https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

Everything everywhere all at once: A single-cell LSTM network unifying multi-frequency, missing data, and discharge assimilation for robust operational flood forecasting

Eduardo Acuña Espinoza[1, 6], Manuel Álvarez Chaves[2], Frederik Kratzert[3], Daniel Klotz[4, 3], Martin Gauch[5], Robert Lang[6], Dominik Elfgang[7], Alexander Dolich[1], Ralf Loritz[1], and Uwe Ehret[1]

[1]Institute of Water and Environment, Karlsruhe Institute of Technology (KIT), Karlsruhe, Germany
[2]Earth and Environmental Sciences, University of Waterloo, Waterloo, Canada
[3]Google Research, Vienna, Austria
[4]Machine Learning in Earth Science, Interdisciplinary Transformation University Austria, Linz, Austria
[5]Google Research, Zurich, Switzerland
[6]Hydron GmbH, Karlsruhe, Germany
[7]Hochwasservorhersagezentrale, Karlsruhe, Germany

Correspondence: Eduardo Acuña Espinoza (eduardo.espinoza@kit.edu)

Abstract. Long short-term memory (LSTM) networks have demonstrated state-of-the-art capabilities in operational flood forecasting, with recent missing-data handling strategies further improving pipeline robustness. Building on these advancements, we introduce the multi-frequency masked-forecasting LSTM (MF^{2}LSTM), a model architecture that combines missing-data workflows with multi-frequency approaches to generate hourly streamflow forecasts. Additionally, our framework features a flexible data assimilation strategy to include real-time discharge information. This approach enhances forecast performance when real-time observations are available, and allows the model to keep operating when the discharge signal is absent, either by gauge failure or for prediction in ungauged basins. We benchmarked the MF^{2}LSTM against the Large Area Runoff Simulation (LARSIM) model, the current operational flood-forecasting model used by several European countries, forcing both models with ICON-D2 meteorological products. Our results indicate that the MF^{2}LSTM yields higher predictive accuracy than LARSIM. Overall, this framework presents a robust operational pipeline that demonstrates the viability of deep learning for hourly forecasting. Furthermore, by relying on a standard single-cell LSTM architecture, the approach highlights that structurally simple deep learning architectures can achieve high operational performance.

1 Introduction

Long short-term memory (LSTM) networks (Hochreiter and Schmidhuber, 1997) have demonstrated state-of-the-art performance in rainfall-runoff hydrological modeling (Kratzert et al., 2019; Lees et al., 2021; Klotz et al., 2022; Loritz et al., 2024), establishing significant potential for operational flood forecasting at daily timescales (Nearing et al., 2024). To further improve resilience in day-to-day deployment, recent research has focused on enhancing the robustness of these frameworks by handling missing input data (Gauch et al., 2025) and generalizing these strategies into operational forecasting pipelines (Cohen et al., 2026). Building on this operational foundation, we propose two further developments to expand the capabilities of these models.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

The first development addresses temporal resolution. Standard daily-resolution models can be limited in capturing peak magnitudes and precise flood timing due to daily input aggregation, a drawback particularly pronounced in small, fast-responding catchments. In the past, high-quality large-sample datasets at hourly resolution were rare (Klingler et al., 2021; Xia et al., 2012), hindering the development of these high-frequency models. However, such datasets have recently become available (Dolich et al., 2026; Coxon et al., 2026; Tran et al., 2025; Bushra et al., 2025; Nijzink et al., 2025). Although multi-frequency LSTM architectures (Acuña Espinoza et al., 2025; Gauch et al., 2021) were primarily proposed to address the computational challenges associated with long temporal sequences, they also provide a framework for integrating high-frequency observations into operational forecasting systems. To date, however, these architectures have not been combined with missing-data handling in a unified operational framework.

The second development focuses on discharge assimilation, where real-time river discharge is injected into the model. Data-assimilation strategies have shown to improve model accuracy (Saint-Fleur et al., 2026), yet deep learning architectures require caution when ingesting these observations for two key reasons. First, because discharge is highly autocorrelated, models can easily become over-dependent on the streamflow signal and less sensitive to meteorological forcings. This dependency can degrade long-term forecasting performance and introduce a severe operational failure risk if gauges go offline. Second, some assimilation strategies are computationally expensive to train and execute (Nearing et al., 2022), presenting a bottleneck for hourly-resolution models where training costs are already high. To address these limitations, we propose the multi-frequency masked-forecasting LSTM (MF^{2}LSTM), which repurposes the masked-mean embedding (MME) strategy (Gauch et al., 2025), originally designed for handling missing data, to assimilate real-time discharge. This approach improves predictive performance when real-time discharge observations are available, while retaining the flexibility to operate seamlessly without them. This capability not only makes the forecasting pipeline robust to telemetry failures, delayed observations, and temporary gauge outages, but also enables seamless deployment across heterogeneous monitoring networks, where real-time discharge information may be available for only a subset of catchments or forecasting periods.

We evaluated our model's performance against the Large Area Runoff Simulation (LARSIM) model (Bremicker, 2000), a well-established process-based hydrological model currently utilized by flood-forecasting agencies across Germany, France, Austria, Luxembourg, and Switzerland. To rigorously test both frameworks within a quasi-operational environment, we forced them using weather forecasts from ICON-D2, a state-of-the-art meteorological model operated by the German Weather Service (DWD) and used in operational practice by the federal German flood forecasting agencies. This benchmarking strategy allows us to directly evaluate our deep learning pipeline against an actively deployed, expertly calibrated operational forecasting system under realistic forecasting conditions, providing a clear assessment of whether deep learning methods can compete with expertly tuned, established operational models.

The remainder of this paper is structured as follows. Section 2 details the model architectures and experimental setup. Section 3 presents the results of the experiments. Section 4 summarizes our key findings and conclusions, and section 5 presents an outlook for future lines of research.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

2 Data and methods

2.1 Data

The experiments were conducted across 138 river gauges located in the state of Baden-Württemberg (BaWü) in southwestern Germany (see Fig. 1). The associated catchment areas range from 31 km^2 to 3253 km^2, with a median area of 165 km^2. This gauge ensemble encompasses a subset of the CAMELS-DE-1h dataset (Dolich et al., 2026), restricted to basins where pre-computed LARSIM benchmarks were accessible and additional manual data quality checks were passed.

The model comparison was conducted for the period spanning from January 2021 to December 2024. The temporal extent of this evaluation period was strictly defined by the availability of the ICON-D2 numerical weather forecast archive. Further details regarding specific data configurations and experimental setups are provided in Section 2.3.2 for LARSIM and Section 2.4.2 for the LSTM.

Overall, the data configurations used in this study are intended to (1) allow direct model comparison and (2) provide conditions as similar as possible to the operational pipeline currently implemented in the flood forecasting center of Baden-Württemberg (Hochwasservorhersagezentrale BaWü, HVZ-BaWü), giving a clear assessment of how ML models operate under quasi-operational conditions.

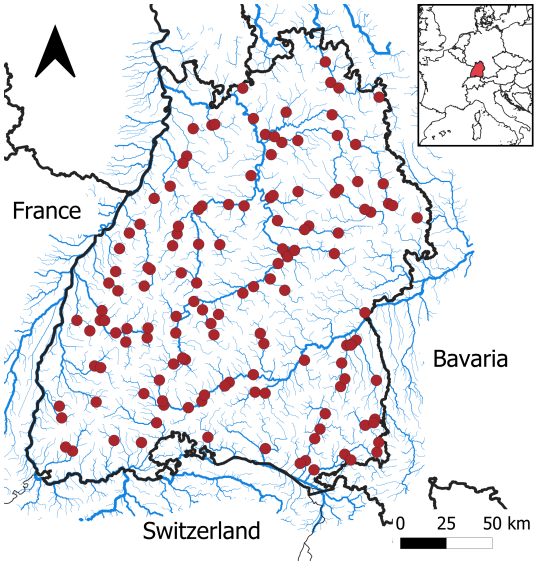

Figure 1. Location of 138 gauges used in the study. Gauges are located in the state of Baden-Württemberg in southwestern Germany.

