# As Simple as Possible, but Not Simpler: When Reservoir Computing Rivals Deep Learning in Hydrological Prediction

Farzad Hosseini* and Javier Del Ser†‡
*Instituto de Hidráulica Ambiental de la Universidad de Cantabria (IHCantabria), 39011 Santander, Cantabria, Spain
†TECNALIA, Basque Research and Technology Alliance (BRTA), 48160 Derio, Bizkaia, Spain
‡Department of Mathematics, University of the Basque Country (UPV/EHU), 48940 Leioa, Bizkaia, Spain
Email: farzad.hosseini@alumnos.unican.es, javier.delser@tecnalia.com

***Abstract*—Deep learning models – especially Long Short-Term Memory networks (LSTMs) – are strong baselines in research related to rainfall–runoff modeling, but their accuracy typically relies on GPU-intensive training, extensive hyperparameter optimization, and multi-seed ensembling. Echo State Networks (ESNs) offer a lightweight reservoir-computing alternative with efficient training, yet their practical potential for operational *hourly* hydrology is underexplored. In this work we benchmark ESNs against competitive LSTM baselines across three axis: (i) predictive skill across flow regimes and high-flow events, (ii) robustness to hyperparameter choices and seasonal variability, and (iii) compute cost. Using an hourly dataset from 40 humid, flashy catchments located in the Basque Country (North of Spain), we train one tuned local ESN per catchment and compare them with a highly optimized regional LSTM ensemble and 40 tuned local LSTMs. Results reveal that well-configured ESNs can approach LSTM skill for flood-relevant dynamics (peak timing/magnitude and wet-season regimes). A cost–skill analysis further indicates that ESNs achieve competitive performance at a fraction of the computational budget (CPU-scale training vs cluster GPU-centric LSTMs). Our findings clarify when ESNs are a sufficient operational choice and when the added complexity of LSTMs yields gains of practical value.**

***Index Terms*—Echo State Networks, Reservoir Computing, Long Short-Term Memory Networks, Hydrological prediction.**

## I. Introduction

Accurate rainfall–runoff modeling underpins flood forecasting, reservoir operation, and early-warning services, particularly in humid flashy catchments where peak magnitude and timing errors have disproportionate operational consequences [1]–[3]. Over the last decade, hydrology has increasingly embraced data-driven modeling, facilitated by expanded data availability and advances in computational hardware, and driven by the need to maintain predictive accuracy under nonstationary forcing conditions. [4]–[6]. Large-sample benchmarks have further accelerated progress by enabling consistent comparisons across basins, regimes, and evaluation metrics [7]–[9].

Among data-driven approaches, LSTMs [10] are nowadays widely used for rainfall–runoff modeling, consistently achieving strong performance across diverse hydro-climatic settings [3], [4], [7]. However, their skill is sensitive to experimental design and typically improves with systematic hyperparameter optimization (HPO) and ensembling to enhance robustness across basins and hydrological events [2], [11]. For instance, in catchments located in the Basque Country (northern Spain), such strategies have been proven to deliver high hourly accuracy and supported post-hoc analyses of learned hydrological signatures, including explainability-oriented diagnostics [12].

On the negative side, predictive performance gains achieved by LSTM-based methodologies often demand GPU devices and long wall-clock times, making cost and energy tangible constraints in applied hydrology and "green" modeling [13]–[16]. Accordingly, training and inference time should be reported to properly assess hardware dependence alongside accuracy performance. This is operationally critical in basin-network deployments where latency, scalability, and limited GPU availability can determine feasibility. In this context, ESNs [17], a core family of approaches within the Reservoir Computing (RC) paradigm, offer an efficiency-oriented alternative for modeling recurrent data. ESNs keep a set of randomly initialized, recurrently connected neurons (*reservoir*), and train only a linear readout, reducing learning to regularized linear regression rather than backpropagation through time [18], [19]. This enables fast, low-memory training while preserving nonlinear temporal representation capacity. This efficiency has, in turn, motivated recent advances aimed at enhancing RC robustness without compromising its computational advantages [20], [21].

Despite early coarse-resolution hydrological investigations [22] and scattered applications (often at monthly/annual scales or hybrid pipelines [23], [24]), ESNs remain underexplored for *operational hourly* rainfall–runoff modeling. Moreover, the cost–skill trade-off relative to strong LSTM baselines is still not well quantified in the literature related to hydrological data modeling. We address this gap with a systematic ESN–LSTM benchmark for hourly prediction across 40 humid, flashy Basque catchments. By considering optimized regional and local LSTM ensembles [2], [11] and tuned local ESNs, our study compares: (i) hydrological skill across regimes,

J. Del Ser acknowledges funding support from the Basque Government through EMAITEK and ELKARTEK funding grants.

emphasizing seasonality and high-flow peaks; (ii) robustness across seasons/events and hyperparameter landscapes; and (iii) computational efficiency (training/inference time and hardware). Our ultimate goal is to shed light on whether the high computational cost required by LSTM training and HPO gives rise to performance gains of practical operational value for hydrological prediction in water resources management.

The rest of the paper is structured as follows: first, Section II reviews related work on deep learning for rainfall–runoff modeling and reservoir computing, and states the contributions of this study. Section III describes the study area, dataset, and the machine learning models considered. Section IV details the experimental setup, including research questions (RQs), hyperparameter optimization, and evaluation protocols. Section V presents and discusses the results in terms of predictive skill, robustness, and computational cost. Section VI concludes the paper and outlines future research directions. A repository with the data and scripts producing the results reported in this research will be linked in the manuscript upon acceptance.

*Data and code availability*: The data and code supporting the findings of this study are publicly available:

- Dataset repository: https://zenodo.org/records/15080569
- Results repository: https://zenodo.org/uploads/19310483
- ESN implementation: https://github.com/farzadhoseini/as.simple.as.possible.ESNs
- Regional LSTM ensemble code: https://github.com/farzadhoseini/ensemble.deep.learning

## II. Related Work and Contribution

To position this study, we briefly summarize prior work on Deep Learning for rainfall–runoff modeling (Subsection II-A), RC and ESNs (Subsection II-B), and ESN applications in hydrology (Subsection II-C). We then state the contribution of this work framed within the reviewed literature.

