences.

Etalab. Les données relatives au COVID-19 en France - data.gouv.fr, 2020. URL https://www.data.gouv.fr/fr/pages/donnees-coronavirus/.

Ezugwu, A. E., Adeleke, O. J., Akinyelu, A. A., and Viriri, S. A conceptual comparison of several metaheuristic algorithms on continuous optimisation problems. *Neural Computing and Applications*, 32(10):6207–6251, May 2020. ISSN 0941-0643, 1433-3058. doi: 10.1007/s00521-019-04132-w. URL http://link.springer.com/10.1007/s00521-019-04132-w.

Ferté, T., Jouhet, V., Griffier, R., Hejblum, B. P., Thiébaut, R., and Bordeaux University Hospital Covid-19 Crisis Task Force. The benefit of augmenting open data with clinical data-warehouse EHR for forecasting SARS-CoV-2 hospitalizations in Bordeaux area, France. *JAMIA open*, 5(4):ooac086, December 2022. ISSN 2574-2531. doi: 10.1093/jamiaopen/ooac086.

Ferté, T., Jouhet, V., Griffier, R., Hejblum, B., Thiébaut, R., and Bordeaux University Hospital Covid-19 Crisis Task Force. The benefit of augmenting open data with clinical data-warehouse EHR for forecasting SARS-CoV-2 hospitalizations in Bordeaux area, France, January 2023. URL https://datadryad.org/stash/dataset/doi:10.5061/dryad.hhmgqnkkx. Artwork Size: 153575 bytes Pages: 153575 bytes.

Gers, F. A., Schmidhuber, J., and Cummins, F. Learning to forget: Continual prediction with lstm. *Neural computation*, 12(10):2451–2471, 2000.

Ghosh, S., Senapati, A., Mishra, A., Chattopadhyay, J., Dana, S. K., Hens, C., and Ghosh, D. Reservoir computing on epidemic spreading: A case study on COVID-19 cases. *Physical Review E*, 104(1):014308, July 2021. ISSN 2470-0045, 2470-0053. doi: 10.1103/PhysRevE.104.014308. URL https://link.aps.org/doi/10.1103/PhysRevE.104.014308.

Goldberg D.E. Genetic algorithms in search, optimization and machine learning. *Addison Wesley*, 1989.

Hinaut, X. and Dominey, P. F. Real-Time Parallel Processing of Grammatical Structure in the Fronto-Striatal System: A Recurrent Network Simulation Study Using Reservoir Computing. *PLOS ONE*, 8(2):e52946, February 2013. ISSN 1932-6203. doi: 10.1371/journal.pone.0052946. URL https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0052946. Publisher: Public Library of Science.

Holland J.H. Adaptation in natural and artificial system. *Ann Arbor University of Michigan Press*, 1975.

Jaeger, H. The" echo state" approach to analysing and training recurrent neural networks-with an erratum note'. *Bonn, Germany: German National Research Center for Information Technology GMD Technical Report*, 148, January 2001.

Jalalvand, A., Demuynck, K., De Neve, W., and Martens, J.-P. On the application of reservoir computing networks for noisy image recognition. *Neurocomputing*, 277:237–248, 2018.

Jong, D. and Alan, K. Analysis of the behavior of a class of genetic adaptive systems. 1975. URL http://deepblue.lib.umich.edu/handle/2027.42/4507. Accepted: 2006-02-03T19:05:13Z.

Liu, B., Xie, Y., Liu, W., Jiang, X., Ye, Y., Song, T., Chai, J., Feng, M., and Yuan, H. Nanophotonic reservoir computing for COVID-19 pandemic forecasting. *Nonlinear Dynamics*, 111(7):6895–6914, 2023. ISSN 0924-090X. doi: 10.1007/s11071-022-08190-z.

Lukoševičius, M. and Jaeger, H. Reservoir computing approaches to recurrent neural network training. *Computer Science Review*, 3(3):127–149, August 2009. ISSN 1574-0137. doi: 10.1016/j.cosrev.2009.03.005. URL https://www.sciencedirect.com/science/article/pii/S1574013709000173.

Maass, W., Natschläger, T., and Markram, H. Real-time computing without stable states: a new framework for neural computation based on perturbations. *Neural Computation*, 14(11):2531–2560, November 2002. ISSN 0899-7667. doi: 10.1162/089976602760407955.

Mohimont, L., Chemchem, A., Alin, F., Krajecki, M., and Steffenel, L. A. Convolutional neural networks and temporal CNNs for COVID-19 forecasting in France. *Applied Intelligence*, April 2021. ISSN 1573-7497. doi: 10.1007/s10489-021-02359-6. URL https://doi.org/10.1007/s10489-021-02359-6.

Paireau, J., Andronico, A., Hozé, N., Layan, M., Crépey, P., Roumagnac, A., Lavielle, M., Boëlle, P.-Y., and Cauchemez, S. An ensemble model based on early predictors to forecast COVID-19 health care demand in France. *Proceedings of the National Academy of Sciences*, 119 (18):e2103302119, May 2022. doi: 10.1073/pnas.2103302119. URL https://www.pnas.org/doi/full/10.1073/pnas.2103302119. Publisher: Proceedings of the National Academy of Sciences.

Pottier, L. Forecast of the covid19 epidemic in France. Technical report, April 2021. URL https://www.medrxiv.org/content/10.1101/2021.04.13.21255418v1. Company: Cold Spring Harbor Laboratory Press Distributor: Cold Spring Harbor Laboratory Press Label: Cold Spring Harbor Laboratory Press Type: article.

Rahimi, I., Chen, F., and Gandomi, A. H. A review on COVID-19 forecasting models. *Neural Computing & Applications*, pp. 1–11, February 2021. ISSN 0941-0643. doi: 10.1007/s00521-020-05626-8.