2.2 Metrics

According to the World Meteorological Organization (WMO), forecast skill is determined by evaluating the accuracy of a forecast against a selected reference or baseline model (World Meteorological Organization, 2025). When using the mean

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

squared error (MSE) as the accuracy measure, the general skill score (SS), called mean squared error skill score (MSESS), can be defined as:

$$\text{MSESS} = 1 - \frac{\text{MSE}_{\text{fcst}}}{\text{MSE}_{\text{ref}}}. \tag{1}$$

where fcst and ref are the subject and reference forecasts, respectively. As detailed in appendix A, during hyperparameter optimization and model-robustness tests carried out for our MF^{2}LSTM architecture, we evaluated two scenarios: with and without data assimilation. To evaluate the standalone rainfall-runoff configurations, operating without data assimilation, we use the Nash–Sutcliffe Efficiency (NSE). The NSE represents a specific case of Eq. 1 where the reference model is defined as the mean of the observed discharge ($\overline{y_{\text{obs}}}$), as expressed in Eq. 2:

$$\text{NSE} = 1 - \frac{\sum_t (y_{\text{obs},t} - y_{\text{sim},t})^2}{\sum_t (y_{\text{obs},t} - \overline{y_{\text{obs}}})^2}. \tag{2}$$

Although NSE is a standard baseline in hydrology, it is less effective for evaluating models that use data assimilation. Because the baseline reference is the historical mean, a data-assimilating model can achieve a high score simply because the streamflow inputs place the simulation in the correct flow magnitude (e.g., maintaining high discharge during a flood event). Consequently, standard NSE values can fail to reveal underlying forecasting errors.

To provide a more rigorous assessment for configurations using data assimilation, we additionally implement the Persistence Nash–Sutcliffe Efficiency (PNSE). In this formulation, the reference model is the last available discharge observation recorded before the forecast issue time (y_{pers}), as shown in Eq. 3. The PNSE provides a much stricter benchmark, particularly across short lead times where persistence is a strong predictor. Formally, we define the PSNE as:

$$\text{PNSE} = 1 - \frac{\sum_t (y_{\text{obs},t} - y_{\text{sim},t})^2}{\sum_t (y_{\text{obs},t} - y_{\text{pers}})^2}. \tag{3}$$

We did not use PNSE for comparing models without data assimilation during the hyperparameter optimisation phase (appendix A) because, for hourly forecasts, the y_{pers} is such a strong predictor for the first 2 to 3 hours that minor timing or magnitude offsets, expected in models without assimilation, were penalized so heavily that they skewed the average metrics and made model comparisons unstable.

2.3 LARSIM model

2.3.1 Model description

LARSIM (Bremicker, 2000; Haag and Luce, 2008; Haag et al., 2022) is a process-based distributed hydrological model used for operational flood forecasting across Germany, France, Austria, Luxembourg, and Switzerland. The model accounts for multiple processes including: interception, snow accumulation, compaction, and melt, evapotranspiration, and soil water storage, with subsequent runoff generation for direct runoff, interflow, and groundwater discharge. Furthermore, it simulates catchment runoff concentration at the sub-catchment scale, incorporating channel flood routing, riverine retention, lake dynamics, and regulated reservoir releases (Bremicker, 2000). Besides the core rainfall-runoff component, LARSIM includes an automated

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

data-assimilation module. To optimize the model output alignment with the observed discharge signal, the assimilation module applies a signal shift to correct the forecast starting point, which progressively transitions to the initial trajectory. Moreover, it implements corrections to the forcing or model states, either by modifying the precipitation signal or by a direct correction of LARSIM-internal linear reservoir storage levels (Ho et al., 2025). Details of the data assimilation subroutines can be found in Luce et al. (2006). Further information about the LARSIM model can be found on its website (https://larsim.info/).

2.3.2 Experimental setup

The LARSIM simulations were carried out using the same model version currently operational at HVZ-BaWü. It was calibrated using historical records spanning from November 1997 to June 2021, with the period from 2010 onward receiving greater weight due to higher data availability and quality (Moretti et al., 2022).

Within the study area (see Fig. 1), the model simulates approximately 45 000 grid cells at a $1 \times 1\,\mathrm{km}^2$ spatial resolution. These cells are further discretized into more than 700 000 hydrological response units (HRUs). Each HRU is individually parameterized using high-resolution datasets for land cover, topography, and soil water capacity, enabling LARSIM to compute distinct water balance components for each individual land use and soil compartment.

Please note that the calibration of the LARSIM model was not part of this study. Instead, we evaluated the pre-calibrated operational model using 2021-2024 ICON-D2 historical weather forecasts to establish the baseline benchmark against which the LSTM is compared. As detailed in Section 2.4.2, we aligned (as far as possible) the training periods and training data of the LSTM, to allow a fair model comparison.

It should be noted that there is a six-month overlap between the end of the LARSIM calibration period (June 2021) and the beginning of the evaluation period of this study (January 2021). While this overlap grants LARSIM a small advantage over the LSTM, we retained these six months in the evaluation dataset to maximize the utilization of the available four-year ICON-D2 archive.

2.4 MF^{2}LSTM architecture

2.4.1 Model description

The MF^{2}LSTM architecture is presented in Fig. 2. The first component of our framework integrates the multi-frequency LSTM (MF-LSTM) proposed by Acuña Espinoza et al. (2025). This architecture allows us to simulate hourly discharge without the computational burden of processing the entire historical sequence at an hourly resolution. From a hindcast sequence length of approximately one year (8736 hours $=$ 364 days), we aggregate the first 360 days (8640 hours) into 24-hour blocks and concatenate the remaining 96 hours at their original hourly resolution. This step reduces the hindcast sequence from 8736 to $360 + 96 = 456$ timesteps, nearly a factor of 20, gaining computational efficiency while maintaining model performance. This preservation of performance stems from the fact that the impact of the input's temporal distribution diminishes the further back in time we look, making hourly resolution unnecessary for the distant past. We then apply the sequential-forecast framework introduced by Cohen et al. (2026), where a single LSTM cell is unrolled across the forecast horizon (t+1, …, t+h) in a

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

sequence-to-sequence mode. Combined, the MF-LSTM and sequential-forecast setup serve as a computationally efficient base model for generating hourly forecasts.

To further improve accuracy, we integrate data assimilation. Because the discharge at time t provides a strong predictive signal for subsequent timesteps ($t+1, t+2, \ldots$), incorporating real-time observations when available is highly beneficial. To handle assimilation, we repurpose the MME strategy to separate the meteorological variables and the discharge records into independent groups. Specifically, we use distinct embedding networks to map (1) the meteorological inputs ($\mathbf{x}_t$, comprising precipitation, temperature, radiation, and relative humidity) and (2) lagged discharges (Q_{t-1}, Q_{t-2}, Q_{t-3}) into a shared latent space. We chose one, two, and three-hour discharge lags based on initial experiments, as this window not only informs the model of the immediate prior streamflow volume but also captures the trajectory of the hydrograph (e.g., a rising or falling tendency). Once both input groups are mapped to the same hidden dimension, we apply an element-wise masked-mean operation (nan_mean) to filter out missing data prior to the LSTM cell. This strategy allows the pipeline to ingest discharge information when available, while enabling the model to remain fully operational when it is absent. To teach the model how to operate with and without the discharge signal, we used probabilistic masking during training, based on step-level (p_{step}) and sequence-level (p_{seq}) NaN-infusion (Gauch et al., 2025). The details of our probabilistic masking strategy, the effect of different masking probabilities and their role in ensuring model robustness in ungauged scenarios are presented in appendix A. As illustrated in Fig. 2, unique embedding networks are utilized not only for each variable group, but also across different temporal resolutions and operational phases (daily hindcast, hourly hindcast, and hourly forecast). This concept of frequency-specific embeddings follows the logic introduced by Acuña Espinoza et al. (2025), which we further extend here to handle multiple variable groups for data assimilation.

