Optimizing Reservoir Computing with Genetic Algorithm for High-Dimensional SARS-CoV-2 Hospitalization Forecasting: Impacts of Genetic Algorithm Hyperparameters on Feature Selection and Reservoir Computing Hyperparameter Tuning

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

Optimizing Reservoir Computing with Genetic Algorithm for High-Dimensional SARS-CoV-2 Hospitalization Forecasting: Impacts of Genetic Algorithm Hyperparameters on Feature Selection and Reservoir Computing Hyperparameter Tuning

Thomas Ferté[1,2,3][0000−0001−8455−4665], Dan Dutartre[3][0000−0001−8210−461X], Boris Hejblum[2,3][0000−0003−0646−452X], Romain Griffier[1,4][0000−0002−1096−137X], Vianney Jouhet[1,4][0000−0001−5272−2265], Rodolphe Thiébaut[1,2,3][0000−0002−5235−3962], Xavier Hinaut[3,5,6][0000−0002−1924−1184], and Pierrick Legrand ✉[7,8]

[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] LaBRI, Univ. Bordeaux, Bordeaux INP, CNRS UMR 5800
[6] Univ. Bordeaux, CNRS, IMN, UMR 5293, Bordeaux, France
[7] ASTRAL, Centre Inria de l'université de Bordeaux
[8] IMB, Institut de Mathématiques de Bordeaux, UMR CNRS 5251
pierrick.legrand@u-bordeaux.fr

Abstract. This study focuses on forecasting SARS-CoV-2 hospitalizations 14 days ahead at Bordeaux University Hospital using a high-dimensional dataset with 409 predictors and 586 observations, combining public data and electronic health records. Previous research showed that integrating reservoir computing (RC) with genetic algorithm (GA) for hyperparameter optimization and feature selection outperformed state-of-the-art methods. However, the behavior of RC-GA under high-dimensional conditions is not well understood. This work examines the impact of GA hyperparameters (GA-HP), specifically the mutation probability of feature selection and the extent of mutation on a critical RC hyperparameter (RC-HP), the leaking rate.

GA-HP significantly influence feature selection and RC-HP optimization. Higher mutation rates led to a more diverse set of selected features and fewer total features, resulting in increased mean absolute error (MAE) on the training set but comparable MAE on the test set, suggesting reduced overfitting. Conversely, lower mutation rates of categorical genes and leaking rates correlated with slightly poorer performance, indicating potential lack of exploration during RC-HP selection. Notably, a bimodal

convergence of the leaking rate was observed, with lower leaking rates enhancing training performance but slightly diminishing test performance. The RC-GA approach demonstrated robust behavior with higher leaking rates, possibly due to increased regularization effects from higher ridge values.
Optimizing RC-GA in a high-dimensional setting remains challenging. Current practices using the median forecast of the top 40 RC-HP sets might not be optimal. Enhancing GA for high-dimensional contexts and exploring complementary RC-HP sets instead of selecting the top 40 might further improve forecasting accuracy.

Keywords: Genetic algorithm · Reservoir Computing · High dimension · Covid-19 · Electronic Health Records · Time series

1 Introduction

Since late 2020, the SARS-CoV-2 pandemic has exerted substantial pressure on healthcare systems, necessitating modifications in the organization of care and the implementation of stringent measures such as lock-downs [26]. To better manage and allocate healthcare resources, particularly in anticipating the need for hospitalization, several short-term forecast models have been developed [4, 18, 16, 17, 3, 15, 8, 6].
Most previous studies used public data [4, 18, 16, 17, 3, 15]. However, our recent work showed that adding hospital Electronic Health Records (EHRs) significantly improves forecast accuracy due to the richness and faster updates of EHR data [8, 6]. However, EHRs create a high-dimensional problem, with hundreds of features extracted from the records.
In previous work, we employed several algorithms to forecast hospitalizations at 14 days using both public and EHR data. Among the methods tested, Reservoir Computing combined with a Genetic Algorithm (RC-GA) outperformed other methods, including elastic-net penalized linear regression, XGBoost, LSTM, Transformers, Prophet, PatchTST, and Informer [6]. RC forecasting relies on projecting the input features onto a high-dimensional reservoir where connections are set at random [11, 14, 13]. The neurons inside the reservoir are then used as features by a linear layer, which is trained to forecast the outcome of interest. Previous RC studies have mostly focused on a low-dimension settings (i.e. with less than 50 input variables), because in general RC is used for features expansion as the projection in high-dimension obtained with Support Vector Machines (SVMs). This small input dimension context includes literature in epidemic forecasting [9, 19, 12]. In the context of microsleep and weather forecasting, previous work used more than 500 input features but with more than 10^4 observations [1, 25]. Furthermore, while metaheuristic approaches for RC hyperparameters (RC-HP) optimization are not new, they have rarely been used in high-dimension settings except in our previous work [2, 6].
In our previous study, we found that in this high-dimension context, using RC alone resulted in medium performance, but combining RC with GA for RC-HP optimization and feature selection significantly improved the performance

[6]. However, little is known about the behavior of RC-GA in this context and the influence of GA-HP. In this work, we propose to explore the robustness of RC-GA performance depending on GA-HP for forecasting the number of SARS-CoV-2 hospitalizations 14 days ahead at Bordeaux University Hospital, France. Specifically, we aim to investigate the influence of GA-HP when RC-HP are selected once for the entire time series versus when they are updated monthly. We also explored modifications to the GA to achieve sparser feature selection.