### A. Deep learning for rainfall–runoff modeling

Research established that Deep Learning (e.g., LSTMs) can learn transferable hydrological behavior from meteorological forcings and catchment attributes, often outperforming traditional baselines on standard hydrological scores, such as the Nash-Sutcliffe (NSE) or the Kling-Gupta (KGE) Efficiency coefficients [3], [4], [7], [8]. Subsequent work has progressed along three main directions: (i) constraint-aware architectures (e.g., physics-informed AI) to improve plausibility and regime robustness [25], [26]; (ii) event-focused evaluation to expose peak and high-flow failure modes important in operations [27]; and (iii) systematic HPO and ensembling to increase accuracy [2], [11]. Explainability studies further relate model behavior to hydrological signatures and internal dynamics [12], [26]. However, computational cost is still treated as a secondary aspect, despite widespread reliance on GPU-centric pipelines.

### B. Reservoir Computing and Echo State Networks

RC leverages random recurrent dynamics by keeping the reservoir weights fixed and training only a lightweight readout, which greatly simplifies learning while preserving rich temporal representations [17], [18], [20], [21]. ESNs formalize RC through the echo state property and guidelines on spectral radius, sparsity, and input scaling to ensure stable yet expressive dynamics, as well as tunable memory depth [17], [19].

Over the years, architectural advances have included leaky-integrator ESNs for finer control of time scales [28], deep ESNs that stack multiple reservoirs [29], attention-based ESN [30], and structured or hardware-oriented reservoirs (e.g., orthogonal, delay-based, or nanoelectronic implementations) to improve conditioning, efficiency, and physical realizability [31]. These variants have been successfully applied to many applications, including nonlinear identification in industrial settings [32], renewable energy prediction [33] and speech processing [34], among others.

### C. ESNs in hydrology

ESN applications in hydrology remain relatively sparse to date. Early work reported operative rainfall–runoff performance [22]. More recent studies embed ESNs in hybrid or ensemble pipelines, frequently focusing on daily/monthly/annual time-steps or site-specific tasks and emphasizing accuracy without a controlled cost–skill comparison against strong deep learning baselines [23], [24], [35]. ESN variants have also been used for precipitation nowcasting and event prediction [36], [37]. Overall, few studies provide an ESN-LSTM benchmark that controls tuning, evaluates regime- and event-relevant diagnostics, and reports compute requirements in an operationally actionable manner.

*Contribution.* Our study offers three main contributions. First, it establishes strong baselines under a fair protocol by conducting an *apples-to-apples* benchmark between local ESN models and optimized regional and local LSTM counterparts for hourly hydrological prediction. Second, it provides operational diagnostics through a comprehensive robustness analysis across seasonal regimes and extracted high-flow event windows. Finally, it delivers a cost–skill quantification, systematically accounting for computational requirements and demonstrating the significant low-cost advantage of ESNs over GPU-intensive LSTM pipelines.

## III. Materials and Methods

We now proceed by outlining the experimental framework. Subsection III-A describes the study area, dataset, and data partitioning strategy. Subsection III-B summarizes the implementation of the ML models considered (ESNs and LSTMs).

### A. Study area, dataset, and data splitting

We consider 40 humid flashy catchments in the Basque Country (Spain; Fig. 1), with a climate influenced by the North Atlantic and the Cantabrian Mountains. The study domain covers $\sim$ 4,494 km$^2$ and catchment areas span $\sim$ 4–1000 km$^2$. We use a two-decade hourly hydro-meteorological dataset [38] with lumped precipitation, temperature, and potential evapotranspiration (PET) as inputs. Each model was implemented as a dual-output regressor with two targets: Water Level

(WL) and Streamflow (SF). While accuracy was comparable across targets, the models, in general, perform better on WL because it corresponds to direct sensor observations, allowing an evaluation against physically observed data. By contrast, SF is typically obtained through a rating-curve transformation from WL, which embeds additional uncertainty due to human-defined *rating curve* fitting and hydraulic complexity. Focusing on WL reduces evaluation ambiguity and supports a reliable performance comparison under operational constraints.

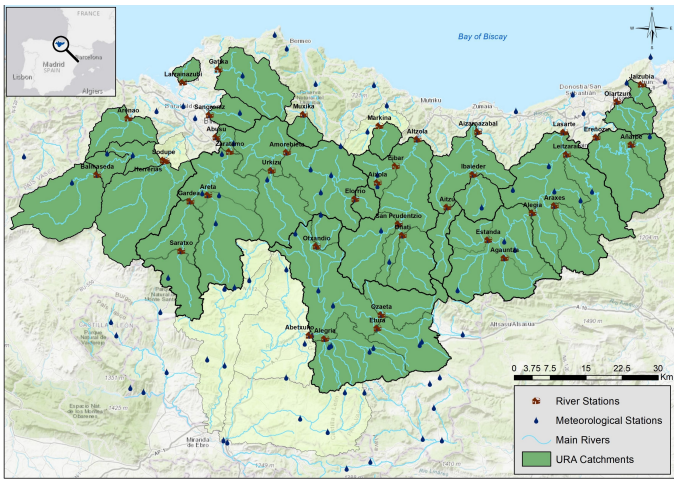


Fig. 1. Study area and 40 Basque catchments considered in this work.

We adopt a strict temporal split into an HPO block (training/validation) and an independent test block. The HPO period spans 1 Oct 2000–30 Sep 2015. We follow the protocol in [2] using an early-period validation block (2000-2005). The use of a long training period is intended to expose the models to a wide range of hydrological variability, including rare and extreme events. While long-term non-stationarity may affect predictive performance, this setup reflects realistic operational conditions in which all available historical data is leveraged for model training. The test period (1 Oct 2015–30 Sep 2021) is fully withheld from model learning and used once for final evaluation.