Ray, A., Chakraborty, T., and Ghosh, D. Optimized ensemble deep learning framework for scalable forecasting of dynamics containing extreme events. *Chaos (Woodbury, N.Y.)*, 31(11):111105, November 2021. ISSN 1089-7682. doi: 10.1063/5.0074213.

Schwefel H.P. Systems analysis, systems design and evolutionary strategies. *System analysis, Modeling and Simulation*, 7:853–864, 1990.

Smith, A., Lott, N., and Vose, R. The Integrated Surface Database: Recent Developments and Partnerships. *Bulletin of the American Meteorological Society*, 92(6):704–708, June 2011. doi: 10.1175/2011BAMS3015.1. URL https://journals.ametsoc.org/view/journals/bams/92/6/2011bams3015_1.xml. Publisher: American Meteorological Society Section: Bulletin of the American Meteorological Society.

Sun, C., Song, M., Hong, S., and Li, H. A Review of Designs and Applications of Echo State Networks, De-

cember 2020. URL http://arxiv.org/abs/2012.02974. arXiv:2012.02974 [cs].

Tanaka, G., Yamane, T., Héroux, J. B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., and Hirose, A. Recent advances in physical reservoir computing: A review. *Neural Networks*, 115:100–123, July 2019. ISSN 0893-6080. doi: 10.1016/j.neunet.2019.03.005. URL https://www.sciencedirect.com/science/article/pii/S0893608019300784.

Trouvain, N. and Hinaut, X. Canary Song Decoder: Transduction and Implicit Segmentation with ESNs and LTSMs. In Farkaš, I., Masulli, P., Otte, S., and Wermter, S. (eds.), *Artificial Neural Networks and Machine Learning – ICANN 2021*, Lecture Notes in Computer Science, pp. 71–82, Cham, 2021. Springer International Publishing. ISBN 978-3-030-86383-8. doi: 10.1007/978-3-030-86383-8_6.

Trouvain, N. and Hinaut, X. reservoirpy: A Simple and Flexible Reservoir Computing Tool in Python, June 2022. URL https://inria.hal.science/hal-03699931.

Vlachas, P. R., Pathak, J., Hunt, B. R., Sapsis, T. P., Girvan, M., Ott, E., and Koumoutsakos, P. Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. *Neural Networks*, 126:191–217, June 2020. ISSN 0893-6080. doi: 10.1016/j.neunet.2020.02.016. URL https://www.sciencedirect.com/science/article/pii/S0893608020300708.

Weddell, S. J., Ayyagari, S., and Jones, R. D. Reservoir computing approaches to microsleep detection. *Journal of Neural Engineering*, 18(4):046021, March 2021. ISSN 1741-2552. doi: 10.1088/1741-2552/abcb7f. URL https://dx.doi.org/10.1088/1741-2552/abcb7f. Publisher: IOP Publishing.

WHO. WHO Coronavirus (COVID-19) Dashboard, 2020. URL https://covid19.who.int.

Yang, L. and Shami, A. On hyperparameter optimization of machine learning algorithms: Theory and practice. *Neurocomputing*, 415:295–316, November 2020. ISSN 0925-2312. doi: 10.1016/j.neucom.2020.07.061. URL https://www.sciencedirect.com/science/article/pii/S0925231220311693.

Yokoyama, A. M., Ferro, M., and Schulze, B. Multi-objective hyperparameter optimization approach with genetic algorithms towards efficient and environmentally friendly machine learning. *AI Communications*, Preprint(Preprint):1–14, January 2023. ISSN 0921-7126. doi: 10.3233/AIC-230063. URL https://content.iospress.com/articles/ai-communications/aic230063. Publisher: IOS Press.

Zeng, A., Chen, M., Zhang, L., and Xu, Q. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Zhang, H. and Vargas, D. V. A Survey on Reservoir Computing and its Interdisciplinary Applications Beyond Traditional Machine Learning. *IEEE Access*, 11:81033–81070, 2023. ISSN 2169-3536. doi: 10.1109/ACCESS.2023.3299296. URL https://ieeexplore.ieee.org/abstract/document/10196105. Conference Name: IEEE Access.

A. Appendix

Algorithm 1 Genetic Algorithm

Input:
$NGeneration$ = Nb of generation
$Npop$ = Nb of individual in the population
$Nchildren$ = Nb of children by generation
$Ntour$ = Nb of individual selected for tournament
$pmutNum$ = Prob of mutation of numeric hp
$pmutCat$ = Prob of mutation of categorical hp
$sigma$ = Scaling factor of mutation of numeric hp

– Initialize algorithm –
Draw first generation of $Npop$ individual with random search

– Iterate over each generation –
for $Gen = 2$ **to** $NGeneration$ **do**
  for $i = 1$ to $Nchildren$ **do**
    – Select parents by tournament –
    $parent_1$ = Sample $Ntournament$ individuals from best $Npop$ individuals of previous generation, take the best of them
    $parent_2$ = same as $parent_1$
    – Cross-over and mutation –
    for $param$ **in** $parameterToOptimize$ **do**
      Draw alp from $\mathcal{U}[0, 1]$
      Draw $mutation$ from $\mathcal{U}[0, 1]$

      if $param$ is $numeric$ **then**
        $child_i(param) = alp * parent_1(param) + (1 - alp) * parent_2(param)$
        if $mutation < pmutNum$ **then**
          $child_i(param) = child_i(param) + sigma * \mathcal{N}(0, 1)$
        else
          Do nothing
        end if

      else if $param$ is $categorical$ **then**
        if $alp > 0.5$ **then**
          $child_i(param) = parent_1(param)$
        else
          $child_i(param) = parent_2(param)$
        end if
        if $mutation < pmutCat$ **then**
          $child_i(param) =$ random modality
        else
          Do nothing
        end if
      end if
    end for
  end for
end for

We evaluated how many reservoir should be aggregated in order to provide stable MAE. Figure 4 shows that after 5 reservoirs, MAE is greatly stabilized and does not evolve much after 10 reservoir are aggregated.

