Eur. Phys. J. Spec. Top.
https://doi.org/10.1140/epjs/s11734-026-02537-y

THE EUROPEAN PHYSICAL JOURNAL SPECIAL TOPICS

Regular Article

State correction via data injection for a reservoir computing-based digital twin of the Lorenz system

Sofia Tkach[a] and Andrey Andreev[b]

Research Institute of Applied Artificial Intelligence and Digital Solutions, Plekhanov Russian University of Economics, 36, Stremyanny Per., Moscow, Russia115054

Received 28 May 2026 / Accepted 7 July 2026

Abstract The concept of a digital twin, a virtual representation of a physical system evolving in real time, holds great promise for monitoring and controlling complex dynamics, yet creating accurate twins for chaotic systems remains challenging due to sensitivity to initial conditions. Reservoir computing (RC) offers a powerful framework for learning chaotic dynamics from data, but autonomous RC models inevitably desynchronize from the true system over long time horizons. Here, we systematically investigate four strategies for coupling a trained RC-based digital twin with real-time observations of the chaotic Lorenz system: (i) periodic data injections, (ii) single-threshold error-triggered injections, (iii) two-threshold hysteretic injections, and (iv) a novel combination of sparse periodic and two-threshold injections. We quantify the trade-off between prediction accuracy and measurement cost. Our results show that the combined approach achieves NRMSE = 0.0262 with only 5 updates per 100 time steps, significantly reducing injection frequency compared to periodic injections while maintaining competitive accuracy. These findings provide a practical framework for designing controllable digital twins under limited measurement budgets.

1 Introduction

The concept of a digital twin, a virtual representation of a physical system that evolves in real time, has emerged as a powerful paradigm for monitoring, prediction, and control of complex dynamics. Creating an accurate digital twin for chaotic systems is particularly challenging due to the exponential divergence of trajectories arising from sensitivity to initial conditions [1]. Recent work has highlighted the relevance of chaos theory in digital twin applications, for instance in facility management and predictive maintenance [2]. Even a well-trained model will eventually desynchronize from the true system if run autonomously, limiting the predictive horizon.

Reservoir Computing (RC) has proven to be a highly effective machine learning framework for learning and predicting chaotic dynamics [1, 3–7]. The seminal work of Jaeger and Haas [3] demonstrated the ability of echo state networks to predict chaotic systems, while subsequent studies established RC as a benchmark for model-free prediction of spatiotemporally chaotic systems [1]. Its architecture is naturally suited to emulating dynamical systems: a fixed, high-dimensional nonlinear reservoir provides a fading memory of past inputs, while training is reduced to a computationally efficient linear regression. This simplicity and the inherent capacity for autonomous dynamical evolution make RC a leading candidate for constructing digital twins of nonlinear systems.

Recent work has significantly expanded the applicability of reservoir computing to complex dynamical systems beyond deterministic chaos. Hramov et al. [8] introduced the concepts of strong and weak prediction for stochastic dynamics, while Badarin et al. [9, 10] demonstrated that RC can recover hidden variables from partially observed systems and reconstruct neuroimaging signals. The flexibility of RC has been extended to irregular time-series classification [11] and stochastic cloning of dynamical systems with hidden variables [12]. However, predicting stochastic systems remains challenging due to intermittency effects [13], and the trade-off between reservoir size and accuracy requires careful optimization [14]. These findings underscore that while RC is a powerful tool for emulating complex dynamics, careful design of synchronization strategies is essential for maintaining long-term fidelity.

[a] e-mail: sofata31@gmail.com
[b] e-mail: andreevandrei1993@gmail.com (corresponding author)

Published online: 13 July 2026

While reservoir computers can be trained to autonomously replicate an attractor (effectively creating a closed-loop digital twin), their long-term stability in the face of perturbations and model inaccuracies remains a critical limitation. Practical deployment often necessitates periodic or event-driven interventions to keep the digital twin close to with the physical asset, especially when using sparse or noisy measurements. This motivates the development of hybrid injection strategies, where a trained autonomous RC model is combined with a data assimilation scheme that occasionally injects observed true states into the reservoir to correct its trajectory. Similar event-triggered approaches have been successfully applied to chaotic Lur'e systems [15, 16], where control updates are performed only when a predefined error threshold is exceeded.

Practical applications of digital twins often operate under constraints where continuous high-frequency measurements are unavailable or prohibitively expensive. For example, in remote environmental monitoring (e.g., oceanic or atmospheric sensing), satellite or buoy data may only be available at discrete intervals [17]. In medical diagnostics, invasive measurements cannot be performed continuously [18]. In industrial settings, sensor faults or communication delays may interrupt the data stream [19]. These scenarios demand digital twins that can maintain useful accuracy with sparse, event-driven, or irregularly timed observations. Our work addresses precisely this challenge: how to design a reservoir-computing digital twin that remains close to a chaotic system using minimal external injections.

In this work, we systematically investigate and compare four strategies for coupling a trained RC model with real-time observations from the chaotic Lorenz system. Beyond a classical open-loop prediction, we study two threshold-based state update methods and a novel combined approach. Such approaches have some similarities with classical pinning control of chaotic systems [20], as well as with intermittency effects in externally synchronized chaotic oscillators [21], but are used to maintain the system close to the required phase trajectory, for example, as in the phase tube method for the generalized synchronization precise detection [22]. Our goal is to delineate the trade-offs between the accuracy of the digital twin (quantified by the Normalized Root-Mean-Square Error) and the cost of measurement (quantified by the average frequency of data injections). We present a comprehensive framework for designing a controllable digital twin that balances these competing objectives.