### B. Machine Learning models

As mentioned in the introduction, we benchmark two recurrent approaches for hourly rainfall–runoff prediction:

- LSTMs, which are gated Recurrent Neural Networks (RNNs) that learn long-range dependencies via input/forget/output gates [10]. We use the multi-timescale LSTM (mtsLSTM) architecture for hourly modeling [3] and train end-to-end with gradient-based optimization. We configure all LSTMs as sequence-to-one forecasters that map a fixed-length lookback window of meteorological forcings to the next-step hydrological targets. For mtsLSTM, the lookback is defined at two temporal resolutions: an hourly sequence length ($seq_h$) and a daily sequence length ($seq_d$). In HPO, we tune the key architectural/regularization hyperparameters reported in Table I: $(seq_h, seq_d)$, hidden size ($h$), dropout ($drop$), ($noise$), forget-gate bias ($b_f$), and the regularization scheme ($reg$). HPO protocols and the evaluation setup for final configurations are described in Section IV. Two LSTMs were employed as benchmarks:
  - *Regional* LSTM: a single model trained jointly across all catchments (labeled as R-LSTM in the benchmark).
  - *Local* LSTMs: one model trained independently per catchment (referred to as L-LSTM in the benchmark).
- ESNs, in which a fixed random recurrent reservoir produces a high-dimensional state from the input sequence, and only a linear readout is trained [19]. Training reduces to regularized linear regression, enabling fast CPU-scale fitting and broad hyperparameter exploration at low memory cost. Here ESNs are only trained locally per catchment, and implemented as a $N$-sized pool of leaky-integrator neurons with random recurrent connections (connectivity $c$), leaking rate $\alpha$, spectral radius $\rho$ and a Ridge readout with parameter $\lambda$.

Both map meteorological forcings to hydrological responses, but they differ fundamentally in how recurrent dynamics are learned and, therefore, in computational cost. All experiments use identical inputs/targets and the temporal split defined in Subsection III-A.

## IV. Experimental Setup

Our experiments answer three research questions (RQs):

- **RQ1** (*Skill*): Can carefully tuned ESNs approach the predictive performance of LSTMs across catchments?
- **RQ2** (*Robustness*): How do ESNs and LSTMs differ in robustness across seasons, flow regimes, and extreme events, and how stable are their performances across the hyperparameter landscape?
- **RQ3** (*Cost-skill*): What is the cost-skill trade-off between ESNs and LSTMs in terms of training and inference time, hardware requirements, and effective model complexity required to achieve a target level of hydrological performance?

*Hyperparameter optimization and ensembling.* Model selection is performed via random-search HPO [39]. Random search was selected due to its strong empirical performance in high-dimensional hyperparameter spaces and its scalability across thousands of configurations in hydrology domain. For LSTMs, we evaluate 1,000 random configurations for the regional model and 150 configurations per catchment for local models; ESNs are tuned locally with 150 random configurations per catchment. In addition to selecting a single best configuration, we evaluate ensemble predictors built from the top-performing configurations (median of predictions on each timestep across the top-10 HP optimized members) to assess whether aggregation improves robustness. A compact HPO summary (search budget and selected configurations) is provided in Table I.

*Evaluation protocol.* When it comes to the predictive performance of the tuned models (RQ1), we compute classical hydrological scores [9] (NSE and KGE) over the held-out test period. We report both NSE and KGE because they

TABLE I
HPO SUMMARY: SEARCH BUDGET, AND KEY TUNED HYPER PARAMETER VALUES FOR SELECTED CONFIGURATIONS.

| Model | HPO budget | Selected configuration summary (median [Interquartile range Q1–Q3]) |
|---|---|---|
| R-LSTM (Regional) | 1000 configs | $h = 64$ [32–128], seq=504h [336–4032], drop=0.2 [0.2–0.4], $b_f$=0 [0–3], noise=0.01 [0–0.05], reg=tie-freq (80%) |
| L-LSTM (Local) | 150/basin | $h$=128 [64–256], seq=672h [336–1344], drop=0.2 [0–0.4], $b_f$=3 [1–3], noise=0.02 [0.01–0.05], reg=tie-freq (53%) |
| ESN (Local) | 150/basin | $N = 1,126$ [894–1530], washout $= 134$ [102–190], $\rho = 0.667$ [0.511–0.758], $\alpha = 0.034$ [0.017–0.058], $\lambda = 1.49 \cdot 10^{-5}$ [$1.53 \cdot 10^{-6}$–$8.67 \cdot 10^{-5}$], $c = 0.26$ [0.16–0.33] |

emphasize different aspects of hydrological performance. NSE is strongly influenced by peak errors due to its squared-error formulation, whereas KGE summarizes agreement in correlation, bias, and variability. Reporting both helps distinguish year-to-year changes driven by peak misfit from those driven by bias or variance shifts. For NSE, a value of 1 indicates perfect agreement, 0 corresponds to the skill of a predictor that always outputs the observed mean, and negative values indicate performance worse than this mean-flow baseline. For KGE, 1 indicates perfect agreement; lower values indicate poorer overall agreement in correlation, bias, and variability.

Robustness (RQ2) is assessed via (i) temporal stratification (water-year and monthly seasonality), (ii) event-focused evaluation over extracted high-flow periods (including basin-level fail-rates), and (iii) sensitivity to hyperparameter choices. High-flow events were extracted from the observed flow in the test period using a percentile threshold. Specifically, an hourly time step is flagged as "high-flow" when observed flow exceeds the basin-specific $98^{\text{th}}$ percentile computed over the test period. Consecutive flagged hours are grouped into the same event if gaps are less than or equal to 24 hours; events with a high-flow core shorter than 3 hours are discarded. For each retained event, we evaluate a window expanded by 12 hours (both pre-event and post-event) around the detected high-flow core.

Finally, for the computational cost (RQ3), we report end-to-end wall-clock time for the full benchmark workflow, including (i) random-search HPO over all evaluated configurations and (ii) the final retrain/test stage. For the sake of fair comparisons, we contextualize runtime by the number of trained configurations (HPO trials) and by whether additional retraining/ensembling is required beyond the already trained HPO candidates.