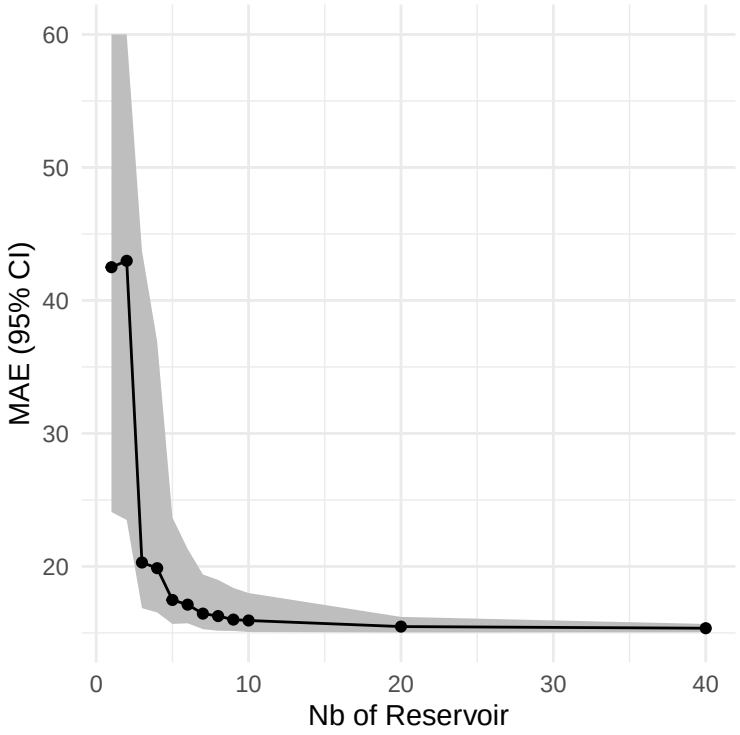

Figure 4. MAE with 250 bootstrap 95% confidence interval of RC-GA model without monthly update. For better visualization, error above 60 were set to 60. After the aggregation of 5 reservoirs, MAE is mostly stable.

Table 3. Model performance and standard deviation. Best model without monthly update of hyperparameters is outlined in bold. Best model with monthly update is outlined in bold and italic.

Model	Update	MAE (± std)	MAEB (± std)	MRE (± std)	MREB (± std)
RC-GA[1]	No	15.27 (± 12.88)	-3.31 (± 6.06)	0.24 (± 0.33)	0.84 (± 1.67)
RC-(GA-500)-2k	No	**14.66 (± 12.63)**	**-3.92 (± 7.70)**	0.25 (± 0.31)	0.83 (± 1.46)
RC-GA (20 HP)	No	14.84 (± 12.53)	-3.75 (± 9.60)	0.25 (± 0.32)	0.81 (± 1.74)
RC-GA	Yes	15.79 (± 13.08)	-2.80 (± 5.23)	***0.27 (± 0.37***	0.90 (± 1.11)
RC-GA (20 HP)	Yes	***14.70 (± 12.22)***	***-3.88 (± 9.66)***	0.27 (± 0.30)	***0.81 (± 1.85)***
RC-RS	No	18.45 (± 13.27)	-0.14 (± 1.13)	0.36 (± 0.38)	1.00 (± 0.31)
RC-RS	Yes	17.81 (± 13.34)	-0.77 (± 2.50)	0.31 (± 0.38)	0.98 (± 0.62)
RC-GA (no FS)	No	17.59 (± 13.35)	-0.99 (± 2.82)	0.31 (± 0.38)	0.96 (± 0.70)
RC-GA (no FS)	Yes	16.95 (± 13.50)	-1.64 (± 4.50)	0.29 (± 0.38)	0.97 (± 1.53)
RC-PCA-GA	No	15.59 (± 12.90)	-3.00 (± 7.70)	0.28 (± 0.33)	0.87 (± 1.76)
RC-PCA-GA	Yes	15.63 (± 12.67)	-2.95 (± 6.37)	0.29 (± 0.35)	0.85 (± 1.68)
Enet-RS	No	15.83 (± 12.41)	-2.76 (± 7.10)	0.29 (± 0.34)	0.86 (± 1.65)
Enet-RS	Yes	16.22 (± 12.54)	-2.36 (± 8.50)	0.30 (± 0.38)	0.88 (± 1.29)
XGB-RS	No	15.45 (± 13.91)	-3.14 (± 9.56)	0.28 (± 0.36)	0.81 (± 2.10)
XGB-RS	Yes	16.32 (± 14.01)	-2.27 (± 9.89)	0.29 (± 0.46)	0.89 (± 2.40)
Transformers-PCA	No	19.28 (± 16.80)	0.69 (± 11.64)	0.32 (± 0.51)	1.00 (± 2.79)
Transformers-RC-GA[2]	No	18.12 (± 15.43)	-0.47 (± 9.33)	0.31 (± 0.44)	0.97 (± 4.02)
Informer-PCA	No	18.70 (± 14.11)	0.12 (± 4.25)	0.34 (± 0.38)	1.00 (± 1.69)
PatchTST-PCA	No	18.34 (± 18.14)	-0.25 (± 13.95)	0.31 (± 0.55)	1.00 (± 3.34)
LSTM-PCA	No	15.74 (± 13.98)	-2.84 (± 11.80)	0.24 (± 0.50)	0.76 (± 3.32)
20 LSTM-PCA[3]	No	16.23 (± 14.37)	-2.35 (± 12.38)	0.25 (± 0.52)	**0.74 (± 3.30)**
LSTM-RC-GA	No	15.08 (± 12.71)	-3.51 (± 9.30)	**0.23 (± 0.48)**	0.80 (± 1.76)
Prophet	No	21.30 (± 16.33)	2.71(± 8.07)	0.39(± 0.35)	1.05 (± 3.15)
RC-GA - dep[4]	No	15.67 (± 13.00)	-2.92 (± 6.19)	0.27 (± 0.36)	0.90 (± 1.51)
RC-GA - weather	No	15.93 (± 12.74)	-2.66 (± 5.73)	0.27 (± 0.35)	0.88 (± 1.43)
RC-GA - emergency	No	15.62 (± 13.10)	-2.97 (± 6.00)	0.28 (± 0.33)	0.85 (± 1.20)
RC-GA - deriv	No	18.74 (± 13.38)	0.15 (± 0.65)	0.36 (± 0.39)	1.01 (± 0.13)
(7 days) RC-GA[5]	No	8.59 (± 7.42)	-2.42 (± 5.87)	0.16 (± 0.19	0.80 (± 4.82)
(7 days) XGB-RS	No	8.96 (± 7.69)	-2.04 (± 6.33)	0.14 (± 0.22)	0.80 (± 5.20)
(21 days) RC-GA[6]	No	25.46 (± 16.20)	0.12 (± 2.32)	0.52 (± 0.50)	1.00 (± 2.93)
(21 days) XGB-RS	No	25.85 (± 22.65)	0.51 (± 23.37)	0.47 (± 1.13)	0.87 (± NaN[7])

