Investigating further use of long short-term memory (LSTM) neural networks in flood forecasting

Mohammed Ameen Dasu
MSc Data Science, BSc (Hons) Mathematics

A dissertation submitted for the degree of
Master of Science in Data Science

Supervisors: *Mr. Plamen Angelov (Lancaster University), Amy Winder (JBA), Sarah Warren (JBA), Paul Wass (JBA)*

School of Computing and Communications
Lancaster University

September, 2025

Declaration

I declare that the work presented in this dissertation is, to the best of my knowledge and belief, original and my own work. The material has not been submitted, either in whole or in part, for a degree at this, or any other university. Estimated word count is: **18923**

Name: **Mohammed Ameen Dasu**
Date: **September, 2025**

Investigating further use of long short-term memory (LSTM) neural networks in flood forecasting

Mohammed Ameen Dasu, MSc Data Science, BSc (Hons) Mathematics.

School of Computing and Communications, Lancaster University

A dissertation submitted for the degree of *Master of Science* in Data Science.
September, 2025

Abstract

This study evaluates the performance of the Handoff Forecast LSTM, implemented via the open-source Neural Hydrology package, for rainfall–runoff prediction in the United Kingdom. Two experiments were conducted using subsets of 256 and 300 basins, with the aim of assessing model skill under varying levels of spatial scale and basin diversity. The experiments were trained for 25 epochs, and model performance was evaluated using a suite of statistical and hydrological metrics, including the Nash–Sutcliffe Efficiency (NSE), Kling–Gupta Efficiency (KGE), root mean squared error (RMSE), and Pearson correlation coefficient.

Results demonstrated consistent convergence of training and validation losses across epochs, with limited evidence of severe overfitting. Both basin subsets achieved moderate predictive accuracy, with NSE and KGE values typically in the range of 0.4–0.5, while Pearson correlation values were higher ($\sim$0.7). The model was able to capture general flow variability but showed persistent challenges in reproducing baseflow conditions and extreme peak events. Comparisons between the two experiments indicated that the 300-basin setup improved correlation but introduced greater variability across runs, reflecting the increased difficulty of learning from a more diverse basin set.

The study highlights both the potential and current limitations of LSTM-based rainfall–runoff models in a UK context. While the Handoff Forecast LSTM demonstrated scalability and robustness across multiple basins, predictive skill remains below the threshold required for operational use. Future work should explore enhanced model architectures, improved representation of baseflow processes, and physics-informed neural networks to achieve more reliable hydrological forecasts.

Acknowledgements

I would like to express my sincere gratitude to my supervisor, Plamen Angelov, for their invaluable guidance, support, and encouragement throughout this project. Their expertise and constructive feedback have been essential in shaping both the direction and quality of this dissertation.

I am also grateful to the JBA Consulting for providing the computational resources and remote access environment that made it possible to conduct large-scale experiments. Special thanks go to my colleagues and peers in the MSc Data Science programme for their advice, discussions, and encouragement. Furthermore, I would like to express my sincere gratitude towards Amy Winder, Sarah Warren and Paul Wass for providing me the data, giving me the correct understanding of the Neural Hydrology package and finally the understanding of hydrological processes that happen within river basins and catchment areas.

Contents

List of Figures

Chapter 1

Introduction

Hydrological forecasting is a critical component of water resource management, flood risk mitigation, and climate resilience planning. Accurate rainfall–runoff prediction supports decision-making across a wide range of domains, from day-to-day reservoir operations to emergency responses during extreme events. However, modelling the rainfall–runoff process remains one of the most challenging problems in hydrology, owing to its non-linear nature, dependence on diverse catchment characteristics, and variability across spatial and temporal scales.

Traditionally, conceptual and physics-based hydrological models have been the dominant tools for streamflow predictions. With the increasing availability of high-resolution hydro-meteorological datasets, there is growing interest in applying machine learning (ML) methods that can learn directly from data without prespecified physical equations.

In recent years, recurrent neural networks (RNNs), and particularly long short-term memory (LSTM) networks, have shown strong promise for rainfall–runoff prediction. Studies using large-sample datasets such as CAMELS-US have demonstrated that LSTMs can outperform traditional models by capturing temporal dependencies and spatial variability in hydrological data. Building on this momentum, the **Neural Hydrology** package provides an open-source framework for training and evaluating deep learning models across large basin sets.

This project applies the Handoff Forecast LSTM, a variant of the LSTM implemented in Neural Hydrology, to UK catchments. The study evaluates model performance across two experiments using 256 and 300 basins, with the aim of assessing how well the model generalises under increasing spatial scale and basin diversity. Results are analysed through a

range of statistical and hydrological metrics, including the Nash–Sutcliffe Efficiency (NSE), Kling–Gupta Efficiency (KGE), root mean squared error (RMSE), and Pearson correlation coefficient. In addition, basin-level case studies are used to explore how the model reproduces baseflow and peak flows at different stages of training.

The key objectives of this project are therefore:

- To evaluate the training and validation performance of the Handoff Forecast LSTM across multiple UK basins.
- To compare results between 256- and 300-basin experiments, assessing the impact of basin diversity on model skill.
- To analyse model behaviour in reproducing both baseflow and extreme events.
- To identify computational challenges and limitations associated with large-scale training.
- To discuss future directions for improving rainfall–runoff forecasting through machine learning approaches.

This dissertation contributes to understanding the applicability of LSTM-based models in a UK hydrological context and explores pathways toward more reliable, data-driven forecasting frameworks.

Chapter 2

Background & Related work

The study titled "Application of Long Short-Term Memory (LSTM) Neural Network for Flood Forecasting" by Le *et al.* [**LeXuan-Hien2019AoLS**] addresses the need for accurate flood forecasts to mitigate the risks and manage water resources. This study acknowledges the complex, non-linear nature of river flow predictions, which is influenced by many factors.

The authors highlight that traditional models have trouble handling sudden fluctuations, which can be caused by discharges from large upstream dams and reservoirs. This is important as upstream behaviour can affect downstream behaviour significantly. Furthermore, traditional models also need extremely large datasets, which can be difficult to obtain, with many required data points frequently missing, unavailable, or difficult to collect. Data-driven methods such as LSTMs can help overcome these challenges by learning directly from the available data, even when the data is incomplete [**kong2025unlocking**]. LSTMs are particularly suited for flood forecasting in regions where data availability is scarce, as they can learn how changes in one part of a system affect another part.

The researchers focused on the Hoa Binh station on VietnamâĂŹs Da River basin, using flow rate and rainfall data. They predicted the flow rate one, two, and three days in advance under two scenarios:

1. Combining rainfall and flow data
2. Using only flow data

LSTMs performed effectively, with NashâĂŞSutcliffe Efficiency (NSE) values reaching 99%

for one-day forecasts, 95% for two-day forecasts, and 87% for three-day forecasts. The research also showed that flow rate data played a more significant role than rainfall in forecast accuracy, particularly for flood peak predictions. This suggested that rainfall data could introduce "noise" that reduced correlation coefficients with the target flow.

Another relevant study was conducted in Central Europe on the Tisza River, where the authors used LSTM neural networks to predict water levels seven days ahead at the Szeged gauging station [**Vizi2023WaterEurope**]. The LSTM was compared with the Discrete Linear Cascade Model (DLCM) to evaluate its applicability across different hydrological situations and prediction horizons. This study employed a multilayer LSTM with an encoderâĂŞdecoder architecture, considered one of the most effective approaches for multi-horizon forecasting [**wang2016learning**]. The encoder LSTM processed historical input data to extract complex patterns, passing this information to the decoder through hidden states. The decoder, also a stacked LSTM, generated predictions for the forecast horizon, initially using observed outputs and later using its own predictions.

The model used daily water levels from 12 gauging systems (January 1951âĂŞDecember 2020), with 76% of the data allocated for training (1951 - 2004) and 23% for validation (2005 - 2020). Each station dataset was normalised separately using parameters from the training set. The model mapped a multivariate input sequence to a univariate target sequence, and performance was evaluated using Mean Absolute Error (MAE), Root Mean Squared Error (RMSE), and R^2.

Results showed that the LSTM outperformed the DLCM across most hydrological situations. In particular, the LSTM achieved a higher proportion of predictions within predefined precision intervals (74.9 - 79.9%) than the DLCM (64.1 - 73.7%) across a six-day forecast horizon. Performance gains were most significant during low stages ($\leq$ 239 cm), with the LSTM achieving 78.9 - 84.2% accuracy compared to the DLCMâĂŹs 70.4 - 80.2%. While both models struggled with medium stages (240 - 649 cm), the LSTM still performed better (53.8 - 62.6% vs. 39.1 - 48.2%). No high-stage events occurred during the main evaluation period (2014 - 2019), but extended validation (2005 - 2019) showed that the LSTM maintained strong performance during historical floods, predicting 60.1 - 73.7% of flood stages within the required range.

The DLCMâĂŹs weaker performance was attributed to its assumption of linear hydrological relationships, which fails to capture the non-linear and dynamic nature of real-world hydrology, particularly under climate variability and downstream disturbances. The LSTM also underestimated less frequently (11.0 - 14.3%) than the DLCM (15.4 - 18.6%).

Our project differs from these studies in several key ways. We employ the `handoff_forecast_lstm`

model from the Neural Hydrology framework [**kratzert2019neuralhydrology**] to forecast flow rate for over 600 gauges across the UK. Whereas existing studies often focus on one or a few catchments, our work spans a broad geographical scale, aiming to generalise across diverse catchments rather than optimising for a small set of locations. Furthermore, we are building a pipeline capable of real-time, batch forecasting across gauges using cleaned and pre-processed input data [**bayliss2021catchment**]. While prior work often emphasises experimental comparisons of model performance, our goal is to create a scalable forecasting solution for potential operational use, with model comparisons serving to contextualise and validate performance.

2.1 Flood Forecasting Techniques

Forecasting techniques have been widely applied to observe and predict natural hazards, particularly water-related disasters. Among all observed natural hazards, water-related disasters are the most frequent and pose major threats to human life [**Jain2018ReviewWater**]. Between 1900 and 2006, floods accounted for approximately 30% of all recorded natural disasters, causing more than 19% of total fatalities and affecting over 48% of the people impacted by natural hazards [**Jain2018ReviewWater**]. Moreover, water-related disasters are responsible for over 60% of the total economic damages from natural disasters, with floods alone contributing a lot of these losses [**Jain2018ReviewWater**]. These impacts are projected to escalate in the future due to climate change, land use changes, deforestation, rising sea levels, and population growth in flood-prone areas. As a result, the global population vulnerable to flood disasters could reach two billion by 2050. So, the development of optimal flood forecasting and viable flood risk management systems have been extremely important so countries can be prepared for critical disasters. These forecasting techniques have been implemented within countries and within communities such that individual communities can have their own defence mechanisms and forecasting methods.

The primary purpose of flood forecasting models is to provide early warnings to the public about impending flood events, enabling communities to prepare in advance and thereby mitigating potential economic losses compared to scenarios without prior warning. Such techniques typically involve the collection of meteorological and hydrological data, followed by the forecasting of future rainfall. These forecasts allow hydrologists to predict subsequent changes in river flow and water levels.

2.1.1 Deterministic Models

In the context of flood forecasting, deterministic models are catchment-based approaches that solve sets of equations representing the physical processes within a watershed, producing a single output for any given set of parameters.

These models rely on conceptual physics to represent processes such as precipitation, infiltration, soil moisture dynamics, evapotranspiration, runoff generation, and streamflow routing. They typically consist of two main components: flood-generating mechanisms, which include precipitation formation, snowmelt modelling, and runoff simulation with precipitation as a crucial input and runoff or flow depth as outputs; and flood routing, which simulates the downstream movement of floodwaters through hydrologic or hydraulic methods, ranging from traditional 1D flow simulations to increasingly common 2D and even 3D models. RainfallâĂŞRunoff (RR) models, central to deterministic systems, usually offer longer lead times than routing-only systems, especially when larger basins are divided into sub-basins, while in smaller basins, Quantitative Precipitation Forecasts (QPFs) are often integrated to extend lead time. Effective RR models must also account for catchment dynamism, reflecting land use, land cover, and infrastructure changes that alter hydrological responses. Model selection depends on forecasting objectives, data availability, institutional capacity, and catchment characteristics [**Hunter2007Flood2D**, **Jain2018ReviewWater**].

However, despite their traditional dominance, deterministic models with constant parameters often fail to capture the complexity and dynamic variability of basins, leading to discrepancies between simulated and observed hydrographs. To address these limitations, techniques such as forecast updating, data assimilation, and ensemble forecastingâĂŤproviding multiple possible future states to quantify uncertaintyâĂŤhave been introduced, marking significant advancements beyond purely deterministic approaches [**Hunter2007Flood2D**].

2.1.1.1 LISFLOOD-2D

LISFLOOD-2D is a GIS-based, spatially distributed hydrological rainfallâĂŞrunoff model, also described as a two-dimensional (2D) hydrodynamic model for rainfallâĂŞrunoff routing [**jrc_lisflood_model**, **Arnal2020LISFLOOD**].

A GIS is a computer-based system that is designed to capture, store, manage, analyse, and visualise spatial or geographic data. It integrates different kinds of information (such as maps, satellite imagery, terrain models, or hydrological data) and links them to specific locations on the EarthâĂŹs surface. GIS is extremely valuable in this context because it allows researchers

to make decisions based on river networks, overlays of different data such as rainfall or soil type, and to simulate the spread of the rainfall in a 2D space [**Hunter2007Flood2D**].

LISFLOOD-2D was developed at the Joint Research Centre (JRC) of the European Commission for operational flood forecasting and forms the basis of the European Flood Awareness System (EFAS). It provides both a pan-European overview of flood probabilities up to 15 days in advance and detailed forecasts at specific gauging stations. By combining large-scale meteorological forecasts with high-resolution hydrodynamic modelling, LISFLOOD-2D enables consistent flood risk assessment across countries, improves trans-boundary flood management, and supports long lead-time warnings that are critical for mitigating human and economic losses [**ecmwf_efas_upgrade_2020**, **Arnal2020LISFLOOD**].

Trans-boundary flood management is extremely important in all types of flood forecasting techniques because rivers often flow across borders or regions; without coordination, upstream activities can significantly affect downstream areas. By promoting cooperation and information sharing between regions, trans-boundary management ensures that forecasts are consistent, warnings are timely, and mitigation measures are more effective across the entire river system [**Arnal2020LISFLOOD**].

2.1.1.2 Limitations

LISFLOOD-2D, while very powerful, requires detailed input datasets such as high-resolution topography, soil type, and rainfall. Since this model is deterministic there is no learning happening; accuracy can be reduced in data-scarce regions [**jrc_lisflood_model**, **Arnal2020LISFLOOD**].

Despite being process-based and physically interpretable, LISFLOOD does not adapt as quickly as data-driven models like LSTMs, which can be retrained or fine-tuned as new observations arrive [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019neuralhydrology**].

2.1.1.3 MIKE 11

This deterministic model was developed by the Danish Hydraulic Institute (DHI) and is classified as a one-dimensional (1D) hydrodynamic model. Known as MIKE 11, it is primarily used for flood routing and simulating the movement of floodwaters through river channels. Unlike simpler hydrologic routing methods, hydraulic models such as MIKE 11 compute hydrographs simultaneously at several cross-sections along the watercourse, though they

require detailed inputs such as channel bathymetry and roughness data [**DHI2009MIKE11**, **DHI_MIKEFloodWatch_1998**].

As a semi-lumped rainfallâĂŞrunoff (RR) model, MIKE 11 has been widely applied for flood inundation mapping, operational forecasting, and integration with other hydrological and meteorological systems. For instance, it has been used in Bangladesh with satellite-based radar altimetry to extend lead times, in Nepal with real-time telemetry and Weather Research and Forecasting (WRF) model inputs, and in Malaysia when coupled with GIS tools for local flood warning systems [**WMO2018Bangladesh-report**, **DHI_MIKEFloodWatch_1998**].

Geographically, its applications extend to India, Malaysia, Bangladesh, Nepal, and China, where it has been incorporated into national flood forecasting systems and combined with hydrological models such as Xinanjiang, URBS, and API for major river basins including the Yangtze, Han, and Dongting Lake systems [**Jain2018ReviewWater**, **DHI_MIKEFloodWatch_1998**]

2.1.1.4 Limitations

Despite its widespread use and robustness, MIKE 11 has several limitations. The model requires detailed input data such as channel geometry, bathymetry, and roughness coefficients, which are not always available or reliable, particularly in data-scarce regions, and it can be computationally demanding for long river networks or high resolution [**DHI2009MIKE11**]. Furthermore, as a primarily 1D routing model, it does not capture floodplain dynamics as effectively as 2D/3D models. Careful calibration is necessary and time-consuming when observations are limited. Unlike data-driven models such as LSTMs, MIKE 11 cannot readily adapt to evolving catchment conditions without explicit recalibration [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019neuralhydrology**].

2.1.2 Data-Driven Models

2.1.2.1 Artificial Neural Networks and Recurrent Neural Networks

Artificial Neural Networks (ANN's) are classified as data-driven, non-parametric, and non-linear time-series models, often described as âĂIJblack-boxâĂİ approaches. They have been shown to be effective in modelling rainfallâĂŞrunoff processes and flood forecasting, as they can represent complex non-linear relationships in hydrological data, adapt to new inputs, learn from data, generalise system behaviour, and handle noisy signals [**ASCE2000ANN**,

Maier2010Review, **Dawson2001ANNhydrology**, **Jain2018ReviewWater**]. A notable application is their use in improving real-time forecasts from deterministic lumped rainfallâĂŞrunoff models; for example, coupling ANN's with ARIMA models has demonstrated substantial improvements in discharge forecasting and updating [**Jain2018ReviewWater**]. Despite these strengths, ANN's have not yet been widely deployed in operational flood warning systems, with only a limited number of prototypes in practice [**Jain2018ReviewWater**].

