Stochastic Environmental Research and Risk Assessment
https://doi.org/10.1007/s00477-025-02996-0

ORIGINAL PAPER

A long short-term memory model for sub-hourly temporal disaggregation of precipitation

Harrison Oates[1,2] · Nayan Arora[1] · Hong Gic Oh[1] · Trevor Lee[1]

Accepted: 12 April 2025

Abstract
High-resolution precipitation data is crucial for modern hydrological and building hygrothermal performance simulation models. In Australia, historical observations are inadequate, as half-hourly recordings only replaced daily observations at many stations from the early 2000s. Moreover, existing machine learning approaches are limited to generating hourly time series data. This paper presents a recurrent neural network using long short-term memory to disaggregate daily precipitation observations into half-hourly intervals. The model leverages temporal dependencies and hourly weather measurements. Our results, based on stations across five Australian climate zones, demonstrate that the model effectively preserves key half-hourly precipitation statistics, including variance and the quantity and distribution of wet half-hours. When aggregated to hourly intervals, our model outperforms other models in most measured metrics.

Keywords Half-hourly precipitation · Temporal disaggregation · Stochastic precipitation generation · Long short-term memory · Neural networks · Machine learning

1 Introduction

Building simulation and modelling software is increasingly used to optimize design parameters for energy efficiency and to predict performance of a building's systems under various conditions (de Wilde 2023). Meaningful weather and climate data forms an essential component of these systems, as well as other systems like hydrological models (Horton et al. 2022). In order to define a climate normal, the World Meterological Organization recommends using at least thirty years of historical data (WMO 2023, p. 25). Especially in the case of precipitation, shorter periods may not produce reliable statistics due to annual variances.

However, thirty years of high-resolution precipitation data suitable for these applications is not always available. In the Australian context, for instance, half-hourly precipitation readings are often only available since the late 1990s and early 2000s, when the Bureau of Meteorology (BoM) installed automatic Tipping Bucket Rain Gauges (Australian Bureau of Meteorology 2010). Prior to this, precipitation data was primarily collected through daily manual readings of the rain gauge by post office staff or volunteers at 0900 local time. As hourly or sub-hourly data is essential for reliable built environment modelling (Brigandì and Aronica 2019), there is a clear need for precipitation disaggregation algorithms that can produce this data.

This paper presents a novel approach to precipitation disaggregation by introducing a long short-term memory (LSTM) network for daily-to-half-hourly precipitation disaggregation. To our knowledge, this is the first machine learning method to achieve this level of temporal resolution in disaggregation. Prior work has focussed on daily-to-hourly disaggregation (Bhattacharyya and Saha 2022) or coarser resolutions. Our model leverages the ability of LSTMs to capture temporal dependencies in sequential data, thus improving the consistency of the disaggregated

✉ Harrison Oates
harrison@harrisonoates.com

Nayan Arora
nayan.arora.work@gmail.com

Hong Gic Oh
honggic.oh@exemplary.energy

Trevor Lee
trevor.lee@exemplary.com.au

[1] Exemplary Energy, 32 Fihelly Street, Canberra, ACT 2604, Australia

[2] School of Computing, The Australian National University, Canberra, ACT 2604, Australia

Published online: 07 May 2025

sequences. Critically, we introduce a novel normalization layer, integrated directly into the network architecture, that guarantees the conservation of the daily precipitation total. This ensures that the sum of the disaggregated half-hourly values exactly matches the observed daily total, a fundamental requirement of disaggregation that is often not strictly enforced in 'black-box' ML models. Furthermore, our model incorporates key hourly meteorological variables (atmospheric pressure, dry-bulb and dew point temperatures, and relative humidity). This integrated approach—combining context, a normalization layer, and meteorological predictors—results in a novel method for daily-to-half-hourly precipitation disaggregation.

A key benefit of generating half-hourly precipitation instead of hourly precipitation is that it enables consistency between different file formats. For instance, the EnergyPlus Weather (EPW) format used in the building energy simulation program EnergyPlus (US Department of Energy 2024), and the Australian Climate Data Bank (ACDB) format used by the Nationwide House Energy Rating Scheme (NatHERS) software (Tan et al. 2023) use different timestamp conventions. Both file types use hourly data, but their timestamp conventions differ by half an hour. EPW records the hour before the timestamp, while ACDB records the hour centred on the timestamp. This difference is neatly illustrated in Fig. 1. Half-hourly precipitation can be re-aggregated to accommodate this difference.

Disaggregation, like downscaling, is the process of producing high-resolution data that is statistically consistent with the original, coarser-scale data (Koutsoyiannis 2003). While similar to downscaling, disaggregation has the additional requirement that the sum of the disaggregated values should closely match the total from the original, coarser-resolution data (Knoesen and Smithers 2009).

Various stochastic methods have been developed for both disaggregation and downscaling. Random cascade models, which distribute precipitation iteratively into successively smaller timescales according to a generator function that stochastically partitions precipitation amounts while adhering to either canonical (average) or microcanonical (strict) conservation of coarser-scale totals, have demonstrated success in maintaining statistical properties of precipitation (Olsson 1998; Müller and Haberlandt 2018). Microcanonical models are particularly relevant to temporal disaggregation, as they preserve precipitation totals across each scale, unlike canonical approaches which do not exhibit this property. Similarly, Poisson cluster models, such as the Bartlett-Lewis and Neymann-Scott models, have been widely applied to disaggregate daily precipitation by simulating rainfall as a sequence of storm events (Onof and Wang 2020; Cowpertwait and O'Connell 1997; Yusop et al. 2013; Koutsoyiannis and Onof 2001; Onof et al. 2000).

While these stochastic approaches have proven effective in many applications, they still display some disagreement with recorded measurements and require a large number of parameters for modelling (Ferrari et al. 2022; Cowpertwait et al. 2007; Rohith et al. 2020). Calibration of these parameters introduces complexity into the modelling process and can impact model performance and reliability. Computational efficiency presents another challenge. Stochastic methods, such as Markov chain Monte Carlo, often require significant resources and time to converge. Moreover, calculating the time to convergence for these methods cannot be done in polynomial time (Bhatnagar et al. 2011). This increased computational burden not only raises infrastructure costs but also limits the practicality of these algorithms.

Given these drawbacks, many researchers have turned to machine learning techniques, particularly neural networks, for precipitation disaggregation. Early work by Burian et al. (2001) investigated the use of feed-forward neural networks to disaggregate hourly data into 15-min intervals. Building on this approach, Bhattacharyya and Saha (2022) extended the application of feed-forward networks to perform daily-to-hourly disaggregation. Their model disaggregated precipitation for day t by inputting the daily totals of days t and $t-1$, along with the day, month, and a category label. This label was derived from k-means clustering ($k = 4$) on the daily totals of days $t, t-1$ and $t+1$. However, the model did not include meteorological variables which are known to be associated with the onset of precipitation, such as atmospheric pressure or humidity (Hintz et al. 2019).