Specifically, we: (i) systematically compare periodic, single-threshold, and two-threshold injection strategies for RC-based digital twins; (ii) introduce a combined periodic and two-threshold approach that balances accuracy and injection cost; (iii) quantify the trade-off using a novel composite metric $\mathrm{NRMSE} \cdot N_{\mathrm{inj}}$.

The remainder of this paper is organized as follows. Section 2 describes the Lorenz system, the reservoir-computing architecture, the training procedure, and the evaluation metrics. Section 3 presents our results, organized by injection strategy: periodic injections (Sect. 3.1), single-threshold error-triggered injections (Sect. 3.2), two-threshold hysteretic injections (Sect. 3.3), and the combined periodic and two-threshold approach (Sect. 3.4). Section 3 concludes with a summary of findings, practical recommendations, and limitations.

2 Methods

2.1 The Lorenz system

The Lorenz system is a benchmark model for chaotic dynamics, defined by the following set of ordinary differential equations [23]:

$$\dot{x} = \sigma(y - x), \tag{1}$$

$$\dot{y} = x(\rho - z) - y, \tag{2}$$

$$\dot{z} = xy - \beta z, \tag{3}$$

where we use the standard parameter set $\sigma = 10$, $\rho = 28$, and $\beta = 8/3$, for which the system exhibits chaotic behavior on a strange attractor. The system is ergodic and mixing, providing a robust benchmark for testing prediction and synchronization strategies.

The system was integrated using the 4th-Order Runge–Kutta numerical scheme with a time step $\Delta t = 0.01$. A time-series of 17,000 points was generated; the first 12,000 points were used for training and the remaining 5,000 for testing.

Table 1 Optimal hyperparameters of the reservoir-computing model

Parameter	Optimal value
Number of reservoir units (N)	500
Leak rate (α)	0.25
Spectral radius (ρ)	0.6
Input scaling (σ_{in})	0.44
Reservoir connectivity (p_{res})	0.02
Input connectivity (p_{in})	0.21
Ridge regularization (λ)	10^{-4}

2.2 Reservoir computing

The proposed digital twin is built upon a standard Echo State Network (ESN) architecture with $N = 500$ reservoir units. The reservoir state $\mathbf{r}(t) \in \mathbb{R}^N$ is updated according to

$$\mathbf{r}(t + \Delta t) = (1 - \alpha)\mathbf{r}(t) + \alpha \tanh(\mathbf{W}\mathbf{r}(t) + \mathbf{W}_{\text{in}}\mathbf{u}(t)), \tag{4}$$

where $\mathbf{u}(t) \in \mathbb{R}^3$ is the input state vector, $\mathbf{W}$ is the $N \times N$ recurrent weight matrix, $\mathbf{W}_{\text{in}}$ is the $N \times 3$ input matrix, and α is the leak rate. The matrices $\mathbf{W}$ and $\mathbf{W}_{\text{in}}$ are randomly generated according to hyperparameters and stay fixed during training; only the output weights $\mathbf{W}_{\text{out}}$ are trained. $\mathbf{W}_{\text{in}}$ values are generated from uniform distribution $[-\sigma_{\text{in}}, \sigma_{\text{in}}]$ with probability p_{in}. $\mathbf{W}$ is defined by spectral radius ρ and Reservoir connectivity (p_{res}) which is the probability of connection existence.

To create a digital twin, the network is trained for autonomous evolution by setting the input $\mathbf{u}(t)$ to the reservoir's own prediction, $\hat{\mathbf{u}}(t)$, in a closed-loop configuration. The output weights are learned via Tikhonov-regularized linear regression, minimizing

$$\sum_t \|\mathbf{W}_{\text{out}}\mathbf{r}(t) - \mathbf{u}_{\text{true}}(t + \Delta t)\|^2 + \lambda\|\mathbf{W}_{\text{out}}\|^2, \tag{5}$$

where $\lambda = 10^{-4}$ is a regularization parameter that prevents overfitting. Therefore, the reservoir's output is defined as

$$\hat{\mathbf{u}}(t) = \mathbf{W}_{\text{out}}\mathbf{r}(t). \tag{6}$$

During the training phase, the input is the current true state $u_{\text{true}}(t)$, and the target output is the true state at the next time step, $u_{\text{true}}(t + \Delta t)$. Thus, the network learns a one-step forward map. During the closed-loop mode, this learned map is applied recursively, with the output $\hat{u}(t)$ fed back as input for the next iteration.

We optimize the hyperparameters to achieve minimal error of prediction defined in (8). The resulting RC's parameters are presented in Table 1.

2.3 Prediction quality metrics

We define the instantaneous prediction error as the Euclidean (L_2) norm

$$e(t) = \|\hat{\mathbf{u}}(t) - \mathbf{u}_{\text{true}}(t)\|_2. \tag{7}$$

Overall performance is quantified by the normalized root-mean-square error (NRMSE) over the testing window

$$\text{NRMSE} = \frac{\sqrt{\frac{1}{T}\sum_{t=1}^{T}\|\hat{\mathbf{u}}(t) - \mathbf{u}_{\text{true}}(t)\|^2}}{\sigma_{\mathbf{u}}}, \tag{8}$$

where $\sigma_{\mathbf{u}}$ is the standard deviation of the true trajectory in each coordinate.

We also define N_{inj} as the mean number of state updates per 100 time steps, averaged over the entire generative run.

Generative runs consist of 5000 steps after a 50-step warmup using test data.