Challenges include long training times, the risk of overfitting to specific datasets, and difficulties in parameter selection [**Maier2010Review**]. Moreover, ANN-based forecasts are often reliable only at short lead times (such as one step ahead), which raises concerns about their suitability for flood management [**Dawson2001ANNhydrology**, **Jain2018ReviewWater**]. Like other data-driven models, they also cannot directly account for changing physical basin characteristics, such as land-use change, and their parameters are entirely dependent on the calibration datasetâĂŹs range [**ASCE2000ANN**, **Maier2010Review**].

This lack of direct physical interpretability has contributed to the continued dominance of process-based hydrological models in flood forecasting. Consequently, it has been suggested that ANN's are best used in combination with other modelling approaches, particularly in pilot applications, to evaluate and strengthen their performance [**Jain2018ReviewWater**, **Hunter2007Flood2D**].

Recurrent Neural Networks (RNN's) are a class of artificial neural networks designed specifically for handling sequential and time-series data. Unlike feedforward ANNs, where information flows in only one direction, RNN's incorporate feedback loops that allow information from previous time steps to influence current outputs. This structure enables them to capture temporal dependencies in data, making them particularly well-suited for applications in hydrology such as rainfallâĂŞrunoff modelling, streamflow forecasting, and flood prediction [**Lipton2015RNNs**].

The strength of RNN's lies in their ability to retain information about past inputs, which they achieve through hidden states. These hidden states effectively allow the network to memorise information, making RNN's powerful tools for modelling the sequential nature of hydrological processes, where current river flow conditions are heavily influenced by preceding rainfall and upstream discharge events [**Lipton2015RNNs**].

Again, traditional RNN's face several limitations. A key challenge is the vanishing gradient problem, which occurs during back-propagation. When gradients shrink or grow exponentially across many time steps, the network struggles to learn long-term dependencies, limiting its effectiveness for long lead-time flood forecasting [**Lipton2015RNNs**]. As a

result, basic RNN's often perform best at short-term forecasting horizons, but their accuracy degrades for longer prediction intervals.

The neural network employed in this project is a type of RNN, specifically the LSTM model. Unlike traditional neural networks, which cannot effectively handle sequential or time dependent data as they treat each input independently and lack a mechanism for retaining information across time steps, LSTMs are designed to overcome this limitation. They incorporate gating mechanisms, namely the input, forget, and output gates, that regulate the flow of information through the network [**Hochreiter1997LSTM**, **Colah2015LSTM**].

These gates allow the model to retain relevant long term dependencies, discard irrelevant information, and update its memory dynamically. As a result, LSTMs successfully address the vanishing gradient problem that hampers traditional RNN's and are considerably more effective at modelling long sequences, making them particularly well suited for applications such as rainfall runoff modelling, streamflow forecasting, and flood prediction [**Hochreiter1997LSTM**, **Kratzert2018RainfallRunoffLSTM**, **kratzert2019neuralhydrology**].

2.1.2.2 Limitations

Despite their strengths, ANNs also suffer from several technical limitations. A well-known challenge is the vanishing gradient problem, which occurs during back-propagation when gradients become very small as they are propagated through many layers. This slows down or prevents effective training of deep networks, making it difficult to capture long-term dependencies in hydrological time series. While techniques such as Long Short-Term Memory (LSTM) networks were developed to counter act this issue, ANNs like the single layer ANN or a multi-layer ANN are extremely susceptible to this problem [**Hochreiter1997LSTM**, **Lipton2015RNNs**].

In addition, ANNs are prone to overfitting, especially when trained on limited hydrological datasets, and they can require long training times and large amounts of data to achieve reliable performance [**Maier2010Review**, **Dawson2001ANNhydrology**]. Their forecasts are often limited to shorter lead times, and because they lack direct physical interpretability, they cannot inherently adapt to changes in catchment characteristics such as land use, infrastructure development, or climate variability [**ASCE2000ANN**, **Maier2010Review**]. These limitations mean that while ANNs are powerful pattern-recognition tools, their application in flood forecasting often benefits from being combined with process-based models or more advanced deep learning architectures [**Jain2018ReviewWater**, **Hunter2007Flood2D**].

Again, traditional RNN's face several limitations. A key challenge is the vanishing

gradient problem, which occurs during back-propagation. When gradients shrink or grow exponentially across many time steps, the network struggles to learn long-term dependencies, limiting its effectiveness for long lead-time flood forecasting. As a result, basic RNN's often perform best at short-term forecasting horizons, but their accuracy degrades for longer prediction intervals [**Lipton2015RNNs**].

2.1.2.3 LSTMs

In this project, an LSTM has been used to predict flow across more than 600 basins in the UK. As previously noted, LSTMs are capable of forecasting future values by learning from historical data, which enables them to capture both short-term fluctuations and long-term dependencies in river flow. Beyond this, LSTMs can also learn the contextual behaviour of a basin and transfer this knowledge to similar basins where data may be missing, thereby generating reliable predictions [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019towards**, **Addor2017CAMELS**]. This approach is particularly effective within a single country, such as England, where geographical landscapes and climatic conditions do not vary drastically between basins, allowing shared patterns to be used for improved accuracy [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019towards**].

Compared to deterministic hydrological models that involve calibration and computation, LSTMs offer a more efficient alternative. This is because they learn directly from historical data without requiring detailed physical parameters, allowing them to adapt quickly to new inputs and generate forecasts with minimal manual intervention [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019neuralhydrology**]. Once trained, LSTMs can produce predictions at a fraction of the computational cost of process-based models, making them especially suitable for large-scale or real-time applications such as national flood forecasting [**Kratzert2018RainfallRunoffLSTM**].

Furthermore, the data provided has many gaps, which is the nature of working with such a large national-scale dataset. With more than 600 basins, differences in monitoring infrastructure, equipment malfunctions, and inconsistent reporting practices often result in missing or incomplete records. This presents a challenge for traditional process-based models that require continuous, high-quality inputs, but LSTMs are more robust in this context, as they can still identify meaningful patterns and produce reliable forecasts even when portions of the data are unavailable [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019towards**].

However, the ability of LSTMs to learn shared hydrological patterns across similar basins allows them to extend predictive skill to areas with sparse or incomplete data,

thereby increasing coverage and resilience of the forecasting system [**kratzert2019towards**, **Addor2017CAMELS**]. This means that reliable forecasts may still be possible in basins with limited observations, reducing the dependence on dense monitoring networks and enabling a more comprehensive national-scale forecasting framework. In practice, this approach improves equity of service by ensuring that data-poor regions, which are often the most vulnerable to flood risks, still benefit from accurate and timely predictions [**kratzert2019towards**].

Evaluating models is extremely important and will allow us to evaluate the performance of the forecasting model. For Hydrological applications, accuracy is typically assessed by using statistical performance metrics such as Nash-Sutcliffe Efficiency (NSE), Root Mean Squared Error (RMSE), Mean Absolute Error (MAE), and the coefficient of the R Squared Value [**Nash1970NSE**, **hodson2022rmsemae**, **Moriasi2007**, **legates1999**]. These measurements provide insights to how closely the predicted flows match the observed data across different the 3 day forecast time. In the context of this project, accuracy measurements are important given the national scale of the project. Also the model will improve accuracy by learning both short-term fluctuations and long-term dependencies, as well as transferring knowledge across similar basins to compensate for missing or noisy data [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019towards**]. This ability not only enhances the reliability of forecasts in well-monitored catchments but also ensures that predictions remain robust in data-scarce regions, making the system more resilient and operationally relevant.

This project also provides a foundation for future hybrid approaches, where data-driven LSTMs could be combined with process-based models to capture both physical realism and data-driven adaptability, further strengthening flood forecasting capabilities [**Hunter2007Flood2D**]. Hybrid systems would allow LSTMs to correct or update process-based model outputs in real time, addressing their calibration and computational limitations, while still preserving the physical interpretability of hydrological processes. Such an approach could be particularly valuable under changing climate and land use conditions, where physical catchment properties evolve over time, but data-driven components can quickly adapt to new patterns. Moreover, integrating LSTMs with established hydrodynamic models such as LISFLOOD or MIKE 11 could enhance both forecast accuracy and forecast times, offering operational agencies more reliable tools for decision-making [**Arnal2020LISFLOOD**, **DHI2009MIKE11**]. This project provides a practical proof of concept for a national-level system that could evolve into a hybrid forecasting framework with broad operational relevance.

2.1.2.4 Limitations

This will be explored further in the results section; however, running the LSTM can take an extremely long time due to the amount of data. Furthermore, cleaning the data so it is suitable for the model is extremely difficult due to the nature of the data and the way it is collected in hydrology. This means that a significant portion of the work involves preprocessing and quality control, where missing values, inconsistent records, and measurement errors must be addressed before training can even begin [**Maier2010Review**].

These challenges highlight that, while LSTMs provide strong predictive capabilities, their effectiveness is closely tied to the availability of high-quality input data and the efficiency of the preprocessing pipeline. In practice, this makes data management just as critical as model development, since inaccuracies or delays in preparing the inputs can reduce forecast accuracy and limit the operational usefulness of the system [**Moriasi2007**].

Although LSTMs are powerful tools for hydrological forecasting, this project also faces several limitations. Firstly, training these models across more than 600 basins is computationally expensive and can take an extremely long time due to the large volume of input data. Secondly, preparing the data presents significant challenges: hydrological observations often contain missing values, inconsistent records, and measurement errors, making data cleaning a time-consuming and complex task [**Maier2010Review**]. These difficulties are amplified at a national scale, where variations in monitoring practices and equipment across basins further increase data heterogeneity.

Moreover, the effectiveness of LSTMs is highly dependent on the quality of the input data, meaning that poor preprocessing can reduce forecast accuracy and reliability [**Moriasi2007**, **legates1999**]. Finally, while the models are data-driven and adaptive, they do not incorporate explicit physical process representations, which can limit interpretability and may reduce their ability to generalise under changing environmental conditions such as climate change or land-use alterations [**Hunter2007Flood2D**].

2.2 Evaluation Metrics

2.2.1 NSE (Nash-Sutcliffe Efficiency)

The Nash–Sutcliffe Efficiency (NSE) is one of the most widely applied statistics in hydrology for evaluating the performance of simulation models, particularly in rainfall–runoff and streamflow forecasting [**Nash1970NSE**, **melsen2023nse**]. NSE is formally defined as a sample statistic; however, until the work of Lamontagne, Barber, and Vogel [**Lamontagne2020**], it lacked a rigorous theoretical foundation. Their study introduced the theoretical statistic E, upon which NSE is based, and clarified that E is a standardized form of the Mean Squared Error (MSE) defined using probability theory. Unlike NSE, which is a sample estimator dependent on observed data, E is independent of the properties of any specific dataset.

$$NSE = 1 - \frac{\frac{1}{n}\sum_{i=1}^{n}(s_i - o_i)^2}{\frac{1}{n-1}\sum_{i=1}^{n}(o_i - \bar{o})^2}$$

Within the NSE formula, s_i represents the simulated flow, o_i represents the observed flow, n is the sample size, and $\bar{o}$ is the sample mean of observations. This clarification helped resolve a common misunderstanding, as many criticisms of NSE were originally directed at the sample statistic rather than the underlying theoretical measure [**melsen2023nse**]. In practice, much of the criticism arose because of the variability of the sample estimator, rather than flaws in the theoretical statistic E [**Lamontagne2020**].

NSE became the "go-to" metric for hydrology because of its intuitive interpretation, ease of calculation, and ability to summarise model performance with a single value [**melsen2023nse**]. A value of NSE = 1 indicates a perfect match between observed and simulated flows, NSE = 0 suggests the model performs no better than using the mean of observations as a predictor, and negative values indicate the model is less accurate than this simple benchmark [**Nash1970NSE**]. This straightforward interpretation, along with strong endorsement from organisations such as ASCE [**ASCE1993**] and Moriasi et al. [**Moriasi2007**], cemented NSE as the standard benchmark for evaluating and calibrating hydrological models.

Since this project uses LSTMs for forecasting flow, NSE is an important metric for analysing and evaluating the performance of the models. However, given the large number of basins (600+) and the presence of missing or skewed data, NSE alone may be misleading due to its

sensitivity to outliers, especially in daily data [**melsen2023nse**]. To overcome this, NSE will be used alongside other evaluation metrics such as RMSE, MAE, and additional measures to provide a more robust evaluation [**hodson2022rmsemae**, **legates1999**].

2.2.2 KGE (Kling-Gupta efficiency)

The Kling-Gupta Efficiency (KGE) score is a metric used in hydrology for goodness of fit between simulations and observations, commonly applied as an alternative to NSE [**Gupta2009**, **Knoben2019**]. The reason this metric came into fruition was to overcome limitations of NSE, notably its sensitivity to outliers and variance structure. KGE introduces correlation, variability error, and bias terms to diagnose performance components [**Gupta2009**].

$$KGE = 1 - \sqrt{(r-1)^2 + (\alpha-1)^2 + (\beta-1)^2}$$

KGE uses r, α, β where r represents the linear correlation between simulation and observations, and α represents the ratio of simulated to observed standard deviations:

$$\alpha = \frac{\sigma_{sim}}{\sigma_{obs}}$$

This measures the modelâĂŹs ability to reproduce the variability of observed flows.

β represents the ratio of simulation mean to the observation mean (bias):

$$\beta = \frac{\mu_{sim}}{\mu_{obs}}$$

When $\beta = 1$, there is no mean bias.

KGE has become popular for calibration and evaluation because it provides a more balanced assessment versus NSE by separating correlation, variability, and bias [**Gupta2009**, **Knoben2019**]. Unlike NSE (which has a natural baseline of 0 corresponding to the mean-flow predictor), KGE has no universal baseline; for example, when the simulated mean equals the observed mean, a particular reference gives $KGE = 1 - \sqrt{2} \approx 0.41$ [**Knoben2019**].

Despite its advantages, KGE also has limitations. Like NSE, it relies on product-moment estimators that can be biased and highly variable for skewed daily streamflow data [**Lamontagne2020**, **legates1999**]. Moreover, KGE lacks a universally accepted

âĂIJgood/poorâĂİ threshold, so practice-specific benchmarks or weighted variants may be needed [**Knoben2019**].

In this project, KGE will be used alongside RMSE, MAE and other metrics to provide a comprehensive evaluation. Whilst NSE is the most established benchmark for hydrological projects, KGE helps diagnose whether errors arise from bias, variability, or correlationâĂŤespecially important at national scale with heterogeneous data quality and catchment characteristics.

2.2.3 RMSE (Root Mean Square Error)

RMSE is a widely used statistical metric for evaluating model performance, particularly when it is important to give greater weight to larger errors [**hodson2022rmsemae**]. It is defined as the square root of the mean of the squared differences between the simulated and observed values:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(s_i - o_i)^2}$$

Here, s_i represents the simulated values, o_i represents the observed values, and n is the number of observations. By squaring the differences before averaging, RMSE penalises larger errors more heavily than smaller ones, making it especially sensitive to outliers and extreme deviations [**hodson2022rmsemae**, **legates1999**].

In this project, RMSE is used alongside other performance metrics to assess the accuracy of LSTM forecasts across more than 600 UK basins. RMSE is particularly valuable in hydrological applications because it emphasises large forecast errors, which can be critical when predicting high-flow or flood events. By incorporating RMSE into the evaluation framework, the project ensures that both typical prediction accuracy and the handling of extreme events are reflected in the model assessment, complementing efficiency-based measures such as NSE and KGE [**Nash1970NSE**, **Gupta2009**].

2.2.4 Pearson-r (Correlation Coefficient)

The Pearson correlation coefficient (r) is widely used in hydrology to evaluate the linear association between simulated and observed streamflow values, indicating how well the model reproduces temporal dynamics regardless of magnitude bias [**legates1999**].

$$r = \frac{\sum_{i=1}^{n}(Q_i^{\text{obs}} - \overline{Q^{\text{obs}}})(Q_i^{\text{sim}} - \overline{Q^{\text{sim}}})}{\sqrt{\sum_{i=1}^{n}(Q_i^{\text{obs}} - \overline{Q^{\text{obs}}})^2}\sqrt{\sum_{i=1}^{n}(Q_i^{\text{sim}} - \overline{Q^{\text{sim}}})^2}}$$

where Q_i^{obs} are observed streamflow values, Q_i^{sim} are simulated values, and $\overline{Q^{\text{obs}}}$ and $\overline{Q^{\text{sim}}}$ are their respective means.

The value of r ranges from -1 to $+1$:

- $r = +1$ indicates a perfect positive linear relationship.
- $r = -1$ indicates a perfect negative linear relationship.
- $r = 0$ indicates no linear correlation.

In this project, Pearson-r evaluates how well the LSTM model captures timing and relative dynamics of streamflow across UK basins. While it does not directly measure magnitude errors (unlike RMSE), a high correlation reflects that the model reproduces observed flow patterns, making it a valuable complement to error- and efficiency-based measures such as RMSE, NSE, and KGE [**legates1999**, **hodson2022rmsemae**, **Gupta2009**].

2.3 Neural Hydrology

Neural Hydrology is an open-source research framework designed for applying deep learning methods to hydrological forecasting tasks. Developed at the Helmholtz Centre for Environmental Research (UFZ), it provides a flexible and reproducible environment for training, testing, and comparing neural network models on large-scale hydrological datasets [**kratzert2019neuralhydrology**].