*Computing environments.* (i) *ESNs (Local):* executed on a personal workstation running Windows 11 Enterprise with an Intel Core i7–4790K CPU (4 cores / 8 threads, 4.0 GHz), using ReservoirPy [40] and CPU-only execution. (ii) *L-LSTMs (Local):* executed on a shared Ubuntu 22.04.5 LTS compute server with an NVIDIA Quadro GV100 GPU (32 GB VRAM), an AMD EPYC 7763 64-core CPU, and 768 GB RAM; due to resource sharing, jobs were limited to 25 CPU cores and 16 GB of GPU memory. LSTMs were implemented in PyTorch (2.7.1+cu118) and trained 12 networks in parallel on the cluster. (iii) *R-LSTMs (Regional):* executed on a Neptune supercomputer, a high-performance cluster with 1,296 compute cores, >5 TB RAM, and an InfiniBand FDR10 interconnect. GPU-accelerated training used dedicated GPU nodes equipped with NVIDIA A30 cards (two nodes with 2×A30 each). Our training jobs, due to resource sharing, allocated 8 CPU cores and 16 GB RAM per CPU, and used 2×A30 GPUs to train 12 networks in parallel.

## V. RESULTS AND DISCUSSION

This section summarizes the predictive skill and cost-skill analysis of ESNs relative to LSTM baselines at regional and local settings, using both aggregate metrics and temporal stratification. Our discussion in what follows is guided by the responses to the three RQs formulated previously.

### A. RQ1 – Overall predictive skill

Table II reports median test performance for the prediction of SF and WL across catchments. As can be observed in this table, ESNs achieve competitive hourly prediction results on the full hydrograph (achieving $\text{KGE}_{\text{SF}} = 0.837$ and $\text{NSE}_{\text{SF}} = 0.814$). LSTM baselines usually perform best in this regard, especially for the regional ensemble setting (R-LSTM: $\text{KGE}_{\text{SF}} = 0.895$, $\text{NSE}_{\text{SF}} = 0.914$). The local ensemble LSTM also outperforms ESN on streamflow (L-LSTM: $\text{KGE}_{\text{SF}} = 0.870$, $\text{NSE}_{\text{SF}} = 0.885$), indicating that the efficiency of ESNs comes with a measurable but not prohibitive accuracy gap in this case study. In general, simpler ESNs achieve competitive performance on the full test hydrograph compared with the computationally expensive LSTM baselines. For ESNs, $\text{KGE}_{\text{WL}} = 0.886$ and $\text{NSE}_{\text{WL}} = 0.856$, whereas the regional LSTM reaches $\text{KGE}_{\text{WL}} = 0.927$ and $\text{NSE}_{\text{WL}} = 0.929$. This systematic advantage in WL may reflect that WL is a direct gauge observation, while SF can embed additional uncertainty due to stage–discharge conversion (human-defined rating-curve effects and potential nonstationarities). This is consistent with LSTMs' adaptive gated dynamics.

TABLE II
MEDIAN TEST PERFORMANCE FOR SF AND WL ACROSS CATCHMENTS.

| Model | $\text{KGE}_{\text{SF}}$ | $\text{KGE}_{\text{WL}}$ | $\text{NSE}_{\text{SF}}$ | $\text{NSE}_{\text{WL}}$ |
|---|---|---|---|---|
| ESN | 0.837 | 0.886 | 0.814 | 0.856 |
| Best L-LSTM | 0.861 | 0.901 | 0.860 | 0.880 |
| L-LSTM | 0.870 | 0.900 | 0.885 | 0.895 |
| R-LSTM | 0.895 | 0.927 | 0.914 | 0.929 |

*Note:* ESN, L-LSTM, and R-LSTM denote ensembles built from the best-performing configurations, whereas Best L-LSTM reports the single top-performing local configuration.

### B. RQ2 – Water-year variability and robustness

To assess temporal robustness, we stratify the prediction results corresponding to WL by water years (WY) over the

TABLE III
WATER-YEAR REGIONAL MEDIAN KGE FOR WL OVER THE TEST PERIOD.

| Water year | ESN | L-LSTM | Best L-LSTM | R-LSTM |
|---|---|---|---|---|
| 2015 | 0.894 | 0.899 | 0.893 | 0.935 |
| 2016 | 0.781 | 0.861 | 0.846 | 0.895 |
| 2017 | 0.893 | 0.902 | 0.908 | 0.933 |
| 2018 | 0.842 | 0.869 | 0.848 | 0.917 |
| 2019 | 0.830 | 0.869 | 0.898 | 0.885 |
| 2020 | 0.810 | 0.837 | 0.859 | 0.871 |

test period. Table III and Fig. 2 report water-year median KGE and NSE, respectively, aggregated across all basins. All model families show a coherent performance degradation in challenging years, most notably WY2016 and WY2020, suggesting that part of the variability is driven by hydro-meteo-climatic conditions and/or data-quality effects rather than model-specific failures alone. However, the magnitude of the drop is architecture-dependent: ESNs are the most sensitive models in these years (e.g., $\mathrm{KGE}_{\mathrm{WL}} = 0.781$ and $\mathrm{NSE}_{\mathrm{WL}} = 0.759$ in WY2016), whereas the regional LSTM remains comparatively stable ($\mathrm{KGE}_{\mathrm{WL}} = 0.895$ and $\mathrm{NSE}_{\mathrm{WL}} = 0.938$ in WY2016). Overall, these results indicate that ESNs can achieve competitive hourly skill, but their performance varies across years, suggesting higher sensitivity to non-stationarity in the data-generating process.