The final stage of the architecture is the autoregressive loop deployed during the forecast period. By definition, observed discharge is unavailable during the forecast window because it is the target variable under simulation. However, given the lagged configuration of our discharge inputs, we can feed the model's own previously generated discharge back into the next timestep. We acknowledge that the predicted discharge is a direct function of the hidden states, which are already propagated forward to the next LSTM step, meaning this information is being passed twice. Nevertheless, explicitly including this autoregressive loop improved our model's performance. We attribute this improvement to the loop's ability to facilitate a smooth sequential transition, preventing abrupt systematic changes when moving from the hindcast phase (which ingests real-time observations) into the forecast phase. However, manually unrolling this autoregressive loop during training introduces a substantial computational bottleneck. To restore training efficiency without introducing exposure bias during inference, we utilize a teacher-forcing (Williams and Zipser, 1989) scheme paired with a customized sigmoid-scaled noise injection function. The details about this formulation and the computational speedup metrics are expanded in appendix B.

2.4.2 Experimental setup

Similarly to LARSIM, we utilized information provided by the HVZ-BaWü for model training. However, as described in Section 2.3.2, Moretti et al. (2022) identified deficiencies in data availability and quality prior to 2010, and therefore decided to give greater weight to the post-2010 period during training.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

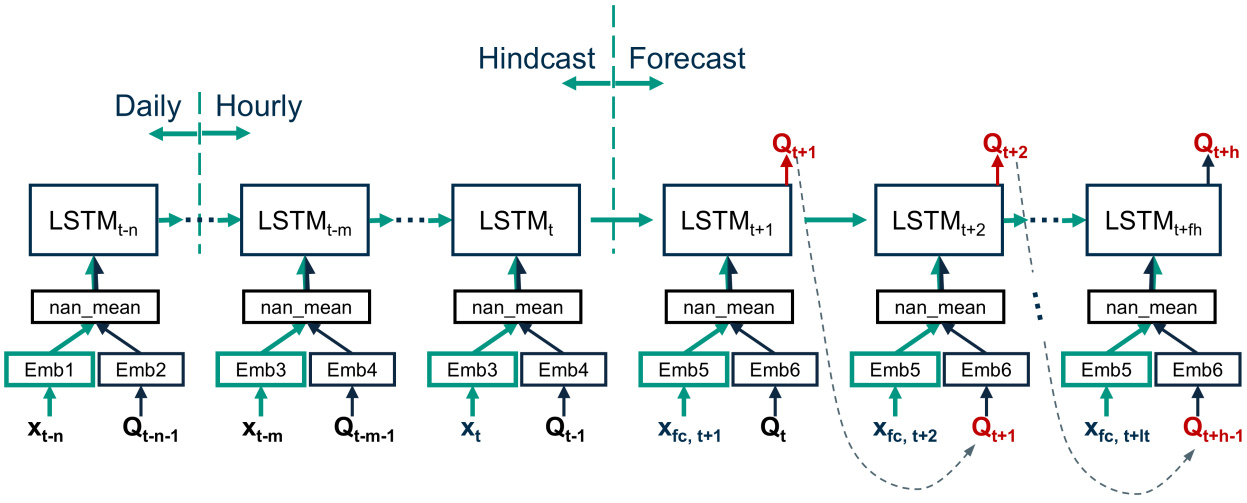

Figure 2. MF^{2}LSTM scheme. The hindcast sequence is processed at dual resolutions: daily for the initial period ($t-n$ to $t-m-1$) and hourly for the subsequent one ($t-m$ to t). At each timestep, the model ingests two distinct groups of variables: meteorological inputs (x) and lagged discharges (Q). To accommodate varying temporal frequencies, independent embedding networks map these groups into a shared latent space. A NaN-ignoring mean operation (nan_mean) is then applied to aggregate the embeddings and filter out missing data prior to LSTM cell processing. During the forecast period ($t+1$ to $t+h$), forecast-specific embeddings are utilized. Here, the model is unrolled across the lead-time horizon in an autoregressive mode, where the predicted discharge output from the previous timestep is fed forward as the input for the current step. Note that frequency flags, forecast counters, and static inputs are omitted for visual clarity.

Conversely, we followed a two-step approach. For the period spanning November 2009 to December 2024, we utilized the same forcing data as LARSIM, with the main difference that we trained the MF^{2}LSTM in a lumped setup, using catchment-average characteristics instead of distributed meteorological fields. However, for the earlier period we took advantage of the CAMELS-DE-1h dataset and extended our timeseries from January 2001 to November 2009. The resulting dataset was divided into two configurations for model development and evaluation:

- Hyperparameter optimization: Models were trained on data from January 2001 to December 2015, and evaluated on a validation period from January 2016 to December 2020.
- Final evaluation: Once the optimal architecture was selected, the models were retrained on the combined period from January 2001 to December 2020. The final, independent testing period spanned January 2021 to December 2024.

3 Results and discussion

Prior to the final model evaluation, we conducted a hyperparameter optimization for the MF^{2}LSTM to determine the optimal configuration for the data-assimilating component of the architecture, focusing specifically on the probabilistic masking rates

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

(p_{step} and p_{seq}). The analysis of these ablation experiments is detailed in Appendix A. Based on the balance between maximizing predictive accuracy during active data assimilation and maintaining a robust fallback performance when the streamflow signal is absent, a configuration of $p_{\text{seq}} = p_{\text{step}} = 0.05$ was selected for the final architecture.

With the model parameters fixed, we retrained the MF^{2}LSTM on the period January 2001 to December 2020. The framework was then evaluated using the deterministic run of the ICON-D2 numerical weather predictions spanning from January 2021 to December 2024. Although the ICON-D2 products provide a 48-hour forecast horizon, the model comparison is restricted to a 21-hour lead time, as this was the maximum horizon available for the LARSIM benchmarks. Additional independent tests confirmed that our pipeline remains stable when extended to the full 48-hour horizon; however, these results are omitted from the manuscript due to the lack of a corresponding operational baseline.

Figure 3 shows the evolution of NSE and PNSE with lead time across the 138 basins in the testing subset. For the NSE case, both models exhibit decreasing performance as lead time increases. This decay is expected due to two main factors: (1) discharge assimilation plays a more prominent and helpful role during the initial timesteps, and (2) the predictive skill of the numerical weather forecast degrades with lead time, directly translating into reduced streamflow accuracy. Nevertheless, the MF^{2}LSTM model consistently outperforms LARSIM, reporting higher average median NSE across all lead times (0.89 vs. 0.85) and a narrower spread, with an average interquartile range (IQR) of 0.12 compared to 0.17 for LARSIM. This not only indicates that the deep learning pipeline produces more accurate results on average, but also suggests that regional training, combined with the structural flexibility to bypass rigid mass-conservation constraints, allows it to perform better in catchments where the process-based model struggles. Notably, the MF^{2}LSTM's median NSE at lead time 21 (0.79) matches LARSIM's performance at lead time 15 (0.79), indicating that the deep learning pipeline extends the forecasting window by up to 6 hours while maintaining an equivalent predictive accuracy.

The evolution of the PNSE metric reflects two opposing effects as lead time increases. On the one hand, we have the same effect as for NSE where the forecast quality degrades due to the diminishing influence of data assimilation, plus the decay of meteorological forecast accuracy with forecast lead time. On the other hand, the baseline reference (denominator in Eq. 3) is fixed to the last available discharge observation (i.e. lead time zero). Consequently, as the forecast horizon extends, this persistence baseline becomes a progressively weaker predictor. Despite these compounding factors, the MF^{2}LSTM model delivers higher average median scores (0.53 vs. 0.40) and narrower IQR (0.14 vs 0.41) than LARSIM. As expected, PNSE values are generally lower than NSE values because evaluating performance against recent persistence (Eq. 3) establishes a much stricter benchmark than comparing against the overall mean (Eq. 2).

Finally, to evaluate performance under high-flow conditions, we recomputed the metrics exclusively for streamflow events exceeding the 95th percentile, calculated specifically for each gauge (Fig. 4). Under these conditions, MF^{2}LSTM outperforms LARSIM across both metrics, achieving higher average median values and narrower IQRs for NSE (averages: 0.64 vs. 0.57; IQR: 0.31 vs. 0.42) and PNSE (averages: 0.55 vs. 0.48; IQR: 0.16 vs. 0.37).