The framework supports a variety of architectures commonly used in time series forecasting, including Long Short-Term Memory (LSTM) networks [**Hochreiter1997LSTM**], Temporal Convolutional Networks (TCNs), and transformer-based models. It is particularly suited to applications where input data consists of meteorological forcings, catchment attributes, and discharge observations, making it an ideal tool for large-sample hydrology studies [**Kratzert2018RainfallRunoffLSTM**].

One of the strengths of Neural Hydrology is its emphasis on reproducibility and scalability. Model configurations are specified using simple YAML files, allowing experiments to be documented and repeated consistently. In addition, the framework facilitates training across hundreds of catchments simultaneously, enabling the development of regional models that can generalise beyond individual basins [**kratzert2019towards**].

2.3.1 Handoff Forecast LSTM

The `NeuralHydrology.modelzoo.handoff_forecast_lstm.HandoffForecastLSTM` model is a specialised forecasting architecture that extends the standard LSTM by introducing a state-handoff mechanism between two sequence models. The design separates the modelling process into two stages: a hindcast LSTM and a forecast LSTM.

The hindcast LSTM is run from the historical period up to the forecast issue time, processing all available past inputs. At the end of this phase, its hidden state and cell state are passed into a nonlinear handoff network. This handoff network, implemented as a configurable fully connected (FC) layer, transforms the hindcast states and uses them to initialise the forecast LSTM [**Olah2015UnderstandingBlog**]. The forecast LSTM then rolls out predictions over the forecast horizon. This separation allows the two LSTMs to learn different dynamics: the hindcast model is optimised for assimilating past information, while the forecast model is tailored to producing forward predictions [**Olah2015UnderstandingBlog**].

In contrast, a standard LSTM forecasting model uses the same network weights for both hindcast and forecast phases [**Hochreiter1997LSTM**]. While effective in many time series problems, this approach can limit flexibility: the same parameters must handle both the assimilation of long input histories and the generation of forward predictions. The Handoff Forecast LSTM overcomes this limitation by decoupling the two phases, allowing each network to specialise in its respective task and improving forecast skill, particularly for longer horizons.

The framework allows for architectural flexibility. The hindcast and forecast LSTMs maintain independent weights, biases, output heads, and embedding networks. Their hidden state sizes can also be set independently via configuration parameters (defaulting to a shared hidden size if unspecified). An additional feature is the option for a delayed handoff, controlled by the forecast-overlap parameter. This overlapping window allows for smoother transitions and can improve forecast stability. To regularise this overlap, the framework provides the Forecast Overlap MSE Regularization option, which penalises disagreement between the overlapping outputs of the hindcast and forecast models [**Olah2015UnderstandingBlog**].

Overall, the Handoff Forecast LSTM provides a powerful extension of the classical LSTM framework by explicitly modelling the transition between past-state assimilation and forward prediction. This makes it especially useful in hydrological forecasting, where models must effectively incorporate long sequences of past meteorological inputs while producing reliable multi-step forecasts of streamflow.

2.3.2 The CAMELS study

The Catchment Attributes and Meteorology for Large-sample Studies (CAMELS) dataset has become a benchmark resource for hydrological modelling. It provides long-term daily hydro-meteorological time series alongside a rich set of static catchment attributes for hundreds of basins in the United States [**Addor2017CAMELS**]. Its design allows for consistent, large-sample evaluations of data-driven and process-based models, making it central to recent advances in machine learning for hydrology.

Kratzert et al. [**Kratzert2018RainfallRunoffLSTM**, **kratzert2019towards**] were among the first to apply LSTM-based models at scale to the CAMELS dataset. Their work demonstrated that LSTMs could outperform traditional hydrological models, achieving median NashâĂŞSutcliffe Efficiency (NSE) values of 0.6–0.7 across basins, with peak values exceeding 0.8. These results highlighted the capacity of deep learning models to capture non-linear hydrological processes across diverse catchments.

The CAMELS studies set an important benchmark for subsequent research using the Neural Hydrology framework. They established that deep learning methods are capable of robust generalisation across heterogeneous basins, while also highlighting the importance of appropriate configuration choices (e.g. sequence length, input variables, and static attributes). As such, CAMELS results provide a useful comparison point for evaluating newer architectures such as the Handoff Forecast LSTM applied in this project.

Chapter 3

Methodology

This chapter outlines the methodology adopted to develop and evaluate Long Short-Term Memory (LSTM) neural networks for river flow forecasting across the UK. The aim of this methodology is to create a framework that can process large hydrological datasets that can handle missing and noisy data whilst also generating suitable forecasts. This approach is staged through different stages: Data Sourcing, Data Preprocessing, Data Cleaning.

3.1 Data Sourcing

3.1.1 Collection of National Dataset

3.1.1.1 Obtaining Catchment Data

The national dataset, which contains data for every basin in the UK, is provided by the Environment Agency. This dataset includes information about catchment locations and rainfall points within each area. From these point rainfall stations, Thiessen polygons are calculated to determine the spatial influence of each gauge and to estimate areal rainfall across catchments. This method ensures that rainfall inputs, derived from discrete point measurements, are appropriately weighted according to proximity and coverage, providing a more representative measure of precipitation for each basin and allowing the hydrological models to incorporate spatial variability in rainfall more effectively.

Using these polygons, an area can first be delineated to represent the extent of each basin within the UK. Based on this delineation, the amount of rainfall concentrated in the basin can then be calculated, along with the temporal distribution of rainfall at regular 15-minute intervals over multiple years, depending on the length of the observational record available. This approach ensures that both the spatial coverage and temporal variability of rainfall, as derived from point-based location of rainfall, are captured for each basin, providing essential inputs for hydrological modelling and flow forecasting.

Rainfall values for each catchment are typically derived from point measurements at rainfall gauges, which are then converted into catchment-average rainfall using spatial weighting methods such as Thiessen polygons. Thiessen polygons assign each gauge an area of influence, ensuring that rainfall contributions are weighted by their proximity and coverage within the catchment. In some cases, additional sources such as radar-based estimates may be incorporated to capture spatial variability more effectively. This process ensures that the rainfall inputs used for model training and evaluation represent both the spatial distribution and temporal variability of precipitation across the catchment, maintaining consistency with official national datasets and operational hydrological practices.

3.1.1.2 Obtaining Catchment Descriptors

The Catchment Descriptors is a dataset which is obtained from a CD ROM from 1999. This CD ROM has every catchment descriptor of every basin within the UK. Altogether, there are 19 Descriptors used within the dataset and these are :

- Name: Catchment name.
- Easting: Most downstream easting coordinate of a catchment.
- Northing: Most downstream northing coordinate of a catchment.
- East: Location based on the British National Grid (BNG).
- North: Location based on the British National Grid (BNG).
- Area: Catchment area (km^2).
- FARL: Index of flood attenuation due to reservoirs (dimensionless, between 0 and 1).
- PROPWET: Index of proportion of time soil is wet (dimensionless, between 0 and 1).
- ALTBAR: Mean catchment altitude (metres above sea level).

- ASPBAR: Index representing the dominant aspect of catchment slopes (dimensionless, between 0 and 1).
- ASPVAR: Index describing the variability of catchment slopes (dimensionless, between 0 and 1).
- BFIHOST: Base Flow Index derived from the HOST (Hydrology of Soil Types) classification (dimensionless, between 0 and 1).
- DPLBAR: Mean drainage path length (km), describing the catchment size and drainage path configuration.
- DPSBAR: Index of mean catchment slope (metres per kilometre, m/km).
- LDP: Longest drainage path length in the catchment (km).
- SAAR: Standard Average Annual Rainfall for the 1961âĂŞ1990 period (mm).
- SPRHOST: Standard Percentage Runoff derived from the HOST classification (dimensionless, percentage).
- URBEXT1990: Fractional urban extent in 1990, as defined in the Flood Estimation Handbook (dimensionless, between 0 and 1).

These are the catchment descriptors obtained from the CD ROM and the explanations can be found in the Flood Estimation Handbook [**bayliss2021catchment**].

These are all extracted and cleaned and placed into a catchment descriptor file to be used for the forecasting.

3.1.1.3 Obtaining flow Data

This data was obtained using factors such as potential evaporation, catchment average rainfall, and the total flow observed within the catchment. These are derived from the catchment dataset. Now, using a rating curve, which provides the empirical relationship between river stage (water level) and discharge (flow), the observed water level records at gauging stations can be transformed into flow values. This allows raw hydrometric measurements to be converted into consistent discharge time series, which are then used for hydrological analysis and as inputs for model calibration and forecasting.

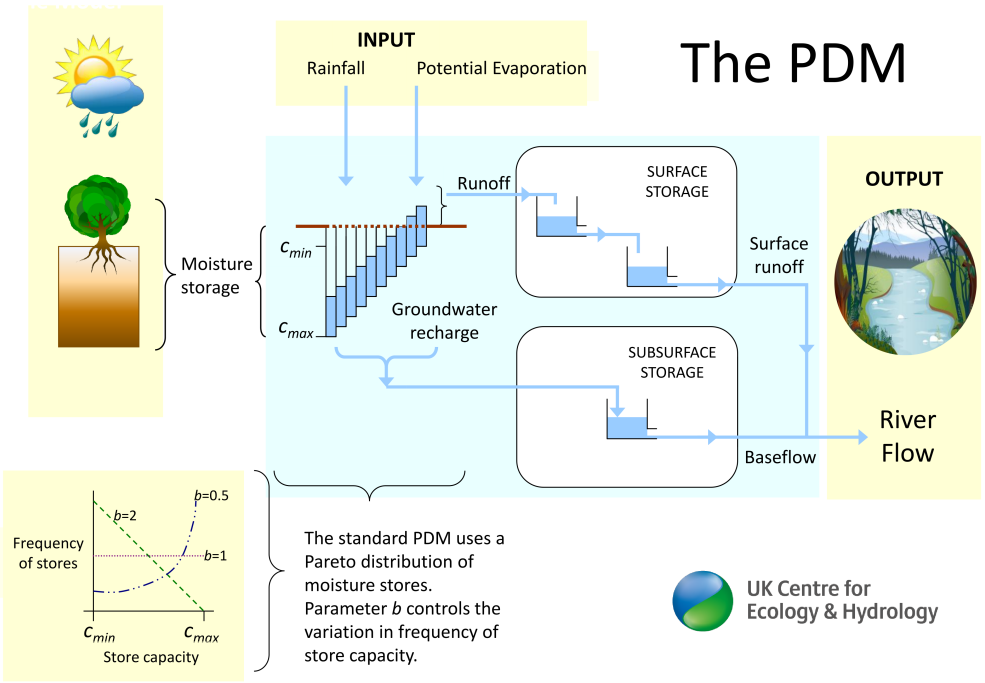


Figure 3.1: How the PDM works by UK Center for Ecology and Hydrology

The Probability Distributed Model (PDM) is a widely used rainfall-runoff model originally developed at the UK Centre for Ecology and Hydrology (CEH) [**Smith2006SoilSystem**]. It is designed to represent the heterogeneity of soil moisture storage within a catchment by assuming that the capacity of the soil to store water follows a probability distribution. This allows the model to account for the fact that different parts of a catchment respond differently to the same rainfall event.

In practice, the PDM takes as inputs time series of catchment average rainfall and potential evaporation. Rainfall is first used to replenish soil moisture up to the distributed storage capacity. Once this capacity is exceeded, excess rainfall becomes effective rainfall, which contributes to runoff generation. The runoff is then separated into quickflow and slowflow components, representing surface and subsurface pathways, and subsequently routed to the catchment outlet to simulate streamflow [**Smith2006SoilSystem**].

The PDM is particularly valuable because it links observed rainfall data to streamflow in a process-based simple framework, making it suitable for both research and operational hydrology. Within catchment modelling, it is often applied to transform periodical rainfall (e.g., 15-minute or daily totals) into corresponding flow series, taking into account soil storage

dynamics and hydrological pathways. This makes the PDM a cornerstone model in UK hydrology and an important benchmark against which newer data-driven models such as LSTMs can be compared.

In this project, the flow data was not transformed directly by the researcher but was instead provided in pre-processed form. This ensured consistency with the official datasets and operational practices employed by the Environment Agency, allowing the focus of this work to remain on the modelling and forecasting aspects rather than on hydrometric data conversion.

3.2 Data Cleaning + Preprocessing

3.2.1 Cleaning the Rainfall Data

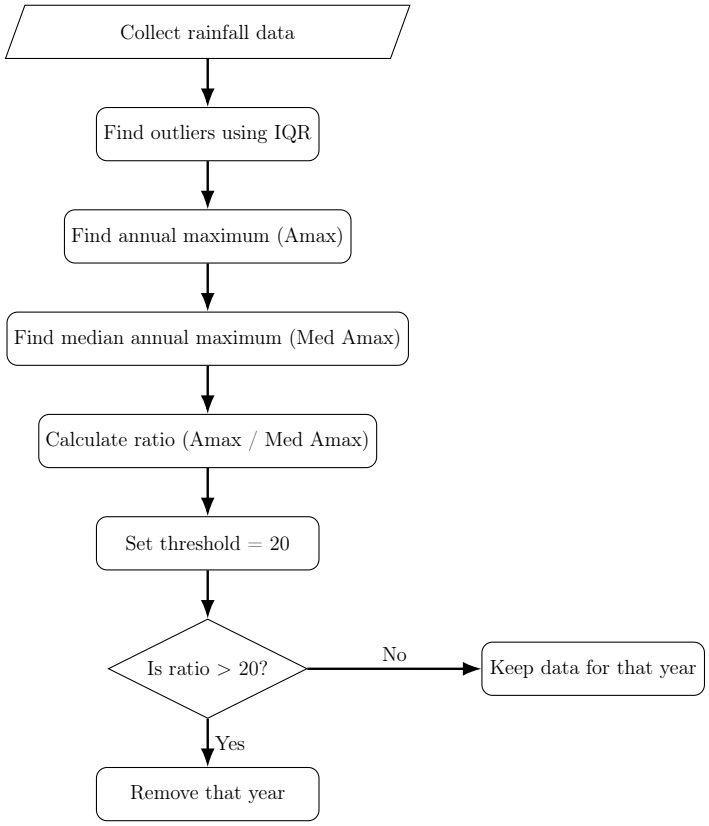


Figure 3.2: Workflow for rainfall quality control and year removal based on annual maximum ratio.

3.2.2 Cleaning Catchment + Flow Data

The catchment data contains many columns and part of the cleaning process was to select the most important columns for the forecast. Furthermore, most of the columns included within the dataset have an extremely large amount. The columns selected included:

- The Basin ID - This is a code which identifies the Basin within the UK.
- Potential Evapotranspiration (PET) - Represents the atmospheric demand for water, i.e., the amount of water that could be evaporated and transpired if sufficient water were available.
- Average Rainfall - This measures the average rainfall which fell in the specific catchment.
- Total Flow -The observed streamflow at the basin outlet, typically measured in cubic metres per second (m^3/s). (**Target Variable**).

The target variable for prediction is streamflow, representing the integrated hydrological response of each catchment to natural processes such as rainfall and potential evapotranspiration. In this project, the forecast horizon is set to 96 timesteps. Given the 15-minute resolution of the data, this corresponds to predicting streamflow up to 24 hours (1 day) ahead.

Prior to model training, the selected variables were cleaned by removing outliers and handling missing values. To distinguish erroneous data points from genuine extreme events, each detected outlier was expressed as a ratio relative to the median annual maximum for that catchment. If this ratio exceeded a predefined threshold (25 in this study), the entire corresponding water year was excluded from further analysis. For consistency with hydrological convention, water years were defined as covering from 1^{st} October to 30^{th} September of the following year.

3.2.2.1 Outlier Detection and Ratio Calculation

To ensure data quality, an outlier detection process was applied to the time series for each catchment. Outliers were identified using the interquartile range (IQR) method. For each catchment, the first quartile (Q_1) and third quartile (Q_3) were calculated from the observed series, and the interquartile range was defined as $IQR = Q_3 - Q_1$. Any value lying outside the bounds

$$\text{Lower Bound} = Q_1 - 2 \times IQR, \quad \text{Upper Bound} = Q_3 + 2 \times IQR$$

was classified as an outlier. This method is robust to skewed distributions and is widely used in hydrological applications where extreme values can otherwise dominate.

Following identification, each outlier value was normalised by the median of the corresponding annual maximum (AMAX) series for that catchment. This produced a dimensionless ratio, defined as

$$\text{Ratio}_{\text{AMAX}} = \frac{\text{Outlier Value}}{\text{Median of Annual Maxima}}$$

The ratio provides a relative measure of extremeness, enabling comparison across the catchments across the UK. Years in which the ratio exceeded a predefined threshold (in this study, a value of 25) were considered unreliable and were removed from subsequent analyses. This approach ensures that the retained dataset reflects realistic hydrological variability while excluding erroneous or highly anomalous records that could bias the forecasting models.

3.2.3 Combining Catchment Descriptors + Flow Data

Finally, to run the model the catchment descriptors and the flow data had to be matched. Since the flow data identified basins using numerical IDs, while the catchment descriptors used basin names, there was initially no direct link between the two datasets. To resolve this, a catchment shapefile containing both basin IDs and names was used as an intermediary. The centroid Easting and Northing coordinates were extracted from this shapefile and matched against the catchment descriptors using a spatial tolerance. In this way, each basin could be linked consistently across the flow data, catchment descriptors, and spatial polygons, ensuring a unified dataset suitable for model training. This also would allow the model to find the correct basin from the data and match them using the IDs and the specific co-ordinates.