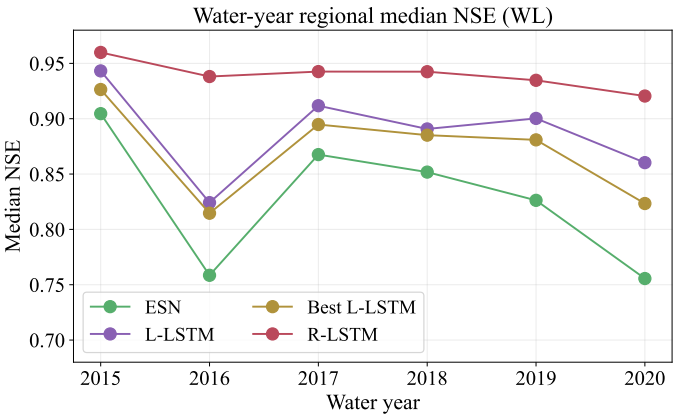


Fig. 2. Water-year regional median NSE for WL (test period).

### C. RQ2 – Monthly performance

We now turn the focus on the intra-annual behavior of the models by aggregating predictive performance per calendar month across the test period.

Fig. 3 reports the monthly representation of the regional median $\mathrm{KGE}_{\mathrm{WL}}$ as a month-by-month seasonality curve. All model families exhibit a consistent seasonal signature: performance is highest during the wet/autumn–winter months (roughly November–March) and decreases toward summer (June–August). This summer degradation may reflect both regime effects (weaker rainfall control and more evapotranspiration/baseflow/operational influences) and metric/observation effects (lower WL variance and higher relative noise at low stages), which we do not disentangle here. However, the amplitude of this seasonal degradation is strongly architecture-dependent. ESNs show the largest summer KGE drop, reaching their minimum around July–August ($\mathrm{KGE}_{\mathrm{WL}} \approx 0.35$ in July and $\approx 0.31$ in August). By contrast, the performance of local LSTM baselines degrades more moderately in the same period (e.g., L-LSTM $\approx 0.52 \rightarrow 0.38$ from July to August). The regional LSTM remains the most stable model overall (R-LSTM $\approx 0.68$ in July and $\approx 0.57$ in August). During the high-skill months (winter), the ESN–LSTM gap narrows, indicating that ESNs can be competitive when hydrometeorological forcing is informative and the WL signal-to-noise ratio is effectively higher, as reflected by the larger seasonal variability. This is crucial for accurate flood prediction in rainy seasons and humid flashy catchments like the ones in Basque Country considered in this paper. Consistent with reduced forcing informativeness in summer, ESNs (fixed reservoir/linear readout) show the largest degradation, while gated LSTMs remain comparatively more stable.

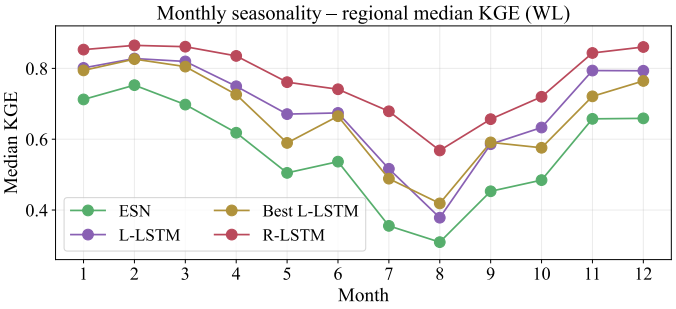


Fig. 3. Monthly WL predictive skill on the test period, summarized as regional medians across basins. The line plot highlights month-to-month differences and the larger summer degradation of ESNs.

### D. RQ2 – Analysis of high-flow events

We next assess robustness on extracted high-flow events using event-wise NSE, and summarize performance through a basin-level *fail-rate*, defined as the fraction of events with NSE below a given threshold.

Fig. 4 reports the Empirical Cumulative Density Function (ECDF) of basin fail-rate for the criterion NSE $\geq 0.5$. Lower fail-rates are better; hence curves that are more left-shifted and higher are preferred). Across basins, the regional LSTM exhibits the lowest fail-rates (most left-shifted ECDF), followed by the local LSTM variants. ESNs show systematically higher fail-rates (right-shifted ECDF), indicating more frequent event failures and weaker cross-basin robustness under this thresholding criterion. Overall, these patterns are consistent with the hypothesis that robustness differences arise from a combination of event difficulty (sharpness of the rising limb, peak magnitude, and post-peak recession) and model sensitivity to hyperparameter configuration. In line with prior evidence, ensembling the top-10 configurations reduces variance relative to selecting a single best configuration [11], supporting the view that part of the robustness gap is mitigated by aggregation (Table II).

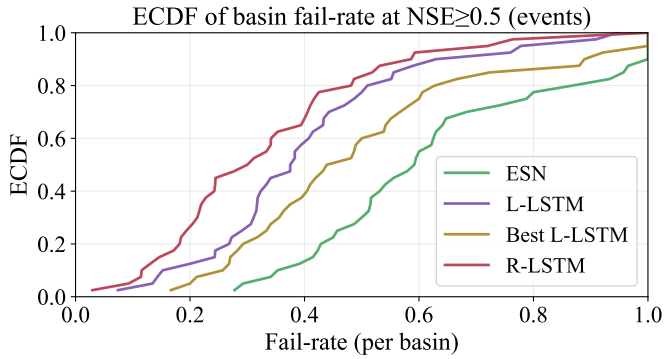


Fig. 4. Empirical CDF of basin fail-rate on high-flow events for the NSE≥0.5 criterion (higher and more left-shifted is better). The regional LSTM is the most robust across basins, while ESNs exhibit higher fail-rates.

A qualitative inspection of event hydrographs supports this statistical picture. When ESNs fail, the dominant error modes are: (i) peak magnitude under/over-estimation during sharp rising limbs, (ii) peak timing offsets (lead/lag) that strongly penalize event-wise NSE; and (iii) less reliable recession tracking immediately after the peak. These behaviors are most apparent in basins with high ESN fail-rates, whereas ESN performance is comparatively stronger in basins with smoother and more repeatable event shapes and/or reliable-data-record observations. By contrast, LSTM-based models generally exhibit better peak alignment and more stable recession behavior, with the regional LSTM being the most robust. The latter model, however, comes along with a substantially higher computational demand, as Section V-E in response to RQ3 will later show.