While not directly addressing disaggregation, the work of Misra et al. (2017) on precipitation downscaling using long short-term memory (LSTM) networks opened new avenues for research. Their success in downscaling precipitation from climatic variables generated by general circulation models suggests that similar recurrent neural network architectures could be effective for disaggregation tasks. This approach's ability to capture temporal dependencies in sequential data makes it particularly promising for precipitation modelling. Inspired by this prior work, we implement a neural network with LSTM layers for sub-hourly disaggregation using

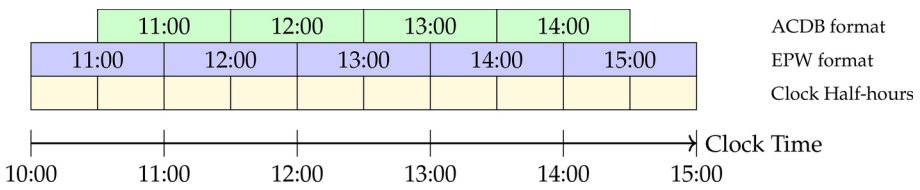


Fig. 1 Timestamp conventions for ACDB and EPW formats

Table 1 Stations used in the study

Location	WMO index	Station coordinates	Climate zone	Half-hourly record start	Half-hourly precipitation missing %
Cairns (CN)	94,287	– 16.87, 145.75	Climate Zone 1	2001–2009	2.81
Brisbane (BR)	94,578	– 27.39, 153.13	Climate Zone 2	2000–2003	0.92
Sydney (SY)	94,768	– 33.86, 151.20	Climate Zone 5	1998–2012	6.77
Melbourne (ME)	94,868	– 37.83, 144.98	Climate Zone 6	1997–2010	8.92
Canberra (CA)	94,926	– 35.30, 149.20	Climate Zone 7	2000–2004	12.01

Table 2 Summary of variables used in study

Variable	Resolution
Precipitation (mm)	Half-hourly
Atmospheric pressure (hPa)	Hourly
Dry-bulb temperature (Tenths of °C)	Hourly
Dew point temperature (Tenths of °C)	Hourly
Relative humidity (%)	Hourly

PyTorch (Paszke et al. 2019), extending the application of these models to finer temporal resolutions.

For our study, we exploit historical weather data for several Australian cities, where hourly weather data and half-hourly precipitation data is available since at least 2001. We show that our approach preserves the statistical metrics examined in our study from the observed time series, and performs favourably compared to other disaggregation approaches. The rest of this paper is organized as follows. Section 2 introduces the stations under study and describes the data utilized by the model, followed by a detailed explanation of the model architecture and implementation specifics. Section 3 presents the model's results and provides a comprehensive analysis of its performance across various metrics. Finally, Sect. 4 summarizes the model's performance, its implications, and outlines directions for future research.

2 Methodology and data

2.1 Areas of study

Our investigation utilizes data from five stations across Australia. Table 1 presents these stations, their locations and their respective climate zones as defined by the Australian Building Codes Board (2024). The table also indicates the year and month when a tipping bucket rain gauge was installed at each station, marking the onset of half-hourly precipitation data availability. For all stations, the data series continues through the end of 2022.

2.2 Data preprocessing and variable selection

Hourly meteorological data (excluding precipitation) for each station was provided to the research team in the TMY2 format. These TMY2 files did not contain missing values for the variables used in our analysis; Exemplary Energy had already filled missing data using techniques prescribed in the TMY2 specification manual (Marion and Urban 1995). After parsing, the data is linearly interpolated to half-hourly intervals and inner joined with the station's precipitation data. Datetimes with missing precipitation values are excluded from the join.

While Cairns' and Brisbane's data required no temporal adjustment, the other locations' daily precipitation measurements needed to be aligned to account for daylight saving time. These locations undergo biannual one-hour shifts, requiring temporal alignment between the daily precipitation readings and other meteorological elements, as the half-hourly precipitation data is consistently recorded in Australian Eastern Standard Time.

To train and test the model, we split the data into three sets. We test the model on all data from 2020 to 2022, use 2018 and 2019 as the validation set to evaluate the model during training, and train the model on all remaining data. This results in a train-test-validation split of roughly 75%–10%–15%. The training dataset was shuffled to allow the model to learn from a more representative sample in each batch.

We selected station atmospheric pressure, dry bulb and dew point temperatures, and relative humidity as the model's input features based on both the reliability of these measurements and their Pearson correlation to precipitation in the Sydney dataset. Table 2 summarises the variables used in the summary and their respective resolutions.

While cloud cover has a strong correlation with precipitation (Mishra 2019), we excluded it from our feature set. In the early years of BoM datasets, cloud cover data is often only available as a derivative of insolation. This results in unreliable linear interpolations between pre-dusk and post-dawn values for nighttime hours.

The selected features were standardized by centering (subtracting the mean) and scaling to unit variance. The preprocessed data was then loaded into a PyTorch `Dataset` class, where it was grouped by day starting from 0900 and processed into three tensors: an input sequence tensor, a target tensor, and a daily total tensor.

2.3 Model architecture and implementation

Feed-forward neural networks are limited to providing a static mapping between input and output, and hence cannot represent context. Context, however, is an important component of time-prediction tasks such as precipitation disaggregation, where each timestep is impacted by preceding timesteps.

In order to model context, signals from previous timesteps can be fed back into the network, with such models known as Recurrent Neural Networks, or RNNs (Staudemeyer and Morris 2019). Theoretically, RNNs should be able to preserve long-term dependencies at an arbitrary timeframe, however, RNNs suffer from problems such as vanishing gradients, which can slow or stop training of the network completely (Pascanu et al. 2013). One such solution that addresses the vanishing gradient problem is long short-term memory (LSTM). An LSTM consists of a memory cell with three gates: an input gate, an output gate, and a forget gate, which together control the flow of information into and out of the cell (Gers et al. 2000).

The basic architecture of our model can be seen in Fig. 2. On a high level, the model processes input sequences of 48 time steps through two successive LSTM layers of 62 units each, before the daily total constraint is enforced by the normalization layer.

To understand how the data is transformed, we now walk through the network layer by layer. Let b represent the batch size of the input. Then, the input is of size $(b, 48, 4)$: b sequences of 48 half-hours, each having 4 input features. This is passed through two LSTM layers, each with 62 units to capture both short-term and long-term temporal dependencies, outputting a tensor of size $(b, 48, 62)$. A fully connected layer transforms this tensor into a single output for each entry in the sequence. This results in a tensor of size $(b, 48, 1)$. A rectified linear unit (ReLU) activation function (Agarap 2019), defined by

$$f(x) = \max(0, x)$$

acts upon each prediction and ensures outputs are non-negative (as precipitation cannot be negative) before the singleton dimension is removed by a squeeze operation. The resulting tensor, alongside a tensor $(b, 1)$ of daily totals, are input into the normalization layer, which scales each day's predictions to ensure they sum to the known daily total. The final tensor of size $(b, 48)$ contains the half-hourly predictions for each day contained in the batch, effectively disaggregating the daily totals into a plausible sub-daily distribution.

The normalization layer is a novel addition to our network in the context of disaggregation. Its primary purpose is to ensure that the predicted half-hourly precipitation values sum exactly to the known daily total, maintaining consistency between our model's output and the observed sequence. For each daily sequence vector $\vec{x}$ and daily total t, we define the layer in the following way:

$$F(\vec{x}, t) = \begin{cases} \vec{0}, & \text{for } t < \varepsilon \\ t \cdot \left(\left(\sum_{i=1}^{48} \vec{x}_i \right) + \varepsilon \right)^{-1} \vec{x} & \text{otherwise} \end{cases}$$

where $\varepsilon > 0$ is a small constant to ensure the function is differentiable across its domain, which is crucial for back-propagation during network training, and to account for floating point error. We set $\varepsilon = 10^{-8}$ in our experiments, though this choice was largely arbitrary. The specific value of ε has minimal impact on the results as long as it is small enough to maintain numerical stability while preserving the intended behaviour of the normalization.