Below will be the workflow shown in a flow chart, which outlines the step-by-step process of matching the Hydrological Database for Catchments (HydbCatchments shape file) with catchment descriptors using a 5000 m Easting/Northing rounding tolerance.

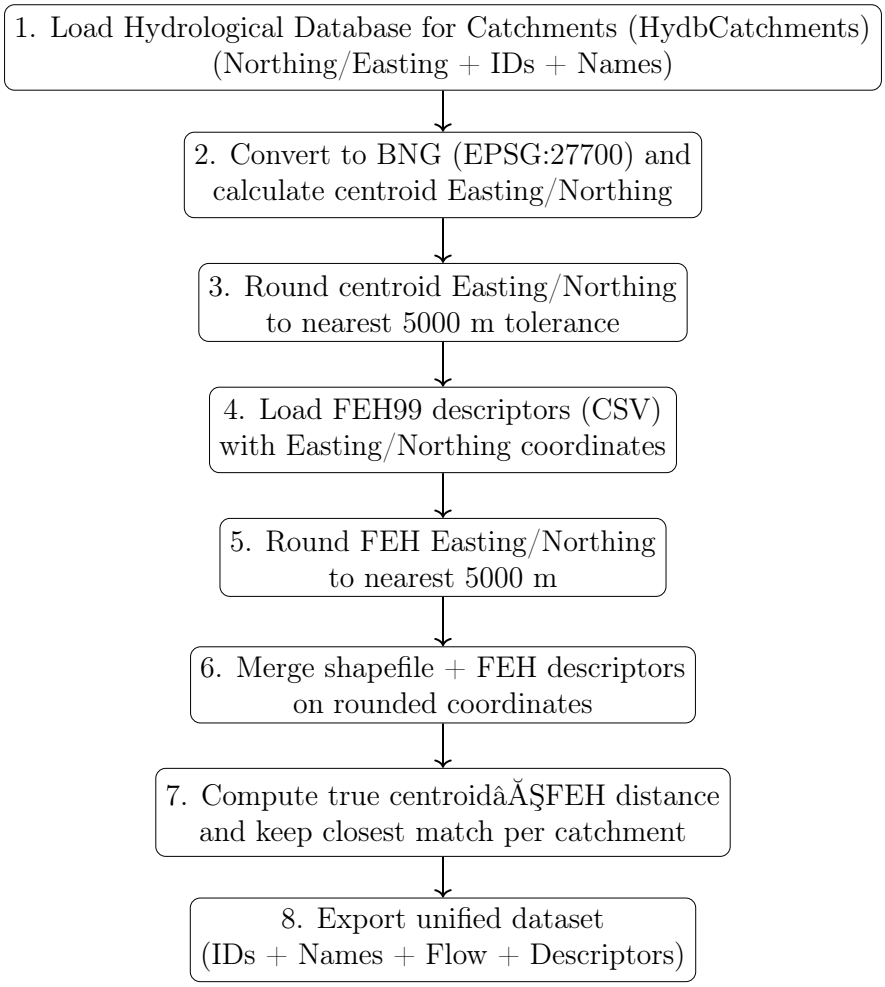


Figure 3.3: Python workflow for matching HydbCatchments shapefile centroids with Catchment descriptors using a 5000 m Easting/Northing rounding tolerance.

3.3 Neural Hydrology

Within Neural Hydrology YAML files are used to define the complete configuration of a model run. These files specify every component of the experiment, including data sources, model choice, training, validation, hyper-parameters, inputs and target variables. This allows the files to be reproducible and repeatable. YAML files are designed to be readable as well as lightweight, making them editable without great effort. Each run is linked to a specific configuration file. This allows the results to be traced directly back to the settings that produced them. These configuration files can be shared and version-controlled,

which improves collaboration. Finally, the YAML files allow manual editing and automated generation of multiple configurations. This means that the hyper-parameter searches and model comparisons can be carried out efficiently, without the need to rewrite code for each experiment. This allows the experiments remain consistent and manageable across the data.

3.3.1 Configuration Setup

After cleaning the data, we create the train, test, and validation split to structure the modelling workflow. The training set is used to fit the model parameters, the validation set is used for hyper-parameter tuning and monitoring performance during training, and the test set is held back to provide an unbiased evaluation of the final model. In this project, the split was defined both by catchments (through basin ID lists) and by time periods, ensuring that the model was evaluated on data it had not previously encountered. So, after the data is split into test, train, and validation with 70% allocated to training, 15% to validation, and 15% to testing, the modelling process balances the need for sufficient training data while still reserving enough unseen data for robust validation and evaluation. This approach improves the robustness of the results and avoids information leakage across the datasets.

The dataset originally began from 2000 to the present day. However, including the entire 25-year period would result in excessive storage and computational demands. To address this, the time window selected was from 2010 to 2022. This provided sufficient variability in the hydrological conditions whilst keeping it manageable for model training and model evaluation.

Finally, a seed was set to ensure reproducibility of the experiments. By fixing the random seed, the training process (including data shuffling, weight initialisation, and batching) produced consistent results across runs, allowing the outcomes to be compared directly and ensuring that differences in performance were due to model configurations.

3.3.2 Model & handoff network

The configuration shown above specifies the architecture of the **Handoff Forecast LSTM** model. The parameter **model: handoff forecast lstm** defines the use of the specialised architecture that separates hindcast and forecast phases with a state-handoff mechanism. The model head is set to **regression**, with a **linear** activation, which is appropriate for continuous outputs such as streamflow. The hidden size of the LSTM is set to 100 units,

determining the dimensionality of the hidden and cell states. An initial forget bias of 1 is used to encourage the model to retain information at the start of training, while an output dropout rate of 0.4 is applied for regularisation.

The block **state handoff network** defines the transformation applied to the hindcast states before passing them to the forecast LSTM. In this case, the handoff network is a fully connected layer with 128 hidden units, a **tanh** activation function, and no dropout. This ensures that the transition between hindcast and forecast phases is smooth while still allowing for nonlinear transformation of the internal states.

```
# --- Model (Handoff Forecast LSTM) ---
model: handoff_forecast_lstm
head: regression
output_activation: linear
hidden_size: 100
initial_forget_bias: 1
output_dropout: 0.4

state_handoff_network:
  type: fc
  hiddens: 128
  activation: tanh
  dropout: 0.0
```

The model employs a regression head, which is appropriate because the prediction target, streamflow, is a continuous variable. A regression head takes the information stored within the LSTMs hidden state and transforms it into a numerical prediction. This is achieved through a linear function, which maps the LSTMs learned features (stored in the hidden state) to the target variable. This allows the model to produce values that are directly comparable to observed discharge measurements (in m^3/s), enabling the use of hydrological performance metrics such as NSE, KGE, and RMSE. Furthermore, a regression head ensures that the model can capture the full range of flows, from low base flow to high flood peaks, without imposing artificial constraints on the output.

The output layer of the model uses a linear activation function. This choice is appropriate for hydrological forecasting because flow is a continuous variable that can take on a wide range of values, from low flows to extreme flood events. Unlike bounded activations such as sigmoid or tanh, which restrict outputs to fixed intervals, a linear activation produces unbounded outputs that remain directly comparable to observed discharge values in their original units (m^3/s). In addition, the use of a linear activation avoids saturation effects, which improves the modelâĂŹs ability to learn from both typical flows and rare extremes.

Several key parameters control the behaviour of the LSTM model. The **hidden size** was set to 100, defining the number of units in the hidden state and thereby the capacity of the model to learn patterns. A larger hidden size allows the model to capture more complex patterns, but also increases computational cost and elevates the risk of over fitting.The **initial forget bias** was set to 1, which encourages the forget gate to remain open at the start of training, allowing the model to retain more past information and improving its ability to capture long-term dependencies. This allows the learning to improve especially when the time series has long term dependencies. Since the dataset will be across the time of 16 years this will be important within this model.Finally, an **output dropout** rate of 0.4 was applied to the outputs of the LSTM during training to reduce over fitting by randomly disabling 40% of the output connections. This prevents the model from being too reliant on specific neurons, reducing over fitting and improving generalisation (the ability for the model to perform on unseen data).

3.3.3 Training

```
# --- Training ---
optimizer: Adam
loss: NSE
learning_rate: 0.0005
batch_size: 32
epochs: 30
max_updates_per_epoch: 500
clip_gradient_norm: 1
```

The optimiser used in this project was Adam (Adaptive Moment Estimation), which is a widely adopted extension of stochastic gradient descent. Adam improves training efficiency by maintaining two moving averages during optimisation: the first of past gradients (momentum), and the second of past squared gradients (adaptive learning rates). Momentum is the way Adam keeps an exponentially weighted average of past gradients. The past squared gradients is used to scale the step size for each parameter, giving the adaptive learning rates where frequently changing parameters get smaller updates and less active parameters get larger updates.This allows the optimiser to adjust the learning rate for each parameter individually, making the training process faster, more stable, and less sensitive to noisy gradients. Adam is therefore well suited for hydrological data, where input sequences such as rainfall and flow can be highly variable.

The loss function defines how well the model predictions match the observed values during training. Here, the Nash–Sutcliffe Efficiency (NSE) was used as the loss function, which is a standard metric in hydrology. NSE evaluates how well the predicted hydrograph reproduces

the observed hydrograph by comparing the variance of the prediction errors to the variance of the observations. Maximising NSE encourages the model to capture the overall shape and variability of streamflow, rather than only minimising raw pointwise errors. This makes it particularly well suited for hydrological forecasting tasks.

The **learning rate** of 0.0005 controls how large each update step is when adjusting the model weight. A smaller value of 0.0005 means the model will converge in a stable manner. This allows the model to avoid overshooting the minima. Furthermore, a batch size of 32 means the number of training sample processed before the model updates its weights. This batch size balances the computational efficiency with stable gradient estimates. The next parameter is **epochs**. This is set to 30 and this is the number of full passes through the training dataset. In this model it is 30, with validation checks performed inbetween.

The training configuration also included two stability measures. First, the **max updates per epoch** parameter was set to 500, which limits the number of mini-batches processed in each epoch. This reduces computational cost in large multi-basin datasets, while still exposing the model to to the full dataset, just in smaller portions. Second, gradient clipping was applied with **clip gradient norm** set to 1. This prevents exploding gradients, a common problem in RNNs, by scaling gradients such that their overall norm does not exceed 1. Together, these measures ensured efficient and stable model training.

3.3.4 Forecast Setup

This section of the configuration defines the sequence settings required for training the LSTM. Recurrent models operate on sequences of timesteps rather than single values, which allows them to capture temporal dependencies in the input data. The parameter **seq length** specifies the length of the input sequence provided to the model. In this project, a value of 336 was chosen, corresponding to 3.5 days of 15-minute observations. The **forecast seq length** was set to 96, meaning that the model predicts streamflow 96 steps ahead, i.e. 24 hours into the future. Together, these parameters determine the memory horizon of the model and the forecast horizon it must generate.

```
# --- Sequence & forecast setup (15-min data) ---
seq_length: 336            # ~3.5 days of hindcast context
forecast_seq_length: 96    # 24 hours ahead (96 * 15min)
forecast_overlap: 0
predict_last_n: 96
```

3.3.5 Variables

```
# --- Data inputs ---
dataset: generic
data_dir: ./data/Formatted_Data_copy
dynamic_inputs: [potentialevaporation, avgrain, flow_copy]
forecast_inputs: [potentialevaporation, avgrain]
hindcast_inputs: [potentialevaporation, avrain, flow_copy]
target_variables: [flow]

static_attributes:
- Easting, Northing, Area, FARL, PROPWET, ALTBAR, ASPBAR, ASPVAR,
  BFIHOST, DPLBAR, DPSBAR, LDP, SAAR, SPRHOST, URBEXT1990

use_basin_id_encoding: false
```

The data configuration specifies the inputs and targets used by the model. The **dynamic inputs** are time-varying variables, including potential evapotranspiration, average rainfall, and past streamflow (**flow copy**). The **forecast inputs** define which of these variables are available to the forecast LSTM during the prediction horizon, while the **hindcast inputs** are those used in the hindcast stage. The **target variables** specify the response to be predicted, which in this project is streamflow. In addition, a set of **static attributes** (e.g., catchment area, elevation, soil and land-use indices) provide physical descriptors that remain constant in time but vary across basins. The option **use basin id encoding** was set to **false**, ensuring that basin characterisation relied on physical descriptors rather than artificial ID encodings.

The streamflow variable is duplicated within the NeuralHydrology package, producing a **flow copy**. This is necessary because streamflow is used both as the target variable to be predicted and as a dynamic input variable. The duplication ensures that the model can safely use past flow values as inputs without directly leaking information from the target variable. In practice, **flow** is reserved as the prediction target, while **flow copy** is used as an autoregressive input, allowing the LSTM to learn how past flow influences future streamflow.

Furthermore, during the testing period a lag of 96 steps was initially added to **flow copy** to prevent the model from accessing future streamflow values that would not be available in a real forecasting scenario. However, the results were unchanged with or without the lag, and therefore the final model was trained without applying this adjustment.

3.3.6 Validation + Evaluations

```
# --- Validation & logging ---
validate_every: 1
validate_n_random_basins: 150
metrics: [NSE, RMSE, KGE]
log_tensorboard: true
log_n_figures: 2
save_validation_results: true
save_weights_every: 1
```

The validation and logging settings control how model performance is monitored during training. The **validate every** parameter was set to 1, meaning validation is performed after each epoch. A random subset of 150 basins (**validate n random basins**) was selected at each validation step to provide a representative yet computationally manageable evaluation. A wide range of performance metrics was computed (there were more metrics but, these were the most important ones), allowing the assessment of both overall model skill and hydrologically relevant behaviours. Logging options included writing results to Tensor Board (**log tensor board: true**) for visual inspection and saving two validation figures per epoch (**log n figures: 2**). Model weights were saved after every epoch (**save weights every: 1**), and validation outputs were also stored (**save validation results: true**) to ensure reproducibility and facilitate later analysis.

Furthermore, after the model has been evaluated on the test set, the predictions can be inspected to assess how well the model performed across the different basins. These predictions are stored in a pickle file as part of the Neural Hydrology output and must be extracted before further analysis or visualisation can be carried out. However, the number of basins contained within the test file determines how many prediction series are stored in the pickle output. In other words, if the test split contains 45 basins, the pickle file will only include results for those 45 basins.

3.4 Evaluation Framework

3.4.1 Metrics

To evaluate model performance, a range of statistical and hydrological metrics were used. Each metric captures a different aspect of model skill, providing a more comprehensive assessment. The key metrics are summarised below:

- **Nash–Sutcliffe Efficiency (NSE)**: See 2.2.1 [**kratzert2019neuralhydrology**]
- **Root Mean Squared Error (RMSE)**: See 2.2.3 [**kratzert2019neuralhydrology**]
- **Kling–Gupta Efficiency (KGE)**: See 2.2.2 [**kratzert2019neuralhydrology**]
- **Mean Squared Error (MSE)**: Penalises large errors more heavily by squaring the differences between predicted and observed values.
- **Alpha-NSE**: A modified NSE that focuses on variability in the predictions relative to observations.
- **Beta-NSE**: A modified NSE that focuses on bias (systematic over- or under-prediction).
- **Beta-KGE**: A component of KGE measuring bias between mean simulated and observed flows.
- **Pearson-r**: Standard correlation coefficient, measuring the linear relationship between simulated and observed values.
- **FHV (High-Flow Bias)**: Evaluates the modelâĂŹs ability to reproduce high flow volumes.
- **FMS (Medium-Flow Bias)**: Evaluates the accuracy of model predictions during medium-flow conditions.
- **FLV (Low-Flow Bias)**: Evaluates the accuracy of model predictions during low-flow conditions.
- **Peak-Timing**: Measures how well the timing of observed flow peaks is captured by the model.
- **Missed-Peaks**: Counts the number of flow peaks in the observed record not captured in the simulation.
- **Peak-MAPE**: Mean Absolute Percentage Error applied to flow peaks, measuring accuracy of peak magnitudes.

Among these metrics, the most important for this project are the **NSE**, **KGE**, **RMSE** annd **Pearson-r**. These were chosen because together they capture a broad range of information about model performance. The NSE is a standard hydrological metric that reflects how well the overall hydrograph is reproduced. The KGE combines correlation, bias, and variability into a single score, offering a more holistic assessment of performance. The RMSE quantifies

the average magnitude of prediction errors in the same units as streamflow, making the results directly interpretable. By considering these three metrics in combination, the evaluation provides both hydrological relevance and statistical robustness.

3.4.2 Validation Process

During training, the model was validated after each epoch (**validate every: 1**) to monitor performance and ensure generalisation. Rather than validating on all catchments, a subset of 150 randomly selected basins was used at each epoch, providing a representative evaluation while reducing computational cost. The same suite of statistical and hydrological metrics outlined in Section 3.4.1 was calculated during validation to maintain consistency with the final evaluation.

In addition to numerical metrics, two basins were randomly chosen at each validation step for visual hydrograph comparison (**log n figures: 2**). This enabled direct inspection of how well simulated streamflow matched observations. Validation results were logged in Tensor Board and saved for later analysis, allowing the training process to be monitored over time and ensuring the model was progressing toward a stable solution rather than over-fitting the training data.

However, since this only printed two plots per epoch, the better approach was to use the pickle file that Neural Hydrology provides. Within this process, the full set of predictions for all basins in the test period can be extracted and analysed. This allows for a more comprehensive assessment of model performance across basins, rather than relying on a limited number of examples produced during training.