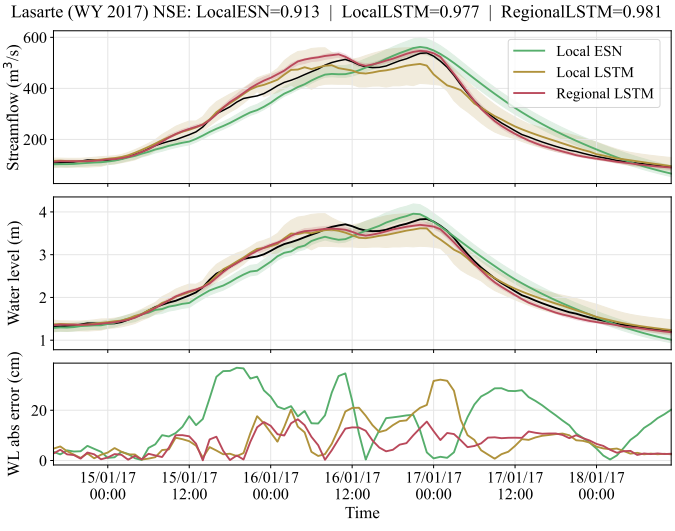


Fig. 5. Illustrative high-flow event in Lasarte catchment. Shown are observed and predicted streamflow (top) and water level (middle), with water-level error (bottom). In this event, the ESN achieves comparable skill to LSTM variants despite substantially lower training cost, illustrating an operational high-flow prediction case by ESNs.

Fig. 5 provides an illustrative example from *Lasarte*, a representative basin with clean and continuous observations in the dataset. This event shows that, when observations are reliable and event dynamics are relatively well-behaved, ESNs can achieve operationally useful levels of performance at a fraction of the computational cost of LSTMs. While this single case should not be over-generalized, it demonstrates a *best-case* regime in which ESNs can be competitive for high-flow event reconstruction.

Overall, ESNs tend to underperform in three main regimes: (i) low-flow summer conditions with weaker forcing signals, (ii) anomalous years characterized by non-stationary dynamics (e.g., WY2016 and WY2020), and (iii) sharp high-flow events with rapid rising limbs and complex recession behavior.

### *E. RQ3 – Computational cost*

To answer RQ3, we report end-to-end *compute* time for the full benchmark workflow (HPO + final retrain/test) implemented over the computing environments detailed previously. The obtained results are normalized by the number of trained configurations, since local and regional protocols differ. Because experiments were executed on different computing systems (ESNs on a personal CPU workstation; local LSTMs on a shared GPU server; regional LSTMs on a private research GPU cluster), absolute runtimes are not strictly comparable across pipelines; we therefore emphasize *time per trained configuration* and reported the hardware context for transparency. Nevertheless, even under heterogeneous compute environments, ESNs consistently require substantially less computational effort per trained configuration than LSTMs, highlighting their suitability when GPU access or energy budgets are constrained. While ESNs and LSTMs were executed on different computing infrastructures due to practical constraints, our comparison focuses on time per trained configuration and overall computational effort. Achieving the level of performance reported in this study typically requires GPU acceleration and extensive hyperparameter optimization. Therefore, our comparison reflects realistic operational settings rather than strictly identical hardware conditions.

Table IV summarizes the resulting cost per configuration. ESNs completed the local multi-basin workflow (150 HPO trials $\times$ 40 basins $=$ 6,000 trained candidates, plus final retrain/test) in $\sim$ 252,512 s ($\approx$ 2.92 CPU-days), corresponding to $\sim$ 42 s/config on the CPU-only personal workstation. In contrast, the analogous local LSTM workflow (150 $\times$ 40 $=$ 6,000 trained configurations, plus final retrain/test) accumulated $\sim$ 834.8 GPU-hours ($\approx$ 34.8 GPU-days) on the shared GV100 server, i.e., $\sim$ 8.35 min/config. This represents an $\approx 12\times$ larger end-to-end compute budget than ESNs for the same configuration count, in addition to requiring GPU resources. For the regional LSTM ensemble, random-search HPO evaluated 1,000 configurations and required $\sim$ 32 GPU-days for training/validation; the final retrain/test stage added 1.78 GPU-days, yielding an end-to-end total of $\sim$ 33.78 GPU-days ($\approx$ 48.6 min/config) on Neptune with 2$\times$A30 GPUs and parallel training.

On a note closing this last analysis, it is important to state that at an hourly resolution, inference is not the bottleneck for

TABLE IV
COMPUTE BREAKDOWN (WORKFLOW = HPO + FINAL RETRAIN/TEST). TIMES ARE END-TO-END WALL-CLOCK AND NORMALIZED BY THE NUMBER OF TRAINED CONFIGURATIONS.

| Pipeline | # configs | Hardware | Total time | Time/config |
|---|---|---|---|---|
| ESN (Local) | 6,000 | CPU (Env-A: i7–4790K) | $\sim$ 2.92 CPU-days | $\sim$ 42 s |
| L-LSTM (Local) | 6,000 | GPU (Env-B: Quadro GV100) | $\sim$ 34.8 GPU-days | $\sim$ 8.35 min |
| R-LSTM (Regional) | 1,000 | GPU (Env-C: 2×A30) | $\sim$ 33.78 GPU-days | $\sim$ 48.6 min |

none of the model families considered in the benchmark. The dominant cost is training/HPO. This makes ESNs attractive under limited computing access, strict wall-clock constraints, or tight energy budgets in operational hydrology.

## VI. CONCLUSIONS AND FUTURE RESEARCH

This study has benchmarked tuned CPU-trained ESNs against optimized GPU-trained LSTM baselines for hourly hydrological prediction across 40 catchments located in the Basque Country (Spain). To this end, we have proposed a consistent operational-style evaluation that included full-hydrograph predictive performance, water-year robustness, monthly/seasonal diagnostics, high-flow event analyses, and computational costs. Our results have revealed that the regional LSTM (R-LSTM) provides the strongest overall predictive skill, while local LSTMs (L-LSTM) remain competitive but more variable across basins and years. ESNs achieve *hydrologically acceptable* performance levels for operational use. Despite a systematic performance gap compared to LSTM-based counterparts, we have demonstrated that ESNs can still support decision making in hydrological asset management, e.g., triggering threshold-based alerts, informing pump/gate operations, and prioritizing inspection and maintenance actions across multiple basins.