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

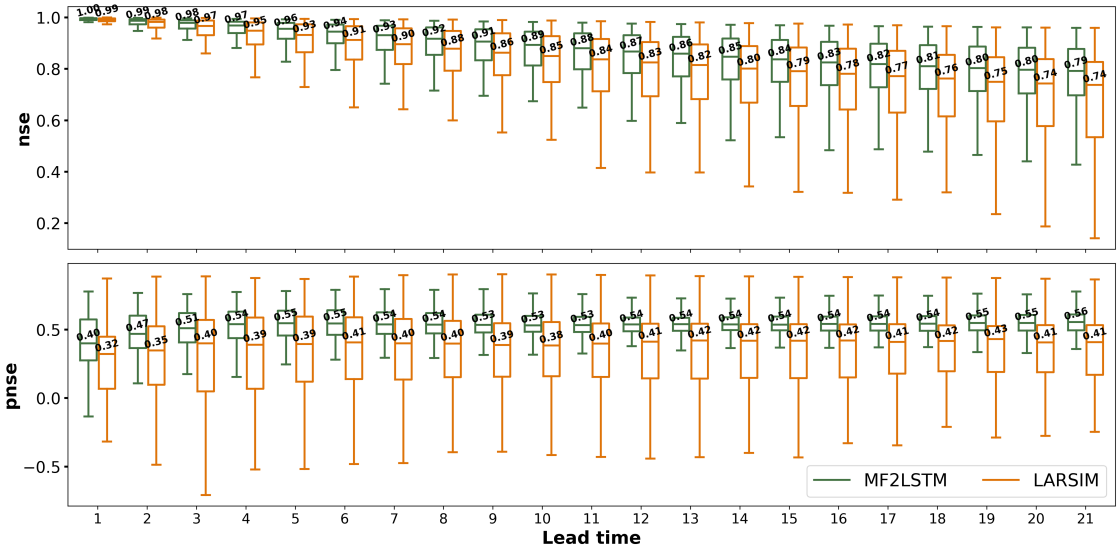

Figure 3. Model performance comparison between the MF^{2}LSTM model and LARSIM across the 138 basins in the testing subset. The upper and lower panels display the evolution of the NSE and PNSE metrics, respectively, as a function of forecast lead time. The annotations indicate the median metric for each lead time. For both metrics, a value of 1.0 indicates a perfect match between simulated and observed discharge.

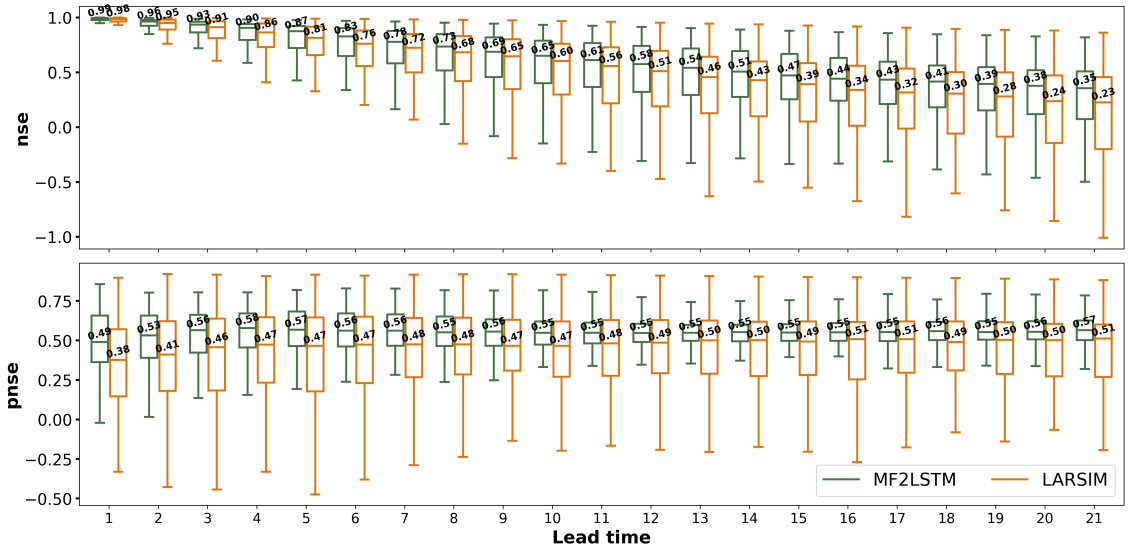

Figure 4. Model performance comparison between the MF^{2}LSTM model and LARSIM across the 138 basins, evaluated exclusively for streamflow events exceeding the 95th percentile (of each basin).

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

4 Summary and conclusions

In this study we introduced MF^{2}LSTM, a deep learning framework suitable for operational hourly flood forecasting that combines multi-frequency input processing with a flexible discharge data assimilation strategy. We then evaluated our framework across 138 catchments in southwestern Germany against LARSIM, the currently operational process-based model used in the region, both forced by deterministic ICON-D2 weather forecast data. This comparison allowed us to establish a rigorous benchmark for deep learning based flood forecasting under quasi-operational conditions.

Combining the MF-LSTM architecture (Acuña Espinoza et al., 2025) with the sequential-forecast setup (Cohen et al., 2026) allowed us to generate hourly forecasts in a computationally efficient and consistent way. Additionally, by extending the masked-mean embedding strategy (Gauch et al., 2025) to multiple frequencies, repurposing it for data assimilation, and applying sequence-level ablation during training, we produced a model that can leverage data assimilation when available, while keeping the ability to operate without it, making it a particularly robust framework for operational implementation.

As shown in Sect. 3, the proposed model consistently outperformed the LARSIM benchmark, yielding higher median performance values (NSE and PNSE) and narrower error distributions across all analyzed lead times. Moreover, the MF^{2}LSTM model at a 21-hour lead time achieves equivalent predictive accuracy (NSE $= 0.79$) to LARSIM at a 15-hour lead time, effectively expanding the forecast window by up to 6 hours. When evaluated under high-flow scenarios, the conclusions were consistent.

Overall, this work demonstrates that the structural simplicity of a standard single-cell LSTM architecture can deliver high predictive accuracy and strong operational resilience. By exceeding the capabilities of a highly tuned, expertly calibrated, process-based operational benchmark, the proposed pipeline highlights the potential of deep learning methods for regional, real-time flood-forecasting deployment in hourly resolution.

5 Outlook

The experiments presented here used a framework in which the models were trained using observed meteorological records as pseudo-forecasts, transitioning to explicit numerical weather predictions from ICON-D2 only during the independent testing phase. This strategy introduces a potential limitation, as raw numerical forecast data can exhibit a different distribution than direct observations (Konold et al., 2025), and several studies have shown that numerical weather archives can be successfully incorporated during training (Nearing et al., 2024; Cohen et al., 2026; Taccari et al., 2026). Nevertheless, our decision to exclude forecast products from the training phase was deliberate and responds to a pragmatic approach to operational implementation. The current operational pipeline deployed by DWD involves a multi-model transition to optimize forecasting accuracy across different horizons. For the first 1.5 to 2 hours of lead time, a radar-based nowcasting strategy projects current precipitation fields using wind patterns. Then the pipeline transitions to the high-resolution, computationally intensive ICON-RUC (Rapid Update Cycle) model up to a 12-hour lead time. Beyond this point, it switches to ICON-D2 up to 48 hours, and then to ICON-EU if the full 5-day window is required.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

An advantage of our deep learning framework is that unique embedding networks can, in theory, be assigned to each segment of this composite forecast sequence, training the model to adapt to shifting data distributions. However, a major practical bottleneck is that these embeddings need to be trained with historical forecast records. Producing these records, especially when reforecasting is necessary, is not always feasible due to the immense computational and storage costs. Furthermore, subsequent updates to the weather forecast systems might alter the data distribution, making retraining necessary. Thus, while exposing networks to multi-model forecast products during training remains an optimal machine learning practice (Nearing et al., 2024; Cohen et al., 2026; Taccari et al., 2026) that we intend to explore, our approach highlights the operational feasibility of training without relying on heavy reforecast infrastructure.