3.4.3 Test Set

After training and validation, the final evaluation was performed on the **test set**, which was completely unseen during model development. The test period was defined from October 2019 to September 2022, covering three full water years. This held-out dataset provides an unbiased assessment of model skill under realistic forecasting conditions.

The test set evaluation was carried out using the same metrics as described in Section 3.4.1, ensuring consistency with the validation process. Unlike validation, which was limited to a random subset of 150 basins, the test set included **all basins specified in the test basin**

file. This allowed model performance to be assessed across the entire spatial domain.

The predictions and corresponding observations from the test set were exported by Neural Hydrology into **pickle files**, which contained the full hydrographs for each basin. These files were then used for detailed analysis and plotting, providing insight into model accuracy at both individual and aggregate levels.

3.4.3.1 Extracting the Plots from the Test Set

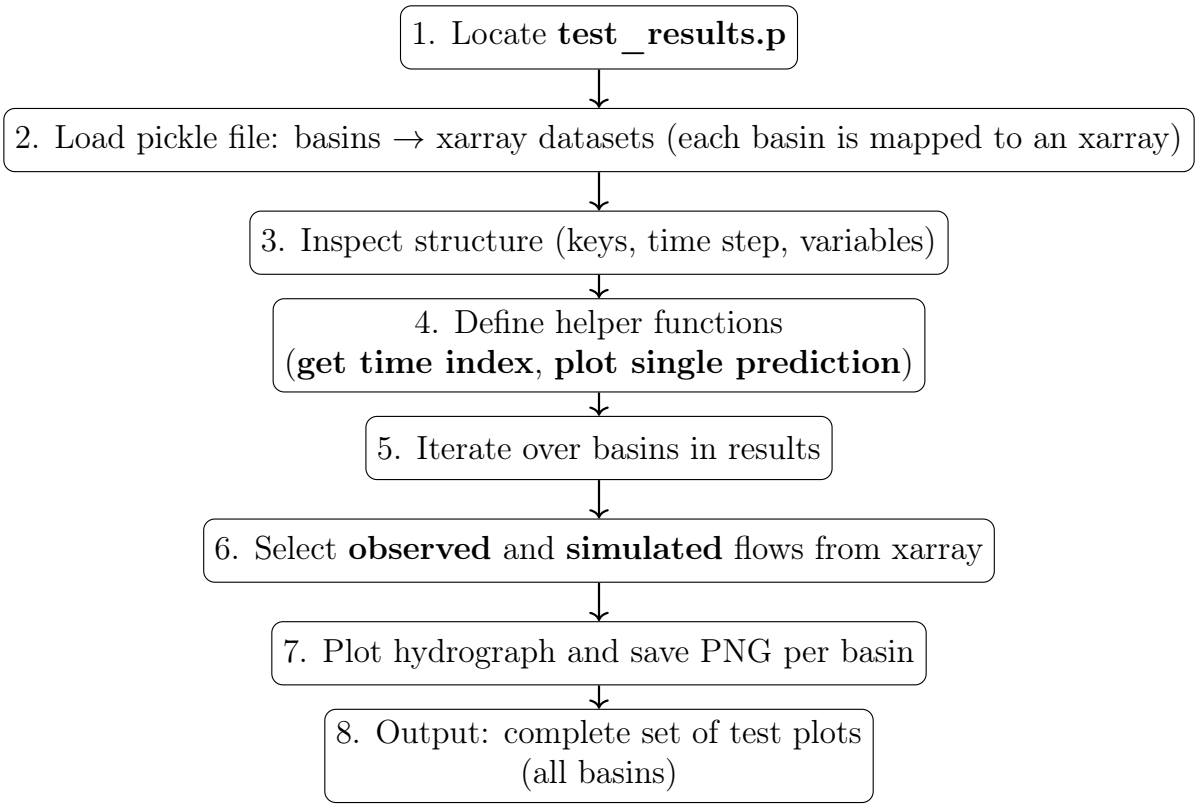


Figure 3.4: Workflow for extracting and plotting Neural Hydrology **test_results.p** outputs across all basins.

This means that, for each test basin, the plots allow direct comparison between observed and simulated flows, making it possible to evaluate how well the model reproduced the hydrograph and to visually assess its performance.

3.5 Runs

Initially, the project was prescribed to run the model across all 600+ basins in the dataset. However, this approach quickly proved to be infeasible due to the large storage requirements generated during training and evaluation. Each basin produces substantial intermediate files, model outputs, and validation results, which collectively exceeded the available storage capacity when scaled to the full dataset. These intermediate files include: the information with the metrics, the pickle file containing the test basins with observed and simulated flows, and the raw data itself. The data is extremely large, as it spans many years at a 15-minute resolution, which multiplies considerably when extended across hundreds of basins.

To address this, a representative subset of basins was chosen. Using a statistical calculation, it was determined that a sample size of ∼246 basins would capture approximately 95% of the variability in the full population, ensuring that the selected basins remained highly representative of the overall dataset. However, we picked to use 300 basins because this provided an additional buffer above the calculated requirement of 246 basins, ensuring that more catchment diversity was captured. This larger sample increased the robustness of the analysis while still remaining computationally feasible within the available storage and processing limits.

3.5.1 The importance of NetCDF files

The raw hydrological datasets used in this project were extremely large, spanning hundreds of basins and covering multiple decades at 15-minute resolution. Storing these data in a traditional **CSV** format would have resulted in excessive file sizes and slow read/write performance. Instead, the data were converted into the **NetCDF** (Network Common Data Form) format, which is specifically designed for storing large, multidimensional scientific datasets. In fact, the dataset contained more than **100 million lines of data**, which made CSV storage infeasible.

NetCDF files allow efficient storage and access to variables such as rainfall, potential evapotranspiration, and streamflow across both space and time. Unlike CSV, NetCDF supports compression, metadata, and direct indexing, meaning only the required portions of the dataset need to be loaded into memory. This significantly reduced storage requirements and improved computational efficiency, making it possible to train deep learning models across hundreds of basins without overwhelming memory or disk capacity.

3.5.2 Configuration Files

In Neural Hydrology, each experiment is defined by a configuration file. The configuration file does not represent a run by itself, but rather provides the complete set of instructions for how a run is carried out. When the model is carried out, the configuration file is read and a run is created, which produces outputs such as validation logs, test results, and saved weights. This design ensures that every run can be traced directly back to the exact configuration that produced it, improving reproducibility as well as repeatability.

To efficiently explore different hyper-parameter settings, Neural Hydrology provides a utility to automatically generate multiple configuration files from a single base template. In this project, a base configuration file was used as the template, and a dictionary of parameters to vary was defined. These included **batch size**, **hidden size**, and whether to use **basin ID encoding**.

The **hidden size** allows the model to capture more complex patterns within the data. Since the network has a greater capacity to store information. However, increasing hidden size also raises the risk of overfitting and makes training more computationally expensive. A smaller hidden size reduces complexity and speeds up the training process.

The **batch size** determines how many training samples are processed before the model updates its weights. Instead of adjusting the network after each individual sample, the model processes a batch of data, computes the average error across that batch, and then performs a weight update. A smaller batch size means more frequent updates, which can make learning more stable but slower per epoch. Larger batch sizes make training faster per epoch and take better advantage of GPU parallelism, but they require more memory and can sometimes lead to poorer generalisation.

Combining the **batch size** and **hidden size** resulted in a total of six configurations for each run of the model. Specifically, the batch sizes tested were **128** and **196**, while the hidden sizes tested were **100**, **128**, and **196**. The Cartesian product of these options gave $2 \times 3 = 6$ unique configurations.

This approach allowed rapid experimentation across different hyper-parameters while ensuring reproducibility, since each run could be traced back to a specific configuration file.

3.5.3 300 Basins

The first run of the model was conducted with **300 basins**, using the handoff Forecast LSTM configuration described previously. This experiment serves as the main basis for the results presented in this dissertation.

This experiment used curated basin lists containing 300 UK catchments and a fixed temporal split(2010–2016 for training, 2016–2019 for validation, and 2019–2022 for testing). A fixed seed (425) and GPU execution ensured reproducibility.

The model was a handoff Forecast LSTM with a regression head and linear output activation. The LSTM hidden size was set to 100 with output dropout of 0.4. A fully connected state handoff network (128 units, tanh) transformed the hindcast states to initialise the forecast LSTM.

Sequence settings reflected the 15-minute resolution: the model ingested 336 steps of context(about 3.5 days) and produced a 96âĂŞstep forecast horizon (24 hours), predicting the last 96 steps with no overlap between hindcast and forecast phases.

Training used the Adam optimiser with a learning rate of 0.0005, batch size of 32, and 25 epochs. To keep epochs computationally manageable on a large multi-basin dataset, updates per epoch were capped at 500 and gradient norms were clipped at 1 to stabilise optimisation.

Dynamic inputs comprised of potential evapotranspiration, average rainfall) and a copy of the flow column (flow copy). The forecasted inputs included only potential evapotranspiration and average rainfall because including flow here would give the model access to future information that is not available in a real forecasting scenario. Finally, the hindcast stage the model only used rainfall, potential evapotranspiration and the flow data. The target variable was observed flow of a river. Static inputs is where the catchment descriptors (e.g., area, FARL,PROPWET, ALTBAR, BFIHOST, SAAR) were included to support cross-basin generalisation (i.e., the ability of the model to transfer what it has learned from some catchments to unseen catchments), while basin IDencoding was disabled to prioritise physical attributes over identifiers.

Validation was run every epoch on a random subset of 150 basins from the validation split. A broad set of metrics was recorded (including NSE, RMSE, and KGE alongside peak-timing and Pearson-r)., results were logged to Tensor Board, and per-epoch figures were saved for qualitative inspection. Final performance was assessed on the held-out 2019–2022 test period.

3.5.4 256 Basins

For the 256 basin experiment, a preprocessing script was used to curate the random sample of basin CSVs before training. Each CSV was scanned for the flow column, the number of missing values was counted, and a removal decision was made based on a configurable missing ratio threshold (here, 40%). Files exceeding the threshold (or empty files) were moved to a âĂIJremovedâĂİ folder; the rest were moved to a âĂIJcleanedâĂİ folder. When a CSV was removed, the script also searched the companion time series directory for matching NetCDF files (case-insensitive, allowing .nc tolerant filename matching) and deleted them to keep the dataset consistent. The process supports a dry-run mode (report only), writes a detailed log with reasons for keep/remove actions, and prints a final summary (kept/removed counts and deleted NetCDFs). This ensured that only basins with usable flow series entered the 256 model training.

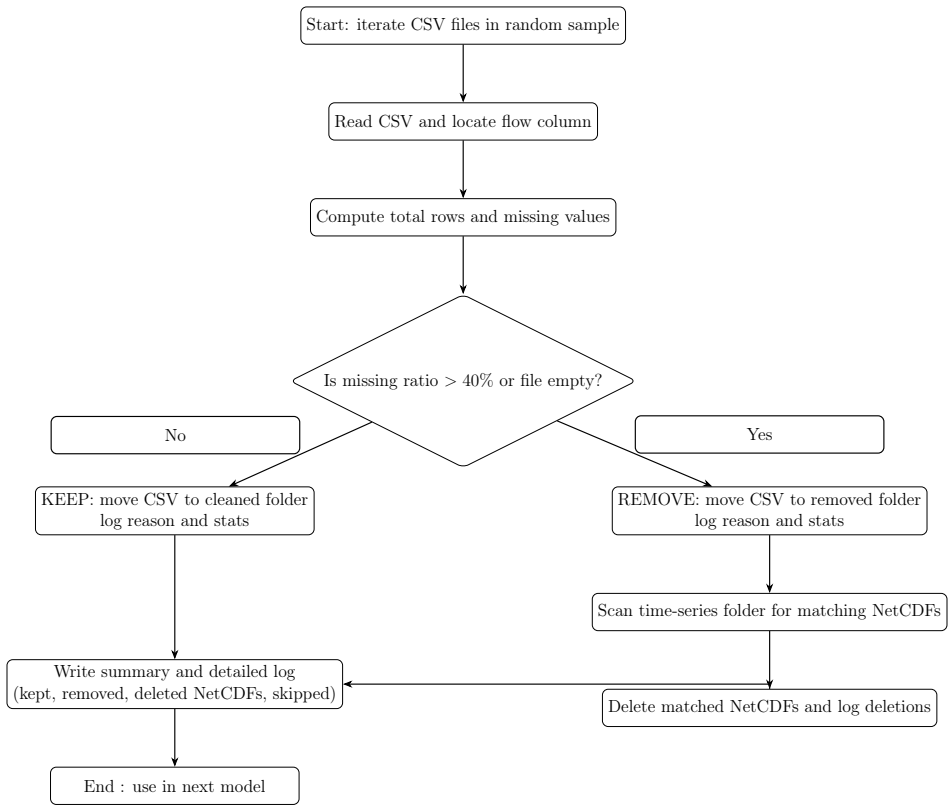


Figure 3.5: Cleaning workflow for the 256-basin experiment.

An example of a basin with substantial missing data is provided in Appendix A.4.

3.5.5 Result Extraction

The pickle files provided by Neural Hydrology enabled the model to be visually seen and then plotted. These files contain the predicted and observed streamflow values for each basin in the test set, stored as xarray datasets. By loading these files, the predictions could be inspected at both basin and epoch level. A custom script was used to generate hydrographs, plotting observed versus simulated flows, which provided a visual means of evaluating model skill. This complemented the statistical metrics and general hydrograph behaviour. The number of basins extracted corresponded directly to those specified in the test split.

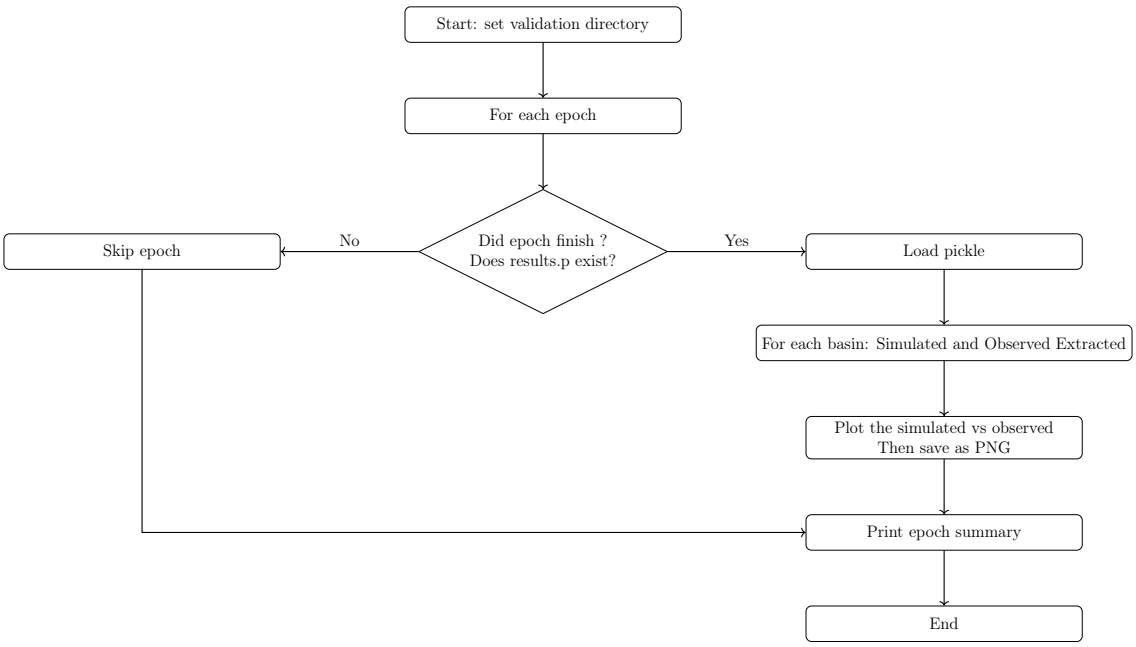


Figure 3.6: Result extraction flow: epochs that ended early or without validation files were skipped, while available results were plotted and summarised.

Furthermore, Neural Hydrology provides an output log that records key information during training and evaluation. This includes the training and validation losses per epoch, selected performance metrics, and details of the model configuration. The log is saved alongside each run, meaning results can always be traced back to the exact settings that produced them. In addition, because the log captures intermediate results, it allows comparisons between different experiments and makes it easier to spot when a model is converging, overfitting, or underperforming.

When combined with Tensor Board outputs, which were enabled in this project, the logs can be visualised interactively, providing insights into how the loss curves and metrics evolve over

time. This makes the training process transparent, reproducible, and easier to interpret.

Overall, the results were extracted using both the pickle files provided by Neural Hydrology and the output logs. Tensor Board was used to visualise the training and validation losses for each configuration, and it also provided information on the exact runtime of the model. The output logs were particularly important, as they contained the full record of performance metrics, making it possible to evaluate and compare the different runs in a structured way.

3.6 Computational Setup

All experiments were conducted using a workstation equipped with an NVIDIA CUDA-enabled GPU and sufficient memory to handle training across hundreds of basins simultaneously. The exact hardware setup included a GPU with and 96 GB of system RAM, ensuring that both large input sequences and model parameters could be processed efficiently.

A fixed random seed was applied to ensure reproducibility across runs. All configuration files were saved in YAML format and version-controlled, ensuring that each set of results could be traced directly back to its training settings. This allowed runs to be repeated exactly and also made it possible to systematically compare the effect of different hyper-parameter choices.