3 Results

First, we train classical RC to predict chaotic dynamics of the Lorenz dynamical system. As one can see on Fig. 1, after optimizing hyperparameters (Table 1), we achieve accurate short-term prediction for $t = 5$ sec and attractor reproduction. Then, we attempt to construct a digital twin of Lorenz system using different approaches of one-point real data injection to the RC.

3.1 Periodic injections

As the first approach for digital twin construction, we apply periodic external injections of real data from the Lorenz system into the reservoir's input. The model runs autonomously for $n-1$ steps, and at every n-th step the true state is injected. This can be expressed as a single update rule

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_{\text{true}}(t), & \text{if } t \bmod (n\Delta t) = 0, \\ \hat{\mathbf{u}}(t-\Delta t), & \text{otherwise,} \end{cases} \tag{9}$$

where $\hat{\mathbf{u}}(t-\Delta t)$ is the reservoir's own prediction from the previous time step. Thus, the true state is fed into the reservoir at a fixed period of n time steps, providing a baseline for performance as a function of measurement frequency.

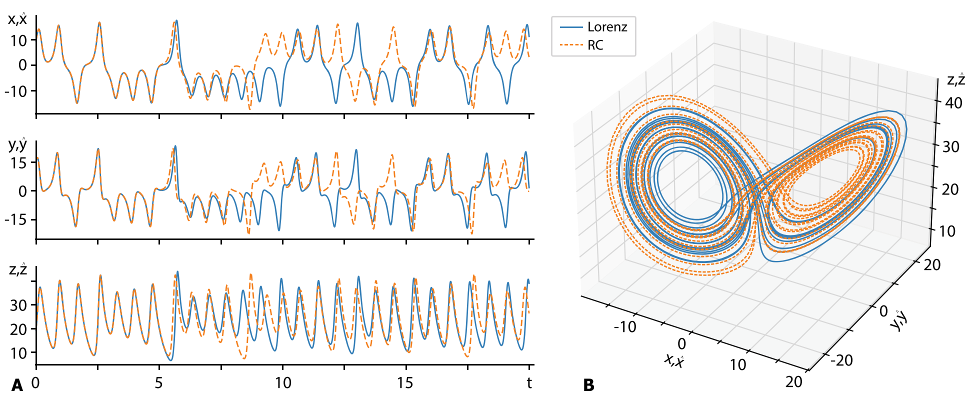

Fig. 1 Prediction of chaotic dynamics of the Lorenz dynamical system by classical RC: (**A**) time-series of Lorenz variables x, y, z and RC's prediction $\hat{x}$, $\hat{y}$, $\hat{z}$ and (**B**) phase portrait

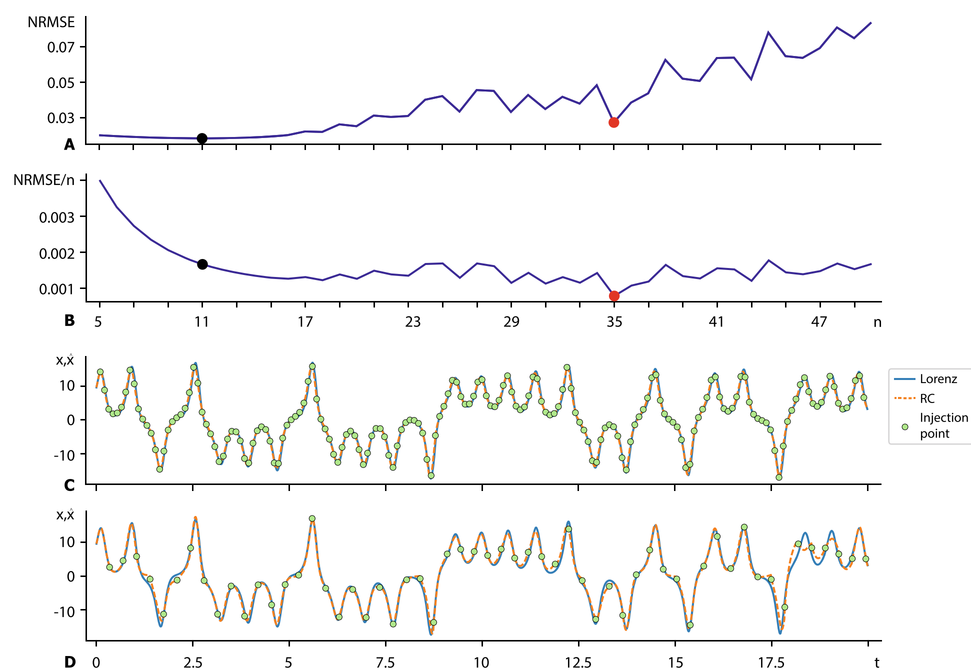

Fig. 2 The quality of Lorenz dynamics prediction using an approach of periodic injections: the dependencies of (**A**) NRMSE and (**B**) normalized NRMSE/n on the injections period n and the examples of true and predicted time-series of x-variable with the injection points for (**C**) $n = 11$ corresponding to a local minimum in (**A**) marked by black point and (**D**) $n = 35$ corresponding for local minimum in (**B**) marked by red point

Figure 2A illustrates the dependence of NRMSE on the period between injections n—lower n means more frequent real values injections to the RC's input. Small NRMSE is achieved for $n \in [5, 16]$ with local minimum of NRMSE $= 0.0182$ for $n = 11$. The true and predicted time-series of x-variable is presented in Fig. 2C. As one can see, achieving such accuracy requires very frequent injections, around 6 times per period. Further increasing n leads to increasing the prediction error. To achieve trade-off between accuracy and injections frequency, we calculate normalized error, NRMSE$/n$ (see Fig. 2B). Such metric has different dependence from n: it decreases when increasing n from 5 to 17, but further increasing n does not almost change normalized NRMSE. It indicates that the total error grows roughly linearly with the injections period. Local minimum is achieved for $n = 35$ with NRMSE $= 0.0272$, the corresponding time-series are presented in Fig. 2D. As one can see, the predicted trajectory stays close to the true one most part of the time. Only small deviations are observed, but they are correcting fast by periodic injections.