A second direction for future research is accounting for predictive uncertainty in operational settings. To evaluate this, we explored integrating a mixture density network (MDN) (Klotz et al., 2022) head with our MF^{2}LSTM approach, enabling the model to generate probabilistic forecasts from deterministic inputs. We got promising initial results (we provide the code for the MF^{2}LSTM + MDN variant in the supplementary material). Nevertheless, fully capturing forecast uncertainty in practice also requires propagating meteorological variability forward. Given that operational products like ICON-D2 provide ensemble predictions, determining how to best incorporate these ensembles, whether through feeding the model ensemble statistics during training or using the ensemble members strictly during inference, remains an active research line.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

Appendix A: Effect of NaN-masking probabilities

As detailed in Sect. 2.4.1, the masked-mean embedding strategy allows the model, in principle, to operate both with and without real-time discharge data. However, this dual behavior must be explicitly learned during the training phase. Following the probabilistic masking framework introduced by Gauch et al. (2025), a portion of the training data is artificially masked with NaN values. This strategy relies on two independent probabilities: p_{step}, the probability of masking a specific timestep, and p_{seq}, the probability of masking an entire group of variables across the complete sequence for a given batch element. In our setup, p_{step} indicates the probability of masking individual discharge observations (simulating random gauge dropouts), whereas p_{seq} defines the probability of forcing the model to process the entire sequence (both hindcast and forecast phases) without any streamflow signal. This sequence-level masking is responsible for training the model to function effectively in ungauged scenarios where real-time discharge observations are unavailable. The total probability that a given step is masked is detailed in Eq. A1, where there is a p_{seq} probability that the step is masked because the whole sequence was masked, plus a $(1 - p_{\text{seq}}) \cdot p_{\text{step}}$ probability that the remaining unmasked sequences are affected at that specific step:

$$P(\text{mask}) = p_{\text{seq}} + (1 - p_{\text{seq}}) \cdot p_{\text{step}}. \tag{A1}$$

To analyze the impact of discharge masking on model performance, we treated these probabilities as hyperparameters and conducted a grid search. Using the optimization split described in Sect. 2.4.2, the models were trained from January 2001 to December 2015 and validated from January 2016 to December 2020. In line with standard deep learning practices for hydrological modeling, we trained an ensemble of three random seeds for each of the six combinations presented in Table A1, and used the median ensemble signal as the final simulated discharge. Limiting the grid search to six configurations and three seeds was a necessary constraint due to the high computational cost of training hourly-resolution models. Even with these restrictions, the hyperparameter tuning required over 5 days of continuous compute time on an NVIDIA H100 GPU.

Table A1 summarizes the performance metrics across the validation period for different p_{step} and p_{seq} combinations under two distinct evaluation scenarios: with and without data assimilation. Specifically, the metrics in the third column represent the scenario where the model received the full discharge signal during evaluation, whereas the fourth column reports metrics derived from evaluating the model with the discharge signal entirely masked out. The first scenario simulates normal operational circumstances, as real-time discharge data is consistently available within the HVZ-BaWü operational pipelines. Conversely, the second scenario illustrates how the masking probabilities affect model resilience if real-time gauge information fails to arrive or if the framework is deployed in ungauged regions. The hyperparameter tuning process used a validation period forced with pseudo-forecast information (i.e., injecting observed meteorological records during the forecast period rather than ICON-D2 products). This choice allowed us to (1) isolate the specific effects of masking probabilities from the compounding errors associated with numerical weather forecast lead times, and (2) preserve the relatively short ICON-D2 record (4 years) to maximize the robustness of our final testing phase.

The summary metrics presented in Table A1 were calculated by taking the median metric value across the 138 basins and then averaging those medians over all forecast lead times. The objective of this aggregated metric is to provide an overview of model performance under varying operational conditions. Section 2.2 explains the evaluation metrics used in each case.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

Table A1. Summary of validation metrics across different missing data probabilities.

p_{step}	p_{seq}	**PNSE (with Q)**	**NSE (without Q)**
0.05	0.05	0.78	0.79
0.10	0.10	0.78	0.78
0.20	0.20	0.76	0.80
0.40	0.40	0.68	0.78
0.60	0.60	0.61	0.79
0.05	0.00	0.78	-0.05

An analysis of the first five rows of Table A1 yields insights into the effect of the masking hyperparameter. Looking at the third column, model performance steadily decreases as the training masking probabilities increase. Lower masking probabilities ensure that the discharge signal is more reliably present during training, allowing the model to learn how to assimilate the streamflow data more effectively. Furthermore, minimizing the masking probability creates a closer alignment between training and evaluation data distributions, keeping the model within a data regime it is highly optimized to handle. Conversely, the fourth column of the first five rows indicates that when evaluated entirely without the discharge signal, all models exhibit similar performance, irrespective of their training masking probabilities. While initially unexpected, subsequent analysis supported our running hypothesis. Due to the inclusion of the sequence-level masking probability (p_{seq}), all five configurations are forced to contend with total discharge ablation during training. Consequently, even a small probability like $p_{\text{step}} = 0.05$ is sufficient to teach the model how to adapt. This aligns with findings by Heudorfer and Loritz (2026), who demonstrated that LSTMs can learn consistent behaviors even with a random data sampling rate as low as 5%. To further test this hypothesis, we conducted two additional baseline experiments.

First, we evaluated a standalone MF-LSTM with sequential forecasting to establish a benchmark for a pure rainfall-runoff model operating without any data assimilation routines. This baseline configuration yielded a summary NSE metric of 0.81, quite similar to the values reported in the final column of Table A1. This confirms that our data-assimilating models fall back to standard rainfall-runoff baselines when the streamflow signal is lost. Figure A1 illustrates this behavior by comparing a specific hydrograph from the MF^{2}LSTM model, with and without data assimilation, against the standalone rainfall-runoff baseline. As expected, integrating data assimilation yields the highest predictive accuracy. However, when the discharge signal is completely withheld, the MF^{2}LSTM model remains robust, matching the stable baseline behavior of the standalone LSTM.

Second, we executed an experiment (see final row in Table A1) where we maintained $p_{\text{step}} = 0.05$ but set $p_{\text{seq}} = 0.00$, effectively depriving the model from experiencing purely rainfall-runoff scenarios during training. Comparing the first and last rows of Table A1 reveals that while both models achieve the same performance when discharge is available, the configuration trained without sequence-level masking ($p_{\text{seq}} = 0.00$) completely fails (NSE $= -0.05$) when discharge is removed. This finding confirms that sequence-level ablation during training is required to enable ungauged operational flexibility.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

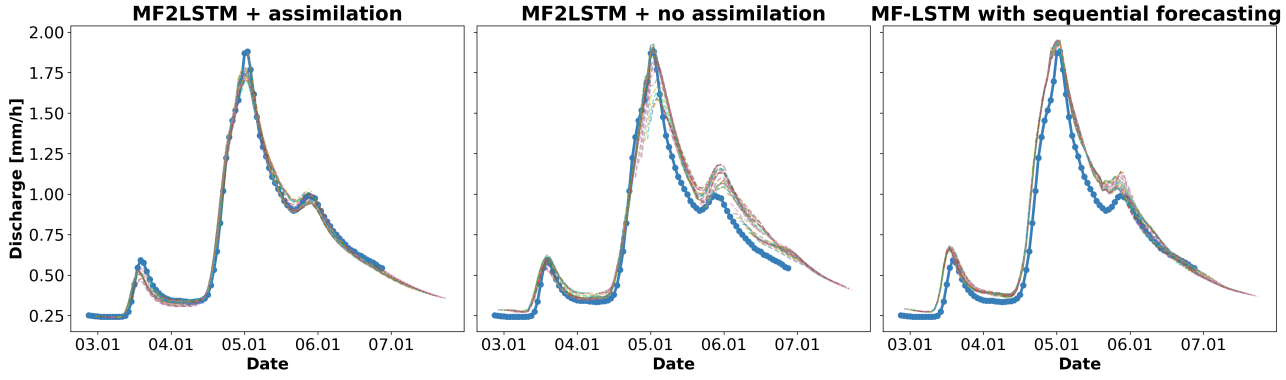

Figure A1. Model behavior across different operational modes. The first two hydrographs display the performance of the MF^{2}LSTM model evaluated with and without data assimilation, respectively, while the third shows the MF-LSTM + sequential forecast benchmark. The MF^{2}LSTM configuration shown here was trained with $p_{\text{seq}} = p_{\text{step}} = 0.05$. The event is selected from the validation period of the first experimental configuration outlined in Sect. 2.1.