We now construct our loss function, which the network will try to minimize. Let $p, q \in \mathbb{R}^{48}$ be the predicted and target vectors respectively. We define the loss function ℓ as

$$\ell(p, q) = \text{MSE}(p, q) + KL(\sigma(p), \sigma(q)) + |V(p) - V(q)| \quad (1)$$

Where

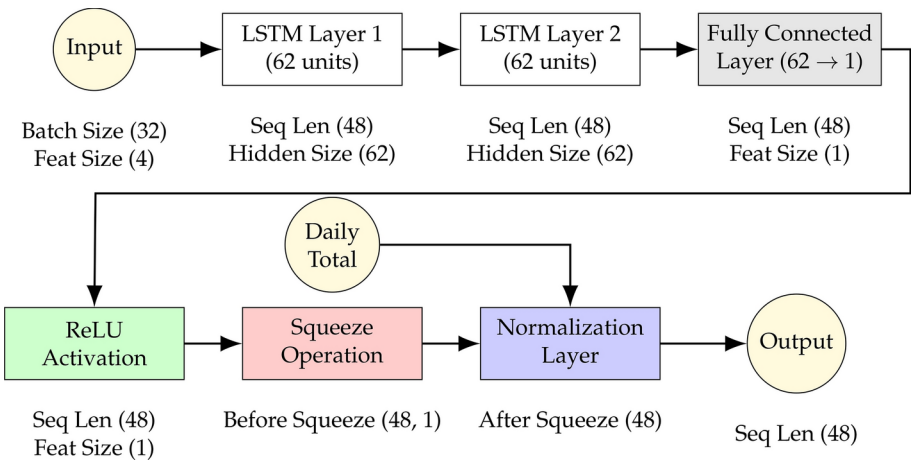


Fig. 2 Model architecture

$$\mathrm{MSE}(p,q) = \frac{1}{48}\sum_{i=1}^{48}(p_i - q_i)^2 \tag{2}$$

$$\mathrm{KL}(\sigma(p),\sigma(q)) = \sum_{i=1}^{48}\sigma(p)_i \log\frac{\sigma(p)_i}{\sigma(q)_i} \tag{3}$$

$$\sigma(z)_i = \frac{e^{z_i}}{\sum_{j=1}^{48} e^{z_j}},\ \text{with}\ \sigma : \mathbb{R}^{48} \to (0,1)^{48} \tag{4}$$

$$V(z) = \frac{1}{48}\sum_{i=1}^{48}(z_i - \mu)^2,\ \text{with}\ \mu = \sum_{i=1}^{48}\frac{z_i}{48} \tag{5}$$

As shown in Eq. (1), our loss function combines three complementary components to address different aspects of modelling. Equation (2) measures the mean squared error between the elements of p and q, providing a measure of overall prediction accuracy. Equation (3) defines Kullback-Liebler divergence (Kullback and Leibler 1951), which measures the dissimilarity between the probability distributions obtained from p and q using the softmax function σ (Goodfellow et al. 2016). KL divergence helps to assess differences in the distribution of rainfall throughout a day. This combination has been successfully employed in various machine learning architectures, particularly in variational autoencoders (Lucas et al. 2019), where it balances reconstruction fidelity with distributional matching. We extended this established framework by incorporating a variance matching term $|V(p) - V(q)|$. This addition is motivated by the need to preserve the statistical characteristics of precipitation, particularly the frequency and intensity of extreme values.

This loss function is minimized by the optimization algorithm Adam (Kingma and Ba 2014). To reduce training times and increase learning stability, we use batch normalization (Ioffe and Szegedy 2015) with a batch size of 32. The model is implemented in PyTorch (Paszke et al. 2019), with data preprocessing done using pandas (McKinney 2010) and scikit-learn (Pedregosa et al. 2011). All training and inference was performed on a single Nvidia 4070 Ti Super GPU. We train a new model for each location to account for local climate characteristics.

Table 3 Model and training hyperparameters

Hyperparameter	Value
LSTM hidden layer size	64
LSTM layer count	2
LSTM output size	1
Batch size	32
Epochs	140 initial + 50 for fine-tuning
Learning rate	10^{-3} initial / $5 \cdot 10^{-6}$ fine-tuning
Scheduler patience	6
Scheduler factor	0.68
Scheduler minimum learning rate	10^{-6}
Dropout	0.24

The validation dataset is used to select the optimal model weights. At the end of each epoch, we compare the epoch's validation loss to the best validation loss achieved so far. If epoch validation loss is lower, the current model state is saved and the best validation loss variable is updated accordingly. The model saved at the conclusion of this process was then used for our final evaluations. This approach ensures that we select the model with the best generalization performance on unseen data, helping to mitigate overfitting to the training set.

The model is trained in two stages, in order to first learn general precipitation patterns and then make fine-grained adjustments to fine-tune the model. In the initial training run of 140 epochs, we use a learning rate scheduler with a reduction on plateau strategy (Loshchilov and Hutter 2017), which monitors validation loss and reduces the learning rate by a factor of 0.68 when the validation loss stops improving for six consecutive epochs. Initially, the learning rate is set at 10^{-3}. This adaptive approach helps prevent the optimization process from stagnating in local minima while allowing for finer parameter updates to occur as training progresses. After this initial training has completed, the saved model state is re-loaded and trained with a fixed learning rate of $5 \cdot 10^{-6}$ for a further 50 epochs.

Each epoch takes about 2.8 s to run, with the entire training and testing pipeline taking just over nine minutes to complete per location.

A summary of all hyperparameters used in the model's architecture and training can be seen in Table 3.

3 Results and discussion

Performance evaluation of the model is done by comparing the generated series with the observed series with regards to the following metrics on the test dataset:

- Temporal mean and variance
- Root mean squared error and normalized mean squared error
- Number of precipitation half-hours
- Proportion of correctly detected precipitation half-hours
- Skill score
- Pearson product-moment correlation coefficient

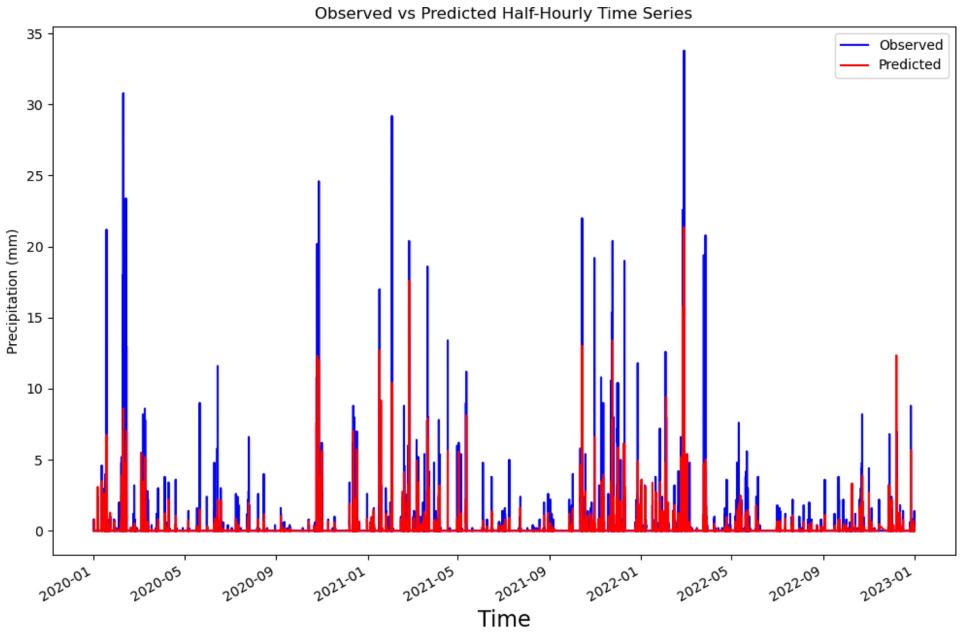
We first present results for the base architecture consisting of two LSTM layers with 62 memory cells in each layer,

before considering the performance of architecture variations. Figures 3 and 4 show comparisons of the precipitation sequences generated by the model and from the observed data at half-hourly and re-aggregated hourly resolution respectively.