[1] RC-GA : Reservoir computing with genetic algorithm. RC-(GA-500)-2k, RC-GA (20 HP), RC-RS, RC-GA (no FS), RC-PCA-GA are similar to RC-GA but with 2000 neurons instead of 500, 20 reservoir for hyperparameter optimization instead of 3, random search instead of genetic algorithm and PCA instead of feature selection respectively.

[2] Transformer-RC-GA and LSTM-RC-GA corresponds to both algorithms trained with the features selected by RC-GA.

[3] 20 LSTM-PCA is similar to LSTM-PCA but takes the median forecast of 20 LSTM instead of 1.

[4] Similar to RC-GA but with department, weather, emergency units or derivatives data removed.

[5] 7 days forecast.

[6] 21 days forecast.

[7] Because relative error to baseline was equal to infinity for some observations, standard deviation could not be computed.

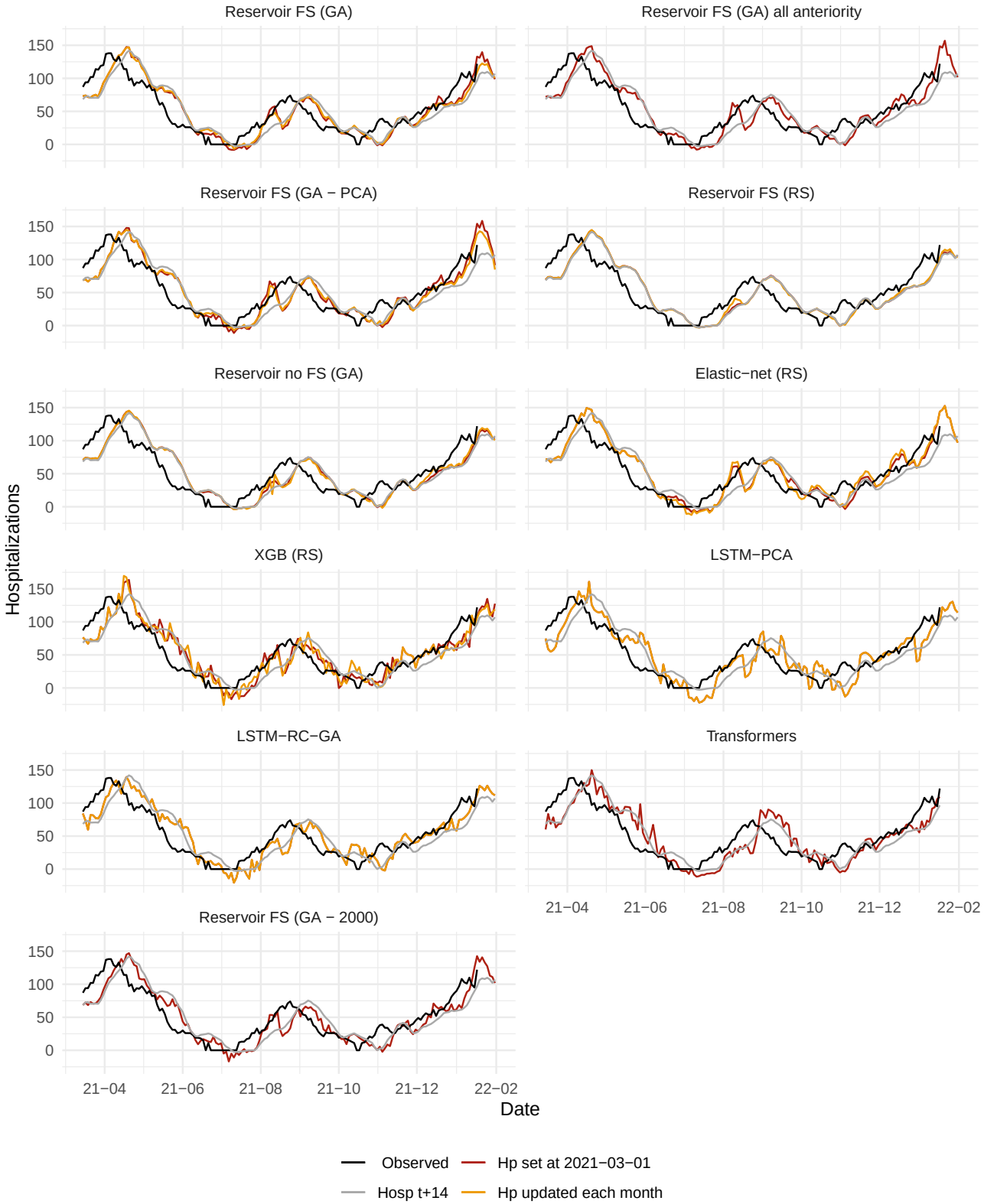

Figure 5. Prediction of all models depending if hyperparameters where updated each month or not.