2 Methods

2.1 Data

A detailed description of the data is available in [8]. The objective was to forecast the number of SARS-CoV-2 hospitalized patients 14 days in advance (h_{t+14}) at Bordeaux University Hospital (BUH). To achieve this, we utilized a diverse ensemble of data sources: department-level COVID-19 data from Santé Publique France [5], department-level weather data from the National Oceanic and Atmospheric Administration (NOAA) [20], and hospital-level COVID-19 data from BUH's EHRs [7]. The study used aggregated data spanning from May 16, 2020, to January 17, 2022. This created a high-dimensional situation with a total of 409 predictors over 586 days, with each observation representing a single day.
Similarly to [6], primary outcome was the mean absolute error (MAE) which we defined as : $MAE = mean(|\widehat{h_{t+14}} - h_{t+14}|)$. Similarly to [6], the MAE was chosen as the primary outcome because it is more sensitive to errors when the number of hospitalizations is high, a situation of great concern since this is when the hospital is under the most strain and decision-making is most critical.
Data preprocessing included computing the first and second derivatives over the last 7 days to enrich the model with additional information. Additionally, the data were smoothed using a local polynomial regression with a span of 21 days to account for daily noise variation, as suggested to improve performance by [8]. Finally, features were scaled between -1 and 1 using maximum absolute scaling, which involves dividing each observed value by the maximum absolute value of the respective feature.
All data utilized in this study are publicly available in the Github repository of previous work `https://github.com/thomasferte/Ferte2024ICML_HighDimensionReservoir` [6]. 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.

2.2 Evaluation framework

The period from May 16, 2020, to March 1, 2021, was used solely for identifying relevant hyperparameters. Similarly to [6], two different approaches were evaluated. First, the hyperparameters were set during the initial period and not

updated afterward. Second, the hyperparameters were updated monthly. This approach aims to specifically change the features selected by the algorithm over time to better adapt to the evolution of the epidemic.
Models were trained and evaluated every other day to ensure the use of the most up-to-date data and to limit computation time. We utilized all data available from the previous year and, as suggested by [6], limiting the learning to a sliding window of one year.
The model was trained to forecast the change in hospitalizations rather than the raw number of hospitalizations, as this task proved easier for machine learning approaches (i.e., $outcome_{t+14} = h_{t+14} - h_t$). However, all metrics and graphical representations will be shown on the raw hospitalization scale.

2.3 Reservoir Computing

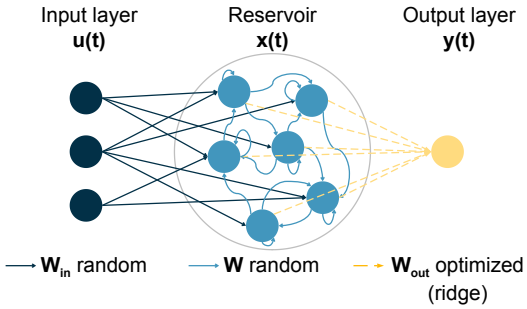

Fig. 1. Reservoir computing is composed of an input layer, a reservoir and an output layer. Connections between input layer and reservoir and inside reservoir are random. Only the output layer is optimized based on a ridge regression. Adapted from [23]

Originally developed by [11] and [14], Reservoir Computing (RC) is a distinctive machine learning paradigm designed to analyze information from dynamic systems [13]. RC consists of a recurrent neural network (RNN) with random internal connections, called *reservoir*, where only the output layer, or *read-out*, undergoes training. This method offers performance comparable to traditional RNNs but with reduced computing time [24]. RC has been successfully applied to various tasks, including analyzing bird songs [22], language processing [10], epidemics [9, 19, 12], power plant monitoring, internet traffic, and stock price prediction [13, 21, 27].
Figure 1 illustrates the main principles of RC. The input layer (dark blue) is randomly projected in a non-linear fashion into a high-dimensional space, the reservoir (light blue). This reservoir allows for non-linear and temporal interactions of input data. Then, the "useful" computations are extracted from the reservoir by a linear output layer (yellow). Additionally, it is possible to directly