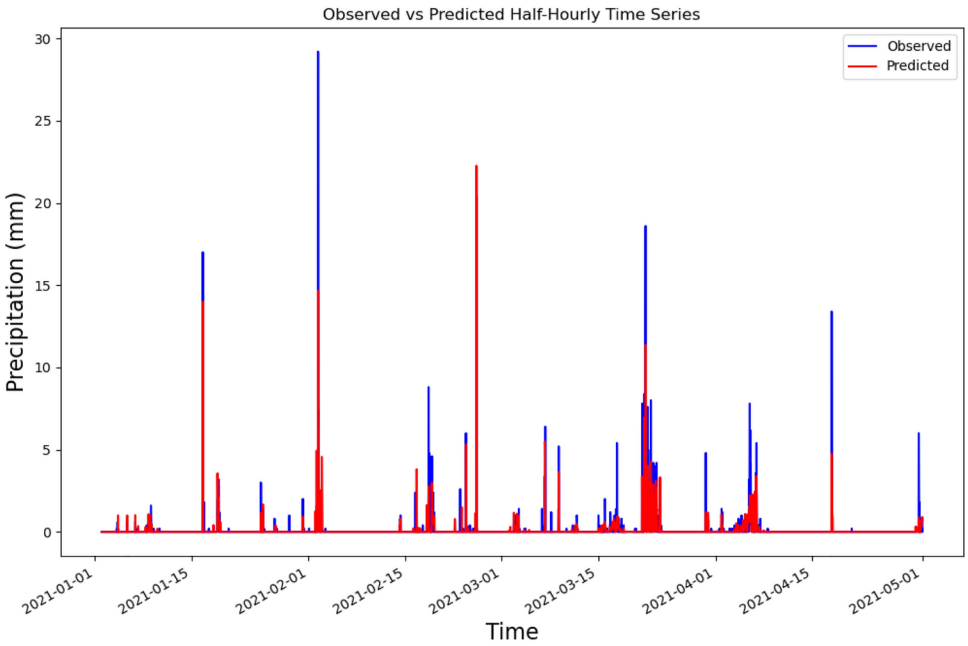
3.1 Comparison of statistical characteristics

Means and variance of the observed precipitation for each location are compared with the disaggregated series for the test dataset. We also calculate the root mean squared error, and normalized mean squared error[1]. Mean squared error is

[1] $NMSE = MSE/\operatorname{Var}(y)$.

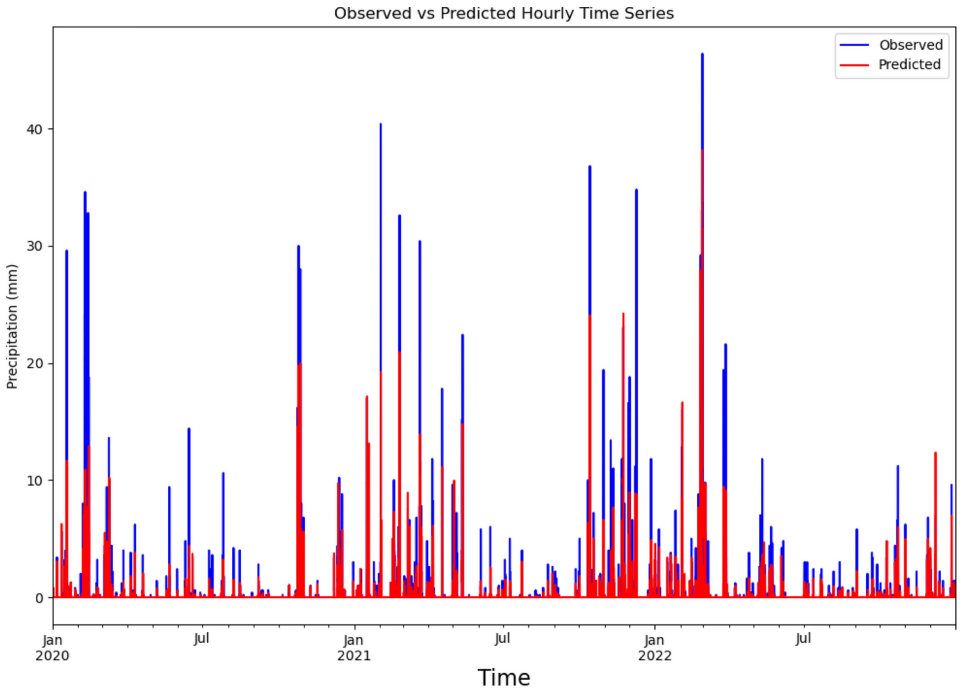


(a) Full test set

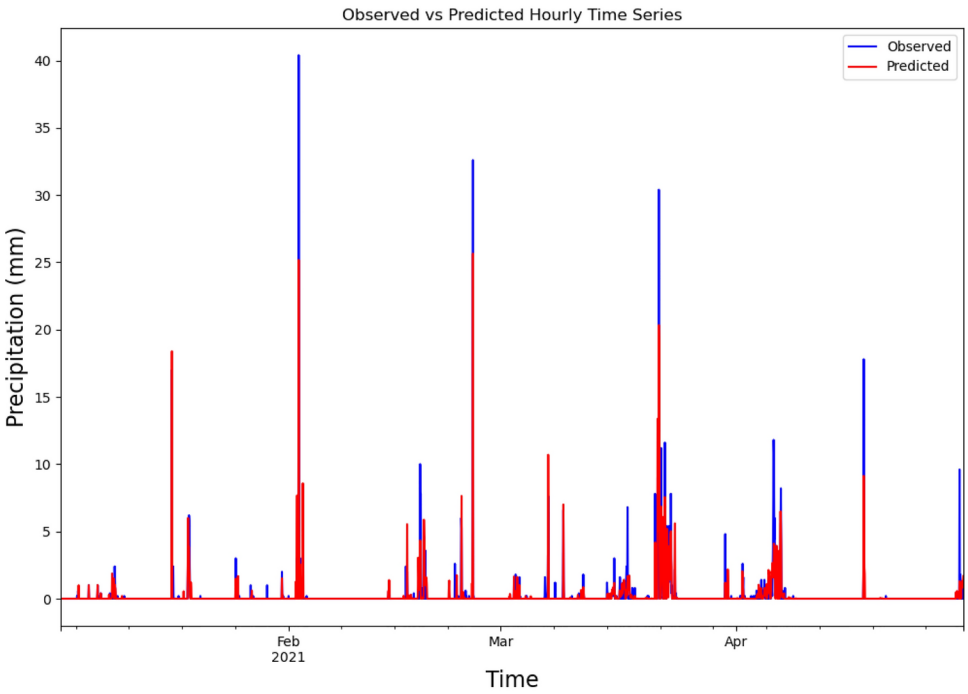


(b) Test set Jan. 2021 - Apr. 2021

Fig. 3 Predicted versus observed half-hourly time series for Brisbane



(a) Full test set



(b) Test set Jan. 2021 - Apr. 2021

Fig. 4 Predicted versus observed hourly time series for Brisbane

Table 4 Statistical characteristics of half-hourly observed and predicted series

	Location					Average
	CN	BR	SY	ME	CA	
Mean (mm)						
Observed	0.1027	0.0788	0.1015	0.0409	0.0512	0.07502
Predicted	0.1027	0.0788	0.1014	0.0409	0.0512	0.075
Variance $(\mathrm{mm})^2$						
Observed	0.7919	0.5060	0.4124	0.0894	0.1149	0.3829
Predicted	0.3039	0.2584	0.1831	0.0409	0.0457	0.1664
Relative error (%)	61.624	48.933	55.601	54.251	60.226	56.127
Root mean squared error (mm)	0.6899	0.567	0.5136	0.2345	0.2857	0.4581
Normalized mean squared error	0.6011	0.6353	0.6397	0.6151	0.7106	0.6404