Training times varied depending on the dataset size and configuration. For example, a full run of 25 epochs across 300 basins required approximately 24.5 hours, whereas the 256-basin run completed in roughly 18.5 hours. The difference highlights the scaling challenges of training on increasingly large basin sets. Storage requirements were also significant, with model checkpoints, pickle result files, and logs generated for each epoch.

Training and validation were monitored using **Tebnsor Board**, which provided real-time visualisation of loss curves and recorded the runtime of each epoch. Neural Hydrology also produced detailed output logs, including performance metrics for each basin, ensuring a transparent evaluation process.

Overall, the computational setup balanced the demands of large-scale hydrological forecasting with reproducibility and transparency, ensuring that results could be reliably interpreted and replicated.

3.6.1 Hardware/Software Environment

The software stack consisted of **Python 3.10** and the open-source **Neural Hydrology** package, built on **PyTorch**. Supporting libraries included **pandas** and **numpy** for data handling, **matplotlib** for visualisation, and **Iris Cube** for managing and preprocessing **NetCDF** files. Training and validation were monitored using **Tensor Board**, which provided real-time logging of training/validation loss and recorded runtime per epoch.

A fixed random seed was applied across experiments to guarantee reproducibility. All YAML configuration files, logs, and model outputs were stored systematically, ensuring that each experiment could be repeated exactly and that hyper-parameter choices were traceable.

Chapter 4

Results

4.1 Introduction

This chapter presents the outcomes of training and evaluating the Handoff Forecast LSTM model on two catchment subsets, comprising 256 and 300 catchments respectively. The aim of this chapter is to demonstrate the predictive skill of the model and to highlight differences in performance across catchments of varying characteristics. Results are assessed using both statistical and hydrological metrics, including the NashâĂŞSutcliffe Efficiency (NSE), KlingâĂŞGupta Efficiency (KGE), and Root Mean Squared Error (RMSE), alongside additional diagnostic measures.

In addition to numerical evaluation, model outputs are inspected visually through hydrograph comparisons between observed and simulated flows. These plots allow assessment of how well the model reproduces streamflow dynamics such as seasonal variability, flood peaks, and low-flow conditions. Performance is further examined at both the aggregated level (across all basins) and the individual basin level, where case studies are used to illustrate examples of strong and weak predictions.

Comparisons between the 256-basin and 300-basin experiments are made to investigate how the size of the training set and basin diversity influence generalisation. Run-time performance, convergence behaviour, and stability across epochs are also reported, making use of both Tensor Board logs and Neural Hydrology output files. Together, these analyses provide a comprehensive view of the modelâĂŹs predictive capability and its limitations in the context of large-sample hydrology.

4.1.1 Experimental Constraints

The remote desktop environment provided ideal conditions for training the model, with sufficient computational power to handle the basin-scale experiments. However, storage capacity became a limiting factor. Although six different configurations were prepared, only three could be executed in practice, since each run required substantial disk space for intermediate files, logs, and model weights. In fact, the completed runs alone consumed approximately **400 GB** of storage, highlighting the data-intensive nature of hydrological deep learning workflows.

This meant that, for each run, only three configurations could be executed in full, each consisting of **25 epochs**. The restriction ensured that experiments remained feasible within the available resources, while still allowing meaningful comparisons between the selected configurations.

4.2 Training and Validation Performance

4.2.1 Training and Validation Loss

The training and validation loss curves for the two experiments (256 basins and 300 basins) are presented in Figures 4.2 and 4.1. These curves provide insight into the model's optimisation process across the 30 training epochs, illustrating how well the model learned from the training data and how effectively it generalised to unseen validation data.

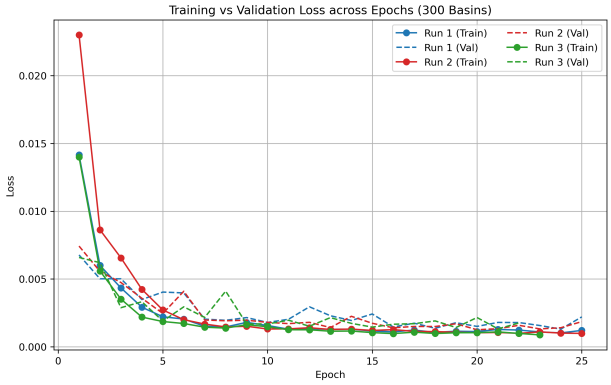


Figure 4.1: Training vs. validation loss for the 300-basin experiment.

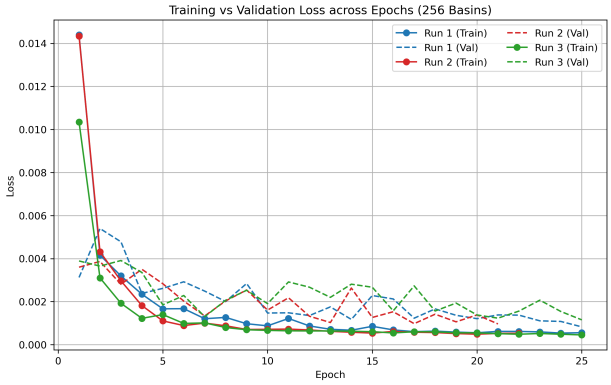


Figure 4.2: Training vs. validation loss for the 256-basin experiment.

For the **256-basin experiment**, the training loss decreased steadily across epochs, indicating that the model was successfully fitting the training data. The validation loss followed a similar trajectory, though it consistently remained above the training curve, as expected for unseen data. The relatively close alignment of the curves suggests that overfitting was limited. By approximately epoch 20, the validation loss had largely stabilised, indicating diminishing returns from further training.

For the **300-basin experiment**, a similar pattern was observed, though the validation loss appeared to plateau slightly earlier compared to the 256-basin run. This may reflect the increased complexity introduced by the larger and more heterogeneous basin set, which makes generalisation more challenging. Nonetheless, the fact that the validation curve remained stable while avoiding significant divergence from the training curve indicates that the model was still able to generalise adequately.

Overall, both experiments demonstrate that the model configuration (hidden size, batch size, dropout, and optimiser) was appropriate for preventing severe overfitting while still learning meaningful hydrological representations. The moderate gap between training and validation losses highlights the balance between capturing catchment-specific dynamics and maintaining generalisability across basins.

In both experiments, the most pronounced improvements in loss occurred during the first 5–10 epochs, after which convergence slowed and the curves approached stability. This pattern is consistent with the behaviour of LSTM-based models, which typically learn dominant temporal dependencies early in training before refining finer-scale patterns in later epochs. The relatively small but consistent gap between training and validation losses across both runs provides evidence that the models achieved good generalisation to unseen basins, a critical requirement for hydrological forecasting. It is also worth noting that the stabilisation of validation loss by epoch 20–25 suggests that early stopping could have been implemented to reduce training time without sacrificing performance. However, all runs were executed for 30 epochs to ensure comparability across experiments.

When comparing the two experiments, the 300-basin configuration plateaued slightly earlier than the 256-basin configuration. This is likely due to the greater variation (massive data set increases variance) introduced by the larger basin set, which increases the difficulty of learning a single representation that generalises across all basins. Despite this, the model remained stable, suggesting that the chosen hyperparameters struck a balance between model complexity and regularisation.

Although training was extended to 30 epochs in all cases for comparability, the observed stabilisation of validation losses by epoch 20–25 indicates that early stopping or adaptive

learning rate scheduling could further optimise computational efficiency without sacrificing predictive performance.

4.3 Overall Model Evaluation

The Handoff Forecast LSTM produced results that, while promising, also highlighted areas for improvement. Across both the 256- and 300-basin experiments, the model achieved competitive values for key hydrological metrics such as NSE, KGE, and RMSE, demonstrating its ability to reproduce streamflow variability across a wide range of UK catchments. The model successfully generalised to unseen basins and avoided severe overfitting, as evidenced by the training and validation loss curves. This will be analysed on both experiments.

Each experiment generated substantial disk usage, which imposed storage constraints on the remote machine. As a result, it was only possible to execute three configuration files per run, rather than the full set originally planned.

4.3.1 256 Basins

4.3.1.1 NSE & KGE

Figures 4.3–4.5 present the Nash–Sutcliffe Efficiency (NSE) and Kling–Gupta Efficiency (KGE) scores across epochs for the three 256-basin runs. Both metrics gradually improve with training, starting from negative values (indicating poor predictive skill) and converging towards values in the range of 0.3–0.5 by epoch 25. This trend demonstrates that the Handoff Forecast LSTM was able to capture meaningful hydrological patterns, though performance remained moderate.

In all three runs, the KGE score tends to be slightly higher than NSE, which reflects that KGE incorporates correlation, bias, and variability, making it more robust in cases where NSE penalises systematic errors more harshly. The consistency across runs highlights that the modelâĂŹs performance is reproducible under different random initialisations, though some fluctuations between epochs remain due to stochastic optimisation.

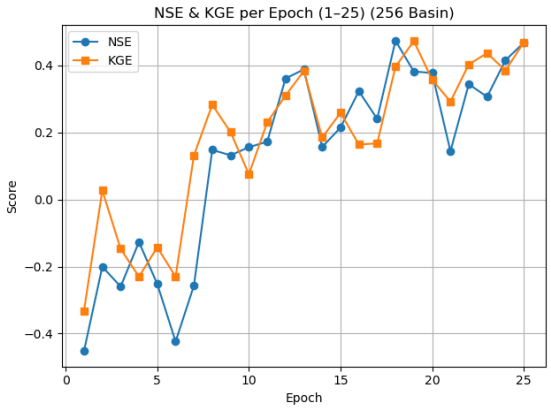


Figure 4.3: NSE and KGE per epoch for the 256-basin experiment, Run 1.

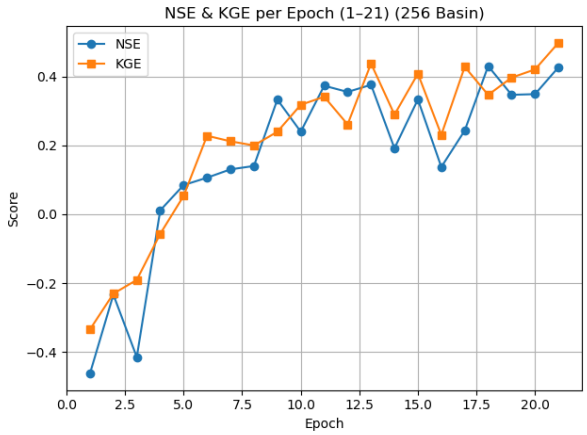


Figure 4.4: NSE and KGE per epoch for the 256-basin experiment, Run 2.

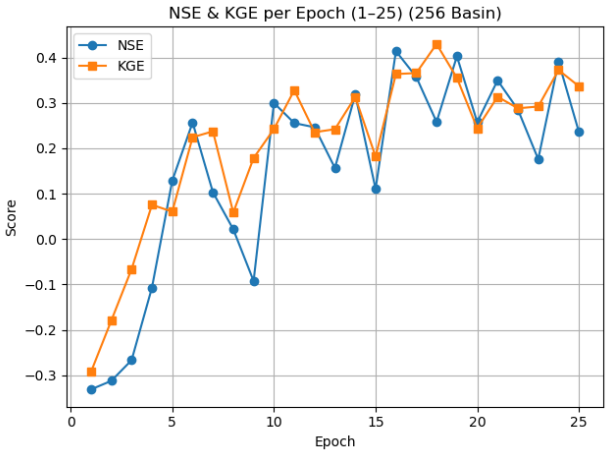


Figure 4.5: NSE and KGE per epoch for the 256-basin experiment, Run 3.

Overall, the results show that while the model was able to achieve stable improvements across epochs, the absolute performance levels suggest room for further optimisation, either through longer training, modified architectures, or additional input features.

4.3.1.2 Pearson-r

The Pearson-r results for the 256-basin experiment are presented in Figures 4.6–4.8. Across all three runs, the metric demonstrates a consistent upward trend during the first 10 epochs,

indicating that the model was quickly learning the linear relationship between predicted and observed flows.

After epoch 10, the Pearson-r stabilised between **0.75 and 0.82**, suggesting a strong correlation between the simulated and observed hydrographs. While small fluctuations occur in individual epochs, the general plateau shows that further training yielded limited improvements beyond epoch 15–20.

This behaviour is consistent across runs, with Run 1 reaching a slightly higher peak correlation (**0.83**) compared to Runs 2 and 3. The stability of Pearson-r across multiple runs suggests that the model generalised reliably across the 256 basins, though there remains room for improvement towards values closer to 1.0, which would indicate near-perfect correlation.

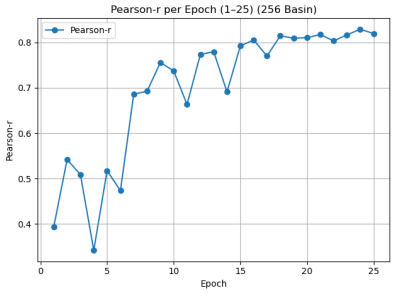


Figure 4.6: Pearson-r per epoch, Run 1.

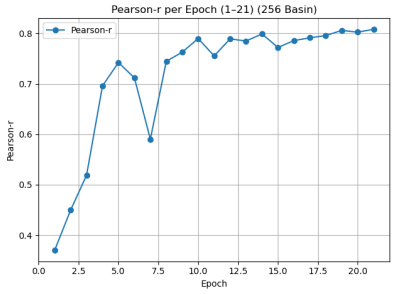


Figure 4.7: Pearson-r per epoch, Run 2.

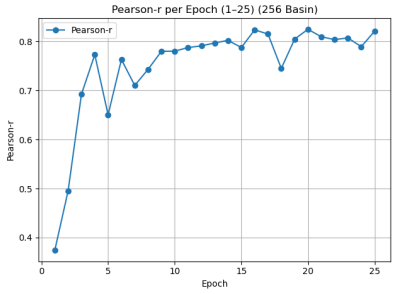


Figure 4.8: Pearson-r per epoch, Run 3.

4.3.1.3 RMSE

Figures 4.9–4.11 present the RMSE across epochs for the three runs of the 256-basin experiment. In **Run 1** (Figure 4.9), RMSE began relatively high (∼0.70) and declined steadily, stabilising near ∼0.45 after epoch 20. This indicates gradual but consistent improvements in predictive accuracy throughout training. **Run 2** (Figure 4.10) followed a similar trajectory, although the decrease was more pronounced during the first 10 epochs, levelling off near ∼0.50. This suggests rapid early learning followed by slower refinement in later epochs. **Run 3** (Figure 4.11) also demonstrated consistent improvement, with minor fluctuations but an overall reduction from ∼0.75 to ∼0.45.

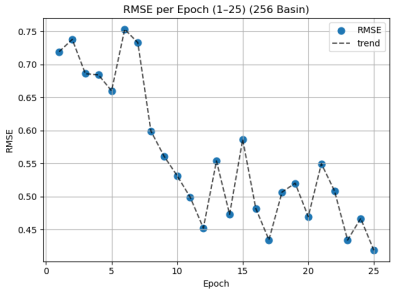


Figure 4.9: RMSE per epoch, Run 1.

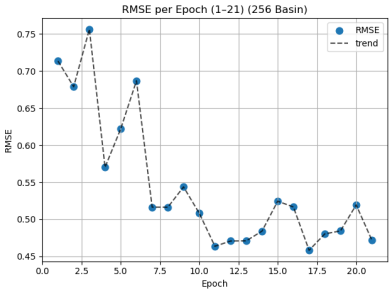


Figure 4.10: RMSE per epoch, Run 2.

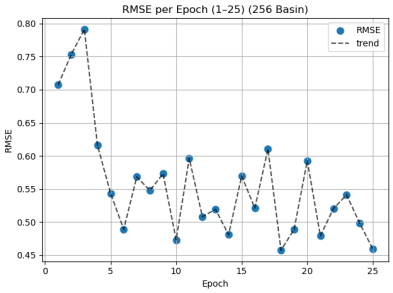


Figure 4.11: RMSE per epoch, Run 3.

Taken together, the three runs show convergence towards comparable RMSE values in the range of 0.45 to 0.50. This convergence highlights the robustness of the training procedure despite minor run-to-run variability. The sharp initial decreases reflect the modelâĂŹs ability to quickly capture catchment-level dynamics, while the plateauing behaviour in later epochs suggests diminishing returns from extended training. Overall, the consistency across runs reinforces the stability of the chosen configuration and confirms the modelâĂŹs capacity to generalise across the basin set.

4.3.2 300 Basins

The 300 basin test including the data with large amounts of missing data. This test can show whether the missing data actually had an impact on performance.

4.3.2.1 NSE & KGE

Figures 4.12–4.14 show the NSE and KGE across epochs for the three runs of the 300-basin experiment. All runs demonstrate steady improvements in both metrics, with convergence around 0.45–0.6 after epoch 20, though some variability is observed between runs.

Run 1 (Figure 4.12) began with both metrics near or below zero, indicating weak predictive skill in the initial epochs. From around epoch 5 onwards, both NSE and KGE showed steady improvements, gradually rising towards 0.4 to 0.5. However, this run exhibited noticeable fluctuations between epochs, suggesting some instability in learning across such a large and diverse basin set.

Run 2 (Figure 4.13) displayed a smoother trajectory. Both NSE and KGE improved consistently through training, surpassing 0.5 by epoch 15 and peaking close to 0.55 to 0.58. This run indicates stronger generalisation compared to Run 1, and highlights the influence of random weight initialisation and stochastic optimisation on training stability.

Run 3 (Figure 4.14) again showed early instability, particularly between epochs 5âĂŞ10, where metric values fluctuated significantly. Despite this, both metrics recovered well in later epochs, converging towards 0.45 to 0.6 by epoch 20. This run demonstrates that even with unstable intermediate phases, the model was able to eventually stabilise and capture basin-level dynamics.