A key finding in our study is that model robustness is not uniform in time. All model families deteriorated in challenging water years (e.g., WY2016 and WY2020), suggesting that part of the performance variability is driven by atypical hydrological conditions and temporal non-stationarity, rather than purely model-specific factors. However, the magnitude of the degradation was noted to be architecture-dependent: ESNs exhibited the largest drops, suggesting a higher sensitivity to temporal shifts and exogenous changes (e.g., sensor/device replacement, rating-curve updates, or operational effects). In addition, all models predicted water level more reliably than streamflow. This observation is consistent with the fact that the observed water level is a direct measurement, whereas streamflow derived through human-defined rating curves that may introduce non-stationary uncertainties. LSTM baselines appeared to be capable of better modeling such latent and time-varying relationships, while ESNs were comparatively less adaptive in this regard.

When it comes to computational effort, ESNs offered a major practical advantage: the full ESN workflow completed in $\sim$ 2.9 CPU-days, whereas the LSTM workflow required more than 33 GPU-days, i.e., an $\sim 12\times$ larger compute-time budget. This efficiency reinforces the core message of this study: in operational hydrology, simplicity can rival complexity when it matters most. ESNs emerge as a fast, low-cost option for rainfall-runoff forecasting pipelines under constraints of GPU access, wall-clock time, or energy budgets, positioning them as a credible baseline or deployment choice. While LSTMs retain a plus in accuracy and robustness, especially during challenging years and high-flow events, such performance gains do not necessarily translate into better flood response by water management decisions. Instead, they demand resources and tuning latencies that, in most operational contexts, are simply not worth it. Closing the remaining skill and robustness gap will likely require strategies beyond merely enlarging reservoirs, such as explicit handling of non-stationarity, missingness and measurement artifacts (e.g., gaps, spikes, drift), and event-driven dynamics. In short, our findings illustrate that Reservoir Computing can be "*as simple as possible, but not simpler*", offering a compelling modelling alternative when rapid iteration and sustainable deployment are priorities.

*Future research directions.* Future work should prioritize making ESNs more reliable for applied hydrology by improving robustness to non-stationarity and potential observation issues (e.g., missing data, drift, artifacts), explicitly accounting for rating-curve uncertainty when estimating streamflow alongside water level, and introducing event-aware calibration to reduce peak-related errors. In this study, statistical significance testing was not conducted, and results are based on a single regional dataset. Therefore, conclusions should be interpreted as indicative within this context rather than universally generalizable. Additional gains may come from richer yet efficient reservoir designs (e.g., leaky/stacked or Deep ESNs) while preserving CPU-scale training, and from operational uncertainty quantification (e.g., multi-seed ensembles, Bayesian readouts, conformal prediction). Additionally, *regional* (multi-basin) ESNs that share representations across catchments may reduce variance and improve generalization, mirroring the accuracy gains observed for R-LSTMs. Extending ESNs to regional (multi-basin) settings is non-trivial, as it requires shared representations or conditioning mechanisms not inherent to standard ESN formulations; this remains an important direction for future work. ESN-specific interpretability [41] via readout attributions traced back through the reservoir states can help link learned dynamics to hydrological regimes, diagnose failure modes, and guide minimal architectural upgrades. Finally, the generality of the compute-skill trade-off should be tested on larger datasets and longer lead times to assess whether our conclusions generalize to other hydrological regimes.

## REFERENCES


[1] K. Beven, "Deep learning, hydrological processes and the uniqueness of place." *Hydrological Processes*, vol. 34, no. 16, 2020.

[2] F. Hosseini, C. Prieto, and C. Álvarez, "Hyperparameter optimization of regional hydrological LSTMs by random search: A case study from Basque Country, Spain," *Journal of Hydrology*, vol. 643, p. 132003, 2024.

[3] M. Gauch *et al.*, "Rainfall–runoff prediction at multiple timescales with a single Long Short-Term Memory network," *Hydrology and Earth System Sciences*, vol. 25, no. 4, pp. 2045–2062, 2021.

[4] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger, "Rainfall–runoff modelling using long short-term memory (LSTM) networks," *Hydrology and Earth System Sciences*, vol. 22, no. 11, pp. 6005–6022, 2018.

[5] C. Shen and K. Lawson, "Applications of deep learning in hydrology," *Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences*, pp. 283–297, 2021.

[6] K. P. Tripathy and A. K. Mishra, "Deep learning in hydrology and water resources disciplines: concepts, methods, applications, and research directions," *Journal of Hydrology*, vol. 628, p. 130458, 2024.

[7] F. Kratzert *et al.*, "Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets," *Hydrology and Earth System Sciences*, vol. 23, no. 12, pp. 5089–5110, 2019.

[8] F. Kratzert, M. Gauch, D. Klotz, and G. Nearing, "HESS Opinions: Never train a Long Short-Term Memory (LSTM) network on a single basin," *Hydrology and Earth System Sciences*, vol. 28, no. 17, pp. 4187–4201, 2024.

[9] M. Gauch *et al.*, "In defense of metrics: Metrics sufficiently encode typical human preferences regarding hydrological model performance," *Water Resources Research*, vol. 59, no. 6, p. e2022WR033918, 2023.

[10] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[11] F. Hosseini, C. Prieto, and C. Álvarez, "Ensemble learning of catchment-wise optimized LSTMs enhances regional rainfall-runoff modelling-case study: Basque Country, Spain," *Journal of Hydrology*, vol. 646, p. 132269, 2025.

[12] F. Hosseini, C. Prieto, and C. Álvarez, "An explainable AI approach for interpreting regionally optimized deep neural networks in hydrological prediction," *Journal of Hydrology*, vol. 661, p. 133689, 2025.

[13] E. Strubell, A. Ganesh, and A. McCallum, "Energy and policy considerations for deep learning in NLP," in *57th Annual Meeting of the Association for Computational Linguistics*, 2019, pp. 3645–3650.