3.2 Single-threshold error-triggered injections

As the second approach, instead of periodic injections, we propose error-triggered injections when the true state is fed only when the error threshold is reached. A threshold value ε_1 on the instantaneous error is set. The error is computed as the L_2-norm (Euclidean distance) between the true and predicted state vectors: $e(t) = \|\mathbf{u}_{true}(t) - \hat{\mathbf{u}}(t)\|_2$. If $e(t) > \varepsilon_1$, the true state $\mathbf{u}_{true}(t)$ is fed into the reservoir at the next time step

$$\mathbf{u}(t) = \begin{cases} \mathbf{u}_{\text{true}}(t), & \text{if } e(t-\Delta t) > \varepsilon_1, \\ \hat{\mathbf{u}}(t-\Delta t), & \text{otherwise.} \end{cases} \tag{10}$$

Figure 3A illustrates the dependence of NRMSE on threshold value ε_1. One can expect that as $\varepsilon_1 \to 0$, the error should approach a minimum at the cost of a high reset rate, while large ε_1 should approach autonomous performance. We found that for low values of $\varepsilon_1 < 1$, NRMSE hardly changes and stays around 0.0245. Increasing of threshold value leads to first decreasing the error and then increasing to high NRMSE. Figure 3B, C demonstrates true and predicted time-series of x-variable for top-2 threshold values $\varepsilon_1 = 1.6$ and $\varepsilon_1 = 3.4$ with NRMSE $= 0.0169$ and 0.0170, respectively. As one can see, lower ε_1 keeps the predicted trajectory close to the true one; exceeding the threshold typically initiates a long sequence of injections until the error drops below ε_1, resulting in $N_{\text{inj}} = 19$ updates per 100 steps. In contrast, higher ε_1 yields more frequent but shorter injections sequences $N_{\text{inj}} = 5$. With a higher threshold, the RC returns to autonomous operation at a larger instantaneous error, causing the next threshold crossing to occur sooner.

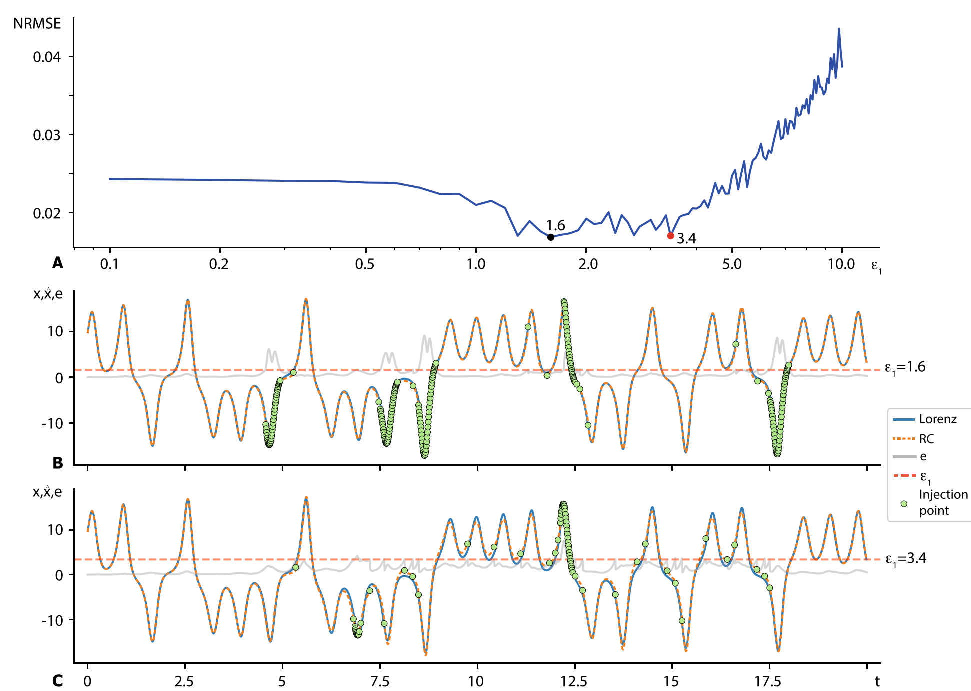

Fig. 3 The quality of Lorenz dynamics prediction using an approach of single-threshold error-triggered injections: (**A**) the dependence of NRMSE on threshold value ε_1 and (**B, C**) the examples of true and predicted time-series of x-variable, error e, and the injection points for (**B**) $\varepsilon_1 = 1.6$ and (**C**) $\varepsilon_1 = 3.4$