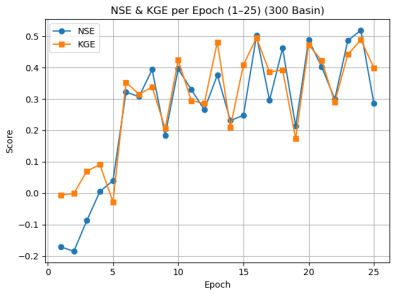


Figure 4.12: NSE and KGE per epoch, 300-basin Run 1.

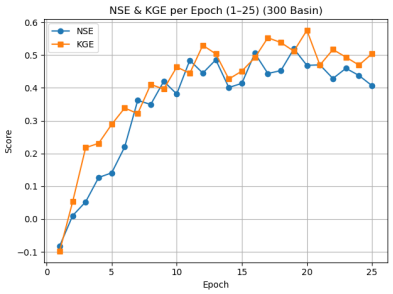


Figure 4.13: NSE and KGE per epoch, 300-basin Run 2.

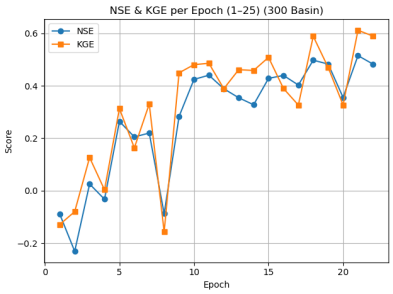


Figure 4.14: NSE and KGE per epoch, 300-basin Run 3.

Overall, all three runs converged towards similar performance levels, with NSE and KGE typically stabilising between 0.45 and 0.6 after epoch 20. The variability in early training phases reflects the challenges of scaling the model to 300 basins, where heterogeneity introduces additional noise. Nonetheless, the consistent convergence across runs suggests that the configuration is robust and capable of generalisation, albeit with slightly noisier optimisation compared to the 256-basin experiment.

4.3.2.2 Pearson-r

Figures 4.15–4.17 present the Pearson-r values across epochs for the three runs of the 300-basin experiment. Pearson-r measures the linear correlation between simulated and observed streamflow, with values closer to 1 indicating stronger agreement.

For **Run 1** (Figure 4.15), the Pearson-r rose steadily from an initial value of ∼0.45, reaching ∼0.83 by epoch 15 and stabilising between 0.81 and 0.85 in later epochs. This indicates that the model was learning the underlying hydrological patterns effectively and consistently converging, as reflected by the strong Pearson-r values across runs. The stability

of convergence highlights the modelâĂŹs ability to generalise across the wide range of basins within the dataset.

In **Run 2** (Figure 4.16), the trajectory was more erratic in the early epochs, with fluctuations between 0.4 and 0.7. However, after epoch 6, the curve stabilised around 0.75–0.82, again demonstrating reliable correlation performance, despite some mid-epoch dips.

For **Run 3** (Figure 4.17), the results were smoother, with a rapid increase from ∼0.47 in the first epoch to ∼0.8 by epoch 10. Afterward, the curve maintained stability, finishing at ∼0.85 by epoch 25, representing the strongest and most stable run.

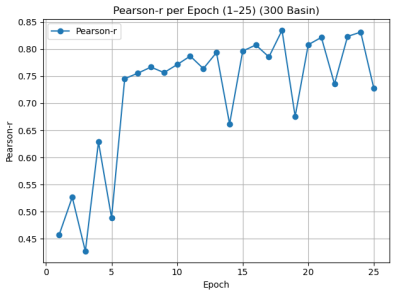


Figure 4.15: Pearson-r per epoch, Run 1 (300 basins).

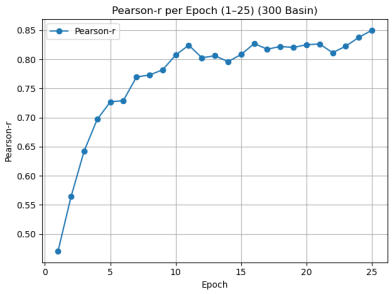


Figure 4.16: Pearson-r per epoch, Run 2 (300 basins).

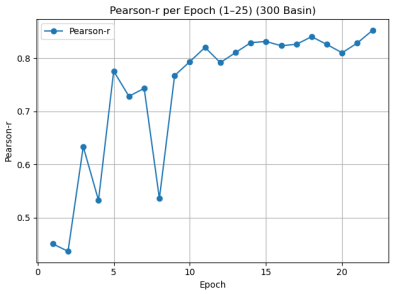


Figure 4.17: Pearson-r per epoch, Run 3 (300 basins).

The plots show that the runs converged towards Pearson-r values above 0.8, confirming that the Handoff Forecast LSTM captured strong linear relationships between simulated and observed flows across the 300 basins. While Run 2 exhibited greater volatility, the consistency of the final correlations across all runs highlights the robustness of the model training process.

4.3.2.3 RMSE

Figures 4.18–4.20 show the RMSE values across epochs for the three 300-basin runs. In Run 1 (Figure 4.18), RMSE began relatively high (∼0.75) but showed a sharp decline in the first 5 epochs, stabilising near 0.40 by epoch 10–15. While some fluctuations were observed, the overall downward trend demonstrates improved predictive accuracy as training progressed.

In Run 2 (Figure 4.19), the model again started with a high RMSE (∼0.70), followed by a steep drop within the first 5 epochs. From epoch 10 onward, the RMSE plateaued around 0.35–0.40, indicating that the model quickly learned general basin dynamics and achieved relatively stable performance thereafter.

For Run 3 (Figure 4.20), a similar trajectory was observed, with RMSE decreasing sharply after epoch 5 and stabilising between 0.35 and 0.40. Despite minor oscillations, the overall pattern again reflects steady improvement and convergence.

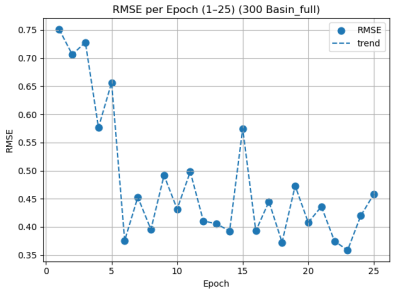


Figure 4.18: RMSE per epoch, Run 1 (300 basins).

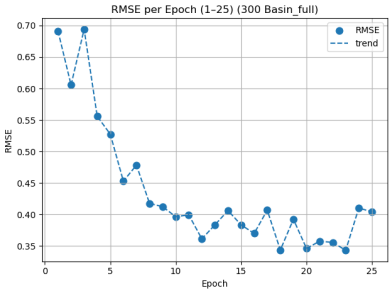


Figure 4.19: RMSE per epoch, Run 2 (300 basins).

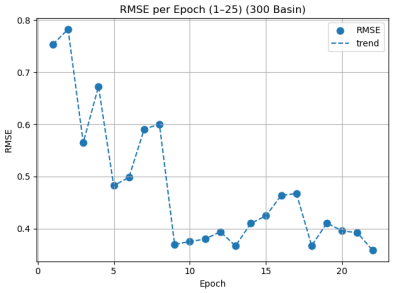


Figure 4.20: RMSE per epoch, Run 3 (300 basins).

Finally, the three runs demonstrate consistent convergence behaviour, with RMSE values stabilising around 0.35–0.40. This reflects robust learning despite differences across runs and suggests that the chosen configuration was able to generalise effectively across the diverse 300-basin dataset. The sharper early declines highlight rapid learning of large-scale basin-level dynamics, while the later plateau reflects diminishing returns from continued training.

4.4 Comparison of 256 vs 300 Basins

Firstly, both models were extremely computationally expensive, taking more than 400GB worth of disk space per experiment. Furthermore, if the number of workers had not been set to 1 (or 0), the model would not have run, as parallel data loading proved too intensive on the available RAM. This limitation highlights the trade-off between computational efficiency and practicality when working with large-scale hydrological datasets. Despite these challenges, all runs were successfully completed, but the resource demands restricted the number of configurations that could be executed. Consequently, only three runs per basin subset were feasible, underscoring the importance of balancing model complexity with available computational infrastructure.

The experiments conducted on the 256-basin and 300-basin subsets allow for a direct comparison of model behaviour under different levels of dataset size and diversity. In both cases, the handoff forecast LSTM exhibited steady improvements across epochs, as shown by decreasing training/validation losses, increasing NSE and KGE scores, and declining RMSE

values. This consistency demonstrates the robustness of the model configuration across datasets of different sizes.

For the **256-basin runs**, performance improvements were clear but more gradual, with NSE and KGE values stabilising in the range of 0.3–0.4, Pearson-r plateauing around 0.80, and RMSE converging to approximately 0.45–0.50. These values indicate reasonable predictive skill but also highlight that convergence required almost the full set of 25 epochs, reflecting the challenge of capturing catchment-specific dynamics with fewer basins.

In contrast, the **300-basin runs** displayed sharper early improvements, with NSE and KGE exceeding 0.4 within the first 10 epochs and RMSE stabilising at lower values of 0.35–0.40. Pearson-r also consistently exceeded 0.80 across all runs. The faster convergence suggests that the larger and slightly more diverse dataset allowed the model to generalise basin-level relationships more efficiently, learning hydrological dynamics quicker and with reduced overfitting risk.

To conclude, the comparison indicates that expanding from 256 to 300 basins led to more stable training dynamics, stronger generalisation, and improved final performance across key metrics. While both setups highlight the modelâĂŹs ability to capture hydrological variability, the 300-basin experiments benefited from greater data diversity, which improved the robustness and convergence speed of the trained models.

4.5 Overall Model Evaluation

The handoff forecast LSTM demonstrated the ability to capture basin-level hydrological dynamics with moderate accuracy, as evidenced by improvements across NSE, KGE, Pearson-r, and RMSE metrics. Both the 256- and 300-basin experiments showed consistent learning behaviour, with rapid improvements during the first 10âĂŞ15 epochs followed by stabilisation, suggesting effective convergence.

Across runs, the model avoided severe overfitting, with training and validation losses remaining reasonably close. This indicates that the chosen configuration (hidden size, batch size, dropout, and optimiser) was adequate for generalisation, even under the increased complexity of the 300-basin setup. While the 300-basin model achieved slightly higher correlation (Pearson-r) and maintained competitive NSE/KGE scores, it also exhibited greater variability across runs, reflecting the challenge of learning from a more diverse set of catchments.

Despite strong convergence patterns, the evaluation metrics highlight that there remains room for improvement. For instance, NSE and KGE values were generally positive but moderate, indicating that while the LSTM captured much of the variability in streamflow, extreme events and fine-scale dynamics were less consistently reproduced. Similarly, RMSE reductions suggest improved predictive accuracy, but not yet at the level of operational hydrological forecasting standards.

In terms of computational cost, both basin subsets were highly demanding, with each full run consuming over 400GB of disk space and requiring careful management of worker processes to avoid memory overload. This constraint limited the total number of configurations explored, reducing opportunities for hyperparameter tuning.

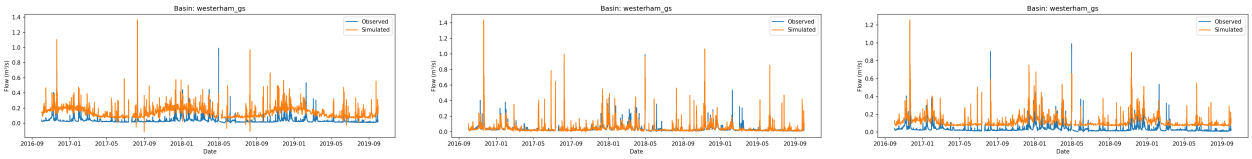
However, if the model were to be deployed in a commercial or operational forecasting context, its performance would need to be substantially improved. While the experiments produced KGE and NSE values in the range of approximately 0.4âĂŞ0.5, these scores remain modest by hydrological standards. A KGE or NSE near 0.5 indicates that the model is only moderately better than simple benchmark models (such as the mean predictor or persistence model) and is not sufficiently reliable for decision-making in real-world water resource management or flood forecasting. For operational use, values closer to 0.7âĂŞ0.8 (or higher) would typically be expected to ensure confidence in predictions, particularly during extreme flow events.

Tying this back to Neural Hydrology, a study using the CAMELS dataset reported median NSE values around 0.6âĂŞ0.7 across basins when using LSTM-based models, with top-performing basins achieving scores above 0.8. In contrast, the Handoff Forecast LSTM applied here achieved NSE and KGE values closer to 0.4âĂŞ0.5, suggesting that while the model was able to learn basin-level dynamics, its performance fell short of the benchmark results reported in the literature. This discrepancy may reflect differences in dataset size, predictor variables, or the absence of extensive hyperparameter tuning, all of which are known to significantly influence LSTM performance in hydrological applications.

4.6 Basin-Level Case Studies

4.6.1 256 Basins

For this section Run 1 was chosen for the 256 basins, this is because the Pearson-r was the highest and most stable furthermore, the RMSE converged with less fluctuations.



Epoch 1 Epoch 19 Epoch 25

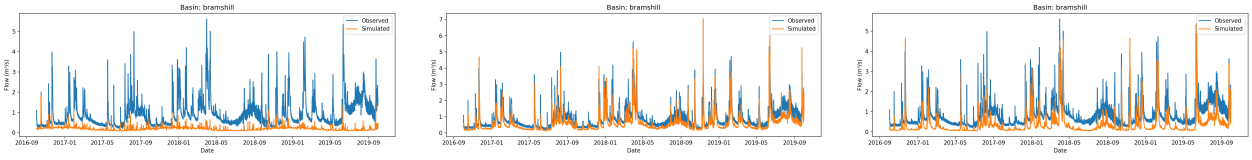
Figure 4.21: Observed vs. simulated streamflow for the Westerham_gs basin at different training epochs.

At epoch 1, the model output is poorly aligned with the observed hydrograph. The simulated flows are systematically biased high, with exaggerated peaks and little correspondence to observed variability. This reflects the modelâĂŹs initialisation stage, where it has not yet learned meaningful hydrological dynamics.

By epoch 19, the predictions show marked improvement. Simulated flows begin to track the observed hydrograph more closely, particularly in the timing of peaks and recession periods. However, peak magnitudes are still overestimated, indicating partial but incomplete learning of catchment-specific runoff processes.

At epoch 25, the model has largely stabilised. The simulated hydrograph follows the observed series more consistently, with improved alignment in both timing and magnitude of flows. While overestimation of extremes remains, the overall representation of baseflow and intermediate events is noticeably more realistic compared to epoch 1.

Visual inspection suggests overfitting after approximately epoch 19: by epoch 25 the simulated series exhibits amplified peaks and a higher bias relative to observations, while validation metrics no longer improve.



Epoch 1 Epoch 19 Epoch 25

Figure 4.22: Observed vs. simulated streamflow for the Bramshill basin at different training epochs.

As we can see with both figures, the model consistently struggles to capture the baseflow across epochs and training periods. While the simulated hydrographs reproduce some of the timing of high-flow peaks, they either overshoot (e.g., epoch 19) or underestimate (epoch 25) the magnitude of events, leading to mismatches during critical flow periods. At epoch 1, the

model performance is particularly poor, with simulated flows failing to align with observed peaks or baseflow levels, reflecting the early stage of training. By epoch 19, the model improves in peak timing but still exaggerates flow magnitudes. At epoch 25, overfitting becomes evident, as the model reproduces certain peaks more closely but diverges from observed baseflow, systematically underpredicting low flows.

This indicates that while the LSTM learned aspects of peak-flow dynamics, it did not generalise well across the full hydrograph, especially for sustaining baseflow conditions. Such misalignment has important hydrological implications, as accurate representation of baseflow is critical for water resource management, drought prediction, and ecosystem sustainability.

4.6.2 300 Basins

After analysing the runs and metrics from the plots and observations from the configurations, the best run within the model is run 3 as it achieved the highest correlation (Pearson-r), the most consistent NSE/KGE values, and the lowest RMSE with fewer fluctuations, showing both better learning and better stability compared to the other runs.

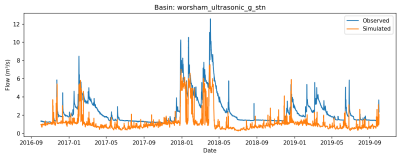


Figure 4.23: Epoch 1

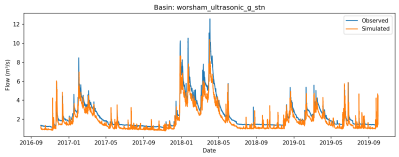


Figure 4.24: Epoch 19

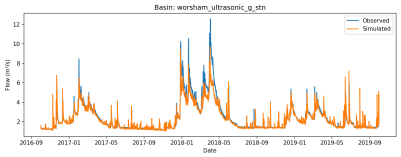


Figure 4.25: Epoch 25

Figure 4.26: Observed vs. simulated streamflow for the Worsham ultrasonic basin at different training epochs.

For the Worsham basin under the 300-basin configuration, Figure 4.23 illustrates the simulated hydrograph after the first epoch. At this stage, the model captures some of the seasonal shape of streamflow, but the simulated flows are poorly aligned with observations, and peak events are either exaggerated or misplaced. This is expected given the minimal training completed.

By epoch 19 (Figure 4.24), the simulations show a marked improvement. The model is able to reproduce both the timing and magnitude of many flow peaks, while also capturing the recession behaviour following events. The baseflow alignment is noticeably better compared to epoch 1, though some peaks remain slightly overestimated. This suggests that the model had effectively learned key basin-level hydrological dynamics by this point.