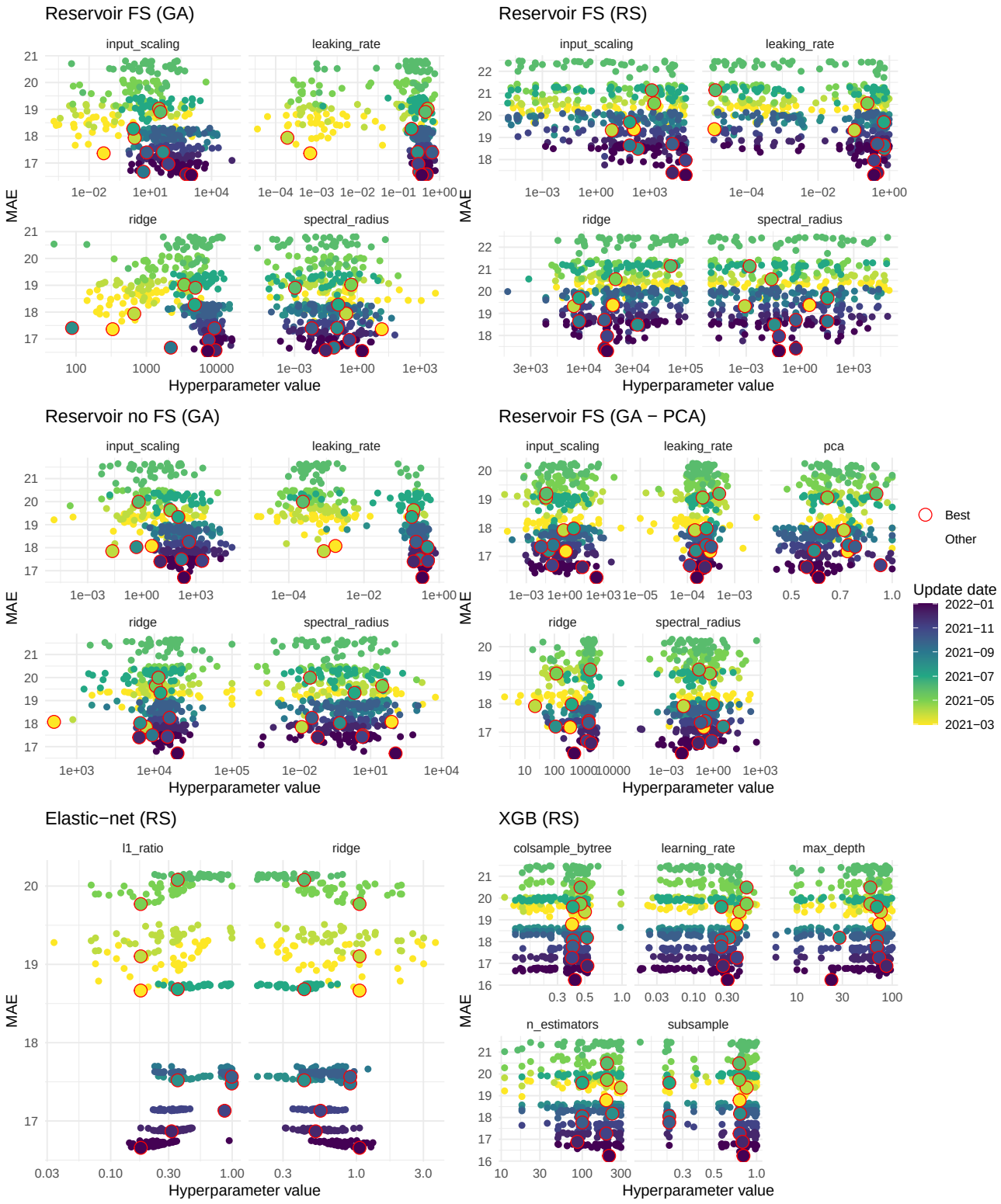

Figure 6. Hyperparameter optimisation of the different algorithms. Only best 40 hyperparameters sets according to Mean Absolute Error (MAE) are shown by month.

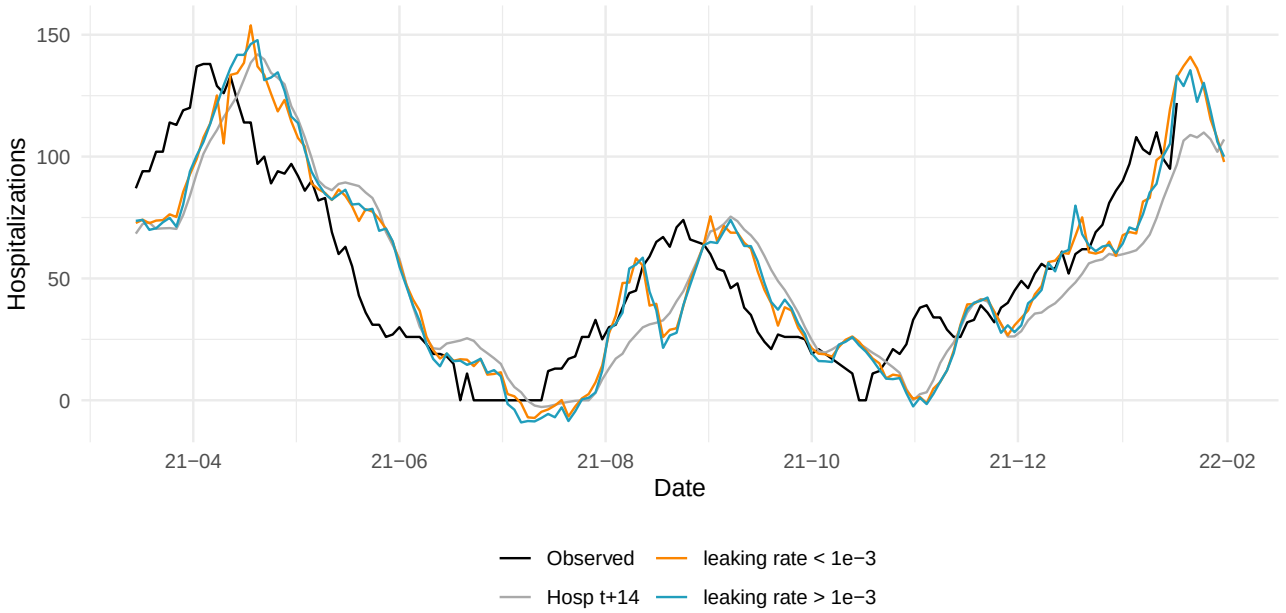

Figure 7. RC-GA without monthly update forecast depending on leaking rate.