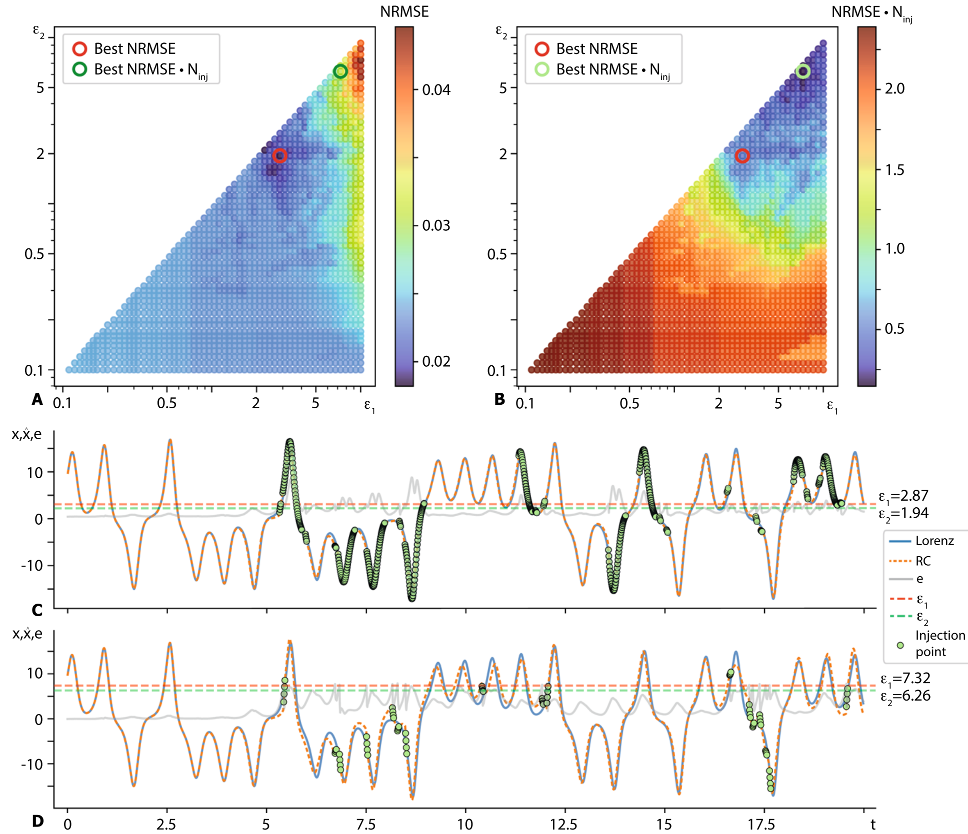

Fig. 4 The quality of Lorenz dynamics prediction using an approach of two-threshold error-triggered injections: the dependencies of (**A**) NRMSE and (**B**) NRMSE $\cdot N_{\text{inj}}$ on the threshold values ε_1 and ε_2 and (**C, D**) the examples of true and predicted time-series of x-variable, error e, and the injection points for (**C**) $\varepsilon_1 = 2.87$, $\varepsilon_2 = 1.94$ and (**D**) $\varepsilon_1 = 7.32$, $\varepsilon_2 = 6.26$

3.3 Two-threshold error-triggered injections

Then, we propose to extend the previous strategy by defining two thresholds: an activation threshold ε_1 and a lower stabilization exit threshold ε_2 (where $\varepsilon_1 \geq \varepsilon_2$). When $e(t) > \varepsilon_1$, a "stabilization mode" is activated: the true states are then fed to the reservoir for a minimum of 4 consecutive time steps. This duration was chosen empirically based on the reservoir's leak rate ($\alpha = 0.25$); it is long enough to allow the reservoir's hidden state to 'flush' the memory of the erroneous autonomous trajectory and fully synchronize with the true dynamics. Fewer than 4 steps were found to be insufficient, often resulting in immediate re-exceedance of the error threshold upon returning to autonomous operation. The system exits this mode only after the error at the last three injected points is at or below ε_2. This provides a hysteretic control mechanism to ensure that the system is firmly re-synchronized before returning to autonomous mode.

We investigate the influence of a couple of threshold values ε_1 and ε_2 on the prediction accuracy. As one can see in Fig. 4A, the leading role plays the activation threshold: low NRMSE is achieved for $\varepsilon_1 < 4$ for any ε_2. Both high ε_1 and ε_2 lead to high errors as expected, while the lowest NRMSE $= 0.0182$ is achieved for mediocre threshold values $\varepsilon_1 = 2.87$ and $\varepsilon_2 = 1.94$. However, such low accuracy has a price of high number of state updates. As one

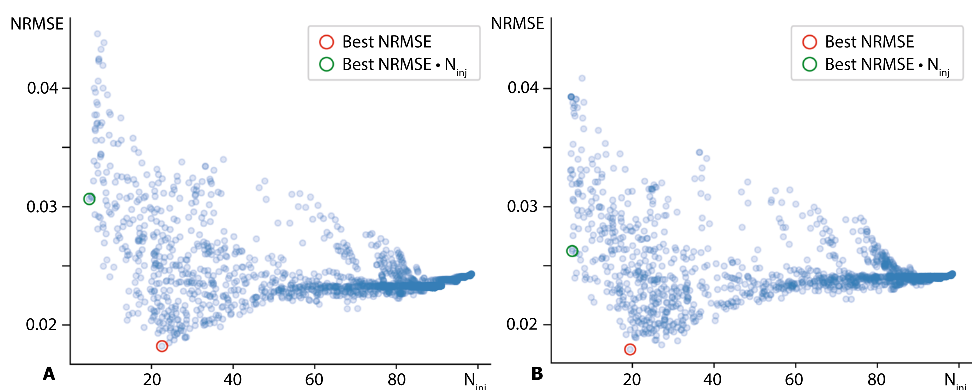

Fig. 5 The dependencies of NRMSE on the mean number of injections N_{inj} for (**A**) the approach of two-threshold error-triggered injections and (**B**) the combined one with periodic and two-threshold error-triggered injections

can see in Fig. 4C, there are a big number of long series of state updates, one or more periods of oscillations are almost fully consist of true states intermittent by short prediction series.