connect the input layer to the output layer, allowing the output layer to leverage both non-linear and temporal interactions within the reservoir and linear instantaneous input interaction.
RC relies on several hyperparameters which influence the internal dynamics of the reservoir. We used a similar setting as in [6]. Namely, we used a reservoir of 500 units where both the input layer and the reservoir are connected to the output layer. We focused on the optimization of the same hyperparameters : (1) the leaking rate, which represents the inertia of the reservoir by balancing new inputs and the previous state – a leaking rate of 1 means that reservoir units are only influenced by the current input and the current state of their recurrent neighbors, and not by their own past states –; (2) the spectral radius, which is the maximum absolute eigenvalue of the reservoir connectivity matrix, where a small spectral radius results in stable dynamics within the reservoir, while a high spectral radius leads to a chaotic regime; (3) the input scaling, which is a gain applied to the input features of the reservoir to adjust their influence; and (4) ridge parameter of the regression, which refers to the regularization penalty applied to the output layer to limit overfitting.
Due to the inherent randomness of RC, it is interesting to aggregate the outcomes of multiple reservoirs. Following a similar approach to [6], the forecasting was conducted using 40 RC instances, each representing the top 40 hyperparameter sets selected by GA. To limit computation time, hyperparameter optimization was conducted using the median forecast of three RC instances with the same hyperparameters (i.e., selected features and hyperparameters are the same, only random connections inside the reservoir change).

2.4 Genetic algorithm

We used the same algorithm as the one proposed by [6] depicted in their supplementary algorithm. In this GA, each individual genotype corresponds to 413 genes: 4 numeric genes corresponding to reservoir structure (i.e., leaking rate, spectral radius, ridge, and input scaling) and 409 binary genes, one for each feature to specify if it is used or not for forecasting. The fitness function to optimize is the MAE for the period from May 16, 2020, to March 1, 2021, as specified in section 2.2. We employed a GA with barycentric crossover, tournament selection, and Gaussian mutation, optimizing the MAE as the objective function. The algorithm is initialized by randomly drawing initial individuals similarly to random search. For each individual, two parents are selected through tournament selection of two individuals 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 and random choice for categorical hyperparameters). We used a total of 30 generations for initial optimization and 10 generations for the monthly update, with a population of 200 and 100 children generated at each generation, respectively. For the first hyperparameter optimization, the first 200 individuals are sampled by random search. For the following month, the first 200 individuals are the best 200 individuals from the previous month, evaluated

on the new period. Mutation of categorical genes was performed with a mutation probability denoted as $pmutCat$. Mutation of numeric genes was performed with a mutation probability of 0.5 and with a Gaussian distribution of variance denoted as $\sigma_{input\_scaling} = \sigma_{spectral\_radius} = \sigma_{ridge} = 1$ and a variance for the leaking rate denoted as $\sigma_{leaking\_rate}$.
We explored the influence of $pmutCat$ and $\sigma_{leaking\_rate}$ (i.e the σ for leaking rate hyperparameter) for several reasons. First, [6] showed no benefit of monthly updating hyperparameters. However, the dynamic of the epidemic changes over time and it has been shown that linear regression with elastic-net penalization leverage different features over time [8] which should also be expected for RC. Second, [6] found convergence of the GA to different leaking rate (i.e the hyperparameter determining the inertia of RC) depending on the number of aggregated RC for hyperparameter optimization. It is possible that different settings of hyperparameters of GA might converge to different hyperparameters values and that some might be stuck in a local optimum.
We explored the influence of $pmutCat$ and $\sigma_{leaking\_rate}$ with $pmutCat \in \{0.016, 0.04, 0.1, 0.25\}$ and $\sigma_{leaking\_rate} \in \{0.1, 0.2, 0.4, 0.8\}$. The influence of those GA-HP is evaluated according to the selected hyperparameter values, the performance on the train set and the performance on the test set.
In addition, we explored three approaches to reduce the number of features selected by the genetic algorithm (GA) to obtain a sparser model. (i) We modified the fitness function from MAE to $MAE + \theta \times$ number of selected features, where $\theta \in \{1 \times 10^{-3}, 5 \times 10^{-3}, 1 \times 10^{-2}\}$. (ii) We adjusted the tournament selection process by using a tournament with three individuals (instead of two), excluding the individual with the highest number of features, and selecting the one with the lowest MAE. (iii) We altered the mutation process by reducing the probability of selecting a feature compared to not selecting it, setting the probability to $\frac{\text{mean number of features of parents}}{2 \times \text{total number of features}}$, instead of using a 50/50 chance.
The hyperparameter distribution of RC was set as follows: the leaking rate was within the range of $[1 \times 10^{-5}; 1]$, the spectral radius was between $[1 \times 10^{-5}; 1 \times 10^{5}]$, the input scaling ranged from $[1 \times 10^{-5}; 1 \times 10^{5}]$, and the ridge parameter spanned $[1 \times 10^{-10}; 1 \times 10^{5}]$. Additionally, feature selection involved a binary indicator for each of the 409 features, specifying whether each feature was selected or not. Therefore, each individual of the genetic population has 413 genes. Notably, leaking rate, spectral radius, input scaling and ridge were log-transformed before processing by GA to ensure a more efficient sampling process.
Code and supplementary results are available at https://github.com/thomasferte/-ChallengesRCGA.