Table 5 Statistical characteristics of re-aggregated hourly series

	Location					Average
	CN	BR	SY	ME	CA	
Mean (mm)						
Observed	0.1933	0.1488	0.1980	0.0758	0.0868	0.14054
Predicted	0.1933	0.1488	0.1979	0.0758	0.0868	0.14052
Variance $(\mathrm{mm})^2$						
Observed	2.1752	1.5035	1.2141	0.2533	0.3128	1.0918
Predicted	1.0627	0.8632	0.6723	0.1415	0.1495	0.5778
Relative Error (%)	51.145	42.587	44.626	44.137	52.206	46.940
Root mean squared error (mm)	1.0269	0.8975	0.8195	0.3637	0.4513	0.7118
Normalized mean squared error	0.4848	0.5358	0.5532	0.5194	0.6512	0.5489

sensitive to outliers, and climate zones vary in the volatility of their rainfall. We normalize mean squared error in order to better facilitate the comparison of error between climate zones.

The results are shown in Table 4. For comparison with existing methods for daily-to-hourly disaggregation, we also present the same statistics for the partially re-aggregated series in Table 5.

The hourly model presented by Bhattacharyya and Saha (2022) has a relative error in mean rainfall of 7.516% and a relative error in variance of 36.2%. In comparison, our LSTM model demonstrates significantly improved performance on the mean, with only a slight increase in error for the variance.

While the model generally captures the temporal patterns of rainfall, it tends to underestimate the magnitude of extreme events, as demonstrated in Fig. 4 for Brisbane. This underestimation is reflected in the measured variance error, and arises from two main causes. The first cause is the smoothing effect of LSTM predictions (Waqas and Humphries 2024), as the model prioritizes long-term dependencies over sharp fluctuations. The second cause is the scarcity of extreme rainfall instances in the training data. In the Brisbane training set, for instance, 6.13% of wet days featured at least one instance of half-hourly precipitation exceeding 10 mm (126 of 2055), with such events comprising just 1.5% of all wet half-hours. This imbalance in the training data naturally biases the model toward more moderate predictions, compounding the inherent smoothing tendency of the LSTM architecture.

For a more direct comparison of performance, we can consider the Markov chain Monte Carlo (MCMC) method for hourly disaggregation presented by Ferrari et al. (2022), which also uses Canberra BoM data for evaluation. Comparing performance on Canberra specifically, our model achieved an RMSE of 0.4513, representing a 30.57% improvement over the RMSE of 0.65 reported by Ferrari et al.. Overall, the results indicate that our model is robust and outperforms existing methods in preserving the studied statistical characteristics of the observed precipitation series.

3.2 Preservation and detection of precipitation intervals

Bhattacharyya and Saha (2022) highlights preservation of the number of dry hours as a useful metric by which to evaluate a disaggregation model. Consequently, we present the number of dry and wet periods for both half-hourly and re-aggregated hourly measurements. A time period is *dry* if precipitation is less than 0.2 mm. A time period is *wet* if it is not dry. This cutoff is chosen because 0.2 mm is the smallest amount of precipitation that can be measured by BoM's equipment.

Table 6 Number of wet and dry half-hours for each location

	Location					Average
	CN	BR	SY	ME	CA	
Number of dry half-hours						
Observed	46,026	46,834	46,805	46,113	41,895	
Predicted	45,580	46,596	46,177	46,326	41,303	
Relative error (%)	0.967	0.508	1.341	0.462	1.413	0.939
Number of wet half-hours						
Observed	3414	2846	4507	2655	2649	
Predicted	3860	3084	5135	2442	3241	
Relative error (%)	13.06	8.36	13.93	8.022	22.35	13.15
Number of dry hours						
Observed	23,859	24,337	23,244	24,414	24,539	
Predicted	23,720	24,168	22,769	24,468	24,128	
Relative error (%)	0.583	0.694	2.044	0.221	1.675	1.043
Number of wet hours						
Observed	2421	1967	3060	1890	1765	
Predicted	2560	2136	3535	1836	2176	
Relative error (%)	5.74	8.59	15.52	2.86	23.29	11.20
Correctly detected wet half-hours (%)						
± 0 half hours	65.88	60.68	70.85	55.86	68.14	64.28
± 1 half hour	76.24	70.41	77.50	65.57	75.20	72.98
± 2 half hours	81.40	75.79	81.34	69.87	78.75	77.43
Correctly detected wet hours (%)						
± 0 half hours	70.92	63.85	75.00	61.16	74.28	69.04
± 1 half hour	81.58	75.29	83.40	72.49	81.42	78.87
± 2 half hours	86.16	80.48	87.29	77.88	85.50	83.46

Table 7 Correlation and skill score measurements

	Location					Average
	CN	BR	SY	ME	CA	
R half-hourly	0.6317	0.6125	0.6035	0.6227	0.5449	0.6031
R hourly	0.7180	0.6852	0.6723	0.6952	0.5980	0.6737
Skill score half-hourly	0.7127	0.7024	0.6759	0.6932	0.7401	0.7049
Skill score hourly	0.6866	0.6533	0.6255	0.6498	0.6961	0.6623

Following Ferrari et al. (2022) we also evaluate the model's timing accuracy by calculating the percentage of correctly-detected wet periods, considering both half-hourly and hourly intervals. We allow for 0, ± 1, and ± 2 interval margins of error. For example, if precipitation occurs at 10:00, predictions at 09:30, 10:00, or 10:30 are considered correct with a ± 1 half-hour tolerance. Table 6 presents these results for each location.

The LSTM model has an average error of 1.04% in the total number of dry hours. This represents a 95.04% reduction in error over the results reported by Bhattacharyya and Saha, who observed an error of 20.96%. Such a substantial enhancement in capturing dry periods indicates that our model more faithfully reproduces the intermittent nature of precipitation patterns than the feed-forward neural network.

The LSTM model also displays improved performance in timing accuracy. It can detect 83.46% of wet hours within a $\pm$ 2 h window, and 69.04% of wet hours with no error. This is a significant improvement over the MCMC model by Ferrari et al., which detects 60% of wet hours with $\pm$ 2 h error, and 20% of wet hours with no error. Even at half-hourly resolution without re-aggregation, the LSTM maintains high accuracy, detecting 64.28% of wet half-hours with no error. The improved timing accuracy suggests a better capability to capture the temporal dynamics of precipitation events compared with the MCMC model.