315 To complement the summary statistics from Table A1, Fig. A2 illustrates a detailed comparison of the full performance distributions across the 138 basins. This comparison shows the metrics at each forecast lead time for both configurations (with and without data assimilation), specifically comparing the $p_{\text{NaN}} = 0.05$ and $p_{\text{NaN}} = 0.20$ masking probabilities.

These ablation experiments presented in this section clarify the impact of specific mask probabilities during model training, and show that probabilistic masking must be applied if one is interested in the model learning how to handle missing data.
320 Based on these results, a configuration of $p_{\text{seq}} = p_{\text{step}} = 0.05$ is selected as the final model configuration.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

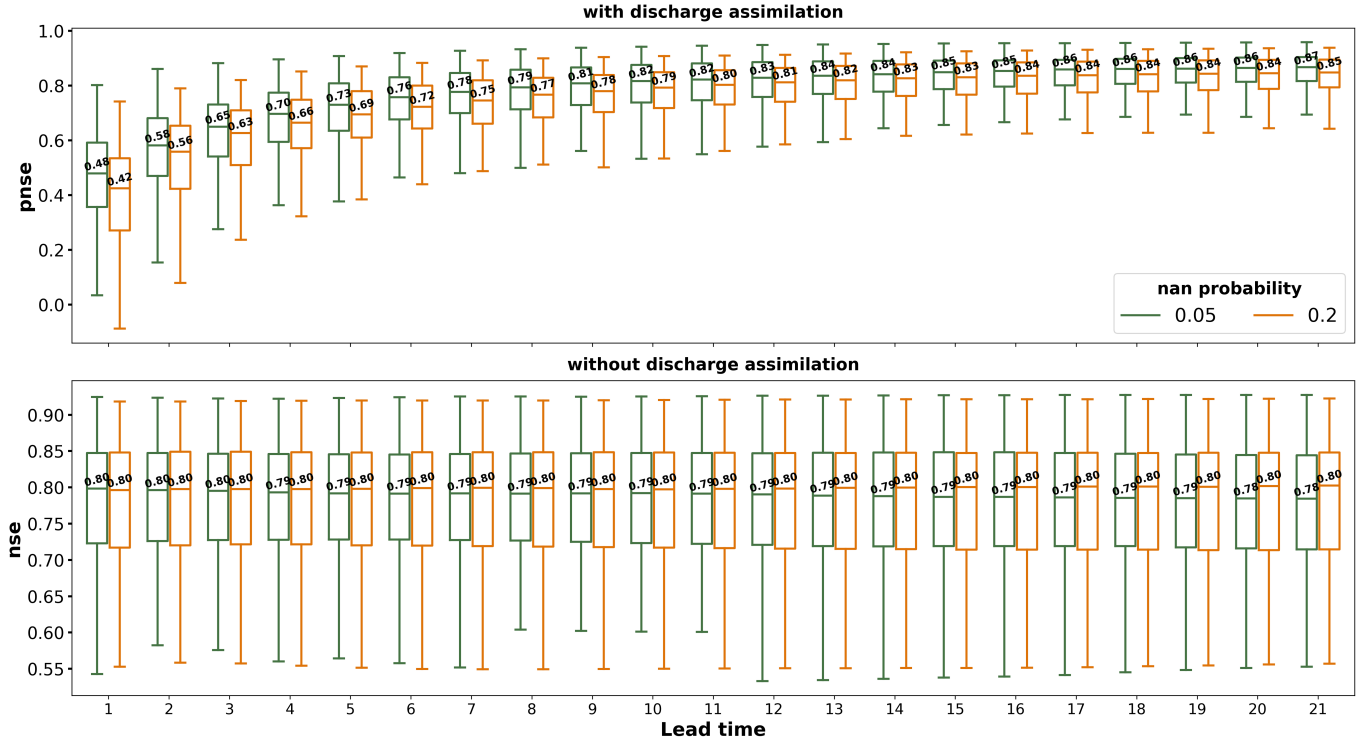

Figure A2. Comparison of the full performance distributions across the 138 basins as a function of forecast lead time. The upper panel displays the case where data assimilation is active, while the lower panel presents the baseline performance without assimilation. Results compare $p_{\mathrm{NaN}} = 0.05$ and $p_{\mathrm{NaN}} = 0.20$

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

Appendix B: Teacher-forcing and noise injection

The architectural framework presented in Fig. 2 is fully capable of generating hourly forecasts while assimilating streamflow observations. However, a practical implementation challenge emerged during training. The standard PyTorch LSTM implementation (torch.nn.LSTM) expects an input tensor of shape (batch_size, sequence_length, input_size), which later loops through the sequence dimension internally via highly optimized CUDA kernels. In contrast, due to the autoregressive nature of our forecast loop, the prediction from the previous timestep must serve as the input for the next. Consequently, the complete input tensor cannot be assembled prior to the forward pass, forcing the forecast sequence to be unrolled manually in Python. A similar performance bottleneck was documented by Nearing et al. (2022), who reported a $6\times$ computational slowdown when transitioning from a standard native LSTM to an autoregressive variation. Given that hourly hydrological models are already computationally expensive to train, this overhead could mean the difference between a training runtime of a few days versus a few weeks.

To circumvent this limitation and restore computational efficiency, we employ a teacher-forcing scheme (Williams and Zipser, 1989). Under this setup, the actual observed discharge is fed into the forecast loop, instead of the autoregressive model-simulated values. Because the historical discharge observations are known at the start of training, the entire input tensor can be assembled beforehand. This strategy allows the model to use the CUDA-optimized pipeline, maintaining a reasonable computational cost. Depending on the forecast sequence length, the computational gain of running teacher-forcing during training, in contrast to unrolling the complete autoregressive loop, yields a performance speedup between $3\times$ and $5\times$.

However, a side effect of teacher-forcing must be accounted for: hiding the model from its own forecasting errors during training prevents it from adapting to error accumulation during operational inference. In a true autoregressive deployment, error propagation is inevitable. To produce reliable forecasts, the model must learn to rely on the real-time discharge signal during the initial lead times, but gradually shift its dependence toward meteorological inputs as the forecast horizon extends. While our original architecture includes a forecast counter to help the LSTM gates regulate this trade-off over time, teacher-forcing fundamentally alters what the model is forced to learn. Without the autoregressive loop during training, the model only needs to map a marginal distribution over multiple timesteps, i.e., $p(y_t|\mathbf{X}_{t-1})$. In contrast, operational inference requires the model to track an evolving conditional distribution, i.e., $p(y_t|\mathbf{X}_{t-1}, \hat{y}_{t-1}, \hat{y}_{t-2}, \dots)$, which behaves much closer to a joint distribution. Teacher-forcing disrupts this learning process because the quality of the injected streamflow signal remains constant across all timesteps, preventing the model from adapting to this dynamic conditional space.