3 Results

3.1 Selected hyperparameters

Panel A of figure 2 presents the distribution of RC-HP for the top 40 hyperparameter sets, categorized by GA-HP and the month of hyperparameter updates. Overall, we observed that input scaling varied between 1 and 1e5 without

a clear optimum. One exception was with a $\sigma_{leaking\_rate}$ of 0.2, where input scaling ranged between 1e-2 and 1e2. The ridge hyperparameter remained relatively consistent across different GA-HP, showing an optimum around 1e3, but it tended to converge to a smaller value during the first month. The spectral radius displayed different optima depending on the leaking rate: it tended to approach 1 when the leaking rate was low (around 1e-4), and it tended to be less than 1 when the leaking rate was high (around 1). Lastly, the leaking rate itself showed two optima in the first month, at 1e-4 and 1. After May 2021, most RC-GA runs converged to a leaking rate close to 1. One exception was observed with a $\sigma_{leaking\_rate}$ of 0.2 and a $pmutCat$ of 0.016.
The number of features selected by RC-GA remains within the same range over time and GA-HP, averaging around 205 features. In panel B of figure 2, we observe that the number of selected features remained mostly consistent after the first month. However, after several updates, RC-GA with the highest $pmutCat$ tended to select fewer features. This trend is consistent with the distribution of feature selection frequency illustrated in panel C of figure 2. During the first month, all RC-GA runs showed a feature selection distribution centered around 0.5 with low variance, indicating that GA assigned similar weights to all features. This behavior persisted over several months when $pmutCat$ was set to 0.25. However, for lower $pmutCat$ values, GA began to discriminate more between features, as indicated by the higher variance in the frequency distribution.

3.2 Performance on train set

Performance on the train set is illustrated in panel A and B in figure 3. The first panel of this figure outlines that RC-GA with higher $pmutCat$ tended to perform worse on the train set compared to others. This is expected, as these configurations are more likely to explore a larger hyperparameter space for the selected feature, which tends to be less optimal for the train set. This observation is also consistent with panel C of figure 2, which shows that higher values of $pmutCat$ tend to be less stringent in feature selection (i.e., most features have approximately a 50% chance of being selected).
Panel B of figure 3 indicates that $\sigma_{leaking\_rate}$ has little impact on performance on the train set. There is one exception: when both $\sigma_{leaking\_rate} = 0.1$ and $pmutCat = 0.016$, there is less mutation and, therefore, less exploration by the genetic algorithm, which appears to prevent finding optimal hyperparameters.

3.3 Performance on test set

Panel C of figure 3 and Table 1 outline the performance on the test set. First, we note that having both a small $\sigma_{leaking\_rate}$ and $pmutCat$ led slightly poorer performance, which might be explained by the lack of algorithmic exploration, as shown at panels A and B in figure 3. Aside from this, the performance of the different RC-GA configurations was similar. Higher values of $\sigma_{leaking\_rate}$ tended to improve performance. Although $pmutCat = 0.25$ did not select features stringently, it demonstrated robust performance even with monthly updates. This