3.3 Correlation and skill score

The temporal correlation between the predicted and observed series and the skill score for each location can be seen in Table 7.

The skill score presented by Perkins et al. (2007) measures the relative similarity of two probability density functions and is an effective metric to capture the ability of the

model to simulate the distribution of precipitation over a day.

$$S_{\text{score}} = \sum_{i=1}^{n} \min(p_i, q_i) \tag{6}$$

where n represents the number of intervals per day (24 for hourly data, 48 for half-hourly data), and $p, q \in \mathbb{R}^{48}$ the predicted and target tensors respectively. A skill score of 1 indicates that the distributions are identical, while a skill score of 0 indicates that the distributions have no common area between them.

The model achieves correlations of 0.60 and 0.67 for the half-hourly and hourly series respectively. This indicates that the model effectively captures the timing and intensity of precipitation events at fine temporal scales. It also demonstrates high skill in reproducing the probability distribution of precipitation intensities. With an average skill score of 0.70 for half-hourly and 0.66 for hourly, the model shows a strong ability to match the observed distribution of precipitation.

3.4 Climate-based performance analysis

To assess whether model performance varies across climate types, we performed a statistical anaylsis comparing RMSE and variance error across wetter and drier regions. Annual rainfall statistics for this purpose were sourced from BoM (2025). Table 8 presents the results of our analysis.

The analysis revealed a strong, statistically significant ($p < 0.05$) positive correlation between RMSE and annual rainfall for both the half-hourly ($r = 0.926, p = 0.024$) and re-aggregated hourly ($r = 0.893, p = 0.042$) outputs. This indicates that the absolute error of the model increases in regions with higher annual precipitation. However, when considering NMSE no statistically significant relationship with annual rainfall was observed (half-hourly: $r = 0.557, p = 0.330$; hourly: $r = -0.606, p = 0.279$). Similarly, variance errors showed weak, non-significant correlations with annual rainfall across both temporal resolutions, suggesting that the model's tendency to underestimate variability is independent of climate type.

Direct comparison between wet and dry climate zones reinforced these findings. Cairns (1982 mm), Brisbane (1080 mm), and Sydney (1044 mm) all experience average annual precipitation over 1000 mm, and so were categorised as 'wet', while Melbourne and Brisbane were categorised as 'dry' due to annual precipitation of 518 mm and 603 mm, respectively. T-tests showed significant differences in RMSE between wet and dry climates for both half-hourly ($t = 5.678, p = 0.014$) and hourly ($t = 6.791, p = 0.007$) disaggregation. The absolute errors were consistently higher in wet climate zones. However, the NMSE showed no statistically significant differences between climate types (half-hourly: $t = -0.761, p = 0.574$; hourly: $t = -0.925, p = 0.508$), indicating that the relative model performance remains consistent across difference climate types when accounting for precipitation variance.

Table 8 Statistical analysis of model performance across different climate types

Metric	Half-hourly	Hourly
Pearson Correlation (RMSE vs. Annual Rainfall)	0.926 ($p = 0.024$)	0.893 ($p = 0.042$)
Pearson Correlation (NMSE vs. Annual Rainfall)	0.557 ($p = 0.330$)	-0.606 ($p = 0.279$)
Pearson Correlation (Variance Error vs. Annual Rainfall)	0.346 ($p = 0.569$)	0.287 ($p = 0.640$)
T-test for RMSE (Wet vs. Dry Climates)	$t = 5.678$, $p = 0.014$	$t = 6.791$, $p = 0.007$
T-test for NMSE (Wet vs Dry Climates)	$t = -0.761$, $p = 0.574$	$t = -0.925$, $p = 0.508$

3.5 Alternative model architectures

We now consider the impact that altering the number of hidden LSTM layers and the number of LSTM units in each layer has on the disaggregation performance of the model, using Sydney data to facilitate the comparison. We call a model with hidden size h and l layers model (h, l) for brevity (Table 9).

The base configuration $(62, 2)$ has the second-lowest RMSE and NMSE of the tested configurations, with $(124, 3)$ outperforming it by a small margin. However, $(62, 2)$ has a smaller error in variance and more faithfully reproduces the number of wet and dry half-hours. $(124, 3)$ suffers from a further problem that is not reflected in the above results: that of dying neurons. Due to the ReLU layer, bad initializations of the model can cause the whole network to become a constant function. This prevents the model from learning, and is more often seen in deeper networks than shallower ones (Lu 2020). Consequently, it often needs to be re-initialized several times in order to produce a satisfactory result. The phenomenon was also observed in $(62, 3)$, albeit to a lesser extent.

Alternate activation functions like the exponential linear unit (Clevert et al. 2016) have been developed to mitigate the dying neuron problem. However, our implementation specifically requires ReLU activation because the normalization layer depends on each time step having non-negative values, which ReLU guarantees. Using alternative activation functions that allow negative values (such as ELU, Leaky ReLU, or tanh) would fundamentally disrupt the daily total constraint enforced by our normalization layer,

Table 9 Statistical characteristics of alternate model predictions

	Number of hidden layers		
	1	2	3
RMSE(mm)			
Hidden size 31	0.5621	0.5133	0.5079
Hidden size 62	0.5211	0.5056	0.5118
Hidden size 124	0.5436	0.5272	**0.5052**
Normalized MSE			
Hidden size 31	0.7662	0.6388	0.6256
Hidden size 62	0.6585	0.6198	0.6353
Hidden size 124	0.7165	0.6740	**0.6194**
Variance error			
Hidden size 31	**50.02%**	68.72%	63.87%
Hidden size 62	61.37%	61.83%	60.96%
Hidden size 124	49.95%	56.23%	62.07%
Dry half-hour error			
Hidden size 31	1.62%	2.96%	2.17%
Hidden size 62	2.01%	1.73%	1.70%
Hidden size 124	1.08%	**1.59%**	1.85%
Wet half-hour error			
Hidden size 31	16.84%	30.73%	22.50%
Hidden size 62	20.92%	17.95%	17.71%
Hidden size 124	**11.23%**	16.46%	19.21%

Bold indicates the best result for each metric. Underlined indicates the second-best result

Table 10 LSTM performance on SY data with and without normalization layer

Metric	Observed value	With layer	Without layer
Mean (mm)	0.1015	0.1014 (0.10%)	0.1752 (72.61%)
Variance $(\text{mm})^2$	0.4124	0.1831 (55.60%)	0.1783 (56.77%)
Number of dry half-hours	46,805	46,117 (1.34%)	40,729 (12.98%)
Number of wet half-hours	4507	5135 (13.93%)	10,583 (134.81%)
RMSE	NA	0.5136	0.6161
NMSE	NA	0.6397	0.9203
R	NA	0.6035	0.3993
Skill score	NA	0.6759	0.6774

Values in brackets indicate relative error from observed value

requiring substantial architectural redesign and potentially compromising the conservation properties of our model.

While deeper and wider networks like $(124, 3)$ can achieve slightly better error rates, they come with the drawback of potential initialization issues and less accurate reproduction of certain precipitation characteristics. This suggests that the base configuration $(62, 2)$ offers a good balance between performance and stability.

3.6 Ablation study into effect of normalization layer

The normalization layer of the LSTM ensures that the predicted half-hourly precipitation values sum exactly to the observed daily total, maintaining physical consistency in disaggregation. To evaluate its impact, we conduct an ablation study (Meyes et al. 2019) by training the model with and without this component on Sydney data and comparing performance metrics. Table 10 presents a comparison of these configurations against observed values.