Table 4. Model performance of RC-GA without monthly hyperparameter update.

model	update	leaking_rate	nb models	MAE	MAEB	MRE	MREB
Reservoir FS (GA)	No	$<$ 1e-3	14	15.07	-3.52	0.26	0.82
Reservoir FS (GA)	No	$>$ 1e-3	26	15.52	-3.07	0.26	0.85

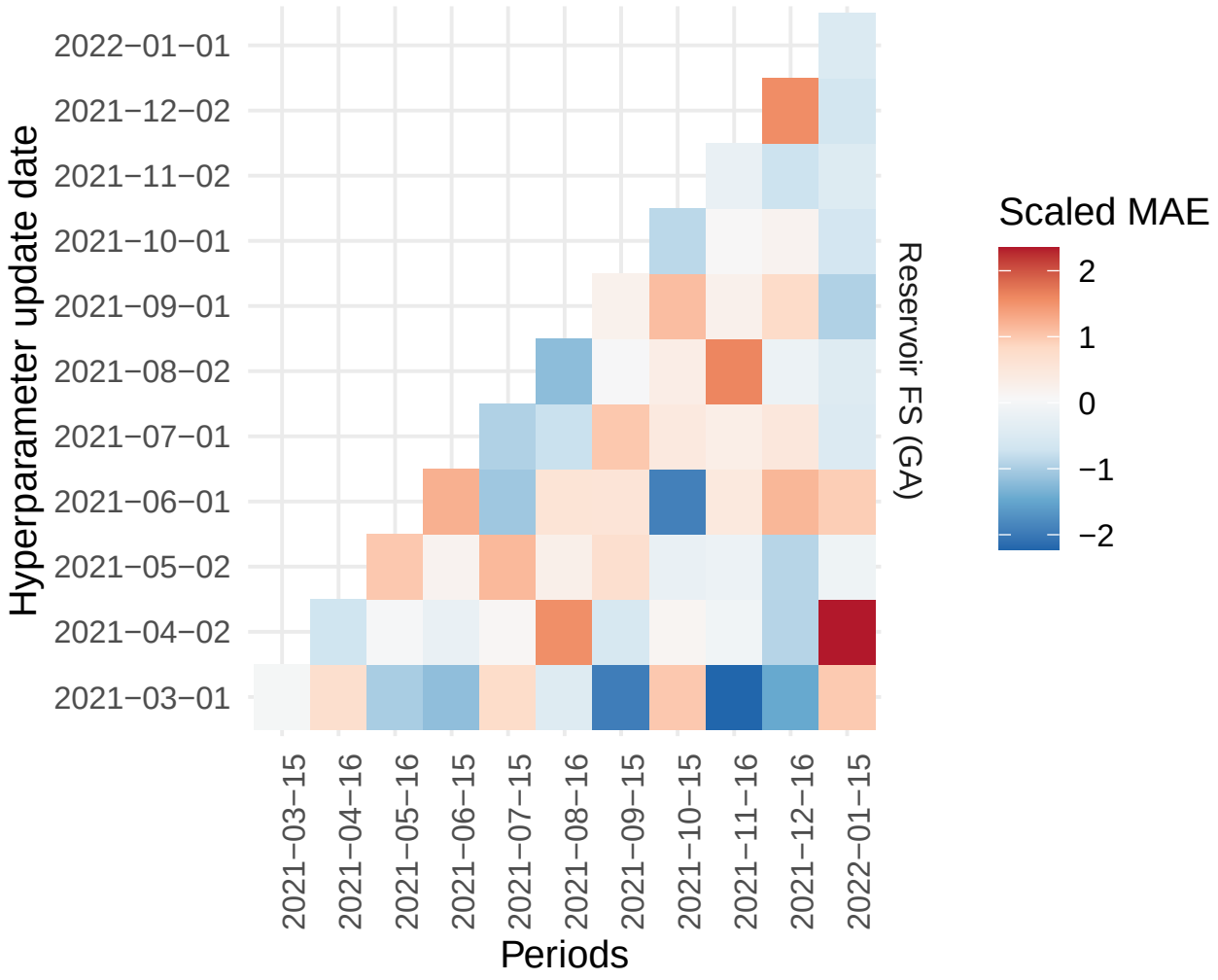

Figure 8. Performance of RC-GA by hyperparameter date update and by period. To facilitate visualization, the Mean Absolute Error (MAE) was normalized by scaling (subtracting the mean and dividing by the standard deviation) for each respective period.