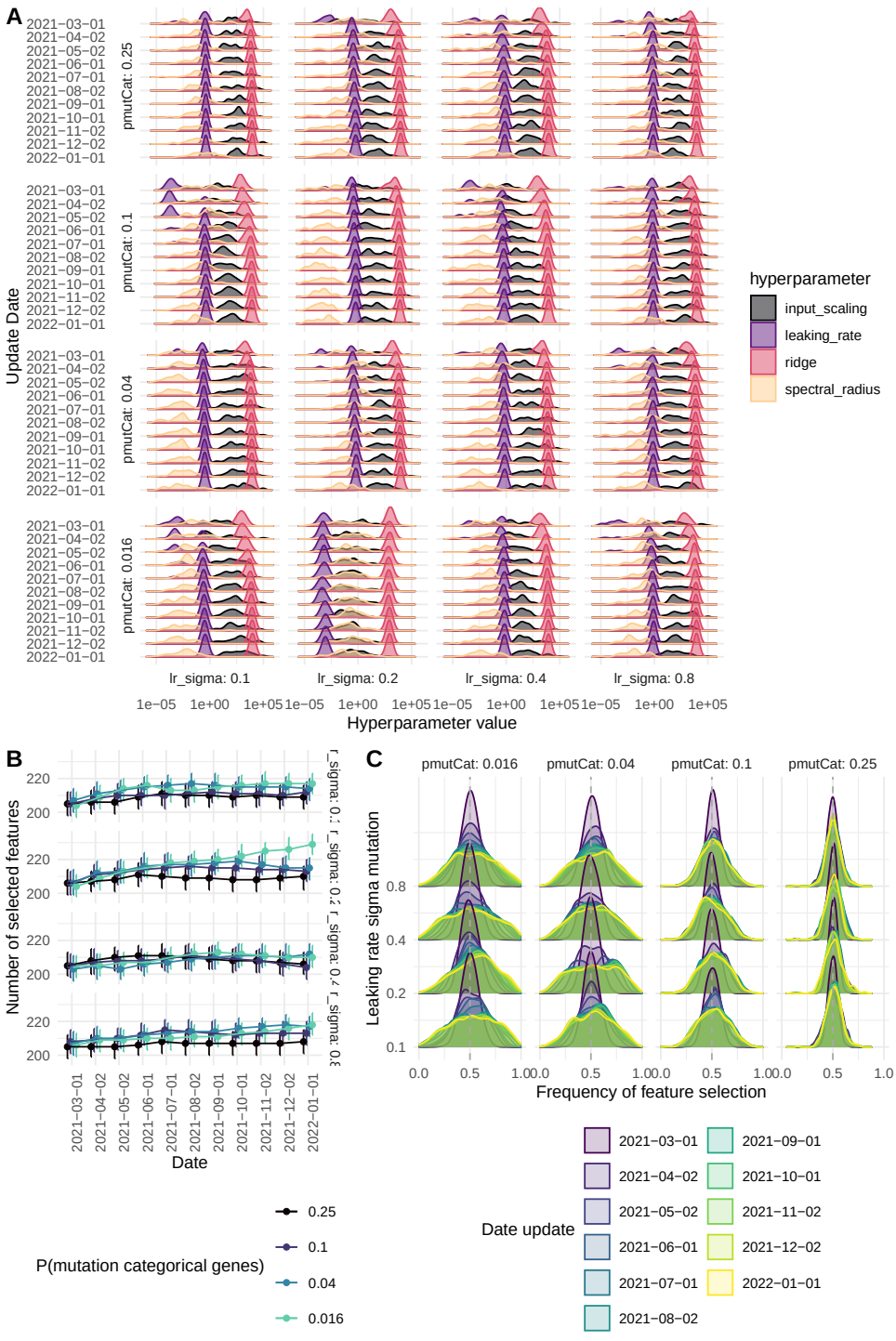

Fig. 2. Hyperparameter selection among the top 40 hyperparameter sets depending on GA-HP by month. Panel A outlines RC-HP distribution. Panel B outlines the number of selected features. Vertical bars represent the 25th and 75th percentiles. Panel C outlines the distribution of frequency of feature selection.