The results indicate that the normalization layer substantially improves model performance over multiple metrics. Most notably, the normalized model reproduces the observed mean precipitation with 0.10% error, whereas the non-normalized model significantly overestimates mean precipitation (72.61% error). This demonstrates the layer's primary function of enforcing the daily total constraint. Both models underestimate precipitation variance to a similar degree (55.60% with normalization vs 56.77% without), suggesting that the normalization process does not significantly affect the model's ability to capture precipitation variability, although it does help. However, the representation of precipitation patterns differs considerably between the two approaches. The normalized model closely approximates the observed frequency of dry periods (1.34% error) while moderately overestimating wet half-hours (13.93%). In contrast, the non-normalized model substantially underestimates the number of dry half-hours (12.98% error) and dramatically overestimates wet half-hours (134.81% error).

The performance metrics further support the value of the normalization layer. The normalized model achieves lower error values for both RMSE (0.5136 vs. 0.6161) and NMSE (0.6397 vs. 0.9203). The correlation coefficient R is substantially higher with the normalization layer (0.6035) than without (0.3993), indicating improved alignment with observed precipitation patterns. The skill score changes minimally.

These ablation results demonstrate that the normalization layer maintains or improves the performance of the model across the evaluation metrics.

3.7 Sensitivity analysis of the loss function

As discussed in Section 3.1, the model tends to underestimate the magnitude of extreme events. One intuitive way to address this problem is to increase the weight of the variance term in the loss function.

For this analysis, we define the loss function ℓ_λ for predicted tensor p and target tensor q as

$$\ell_\lambda(p, q) = \text{MSE}(p, q) + KL(\sigma(p), \sigma(q)) + \lambda \cdot |V(p) - V(q)| \tag{7}$$

Table 11 Sensitivity analysis on SY data

Metric	Observed value	$\lambda = 1$	$\lambda = 2$	$\lambda = 3$
Mean (mm)	0.1015	0.1014 (0.10%)	0.1014 (0.10%)	0.1014 (0.10%)
Variance $(\mathrm{mm})^2$	0.4124	0.1831 (55.60%)	0.165 (60.00%)	0.1722 (58.24%)
Number of dry half-hours	46,805	46,117 (1.34%)	46,036 (1.64%)	46,117 (1.47%)
Number of wet half-hours	4507	5135 (13.93%)	5276 (17.06%)	5195 (15.27%)
RMSE	NA	0.5136	0.5129	0.5146
NMSE	NA	0.6397	0.6379	0.6422
R	NA	0.6035	0.6025	0.60
Skill score	NA	0.6759	0.6749	0.6743

Values in brackets indicate relative error from observed value

Noting that $\lambda = 1$ corresponds to the base model. Results are presented in Table 11.

Despite increasing the weight of the variance term, our sensitivity analysis reveals significant variance suppression persists across all tested λ values, with the base model $(\lambda = 1)$ exhibiting a 55.60% underestimation of observed variance. Surprisingly, increasing λ to 2 worsens variance representation (60.00% error) despite the stronger penalty, while further increasing to $\lambda = 3$ slightly improves it (58.24% error), indicating a non-monotonic relationship. This pattern mirrors the wet half-hour classification errors, suggesting direct influence of the variance penalty term on precipitation event detection. Despite these substantial differences in variance representation, overall performance metrics (RMSE, NMSE, R, and skill score) remain notably stable across all λ values, with the base model achieving marginally superior results. This stability amid statistical discrepancies indicates that our model finds different local minima with similar global performance characteristics but distinct distributional properties. We conclude that the base model $(\lambda = 1)$ provides the optimal balance between variance representation and prediction accuracy of the loss functions tested, and that simply increasing the variance penalty coefficient does not effectively address the underestimation of extreme events.

4 Conclusion

This paper extends prior work on machine learning approaches to precipitation disaggregation and downscaling by presenting an LSTM model that can effectively generate half-hourly precipitation statistics from a recorded daily total and hourly meteorological variables. We analyse our model with regard to a number of performance metrics previously established for hourly precipitation and show that its performance is on par with existing methods while both operating on a finer time scale and more effectively enforcing the daily total constraint. This makes our work valuable in ensuring that precipitation can be reliably used for modelling and simulation of built environments.

There are several avenues for future research. Our present approach could be refined by incorporating additional meteorological variables, applying the model to more locations and conducting further performance evaluations, or investigating the performance of the model when trained on an entire climate zone instead of individual stations. Additionally, exploring the performance of the model in low-data environments could be valuable in contexts where extensive historical data is unavailable. These improvements would be beneficial to better understand the limitations of the LSTM approach. Another promising direction is the investigation of foundation model performance in the context of precipitation disaggregation. Foundation models, which are large-scale machine learning models pre-trained on vast amounts of diverse data, have shown promising performance in transfer learning across various domains (Schneider et al. 2024). Future research could explore how these models might capture more nuanced temporal dependencies.

Acknowledgements The authors would like to thank the Bureau of Meteorology for providing the data used in this study.

Author Contributions All authors contributed to the conception and design of the work. Methodology development, software creation, and data analysis were performed by HO and NA. Supervision and data acquisition were performed by TL and HGO. The first draft of the manuscript was written by HO and all authors commented on previous versions of the manuscript. All authors read and approved the final manuscript.

Funding Open Access funding enabled and organized by CAUL and its Member Institutions

Data availability No datasets were generated or analysed during the current study.

Declarations

Conflict of interest The authors declare no competing interests.

References

Agarap AF (2019) Deep learning using rectified linear units (RELU). arXiv:1803.08375

Australian Building Codes Board (2024) Australian climate zone map. https://data.gov.au/data/dataset/australian-climate-zone-map

Australian Bureau of Meteorology (2010) Observation of rainfall. http://www.bom.gov.au/climate/how/observations/rain-measure.shtml

Bhatnagar N, Bogdanov A, Mossel E (2011) The computational complexity of estimating MCMC convergence time. In: Goldberg LA, Jansen K, Ravi R, Rolim JDP (eds) Approximation, randomization, and combinatorial optimization. Algorithms and techniques. Springer, Berlin, pp 424–435. https://doi.org/10.1007/978-3-642-22935-0_36

Bhattacharyya D, Saha U (2022) Deep learning application for disaggregation of rainfall with emphasis on preservation of extreme rainfall characteristics for indian monsoon conditions. Stoch Environ Res Risk Assess 37:1021–1038. https://doi.org/10.1007/s00477-022-02331-x

BoM (2025) Australian capital and major city climate data. http://www.bom.gov.au/climate/australia/cities/

Brigandì G, Aronica GT (2019) Generation of sub-hourly rainfall events through a point stochastic rainfall model. Geosciences 9:226. https://doi.org/10.3390/geosciences9050226

Burian SJ, Durrans SR, Nix SJ, Pitt RE (2001) Training artificial neural networks to perform rainfall disaggregation. J Hydrol Eng 6:43–51. https://doi.org/10.1061/(ASCE)1084-0699(2001)6:1(43)

Clevert DA, Unterthiner T, Hochreiter S (2016) Fast and accurate deep network learning by exponential linear units (ELUS). arXiv:1511.07289

Cowpertwait PSP, O'Connell PE (1997) A regionalised Neyman-Scott model of rainfall with convective and stratiform cells. Hydrol Earth Syst Sci 1:71–80. https://doi.org/10.5194/hess-1-71-1997