To achieve a reasonable accuracy with small number of injections, we consider a new metric – NRMSE $\cdot$ N_{inj}. Due to low thresholds leads to frequent injections, the dependence of the new metric on ε_1 and ε_2 strictly differs from NRMSE (see Fig. 4B). The minimum is achieved for $\varepsilon_1 = 7.32$ and $\varepsilon_2 = 6.26$ (NRMSE $= 0.0306$). As one can see in Fig. 4D, the number of injections series is significantly reduced ($N_{\text{inj}} = 4$ for panel D compared to $N_{\text{inj}} = 18$ for panel C), each series typically consists of the minimum of 4 points, but the predicted and real trajectories sometimes reach a large divergence between themselves.

Figure 5A illustrates how NRMSE depends on the mean number of injections N_{inj}. As one can see, the highest error values are achieved at low N_{inj} values. However, for $N_{\text{inj}} < 40$, a wide range of error values is observed: errors can range from very low (the lowest NRMSE is achieved at $N_{\text{inj}} = 22$) to high ones. The optimal trade-off between error and injections frequency is observed for $N_{\text{inj}} = 4$ that corresponds to the minimal length of the stabilization series. Therefore, only the shortest series are triggered for such case. For $N_{\text{inj}} > 50$, most of the points align around NRMSE $= 0.024$, no longer depending strongly on N_{inj}.

3.4 Combined periodic and two-threshold injections

As a final strategy, we propose to combine a sparse periodic injections ($n = 150$) with the two-threshold method. This ensures a baseline level is maintained via rare periodic injections, while the dual-threshold mechanism provides rapid correction during large excursions. The period $n = 150$ was chosen empirically to serve as a sparse baseline safety net; it is significantly larger than the optimal standalone periodic injections ($n = 11$ and $n = 35$), ensuring that it does not dominate the correction mechanism but rather prevents complete desynchronization over long runs.

Fixing n and varying threshold values in the same boundaries, we observe dependencies similar to the two-threshold only approach. Figure 5B has the same main trends as A. The only differences are lower maximal, optimal and minimal NRMSEs, while the last one (NRMSE $= 0.0179$) is achieved for lower $N_{\text{inj}} = 20$.

Two-parametric dependencies of NRMSE and NRMSE $\cdot$ N_{inj} on ε_1 and ε_2 are illustrated in Fig. 6A, B. As one can see, all trends remain the same as in Fig. 4A, B. The minimum NRMSE $= 0.0179$ is also achieved with mediocre threshold values: the activation threshold $\varepsilon_1 = 2.87$ remains the same, and the exit threshold $\varepsilon_2 = 2.10$ is slightly higher. Figure 6C illustrates the true and predicted time-series of x-variable. One can see that introducing

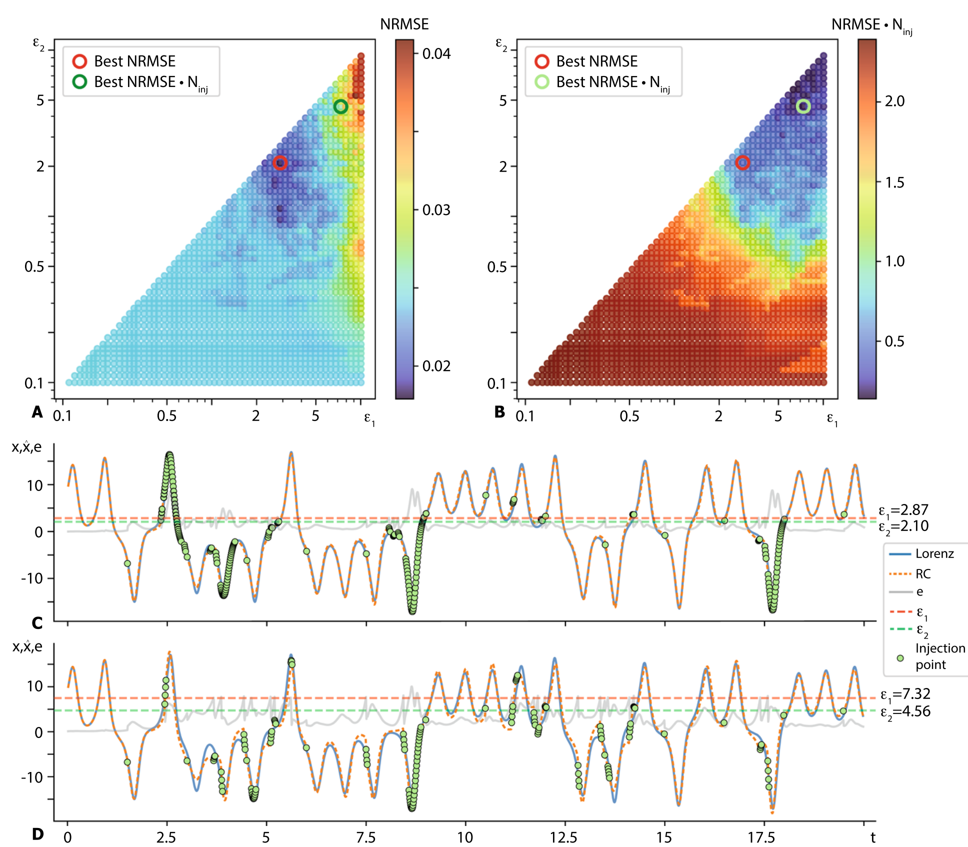

Fig. 6 The quality of Lorenz dynamics prediction using combined approach with periodic and two-threshold error-triggered injections: the dependencies of (**A**) NRMSE and (**B**) NRMSE $\cdot$ N_{inj} on the threshold values ε_1 and ε_2 and (**C, D**) the examples of true and predicted time-series of x-variable, error e, and the injection points for (**C**) $\varepsilon_1 = 2.87$, $\varepsilon_2 = 2.10$ and (**D**) $\varepsilon_1 = 7.32$, $\varepsilon_2 = 4.56$