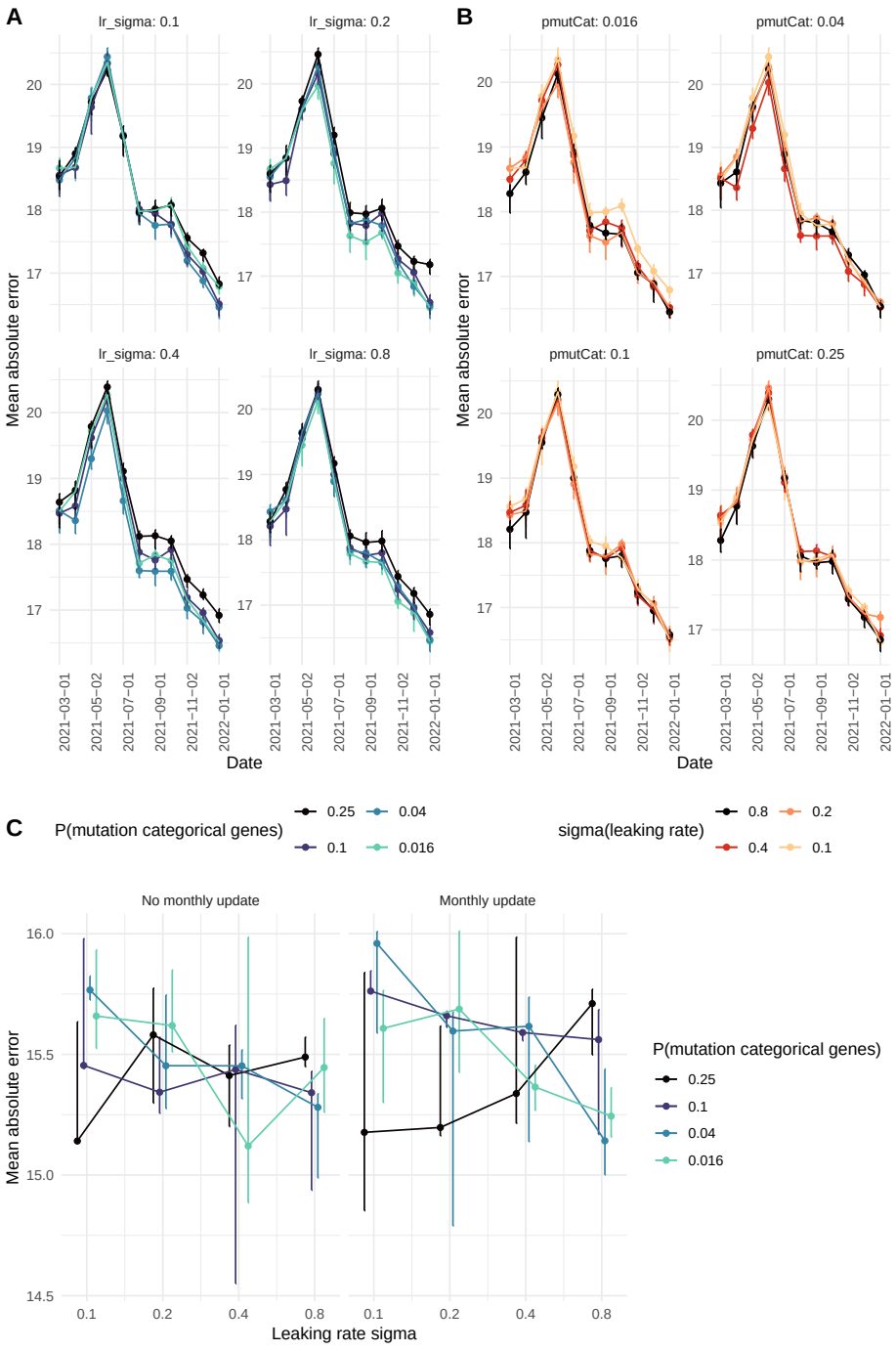

Fig. 3. Performance of RC-GA depending on GA-HP. Panel A and B outline the performance on the train set. Vertical bar indicates 25th and 75th percentiles. Panel C outlines the performance on the test set when hyperparameter are monthly updated or not. For panel C, the experiment was repeated three times. The points and vertical bars represent the median, minimum, and maximum MAE, respectively, across the three experiments.

Table 1. Model performance and standard deviation. Performance is represented by the median and the range of MAE over three repetitions of the experiment. Best model without monthly update of hyperparameters is outlined in bold. Best model with monthly update is outlined in bold and italic. MAE = Mean Absolute Error.

pmutCat	$\sigma_{leaking\_rate}$	No monthly update MAE (range)	Monthly update MAE (range)
0.016	0.1	15.66 (15.52 ; 15.93)	15.61 (15.3 ; 15.77)
0.016	0.2	15.62 (15.51 ; 15.85)	15.69 (15.43 ; 16.01)
0.016	0.4	**15.12 (14.89 ; 15.99)**	15.36 (15.27 ; 15.46)
0.016	0.8	15.45 (15.26 ; 15.65)	15.24 (15.16 ; 15.36)
0.040	0.1	15.77 (15.73 ; 15.82)	15.96 (15.59 ; 16.01)
0.040	0.2	15.45 (15.28 ; 15.75)	15.6 (14.79 ; 15.67)
0.040	0.4	15.45 (15.32 ; 15.52)	15.62 (15.14 ; 15.74)
0.040	0.8	15.28 (14.99 ; 15.34)	***15.14 (15 ; 15.44)***
0.100	0.1	15.45 (15.44 ; 15.98)	15.76 (15.75 ; 15.85)
0.100	0.2	15.34 (15.26 ; 15.57)	15.66 (15.61 ; 15.68)
0.100	0.4	15.44 (14.55 ; 15.62)	15.59 (15.56 ; 15.6)
0.100	0.8	15.34 (14.94 ; 15.43)	15.56 (15.17 ; 15.69)
0.250	0.1	15.14 (15.14 ; 15.64)	15.18 (14.85 ; 15.84)
0.250	0.2	15.58 (15.3 ; 15.77)	15.2 (15.16 ; 15.62)
0.250	0.4	15.41 (15.2 ; 15.54)	15.34 (15.21 ; 15.99)
0.250	0.8	15.49 (15.45 ; 15.57)	15.71 (15.5 ; 15.77)

robustness might be due to relying on a larger number of features, making it more resilient to changes in the behavior of COVID-19 epidemics.

3.4 Influence of leaking rate

Panel A of figure 4 shows that other hyperparameters varied depending on the leaking rate selected by the GA. Specifically, this figure highlights a bimodal distribution of optimal leaking rates. When the optimal leaking rate was below $1e-2$, both input scaling and ridge regression values tended to be lower, at approximately 1 and $1e3$ respectively. In contrast, for leaking rates above $1e-2$, these values were higher, at $1e2$ and $1e3$ respectively. The distribution of the spectral radius was more spread out when the leaking rate was below $1e-3$.
Regarding the impact of the leaking rate on the training set, panel B of Figure 4 indicates that a lower leaking rate resulted in better performance compared to higher leaking rate values. However, this trend was reversed on the test set, where higher leaking rates provided better performance, with mean absolute errors and standard deviation (MAE ($\pm$ SD)) of 15.48 ($\pm$ 12.79) and 15.21 ($\pm$ 12.86) without a monthly update, and 15.88 ($\pm$ 12.76) and 15.4 ($\pm$ 13.04) with a monthly update. This suggests that lower leaking rates may lead to overfitting,

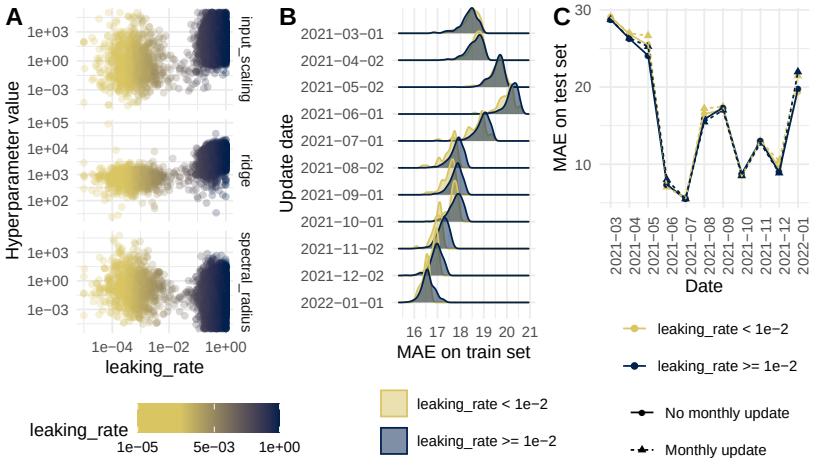

Fig. 4. Influence of the GA-selected leaking rate hyperparameter on performance. Panel A displays the top 40 sets of hyperparameters across all GA metaparameter experiments. Panel B shows the relationship between the leaking rate and performance on the training set. Panel C shows the relationship between the leaking rate and performance on the test set.