Cowpertwait P, Isham V, Onof C (2007) Point process models of rainfall: developments for fine-scale structure. Proc Math Phys Eng Sci 463:2569–2587

de Wilde P (2023) Building performance simulation in the brave new world of artificial intelligence and digital twins: a systematic review. Energy Build 292:113171. https://doi.org/10.1016/j.enbuild.2023.113171

Ferrari D, Mahmoodi M, Kodagoda C, Hameed NA, Lee T, Anderson G (2022) Disaggregation of precipitation data applicable for climate-aware planning in built enviroments. In: Australasian building simulation conference proceedings

Gers FA, Schmidhuber J, Cummins F (2000) Learning to forget: continual prediction with LSTM. Neural Comput 12:2451–2471. https://doi.org/10.1162/089976600300015015

Goodfellow I, Bengio Y, Courville A (2016) Deep learning. adaptive computation and machine learning series. MIT Press, London

Hintz KS, Vedel H, Kaas E (2019) Collecting and processing of barometric data from smartphones for potential use in numerical weather prediction data assimilation. Meteorol Appl 26:733–746. https://doi.org/10.1002/met.1805

Horton P, Schaefli B, Kauzlaric M (2022) Why do we have so many different hydrological models? A review based on the case of Switzerland. WIREs Water 9:e1574. https://doi.org/10.1002/wat2.1574

Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. https://doi.org/10.48550/ARXIV.1502.03167. arXiv:1502.03167

Kingma DP, Ba J (2014) Adam: a method for stochastic optimization. https://doi.org/10.48550/ARXIV.1412.6980. arXiv:1412.6980

Knoesen D, Smithers J (2009) The development and assessment of a daily rainfall disaggregation model for South Africa. Hydrol Sci J 54:217–233. https://doi.org/10.1623/hysj.54.2.217

Koutsoyiannis D (2003) Rainfall disaggregation methods: theory and applications. In: Workshop on statistical and mathematical methods for hydrological analysis. https://doi.org/10.13140/RG.2.1.2840.8564

Koutsoyiannis D, Onof C (2001) Rainfall disaggregation using adjusting procedures on a Poisson cluster model. J Hydrol 246:109–122. https://doi.org/10.1016/S0022-1694(01)00363-8

Kullback S, Leibler RA (1951) On information and sufficiency. Ann Math Stat 22:79–86. https://doi.org/10.1214/aoms/1177729694

Loshchilov I, Hutter F (2017) SGDR: stochastic gradient descent with warm restarts. https://doi.org/10.48550/arXiv.1608.03983. arXiv:1608.03983

Lu L (2020) Dying Relu and initialization: theory and numerical examples. Commun Comput Phys 28:1671–1706. https://doi.org/10.4208/cicp.oa-2020-0165

Lucas J, Tucker G, Grosse R, Norouzi M (2019) Don't Blame the ELBO! A linear VAE perspective on posterior collapse. https://doi.org/10.48550/arXiv.1911.02469. arXiv:1911.02469

Marion W, Urban K (1995) User's manual for TMY2s (typical meteorological years)—derived from the 1961-1990 national solar radiation data base. Technical Report NREL/TP-463-7668, 87130, ON: DE95004064. https://doi.org/10.2172/87130

McKinney W (2010) Data structures for statistical computing in Python. In: van der Walt Stéfan MJ (eds), Proceedings of the 9th Python in science conference, pp 56–61. https://doi.org/10.25080/Majora-92bf1922-00a

Meyes R, Lu M, de Puiseau CW, Meisen T (2019) Ablation studies in artificial neural networks. arXiv:1901.08644

Mishra AK (2019) On the linkage between changes in cloud cover and precipitation extremes over central India. Dyn Atmos Oceans 86:163–171. https://doi.org/10.1016/j.dynatmoce.2019.05.002

Misra S, Sarkar S, Mitra P (2017) Statistical downscaling of precipitation using long short-term memory recurrent neural networks. Theor Appl Climatol 134:1179–1196. https://doi.org/10.1007/s00704-017-2307-2

Müller H, Haberlandt U (2018) Temporal rainfall disaggregation using a multiplicative cascade model for spatial application in urban hydrology. J Hydrol 556:847–864. https://doi.org/10.1016/j.jhydrol.2016.01.031

Olsson J (1998) Evaluation of a scaling cascade model for temporal rain-fall disaggregation. Hydrol Earth Syst Sci 2:19–30. https://doi.org/10.5194/hess-2-19-1998

Onof C, Wang LP (2020) Modelling rainfall with a Bartlett-Lewis process: new developments. Hydrol Earth Syst Sci 24:2791–2815. https://doi.org/10.5194/hess-24-2791-2020

Onof C, Chandler RE, Kakou A, Northrop P, Wheater HS, Isham V (2000) Rainfall modelling using Poisson-cluster processes: a review of developments. Stoch Environ Res Risk Assess 14:0384–0411. https://doi.org/10.1007/s004770000043

Pascanu R, Mikolov T, Bengio Y (2013) On the difficulty of training recurrent neural networks. In: Dasgupta S, McAllester D (eds), Proceedings of the 30th international conference on machine learning, PMLR, Atlanta, Georgia, USA, pp 1310–1318. https://proceedings.mlr.press/v28/pascanu13.html

Paszke A, Gross S, Massa F, Lerer A, Bradbury J, Chanan G, Killeen T, Lin Z, Gimelshein N, Antiga L, Desmaison A, Köpf A, Yang E, DeVito Z, Raison M, Tejani A, Chilamkurthy S, Steiner B, Fang L, Bai J, Chintala S (2019) Pytorch: an imperative style, high-performance deep learning library. https://doi.org/10.48550/ARXIV.1912.01703. arXiv:1912.01703

Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E

(2011) Scikit-learn: machine learning in Python. J Mach Learn Res 12:2825–2830. https://doi.org/10.5555/1953048.2078195

Perkins SE, Pitman AJ, Holbrook NJ, McAneney J (2007) Evaluation of the ar4 climate models' simulated daily maximum temperature, minimum temperature, and precipitation over Australia using probability density functions. J Climate 20:4356–4376. https://doi.org/10.1175/JCLI4253.1

Rohith AN, Gitau MW, Chaubey I, Sudheer KP (2020) A multistate first-order Markov model for modeling time distribution of extreme rainfall events. Stoch Environ Res Risk Assess 35:1205–1221. https://doi.org/10.1007/s00477-020-01939-1

Schneider J, Meske C, Kuss P (2024) Foundation models: a new paradigm for artificial intelligence. Bus Inf Syst Eng 66:221–231. https://doi.org/10.1007/s12599-024-00851-0

Staudemeyer RC, Morris ER (2019) Understanding LSTM—a tutorial into long short-term memory recurrent neural networks. arXiv:1909.09586

Tan H, Lee T, Ferrari D (2023) Enhancing Australia's weather and climate data for benchmarking simulations. In: Proceedings of the asia-pacific solar research conference, Australian Photovoltaic Institute

US Department of Energy (2024) Energyplus. https://energyplus.net/

Waqas M, Humphries UW (2024) A critical review of RNN and LSTM variants in hydrological time series predictions. MethodsX 13:102946. https://doi.org/10.1016/j.mex.2024.102946

WMO (2023) Guide to climatological practices, 3rd edn. World Meteorological Organisation, Geneva

Yusop Z, Nasir H, Yusof F (2013) Disaggregation of daily rainfall data using Bartlett Lewis rectangular pulse model: a case study in central peninsular malaysia. Environ Earth Sci 71:3627–3640. https://doi.org/10.1007/s12665-013-2755-7