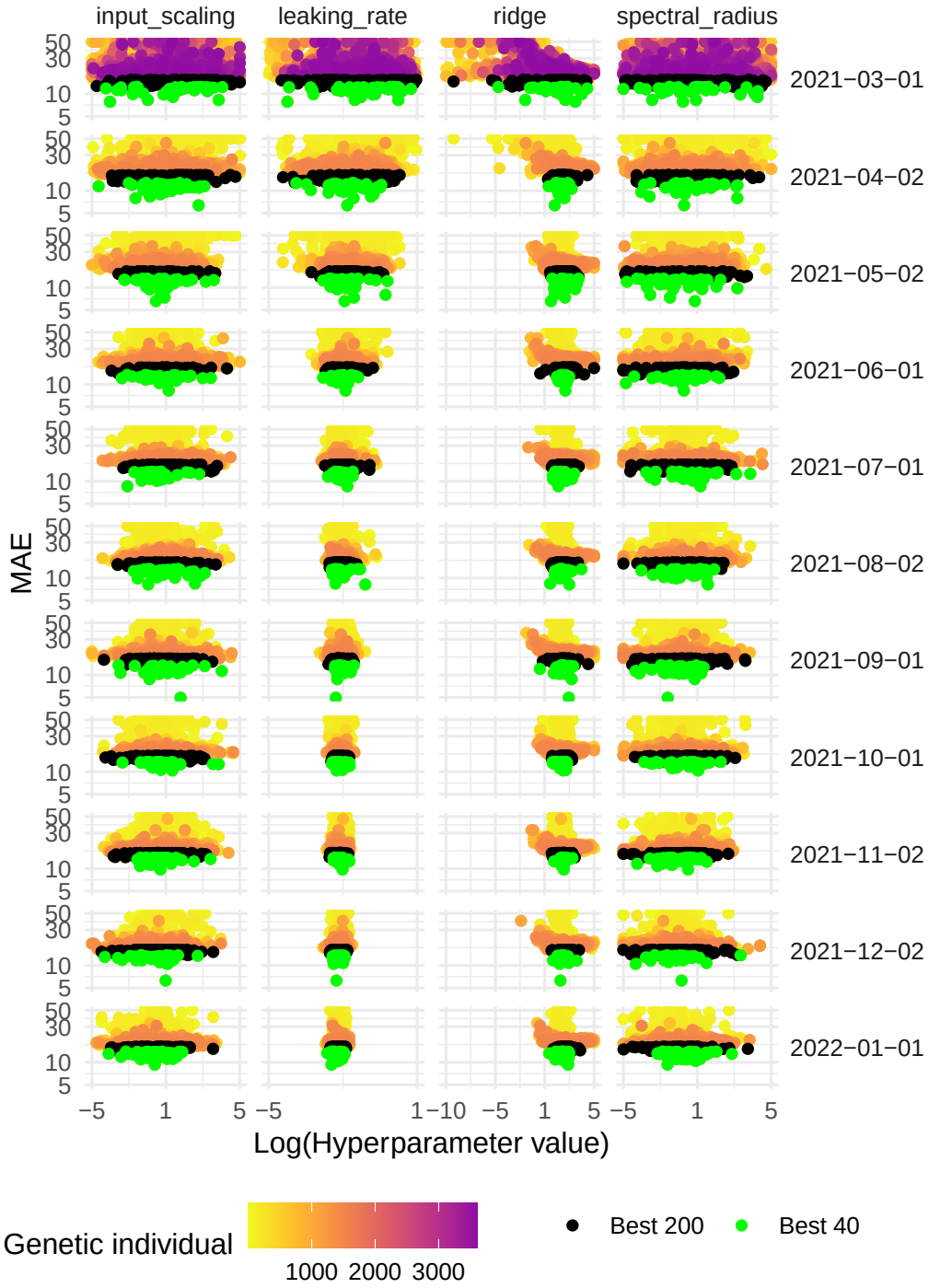

Figure 9. RC-GA with 20 reservoirs for hyperparameter optimization.