To resolve this issue, we inject synthetic noise into the training discharge signal during the forecast period. We initially evaluated constant and linear noise injection strategies. Although both methods yielded reasonable results, constant noise artificially degraded the signal quality during the early timesteps, causing the model to reject short-term assimilation. In contrast, a linear noise profile, which scales indefinitely with lead time, introduced excessive variance at longer horizons, compromising the model's stability. Therefore, we designed a noise function that remains low at the start, grows over time, and eventually plateaus. Under this logic, we implement the sigmoid-scaled noise function as described in Eq. B1, where $\text{std}(0, 1)$ represents a sample from a standard normal distribution and t is the current step in the forecast sequence (1, 2, …, h). This

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

formulation allows us to have near-zero noise during the initial timesteps, boosting model confidence in data assimilation at the beginning of the forecast, before smoothly scaling up and stabilizing at an upper bound of $0.2 \cdot \text{std}(0,1)$. The 0.2 scaling limit is derived from the hyperparameter optimization benchmarks established by Klotz et al. (2022) for noise-injection strategies.

$$noise = std(0,1) \cdot \frac{0.2}{1 + e^{6-t}}. \tag{B1}$$

Appendix C: Model hyperparameters

For the model architecture, we used a standard single-cell LSTM with 128 hidden states. Training was conducted over 12 epochs with a batch size of 256 and a dropout rate of 0.4. We applied a piecewise constant learning rate schedule, starting at 5×10^{-4}, which dropped to 1×10^{-4} after 5 epochs and to 1×10^{-5} after 10 epochs. The input sequence length was set to 8736 hours. Within this window, the first 8640 hours were aggregated into 360 consecutive 24-hour blocks, while the remaining 96 hours were processed at their original hourly resolution. To assist the network, we included a binary flag to help the LSTM distinguish between input frequencies, alongside a step counter deployed during the forecast period. For the data assimilation component, the embedding dimensions of the masked-mean embedding architecture were mapped using a three-layer, fully connected, feed-forward neural network with a hidden layer structure of 10-10-6. These hyperparameters were not subject to structural optimization, instead, they were defined a priori based on an extensive dataset of previous modeling experiments. Further details regarding the experimental configurations can be found in the config.yml file within the Zenodo repository associated with this study (Acuna Espinoza, 2026). Details for the nan-masking probability hyperparameter optimization can be found in appendix A

Appendix D: Benchmarking model development

The importance of driving model improvement through community benchmarks has been discussed in (Donoho, 2017; Nearing et al., 2021; Klotz et al., 2022; Kratzert et al., 2024). This practice implies reproducing experiments from reference studies to confirm that the model implementation operates as expected. In our case, two components of the proposed model framework were benchmarked. First, the MF-LSTM (Acuña Espinoza et al., 2025) was evaluated against Gauch et al. (2021) to validate our multi-frequency approach. Second, we benchmarked our masked-mean embedding implementation against Gauch et al. (2025). The results of both benchmarks, along with all files required to reproduce these experiments, are available in the benchmark section of our Hy2DL library (https://github.com/eduardoAcunaEspinoza/Hy2DL).

Furthermore, to validate our metric subroutines, we compared our outputs against the metrics generated by the PROFOUND software, the computational program used by the HVZ-BaWü to evaluate their operational models.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

Code availability. The MF^{2}LSTM model architecture, configuration files to run the experiments, and code to generate all figures presented in the paper are available at https://doi.org/10.5281/zenodo.21907865 (Acuna Espinoza, 2026).

Data availability. The CAMELS-DE-1h dataset is available at https://doi.org/10.5880/fidgeo.2026.045 (Dolich et al., 2026). The observed meteorological data (2009-2024), ICON-D2 forecasts, LARSIM simulation results, and all the results generated for this publication are available at https://doi.org/10.5281/zenodo.21907865 (Acuna Espinoza, 2026).

Author contributions. The model architecture was developed by EAE, FK, MG, DK, RfLo and UE. The codes were written by EAE with support from MAC. Data processing was carried out by EAE, AD and DE. The LSTM simulations were conducted by EAE. The LARSIM simulations were conducted by RbLa. The results were discussed by all the authors. The draft of the manuscript was prepared by EAE. Reviewing and editing were provided by all the authors. Funding was acquired by UE. All the authors read and agreed to the current version of the paper.

Competing interests. At least one of the (co-)authors is a member of the editorial board of Hydrology and Earth System Sciences. The authors have no other competing interests to declare.

Acknowledgements. We would like to thank the Google Cloud Program (GCP) team for awarding us credits to support our research and run the models. The authors also acknowledge the support of the state of Baden-Württemberg through bwHPC. We would also like to thank the people of Baden-Württemberg, who, through their taxes, provided the basis for this research.

Financial support. This work was supported by the KIT Center MathSEE and the KIT Graduate School for Computational and Data Science under the EPO4Hydro Bridge PhD grant and by the EPO4Hydro project funded by LUBW-HVZ.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

References

Acuña Espinoza, E., Kratzert, F., Klotz, D., Gauch, M., Álvarez Chaves, M., Loritz, R., and Ehret, U.: Technical note: An approach for handling multiple temporal frequencies with different input dimensions using a single LSTM cell, Hydrology and Earth System Sciences, 29, 1749–1758, https://doi.org/10.5194/hess-29-1749-2025, 2025.

Acuna Espinoza, E.: Everything everywhere all at once: A single-cell LSTM network unifying multi-frequency, missing data, and discharge assimilation for robust operational flood forecasting, https://doi.org/10.5281/zenodo.21907865, 2026.

Bremicker, M.: Das Wasserhaushaltsmodell LARSIM - Modellgrundlagen und Anwendungsbeispiele, Tech. rep., Institut für Hydrologie, https://www.hydrology.uni-freiburg.de/publika/band11.html, 2000.

Bushra, S., Shakya, J., Cattoën, C., Fischer, S., and Pahlow, M.: CAMELS-NZ: hydrometeorological time series and landscape attributes for New Zealand, Earth System Science Data, 17, 5745–5760, https://doi.org/10.5194/essd-17-5745-2025, 2025.

Cohen, D., Amira, R., Aschner, R., Carny, Y., Feinstein, B., Fester, H., Fronman, S., Gauch, M., Gilon, O., Green, R., Hassidim, A., Klotz, D., Kratzert, F., Korenfeld, D., Loike, G., Markel, A., Matias, Y., Mayo, R., Metzger, A., Mosheyev, B., Niego, A., Rees, S., Reinstein, E., Sicherman, A., Shalev, G., Shefi, O., Shildan, Y., Zemach, I., Zlydenko, O., and Nearing, G.: Extending Medium-Range Global Flood Forecasts: The Google Global Flood Forecasting Model Version 2, EGUsphere, 2026, 1–31, https://doi.org/10.5194/egusphere-2026-2283, 2026.

Coxon, G., Zheng, Y., Barbedo, R., Cooper, H., Fileni, F., Fowler, H. J., Fry, M., Green, A., Gribbin, T., Harfoot, H., Lewis, E., Neto, G. G. R., Qiu, X., Salwey, S., and Wendt, D. E.: CAMELS-GB v2: hydrometeorological time series and landscape attributes for 671 catchments in Great Britain, Earth System Science Data, 18, 4345–4371, https://doi.org/10.5194/essd-18-4345-2026, 2026.

Dolich, A., Acuña Espinoza, E., Ehret, U., Kraft, M., Bondy, J., Meuer, J., and Loritz, R.: CAMELS-DE-1h: hourly hydro-meteorological time series, weather forecasts, and attributes for 1611 catchments in Germany, Earth System Science Data Discussions, 2026, 1–37, https://doi.org/10.5194/essd-2026-289, 2026.

Donoho, D.: 50 years of data science, Journal of Computational and Graphical Statistics, 26, 745–766, https://doi.org/10.1080/10618600.2017.1384734, 2017.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Lin, J., and Hochreiter, S.: Rainfall–runoff prediction at multiple timescales with a single Long Short-Term Memory network, Hydrology and Earth System Sciences, 25, 2045–2062, https://doi.org/10.5194/hess-25-2045-2021, 2021.

Gauch, M., Kratzert, F., Klotz, D., Nearing, G., Cohen, D., and Gilon, O.: How to deal w___ missing input data, Hydrology and Earth System Sciences, 29, 6221–6235, https://doi.org/10.5194/hess-29-6221-2025, 2025.

Haag, I. and Luce, A.: The integrated water balance and water temperature model LARSIM-WT, Hydrological Processes, 22, 1046–1056, https://doi.org/https://doi.org/10.1002/hyp.6983, 2008.

Haag, I., Krumm, J., Aigner, D., Steinbrich, A., and Weiler, M.: Simulation von Hochwasserereignissen in Folge lokaler Starkregen mit dem Wasserhaushaltsmodell LARSIM, Hydrologie & Wasserbewirtschaftung, 66, 6–27, https://doi.bafg.de/HyWa/2022/HyWa_2022.1_1.pdf, 2022.