whereas higher leaking rates, possibly due to increased ridge penalization, offer greater robustness as depicted in Panel C of Figure 4.

3.5 Sparse model approaches

As discussed in section 2.4, we explored several strategies to achieve sparser models, including: (i) introducing a penalty in the fitness function based on the number of selected features, (ii) modifying the tournament selection process to penalize individuals with a higher number of features, and (iii) adapting the mutation process to favor sparsity. Among these methods, only the mutation adaptation approach led to a significant reduction in the mean (± SD) number of selected features, with 150.3 (± 28.3) in March 2021 and 23.6 (± 6.3) in January 2022, compared to over 199 features for the other approaches. Despite this reduction in features, model performance remained stable, with an MAE of 15.32 (± 12.99) and 15.04 (± 12.02) with and without monthly RC-HP updates, respectively. Additional results are available at https://github.com/thomasferte/ChallengesRCGA.

4 Discussion

This study examined the impact of GA-HP on feature and RC-HP selection for forecasting COVID-19 cases 14 days ahead at Bordeaux University Hospital in a high-dimensional setting, yielding several key findings.

We observed that GA-HP affected both the number and specificity of selected features. Higher mutation rates in the GA led to fewer, more diverse features and increased MAE on the training set, but similar MAE on the test set, suggesting better generalization and reduced overfitting in non-stationary time series. In contrast, lower mutation rates and $\sigma_{leaking\_rate}$ slightly hindered performance, likely due to insufficient exploration. Thus, higher mutation rates may enhance performance by promoting feature diversity and improving RC-HP.
We observed bimodal convergence of $\sigma_{leaking\_rate}$ to both $1e-3$ and 1, as noted by [6]. Lower leaking rates, paired with smaller ridge and input scaling values, improved training performance but slightly worsened test performance. In contrast, the slightly better performance at higher leaking rates might be due to the larger ridge value penalizing the RC output layer.
Furthermore, when exploring GA adaptations for sparser feature selection, we found that even methods resulting in sparse feature selection (specifically through modifying the GA mutation process) yielded similar performance.
In the literature, RC has been used with GA in various contexts. Bala et al. [2] review RC-GA applications, emphasizing the optimization of multiple hyperparameters like reservoir size, spectral radius, scaling, shifting, activation functions, and connectivity. However, RC's use in high-dimensional contexts is rare, and applying metaheuristic algorithms in these settings presents challenges that require further adaptation.
This study has several limitations. We optimized only the main RC-HP, so further optimization could improve performance. Most experiments were based on a single RC-GA run, meaning small differences may stem from the inherent randomness of RC and GA. The GA used was standard and not specifically designed for high-dimensional settings or RC-GA. Additionally, there was a mismatch between the number of reservoirs used for hyperparameter optimization (three) and forecasting (forty). Rather than using the median forecast from the top forty hyperparameter sets, exploring combinations of hyperparameter sets could enhance results by leveraging more complementary reservoirs.
It is noteworthy that combining RC with GA resulted in the best performance for forecasting COVID-19 hospitalizations 14 days ahead at Bordeaux University Hospital. This study highlights the impact of different GA-HP on RC-GA performance, and future work could improve results by tailoring GA for high-dimensional RC contexts.

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

Disclosure of Interests. The authors have no competing interests to declare that are relevant to the content of this article.

Bibliography

[1] Arcomano, T., Szunyogh, I., Pathak, J., Wikner, A., Hunt, B.R., Ott, E.: A Machine Learning-Based Global Atmospheric Forecast Model. Geophysical Research Letters **47**(9) (2020). https://doi.org/10.1029/2020GL087776

[2] Bala, A., Ismail, I., Ibrahim, R., Sait, S.M.: Applications of Metaheuristics in Reservoir Computing Techniques: A Review. IEEE Access **6** (2018). https://doi.org/10.1109/ACCESS.2018.2873770

[3] Carvalho, K., Vicente, J.P., Jakovljevic, M., 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) (Jun 2021). https://doi.org/10.3390/bioengineering8060084

[4] Cramer, E.Y., Ray, E.L., Lopez, V.K., Bracher, J., Brennen, A., 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) (Apr 2022). https://doi.org/10.1073/pnas.2113561119

[5] Etalab: Les données relatives au COVID-19 en France - data.gouv.fr (2020)