[14] R. Schwartz, J. Dodge, N. A. Smith, and O. Etzioni, "Green AI," *Communications of the ACM*, vol. 63, no. 12, pp. 54–63, 2020.

[15] P. Henderson *et al.*, "Towards the systematic reporting of the energy and carbon footprints of machine learning," *Journal of Machine Learning Research*, vol. 21, no. 248, pp. 1–43, 2020.

[16] D. Patterson *et al.*, "Carbon emissions and large neural network training," *arXiv:2104.10350*, 2021.

[17] H. Jaeger, "The echo state approach to analysing and training recurrent neural networks," *German National Research Center for Information Technology gmd, Technical Report 148.34*, 2001.

[18] W. Maass, T. Natschläger, and H. Markram, "Real-time computing without stable states: A new framework for neural computation based on perturbations," *Neural Computation*, vol. 14, no. 11, pp. 2531–2560, 2002.

[19] M. Lukoševičius, "A practical guide to applying echo state networks," in *Neural Networks: Tricks of the Trade: Second Edition*. Springer, 2012, pp. 659–686.

[20] C. Sun *et al.*, "A systematic review of echo state networks from design to application," *IEEE Transactions on Artificial Intelligence*, vol. 5, no. 1, pp. 23–37, 2022.

[21] J. Del Ser *et al.*, "Deep echo state networks for short-term traffic forecasting: Performance comparison and statistical assessment," in *IEEE International Conference on Intelligent Transportation Systems (ITSC)*, 2020, pp. 1–6.

[22] N. De Vos, "Echo state networks as an alternative to traditional artificial neural networks in rainfall–runoff modelling," *Hydrology and Earth System Sciences*, vol. 17, no. 1, pp. 253–267, 2013.

[23] Y. Tian, Y.-P. Xu, Z. Yang, G. Wang, and Q. Zhu, "Integration of a parsimonious hydrological model with recurrent neural networks for improved streamflow forecasting," *Water*, vol. 10, no. 11, p. 1655, 2018.

[24] J. Wang *et al.*, "A hybrid annual runoff prediction model using echo state network and gated recurrent unit based on sand cat swarm optimization with markov chain error correction method," *Journal of Hydroinformatics*, vol. 26, no. 6, pp. 1425–1453, 2024.

[25] J. Donnelly, A. Daneshkhah, and S. Abolfathi, "Physics-informed neural networks as surrogate models of hydrodynamic simulators," *Science of the Total Environment*, vol. 912, p. 168814, 2024.

[26] B. Heudorfer, H. V. Gupta, and R. Loritz, "Are deep learning models in hydrology entity aware?" *Geophysical Research Letters*, vol. 52, no. 6, p. e2024GL113036, 2025.

[27] K. Xie *et al.*, "Physics-guided deep learning for rainfall-runoff modeling by considering extreme events and monotonic relationships," *Journal of Hydrology*, vol. 603, p. 127043, 2021.

[28] H. Jaeger, M. Lukoševičius, D. Popovici, and U. Siewert, "Optimization and applications of echo state networks with leaky-integrator neurons," *Neural Networks*, vol. 20, no. 3, pp. 335–352, 2007.

[29] C. Gallicchio, A. Micheli, and L. Pedrelli, "Deep reservoir computing: A critical experimental analysis," *Neurocomputing*, vol. 268, pp. 87–99, 2017.

[30] C. Liu, R. Yao, L. Zhang, and Y. Liao, "Attention based echo state network: A novel approach for fault prognosis," in *11th International Conference on Machine Learning and Computing*, 2019, pp. 489–493.

[31] G. Tanaka *et al.*, "Recent advances in physical reservoir computing: A review," *Neural Networks*, vol. 115, pp. 100–123, 2019.

[32] S. Dettori, I. Matino, V. Colla, and R. Speets, "Deep echo state networks in industrial applications," in *IFIP International Conference on Artificial Intelligence Applications and Innovations*, 2020, pp. 53–63.

[33] J. Del Ser *et al.*, "Randomization-based machine learning in renewable energy prediction problems: Critical literature review, new results and perspectives," *Applied Soft Computing*, vol. 118, p. 108526, 2022.

[34] M. Khan, A. El Saddik, F. S. Alotaibi, and N. T. Pham, "AAD-Net: Advanced end-to-end signal processing system for human emotion detection & recognition using attention-based deep echo state network," *Knowledge-Based Systems*, vol. 270, p. 110525, 2023.

[35] W.-C. Wang *et al.*, "A stacking ensemble machine learning model for improving monthly runoff prediction," *Earth Science Informatics*, vol. 18, no. 1, p. 120, 2025.

[36] L. N. Moncada Morales, M. Bonas, S. Castruccio, and P. Crippa, "Forecasting high resolution precipitation events with logistic echo state networks," *Journal of Geophysical Research: Machine Learning and Computation*, vol. 1, no. 3, p. e2024JH000291, 2024.

[37] M. Funato and Y. Sawada, "Multi-model ensemble and reservoir computing for river discharge prediction in ungauged basins," *arXiv:2507.18423*, 2025.

[38] F. Hosseini, "URA dataset: 40 Basque Country catchments hourly hydrometeorological data + catchments attributes," 10.5281/zenodo.15080569, 2025.

[39] J. Bergstra and Y. Bengio, "Random search for hyper-parameter optimization," *Journal of Machine Learning Research*, vol. 13, p. 281–305, 2012.

[40] N. Trouvain, L. Pedrelli, T. T. Dinh, and X. Hinaut, "Reservoirpy: an efficient and user-friendly library to design echo state networks," in *International Conference on Artificial Neural Networks*. Springer, 2020, pp. 494–505.

[41] A. Barredo Arrieta, S. Gil-Lopez, I. Laña, M. N. Bilbao, and J. Del Ser, "On the post-hoc explainability of deep echo state networks for time series forecasting, image and video classification," *Neural Computing and Applications*, vol. 34, no. 13, pp. 10 257–10 277, 2022.