Heudorfer, B. and Loritz, R.: Is smart sampling worth it? Impact of training data selection on the performance of LSTMs in streamflow prediction, Hydrology Research, 57, 501–517, https://doi.org/https://doi.org/10.2166/nh.2026.119, 2026.

Ho, S. Q.-G., Lang, R., and Ehret, U.: Forecast-based operation of re-purposed small reservoirs for floods, farms, and (low) flows, EGUsphere, 2025, 1–28, https://doi.org/10.5194/egusphere-2025-4739, 2025.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

Hochreiter, S. and Schmidhuber, J.: Long Short-Term Memory, Neural Computation, 9, 1735–1780, https://doi.org/10.1162/neco.1997.9.8.1735, 1997.

Klingler, C., Schulz, K., and Herrnegger, M.: LamaH-CE: LArge-SaMple DAta for Hydrology and Environmental Sciences for Central Europe, Earth System Science Data, 13, 4529–4565, https://doi.org/10.5194/essd-13-4529-2021, 2021.

Klotz, D., Kratzert, F., Gauch, M., Keefe Sampson, A., Brandstetter, J., Klambauer, G., Hochreiter, S., and Nearing, G.: Uncertainty estimation with deep learning for rainfall–runoff modeling, Hydrology and Earth System Sciences, 26, 1673–1693, https://doi.org/10.5194/hess-26-1673-2022, 2022.

Konold, O., Feigl, M., Podest, P., Klingler, C., and Schulz, K.: BiasCast: Learning and adjusting real time biases from meteorological forecasts to enhance runoff predictions, EGUsphere, 2025, 1–23, https://doi.org/10.5194/egusphere-2025-4978, 2025.

Kratzert, F., Klotz, D., Shalev, G., Klambauer, G., Hochreiter, S., and Nearing, G.: Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets, Hydrology and Earth System Sciences, 23, 5089–5110, https://doi.org/10.5194/hess-23-5089-2019, 2019.

Kratzert, F., Gauch, M., Klotz, D., and Nearing, G.: HESS Opinions: Never train a Long Short-Term Memory (LSTM) network on a single basin, Hydrology and Earth System Sciences, 28, 4187–4201, https://doi.org/10.5194/hess-28-4187-2024, 2024.

Lees, T., Buechel, M., Anderson, B., Slater, L., Reece, S., Coxon, G., and Dadson, S. J.: Benchmarking data-driven rainfall-runoff models in Great Britain: a comparison of long short-term memory (LSTM)-based models with four lumped conceptual models, Hydrology and Earth System Sciences, 25, 5517–5534, https://doi.org/10.5194/hess-25-5517-2021, 2021.

Loritz, R., Dolich, A., Acuña Espinoza, E., Ebeling, P., Guse, B., Götte, J., Hassler, S. K., Hauffe, C., Heidbüchel, I., Kiesel, J., Mälicke, M., Müller-Thomy, H., Stölzle, M., and Tarasova, L.: CAMELS-DE: hydro-meteorological time series and attributes for 1582 catchments in Germany, Earth System Science Data, 16, 5625–5642, https://doi.org/10.5194/essd-16-5625-2024, 2024.

Luce, A., Haag, I., and Bremicker, M.: Einsatz von Wasserhaushaltsmodellen zur kontinuierlichen Abflussvorhersage in Baden-Würtemberg, Hydrologie und Wasserbewirtschaftung, 50, 58–66, https://www.hywa-online.de/einsatz-von-wasserhaushaltsmodellen-zur-kontinuierlichen-abflussvorhersage-in-baden-wuerttemberg/, hyWa Heft 2, April 2006, 2006.

Moretti, G., Teltscher, K., Regenauer, J., Detering, A., Lier, J., Seibert, M., and Haag, I.: Nachkalibrierung LARSIM-Wasserhaushaltsmodelle Baden-Würtemberg, Technical report, Hydron Ingenieurgesellschaft für Umwelt und Wasserwirtschaft GmbH, prepared for the State Agency for Environment Baden-Württemberg (LUBW) Available upon request via hvz@lubw.bwl.de, 2022.

Nearing, G., Cohen, D., Dube, V., Gauch, M., Gilon, O., Harrigan, S., Hassidim, A., Klotz, D., Kratzert, F., Metzger, A., Nevo, S., Pappenberger, F., Prudhomme, C., Shalev, G., Shenzis, S., Tekalign, T. Y., Weitzner, D., and Matias, Y.: Global prediction of extreme floods in ungauged watersheds, Nature, 627, 559–563, https://doi.org/10.1038/s41586-024-07145-1, 2024.

Nearing, G. S., Kratzert, F., Sampson, A. K., Pelissier, C. S., Klotz, D., Frame, J. M., Prieto, C., and Gupta, H. V.: What role does hydrological science play in the age of machine learning?, Water Resources Research, 57, e2020WR028 091, https://doi.org/10.1029/2020WR028091, 2021.

Nearing, G. S., Klotz, D., Frame, J. M., Gauch, M., Gilon, O., Kratzert, F., Sampson, A. K., Shalev, G., and Nevo, S.: Technical note: Data assimilation and autoregression for using near-real-time streamflow observations in long short-term memory networks, Hydrology and Earth System Sciences, 26, 5493–5513, https://doi.org/10.5194/hess-26-5493-2022, 2022.

https://doi.org/10.5194/egusphere-2026-4885
Preprint. Discussion started: 3 September 2026

Nijzink, J., Loritz, R., Gourdol, L., Zoccatelli, D., Iffly, J. F., and Pfister, L.: CAMELS-LUX: Highly Resolved Hydro-Meteorological and Atmospheric Data for Physiographically Characterized Catchments around Luxembourg, Earth System Science Data Discussions, 2025, 1–34, https://doi.org/10.5194/essd-2024-482, 2025.

Saint-Fleur, B. E., Gaume, E., Surmont, F., Akil, N., and Theriez, D.: Testing discharge assimilation strategies to enhance short-range AI-based operational rainfall–runoff forecasts, Hydrology and Earth System Sciences, 30, 3497–3527, https://doi.org/10.5194/hess-30-3497-2026, 2026.

Taccari, M. L., Tazi, K., Morrison, O. M., Grafberger, A., Colonese, J., Carton de Wiart, C., Prudhomme, C., Mazzetti, C., Chantry, M., and Pappenberger, F.: AIFL: A global daily streamflow forecasting model using deterministic an LSTM pre-trained on ERA5-Land and fine-tuned on IFS, Journal of Hydrology, 678, 136 064, https://doi.org/https://doi.org/10.1016/j.jhydrol.2026.136064, 2026.

Tran, V. N., Xu, D., Van Nguyen, T., Kim, T., and Ivanov, V. Y.: CAMELSH: A Large-Sample Hourly Hydrometeorological Dataset and Attributes at Watershed-Scale for CONUS, Scientific Data, 12, 1307, https://doi.org/10.1038/s41597-025-05612-6, 2025.

Williams, R. J. and Zipser, D.: A Learning Algorithm for Continually Running Fully Recurrent Neural Networks, Neural Computation, 1, 270–280, https://doi.org/10.1162/neco.1989.1.2.270, 1989.

World Meteorological Organization: Guidelines on the Verification of Hydrological Forecasts, Technical Guide WMO-No. 1364, World Meteorological Organization (WMO), Geneva, Switzerland, https://library.wmo.int/idurl/4/69478, commission for Weather, Climate, Hydrological, Marine and Related Environmental Services and Applications (SERCOM), 2025.

Xia, Y., Mitchell, K., Ek, M., Sheffield, J., Cosgrove, B., Wood, E., Luo, L., Alonge, C., Wei, H., Meng, J., Livneh, B., Lettenmaier, D., Koren, V., Duan, Q., Mo, K., Fan, Y., and Mocko, D.: Continental-scale water and energy flux analysis and validation for the North American Land Data Assimilation System project phase 2 (NLDAS-2): 1. Intercomparison and application of model products, Journal of Geophysical Research: Atmospheres, 117, https://doi.org/https://doi.org/10.1029/2011JD016048, 2012.