[6] Ferté, T., Dutartre, D., Hejblum, B.P., Griffier, R., Jouhet, V., Thiébaut, R., Legrand, P., Hinaut, X.: Reservoir computing for short high-dimensional time series: an application to SARS-cov-2 hospitalization forecast. In: Forty-first International Conference on Machine Learning (2024)

[7] Ferté, T., Jouhet, V., Griffier, R., Hejblum, B., Thiébaut, R., 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 (Jan 2023). https://doi.org/10.5061/DRYAD.HHMGQNKKX

[8] Ferté, T., Jouhet, V., Griffier, R., Hejblum, B.P., Thiébaut, R., 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) (Dec 2022). https://doi.org/10.1093/jamiaopen/ooac086

[9] Ghosh, S., Senapati, A., Mishra, A., Chattopadhyay, J., Dana, S.K., Hens, C., Ghosh, D.: Reservoir computing on epidemic spreading: A case study on COVID-19 cases. Physical Review E **104**(1) (Jul 2021). https://doi.org/10.1103/PhysRevE.104.014308

[10] Hinaut, X., 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) (Feb 2013). https://doi.org/10.1371/journal.pone.0052946

[11] 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** (Jan 2001)

[12] Liu, B., Xie, Y., Liu, W., Jiang, X., Ye, Y., Song, T., Chai, J., Feng, M., Yuan, H.: Nanophotonic reservoir computing for COVID-19 pandemic forecasting. Nonlinear Dynamics **111**(7) (2023). https://doi.org/10.1007/s11071-022-08190-z

[13] Lukoševičius, M., Jaeger, H.: Reservoir computing approaches to recurrent neural network training. Computer Science Review **3**(3) (Aug 2009). https://doi.org/10.1016/j.cosrev.2009.03.005

[14] Maass, W., Natschläger, T., Markram, H.: Real-time computing without stable states: a new framework for neural computation based on perturbations. Neural Computation **14**(11) (Nov 2002). https://doi.org/10.1162/089976602760407955

[15] Mohimont, L., Chemchem, A., Alin, F., Krajecki, M., Steffenel, L.A.: Convolutional neural networks and temporal CNNs for COVID-19 forecasting in France. Applied Intelligence (Apr 2021). https://doi.org/10.1007/s10489-021-02359-6

[16] Paireau, J., Andronico, A., Hozé, N., Layan, M., Crépey, P., Roumagnac, A., Lavielle, M., Boëlle, P.Y., 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) (May 2022). https://doi.org/10.1073/pnas.2103302119

[17] Pottier, L.: Forecast of the covid19 epidemic in france. medRxiv (2021). https://doi.org/10.1101/2021.04.13.21255418, https://www.medrxiv.org/content/early/2021/04/20/2021.04.13.21255418

[18] Rahimi, I., Chen, F., Gandomi, A.H.: A review on COVID-19 forecasting models. Neural Computing & Applications (Feb 2021). https://doi.org/10.1007/s00521-020-05626-8

[19] Ray, A., Chakraborty, T., Ghosh, D.: Optimized ensemble deep learning framework for scalable forecasting of dynamics containing extreme events. Chaos (Woodbury, N.Y.) **31**(11) (Nov 2021). https://doi.org/10.1063/5.0074213

[20] Smith, A., Lott, N., Vose, R.: The Integrated Surface Database: Recent Developments and Partnerships. Bulletin of the American Meteorological Society **92**(6) (Jun 2011). https://doi.org/10.1175/2011BAMS3015.1

[21] Tanaka, G., Yamane, T., Héroux, J.B., Nakane, R., Kanazawa, N., Takeda, S., Numata, H., Nakano, D., Hirose, A.: Recent advances in physical reservoir computing: A review. Neural Networks **115** (Jul 2019). https://doi.org/10.1016/j.neunet.2019.03.005

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

[23] Trouvain, N., Pedrelli, L., Dinh, T.T., Hinaut, X.: ReservoirPy: an Efficient and User-Friendly Library to Design Echo State Networks. In: ICANN 2020 - 29th International Conference on Artificial Neural Networks. Bratislava, Slovakia (Sep 2020), https://inria.hal.science/hal-02595026

[24] Vlachas, P.R., Pathak, J., Hunt, B.R., Sapsis, T.P., Girvan, M., Ott, E., Koumoutsakos, P.: Backpropagation algorithms and Reservoir Computing in Recurrent Neural Networks for the forecasting of complex spatiotemporal dynamics. Neural Networks **126** (Jun 2020). https://doi.org/10.1016/j.neunet.2020.02.016

[25] Weddell, S.J., Ayyagari, S., Jones, R.D.: Reservoir computing approaches to microsleep detection. Journal of Neural Engineering **18**(4) (Mar 2021). https://doi.org/10.1088/1741-2552/abcb7f

[26] WHO: WHO Coronavirus (COVID-19) Dashboard (2020)

[27] Zhang, H., Vargas, D.V.: A Survey on Reservoir Computing and its Interdisciplinary Applications Beyond Traditional Machine Learning. IEEE Access **11** (2023). https://doi.org/10.1109/ACCESS.2023.3299296