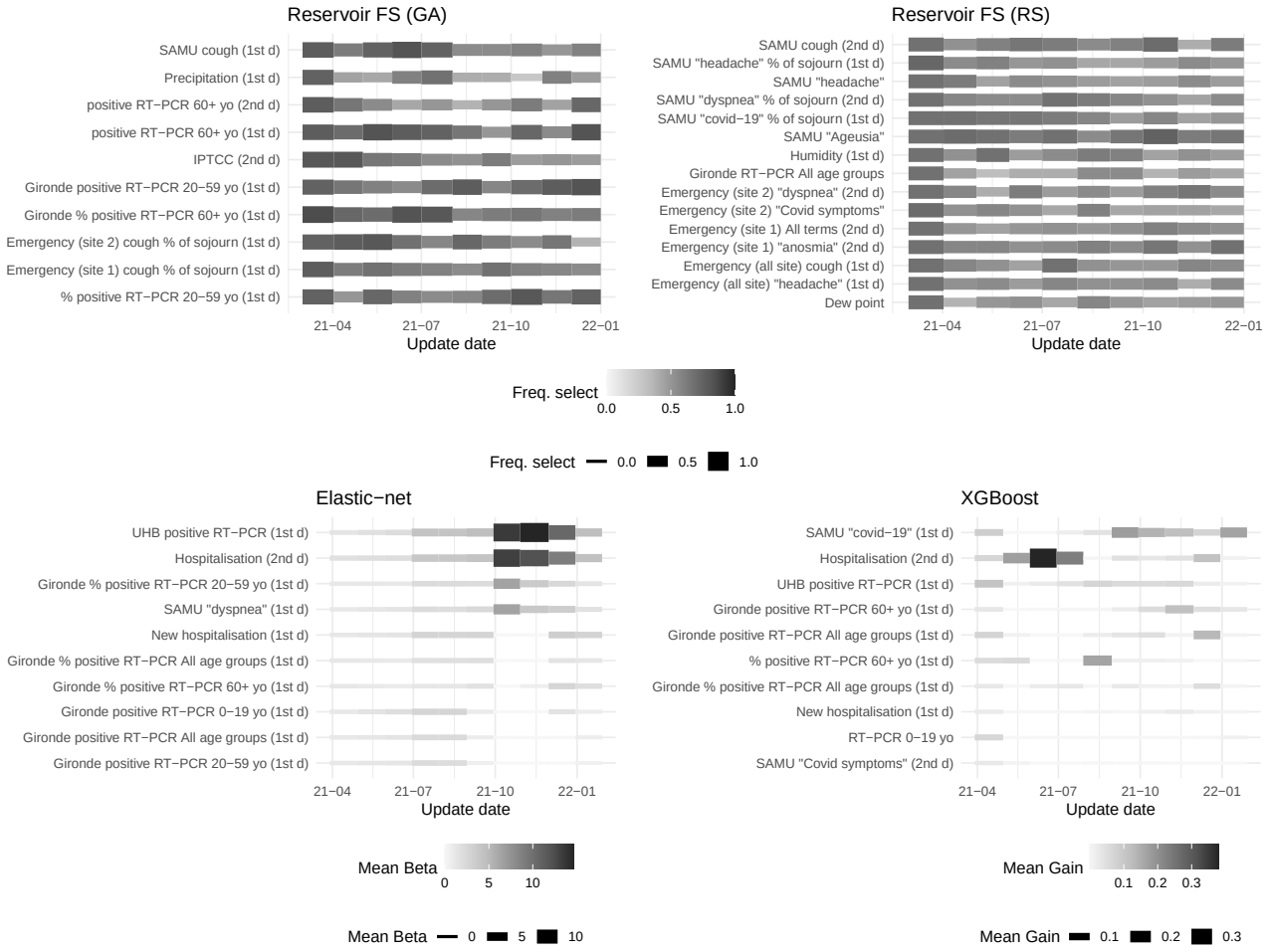

Figure 10. Feature importance of the 10 most important features according to first month (i.e 2021-03-01). Importance of Reservoir is given by the frequency of feature selection. Importance according to Elastic-net is given by the mean absolute value of the feature coefficient. Importance according to XGBoost is given by the mean gain of the feature. By construction, only 10 most important features at 2021-03-01 are shown, therefore initial importance is high.

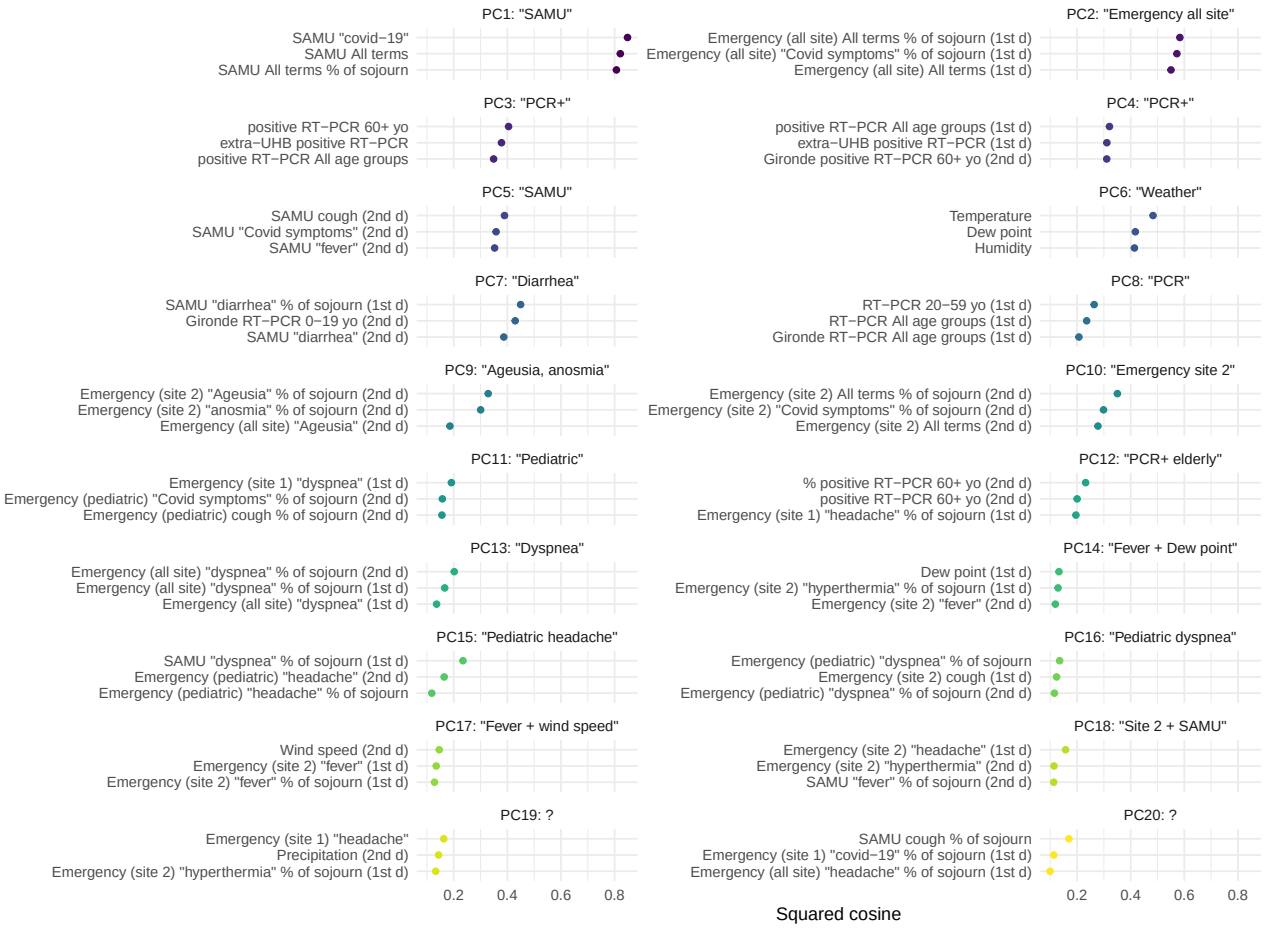

Figure 11. Principal component analysis of the dataset at 2022-01-17. First 20 components are shown (73% of variance explained). For each component, we outlined the three features with the greatest squared cosine and labeled the component with its corresponding epidemiology representation. We observe that first components are homogeneous in terms of represented features which is not the case after the 14th principal component.