Table 2 Comparison of the four injection strategies for the RC-based digital twin of the Lorenz system

Strategy	NRMSE	$\mathbf{N}_{\text{inj}}$	Parameters
Periodic injections	**0.0182**	9	$n = 11$
	0.0272	**3**	$n = 35$
Single-threshold error-triggered	**0.0169**	19	$\varepsilon_1 = 1.6$
	0.0170	**5**	$\varepsilon_1 = 3.4$
Two-threshold error-triggered	**0.0182**	18	$\varepsilon_1 = 2.87,\ \varepsilon_2 = 1.94$
	0.0306	**4**	$\varepsilon_1 = 7.32,\ \varepsilon_2 = 6.26$
Combined (periodic + two-threshold)	**0.0179**	20	$n = 150,\ \varepsilon_1 = 2.87,\ \varepsilon_2 = 2.10$
	0.0262	**5**	$n = 150,\ \varepsilon_1 = 7.32,\ \varepsilon_2 = 4.56$

For each strategy, two operating points are reported: the configuration minimizing NRMSE (maximum accuracy) and the configuration minimizing injection cost (lowest N_{inj}). Bold values indicate the best performance in each column

rare periodic injections decreases the frequency of error-triggered injections, but the last ones remain long as for the two-threshold only approach ($N_{\text{inj}} = 20$). $\varepsilon_1 = 7.32$ for optimal NRMSE $\cdot\, N_{\text{inj}}$ is also remains the same, but $\varepsilon_2 = 4.56$ becomes much lower. It creates a wide gap between the thresholds which is expected to longer the stabilization sequences. However, as one can see on Fig. 6D, most part of them remains minimal and consists of 4 point, and $N_{\text{inj}} = 5$ with NRMSE $= 0.0262$, which is lower than for Fig. 4D. Therefore, we can conclude that introducing sparse periodic injections allows to achieve the same or better accuracy of prediction with lower state updates.

To provide a clear overview of the trade-offs between prediction accuracy and injection cost, Table 2 summarizes the key performance indicators for all four strategies at their optimal operating points.

4 Conclusions

In this work, we systematically investigate and compare four strategies for coupling a trained RC model with real-time observations from the chaotic Lorenz system: periodic injections, single-threshold error-triggered injections, two-threshold hysteretic injections, and a combined approach. Our goal is to delineate the trade-offs between the accuracy of the digital twin, quantified by the Normalized Root-Mean-Square Error (NRMSE), and the measurement cost, quantified by the average number of data injections per 100 time steps (N_{inj}).

The key results are summarized in Table 2. Periodic injections provide a straightforward baseline: the lowest NRMSE of 0.0182 is achieved with $n = 11$ (9 injections per 100 steps), while the minimal injection cost of $N_{\text{inj}} = 3$ is attained at $n = 35$ with NRMSE $= 0.0272$. Single-threshold error-triggered injections yield the overall best accuracy (NRMSE $= 0.0169$) but require frequent injections ($N_{\text{inj}} = 19$), yet a balanced configuration achieves nearly the same accuracy (NRMSE $= 0.0170$) with only $N_{\text{inj}} = 5$. Two-threshold hysteretic injections reduce the injection cost further to $N_{\text{inj}} = 4$ at the expense of higher error (NRMSE $= 0.0306$). Finally, our proposed combined strategy (sparse periodic injections with $n = 150$ plus two-threshold control) achieves NRMSE $= 0.0262$ with $N_{\text{inj}} = 5$, offering an excellent balance between accuracy and measurement cost. These findings provide a practical framework for designing controllable digital twins under limited measurement budgets.

The composite metric NRMSE $\cdot\, N_{\text{inj}}$ introduced here provides a practical way to balance prediction accuracy against measurement cost, which is particularly relevant for applications where observations are expensive, infrequent, or delayed (e.g., remote sensing, medical diagnostics, or industrial monitoring). Our results demonstrate that a digital twin need not be continuously synchronized to remain useful; sparse, event-driven corrections can maintain acceptable accuracy at a fraction of the update cost.

Several limitations should be noted. All results are obtained for the canonical Lorenz-63 system; generalization to higher-dimensional chaotic systems (e.g., Lorenz-96 or real-world time-series) remains to be tested. Regarding the influence of attractor topology, the two-sheeted nature of the Lorenz attractor causes abrupt inter-wing transitions in autonomous RC predictions, making it an ideal case for demonstrating the necessity of data injections. For systems with band-type attractors, such as the Rössler system, trajectory divergence is more gradual. Thus, we hypothesize that while the qualitative trade-offs between NRMSE and N_{inj} would remain unchanged, the specific error-triggered injection frequencies may differ depending on the attractor geometry. A detailed investigation of other chaotic systems is a promising direction for future work. Additionally, our study assumes noise-free, full-state observations. In practical settings, measurements are often noisy and partial, which would require extensions such as adaptive observer-based injections [24]. Finally, the thresholds ε_1, ε_2 were optimized offline; online adaptive threshold tuning could further improve performance.

Acknowledgements The article was prepared as part of research project No. FSSW-2026-0008 "Development of fundamental principles for constructing digital twins of infrastructure systems in the data economy" using funds from a subsidy from the federal budget within the framework of the state assignment of the Ministry of Education and Science of Russia.

Data Availability No data are associated in the manuscript.

References

1. J. Pathak, B. Hunt, M. Girvan, Z. Lu, E. Ott, Model-free prediction of large spatiotemporally chaotic systems from data: a reservoir computing approach. Phys. Rev. Lett. **120**(2), 024102 (2018)
2. A. Ermis, Chaos theory in digital twin-based facility management for maintenance and retrofitting. Master's thesis Politecnico di Milano, (2025)
3. H. Jaeger, H. Haas, Harnessing nonlinearity: predicting chaotic systems and saving energy in wireless communication. Science **304**(5667), 78–80 (2004)
4. Z. Lu, B.R. Hunt, E. Ott, Reservoir computing: a tutorial. IEEE Trans. Neural Netw. Learn. Syst. **33**(2), 506–525 (2021)
5. M. Yan, C. Huang, P. Bienstman, P. Tino, W. Lin, J. Sun, Emerging opportunities and challenges for the future of reservoir computing. Nat. Commun. **15**(1), 2056 (2024)
6. J. Pathak, Z. Lu, B.R. Hunt, M. Girvan, E. Ott, Using machine learning to replicate chaotic attractors and calculate lyapunov exponents from data. Chaos Interdisc. J. Nonlinear Sci. **27**(12) (2017)
7. A.A. Emelianova, O.V. Maslennikov, V.I. Nekorkin, Higher-order interactions, adaptivity, and phase transitions in a novel reservoir computing model. Chaos Interdisc. J. Nonlinear Sci. **35**(10), (2025)
8. A.E. Hramov, N. Kulagin, A.N. Pisarchik, A.V. Andreev, Strong and weak prediction of stochastic dynamics using reservoir computing. Chaos Interdisc. J. Nonlinear Sci. **35**(3) (2025)
9. A. Badarin, A. Andreev, V. Klinshov, V. Antipov, A.E. Hramov, Hidden data recovery using reservoir computing: Adaptive network model and experimental brain signals. Chaos Interdisc. J. Nonlinear Sci. **34**(10) (2024)
10. A. Badarin, V. Klinshov, P. Smelov, A. Andreev, Reservoir computing reconstructs blood-oxygen-level-dependent signals: whole-brain modeling study. Eur. Phys. J. Spec. Top. **234**(15), 4497–4504 (2025)
11. Z. Li, A. Andreev, A. Hramov, O. Blyuss, A. Zaikin, Novel efficient reservoir computing methodologies for regular and irregular time series classification. Nonlinear Dyn. **113**(5), 4045 (2024)
12. A.V. Andreev, A.N. Pisarchik, N. Kulagin, R. Jaimes-Reátegui, G. Huerta-Cuellar, A.A. Badarin, A.E. Hramov, Stochastic cloning of dynamical systems with hidden variables. Phys. Rev. E **112**(1), 015303 (2025)
13. N. Kulagin, A. Andreev, A.A. Koronovskii, O.I. Moskalenko, A.P. Sergeev, A. Badarin, A.E. Hramov, Intermittency in predicting the behavior of stochastic systems using reservoir computing. Phys. Rev. E **111**(2), 024209 (2025)
14. N. Kulagin, A. Andreev, A. Hramov, Analysis of the effect of a recurrent neural network size on modeling and prediction accuracy of a stochastic fitzhugh-nagumo neuron. Bull. Russ. Acad. Sci. Phys. **89**(3), 471–475 (2025)
15. W. Wu, L. He, J. Zhou, Z. Xuan, S. Arik, Disturbance-term-based switching event-triggered synchronization control of chaotic lurie systems subject to a joint performance guarantee. Commun. Nonlinear Sci. Numer. Simul. **115**, 106774 (2022)
16. S. Liu, L. Zhou, Network synchronization and application of chaotic lure systems based on event-triggered mechanism. Nonlinear Dyn. **83**(4), 2497–2507 (2016)
17. M. Rani, M. Masroor, P. Kumar, *Remote sensing of ocean and coastal environment-overview. Remote sensing of Ocean and Coastal environments*, pp. 1–15 (2021)
18. X. Zhang, J.E. Medow, B.J. Iskandar, F. Wang, M. Shokoueinejad, J. Koueik, J.G. Webster, Invasive and noninvasive means of measuring intracranial pressure: a review. Physiol. Meas. **38**(8), 143–182 (2017)
19. A.E. Hramov, A.N. Pisarchik, Big data management and quality evaluation for the implementation of ai technologies in smart manufacturing. Appl. Sci. **15**(22), 11905 (2025)
20. R. Grigoriev, M. Cross, H. Schuster, Pinning control of spatiotemporal chaos. Phys. Rev. Lett. **79**(15), 2795 (1997)
21. A.E. Hramov, A.A. Koronovskii, M.K. Kurovskaya, A.A. Ovchinnikov, S. Boccaletti, Length distribution of laminar phases for type-i intermittency in the presence of noise. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. **76**(2), 026206 (2007)
22. A.A. Koronovskii, O.I. Moskalenko, A.E. Hramov, Nearest neighbors, phase tubes, and generalized synchronization. Phys. Rev. E Stat. Nonlinear Soft Matter Phys. **84**(3), 037201 (2011)
23. E.N. Lorenz, Deterministic nonperiodic flow 1. In: Universality in Chaos, 2nd Edition, pp. 367–378. Routledge, (2017)
24. A. Fradkov, H. Nijmeijer, A. Markov, Adaptive observer-based synchronization for communication. Int. J. Bifurc. Chaos **10**(12), 2807–2813 (2000)