However, by epoch 25 (Figure 4.25), evidence of overfitting begins to emerge. While many peaks are still reproduced, the simulated flows show exaggerated magnitudes during certain flood events and less consistent baseflow representation. This indicates that extending training beyond epoch 20âĂŞ25 did not yield further generalisation benefits and may instead have reduced robustness across hydrological conditions.

The Worsham basin analysis reinforces earlier findings that the model converges effectively within the first 20 epochs, but excessive training risks overfitting, particularly in basins with more variable hydrological responses.

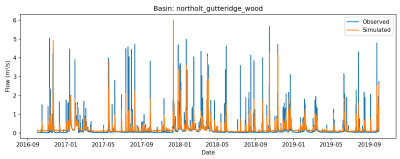


Figure 4.27: Epoch 1

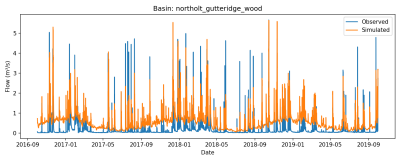


Figure 4.28: Epoch 19

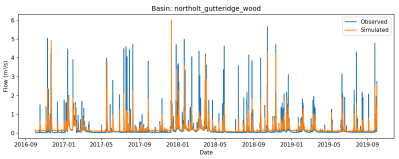


Figure 4.29: Epoch 25

Figure 4.30: Observed vs. simulated streamflow for the Northholt Gutridge basin at different training epochs.

At epoch 1, the model clearly underfits: the simulated series stays close to a flat baseline and fails to capture both the timing and magnitude of peak flows, which makes the predictions essentially unusable. By epoch 19, the performance improves somewhat, with the model able to reproduce more peak events and their approximate timing. However, even at this stage, the simulated flows remain inconsistent âĂŤ some peaks are exaggerated while others are missed entirely. By epoch 25, instead of refining predictions, the model begins to overfit, with simulated flows frequently overshooting both baseflow and flood peaks. This results in systematic biases, with the model showing too much variability compared to the observed record.

Compared to basins such as Bramshill or Worsham, Northolt Gutteridge Wood represents a worse outcome. The model struggles to stabilise at any epoch: underfitting dominates the early stages, while later training introduces excessive variance and over-prediction. This suggests that the basinâĂŹs hydrological behaviour may be harder to learn due to higher variability in flow dynamics or noisier input data, and that the model configuration used here is insufficient to achieve reliable forecasts in this case.

Putting all the results together, from the 256 Basin and the 300 Basin it is definitely noted that the model did not capture the baseflow of the rainfallâĂŞrunoff relationship well. In the future, to counteract this limitation, a cumulative baseflow component could be implemented. This way, the baseflow of the gauge is always captured, ensuring that the

model does not systematically underestimate streamflow during lowâĂŘflow periods. This would help stabilise the lower end of the hydrograph, reducing systematic underestimation during dry periods. And finally, this adjustment would improve the modelâĂŹs reliability across different hydrological regimes, making its outputs more robust for both research and potential operational forecasting applications.

4.7 Additional Observations

In addition to the primary evaluation, several wider observations were made during the experiments. Firstly, variability across basins was clear. While certain catchments such as Worsham demonstrated good alignment between observed and simulated flows in later epochs, others such as Bramshill and Northolt consistently showed difficulties, particularly in representing baseflow and peak extremes. This suggests that basin-specific characteristics, including size, climate, and land use, strongly influence how well the model is able to generalise.

A further observation was the effect of training epochs. Early epochs, such as Epoch 1, consistently underfit, producing overly smooth hydrographs that failed to capture event-scale variability. Later epochs, particularly around Epoch 25, occasionally displayed signs of overfitting, exaggerating peaks relative to observations. The best balance was generally observed between Epochs 15âĂŞ20, where the model captured both peak timing and magnitudes more reliably. This pattern indicates that early stopping could potentially improve generalisation while reducing training time.

The training process also exhibited moderate instability across runs. Metrics such as RMSE and Pearson-r revealed small fluctuations between epochs and runs, which may reflect the stochastic optimisation of LSTMs or the added complexity of training on hundreds of basins. Despite this, convergence trends were broadly consistent, pointing to stable overall learning dynamics.

Finally, the experiments highlighted key limitations in computational cost and hydrological representation. Both the 256 and 300-basin runs required over 400GB of storage per experiment and strict control over the number of workers to avoid exhausting RAM. This restricted the number of hyperparameter configurations that could be tested. From a hydrological standpoint, the model struggled systematically with representing baseflow across basins, often underestimating low flows while focusing on higher flow variability. This limitation suggests that incorporating a cumulative baseflow component or hybridising with physically based constraints could be valuable directions for future work.

4.8 Summary of Results

The experiments conducted with the handoff forecast LSTM model on both the 256-basin and 300-basin datasets provided valuable insights into its ability to generalise across diverse hydrological settings. The results can be summarised as follows:

For the **256-basin experiments**, the model demonstrated steady convergence across all three runs. Training and validation losses decreased consistently, stabilising after approximately 20 epochs, with minimal evidence of severe overfitting. Performance metrics indicated moderate skill: NSE and KGE values stabilised around 0.4 to 0.5, RMSE declined from initial values near 0.7âĂŞ-0.75 to approximately 0.45âĂŞ-0.50, and Pearson-r values confirmed strong correlation between observed and simulated flows. These findings suggest that the model successfully captured broad hydrological dynamics, but its ability to reproduce finer-scale processes âĂŤ particularly baseflow and peak flows âĂŤ was limited.

In the **300-basin experiments**, the model exhibited similar convergence behaviour but with greater variability between runs. Validation losses plateaued earlier compared to the 256-basin setup, suggesting that the increased diversity of basins introduced additional complexity. Pearson-r values were slightly higher, reflecting improved alignment of temporal dynamics, but NSE and KGE scores remained at comparable levels to the 256-basin results. This indicates that while the larger dataset provided more training information, it also posed greater challenges in achieving consistently strong generalisation across all catchments.

Hydrograph comparisons highlighted a recurring weakness across both basin subsets: the model systematically underestimated baseflow and often struggled to align with peak flow magnitudes and timings. While the model captured the overall patterns of streamflow variability, these deficiencies reduce its reliability in operational forecasting, particularly in contexts where accurate low-flow and flood predictions are critical.

From a computational perspective, both experiments were highly demanding. Each full run consumed over 400GB of disk space, and limitations in available memory required worker processes to be set at 0 or 1. These constraints restricted the number of configurations that could be tested, with only three out of six specified configurations successfully executed. Training times varied between experiments, with the 256-basin model requiring approximately 18.5 hours and the 300-basin model 24.5 hours for 25 epochs.

In summary, the results demonstrate that the handoff forecast LSTM achieved moderate but consistent performance across both basin subsets, avoiding severe overfitting while generalising reasonably well. However, the moderate NSE and KGE values (0.4 to 0.5)

suggest that significant improvements are needed before the model could be used in an operational setting. In particular, enhancements in baseflow representation, peak event modelling, and overall predictive accuracy will be required to meet the higher performance standards expected in hydrological forecasting applications.

4.8.1 A Final Remark

Note: Due to the model being highly data intensive, the run tested within this project exhausted available storage, which is why training concluded prematurely at epoch 22 instead of the planned 25.

Chapter 5

Discussion

5.1 Restating Aim and Key Findings

The aim of this study was to evaluate the Handoff Forecast LSTM, implemented via the Neural Hydrology package, for rainfall–runoff prediction across UK basins. Two experiments were performed using subsets of 256 and 300 basins, with the objective of assessing model skill under increasing spatial scale and catchment diversity.

The results demonstrated that training and validation losses converged steadily across epochs, with no strong evidence of overfitting. Evaluation metrics, including NSE, KGE, RMSE, and Pearson-r, indicated moderate predictive accuracy. The model captured much of the flow variability but struggled to reproduce baseflow and extreme flows consistently. Comparing the two experiments, the 300-basin setup introduced greater variability across runs but also achieved slightly higher correlation scores, suggesting improved ability to capture broader hydrological dynamics at the expense of stability.

In addition, the experiments showed that the model was able to learn quickly during the first 10âĂŞ15 epochs, with sharp improvements in performance, before reaching a plateau around epoch 20. This suggests that the chosen configuration was sufficient to stabilise training, but further epochs did not yield major gains. This pattern was consistent across multiple runs, demonstrating reproducibility of training behaviour even if performance levels were modest.

5.2 Interpretation of Results

The training and validation loss curves confirmed that the LSTM architecture was able to learn meaningful hydrological representations without severe divergence between training and validation performance. This reflects a reasonable degree of generalisation, even under more complex conditions in the 300-basin experiment. The relatively close alignment of the training and validation curves suggests that dropout and batch size were appropriate choices to prevent overfitting.

However, the moderate scores in NSE and KGE highlight clear limitations: while the model reproduced overall flow variability, it frequently failed to align baseflow and peak flows with observations. The basin comparisons confirmed this behaviour, with examples showing systematic underestimation of baseflow and timing mismatches in peak events. These patterns suggest that although the network extracted basin-level signals, it lacked robustness in capturing long-term water balance and high-magnitude extremes.

It is also important to note that the variability across runs was higher in the 300-basin experiment. This indicates that while training on more diverse catchments improved correlation (Pearson-r), the model struggled to balance the competing hydrological signals across basins, leading to reduced stability. In practice, this means that increasing the training set size can be beneficial, but it also increases the risk of uneven performance unless carefully regularised or constrained.

5.3 Comparison with Existing Literature

Previous work with the CAMELS dataset and other LSTM-based rainfall–runoff studies has reported higher NSE and KGE values, often in the range of 0.6–0.8. In comparison, the results in this study (typically 0.4–0.5) are noticeably lower. This discrepancy may stem from differences in data quality, preprocessing, or the larger heterogeneity of UK basins compared to CAMELS-US. For example, CAMELS-US contains long, high-quality daily records and relatively homogeneous catchments, while the UK dataset used here involved shorter records, 15-minute intervals, and a wider variety of catchment descriptors.

Despite the lower absolute performance, the trends observed here align with the literature in key ways: rapid early learning during initial epochs, diminishing improvements after around 20 epochs, and improved correlation at larger basin scales. The relatively modest scores highlight the difficulty of transferring LSTM approaches directly to different hydrological

contexts without substantial adaptation. Similar observations have been made in studies applying neural networks to CAMELS-GB, where performance tends to lag behind CAMELS-US, reflecting differences in hydrological regimes and data consistency.

Another difference is methodological: many CAMELS studies have used daily timesteps, which smooth short-term variability and can lead to higher performance metrics. In contrast, the use of 15-minute timesteps in this study made the prediction task more demanding, as the model had to reproduce both rapid storm responses and longer-term baseflow patterns. This likely contributed to lower NSE and KGE scores but provides a more challenging and realistic assessment of model capability.

5.4 Limitations

A major limitation was the computational expense of the experiments. Each run required more than 400GB of disk space, limiting the number of configurations to three per experiment. Storage and memory constraints also forced the number of workers to be set at 0 or 1, reducing computational efficiency. These restrictions hindered broader hyperparameter exploration and reduced opportunities for optimisation.

Another limitation relates to model accuracy. Predictive skill was only moderate, with NSE and KGE values insufficient for operational forecasting. The inability to consistently reproduce baseflow and extremes further constrained model reliability. For instance, basin-level plots showed clear systematic underestimation of baseflow across multiple epochs, while high-flow peaks were often missed or poorly timed.

5.5 Future Work and Implications

Future research should focus on improving baseflow representation, potentially by incorporating cumulative baseflow constraints or hybrid physics-informed approaches. Expanding hyperparameter searches, including larger hidden sizes, alternative loss functions (e.g., combined NSE-RMSE objectives), and more varied dropout schemes, may also improve accuracy.

Another promising direction would be physics-guided neural networks, which embed hydrological process constraints into the model architecture. These have been shown in

recent studies to improve both interpretability and reliability of predictions, particularly for extreme flows. Ensemble approaches could also be explored, combining multiple runs to smooth out variability and improve robustness.

The implications of this work extend beyond the specific experiments. While the Handoff Forecast LSTM showed only moderate skill, it demonstrated the feasibility of large-scale, multi-basin training in a UK context. Improving such models has the potential to transform hydrological forecasting, offering more adaptive and data-driven alternatives to traditional conceptual models. With sufficient refinement, these approaches could support more reliable flood forecasting, water resource management, and climate resilience planning.

Another promising direction for future work is the exploration of emerging state space models, such as **Mamba**. Unlike traditional recurrent networks such as LSTMs, Mamba is designed to efficiently capture long-range dependencies by combining the scalability of convolutional approaches with the expressiveness of recurrent dynamics. This makes it particularly attractive for hydrological forecasting, where processes such as groundwater flow, catchment memory, and delayed runoff inherently depend on long-term temporal dependencies.

In this project, the Handoff Forecast LSTM struggled to fully reproduce baseflow and peak flows, reflecting limitations in its ability to capture extended temporal dynamics. Mamba, by contrast, has the potential to overcome this by providing a more efficient mechanism for learning dependencies over hundreds of time steps without suffering from vanishing gradients or excessive computational cost. Moreover, MambaâĂŹs efficiency in training and inference could reduce the extreme storage and runtime requirements encountered during this study, making it a practical alternative for large multi-basin experiments.

Furthermore, another way to extend this project would be to group the regional catchments based on their geographic location within England (e.g., North, South, East, and West). This regionalisation approach could reduce heterogeneity across basins, as catchments in closer proximity often share similar climatic conditions, geological characteristics, and hydrological responses. Training models on these regional subsets may therefore allow for improved predictive skill, particularly in capturing region-specific rainfall–runoff dynamics.

This approach would reveal whether the model architecture struggles more in certain regions (e.g., lowland versus upland catchments, wetter versus drier climates). In practice, regional grouping could strike a balance between fully local models and a single nationwide model, improving generalisation while still retaining sensitivity to regional hydrological behaviour.

Finally, it is worth emphasising that this study contributes to the growing evidence base for deep learning in hydrology outside of the CAMELS-US dataset. Even though the

scores were lower, the fact that meaningful performance was achieved across 300 UK basins highlights the adaptability of LSTMs to different hydrological contexts. This paves the way for future research into regionalisation, transfer learning, and hybrid models that blend physical hydrology with machine learning to address current shortcomings.

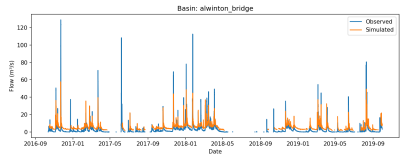
5.6 Closing Statements

After completing this project, it has been clearly demonstrated that the Handoff Forecast LSTM holds significant potential as a data-driven tool for rainfallâĂŞrunoff prediction. While the model did not achieve performance levels sufficient for operational deployment, its ability to learn basin-level hydrological dynamics across hundreds of UK catchments shows the promise of deep learning methods in this field. The experiments highlighted both the strengths of the approach âĂŤ such as rapid early convergence and stable generalisation across diverse basins âĂŤ and the limitations, including challenges in reproducing baseflow and extremes, as well as the substantial computational demands.

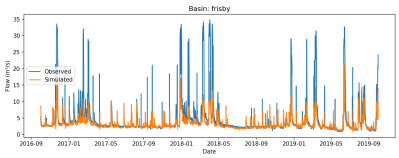
This understanding reinforces the importance of continued development, particularly in the areas of physics-informed architectures, improved baseflow modelling, and broader hyperparameter optimisation. The project thus contributes not only to evaluating Neural Hydrology within a UK context but also to laying the groundwork for future improvements that may bring such models closer to operational forecasting standards.

Appendix A

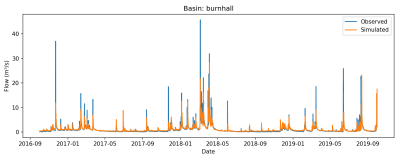
Appendix



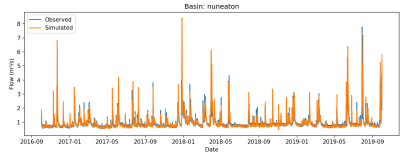
Plot 1



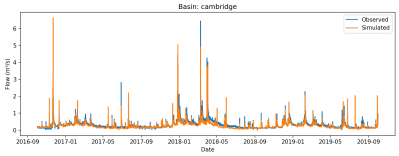
Plot 2



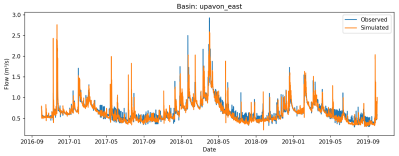
Plot 3



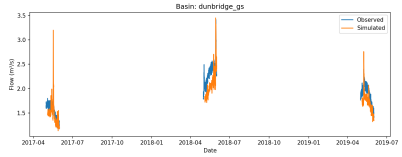
Plot 4



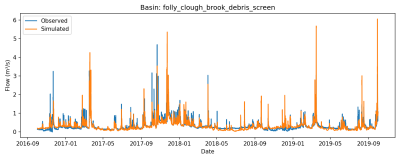
Plot 5



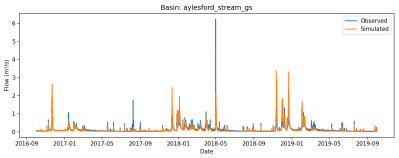
Plot 6



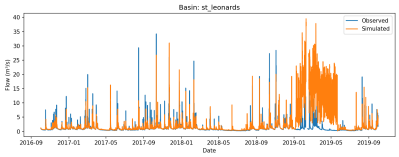
Plot 7



Plot 8

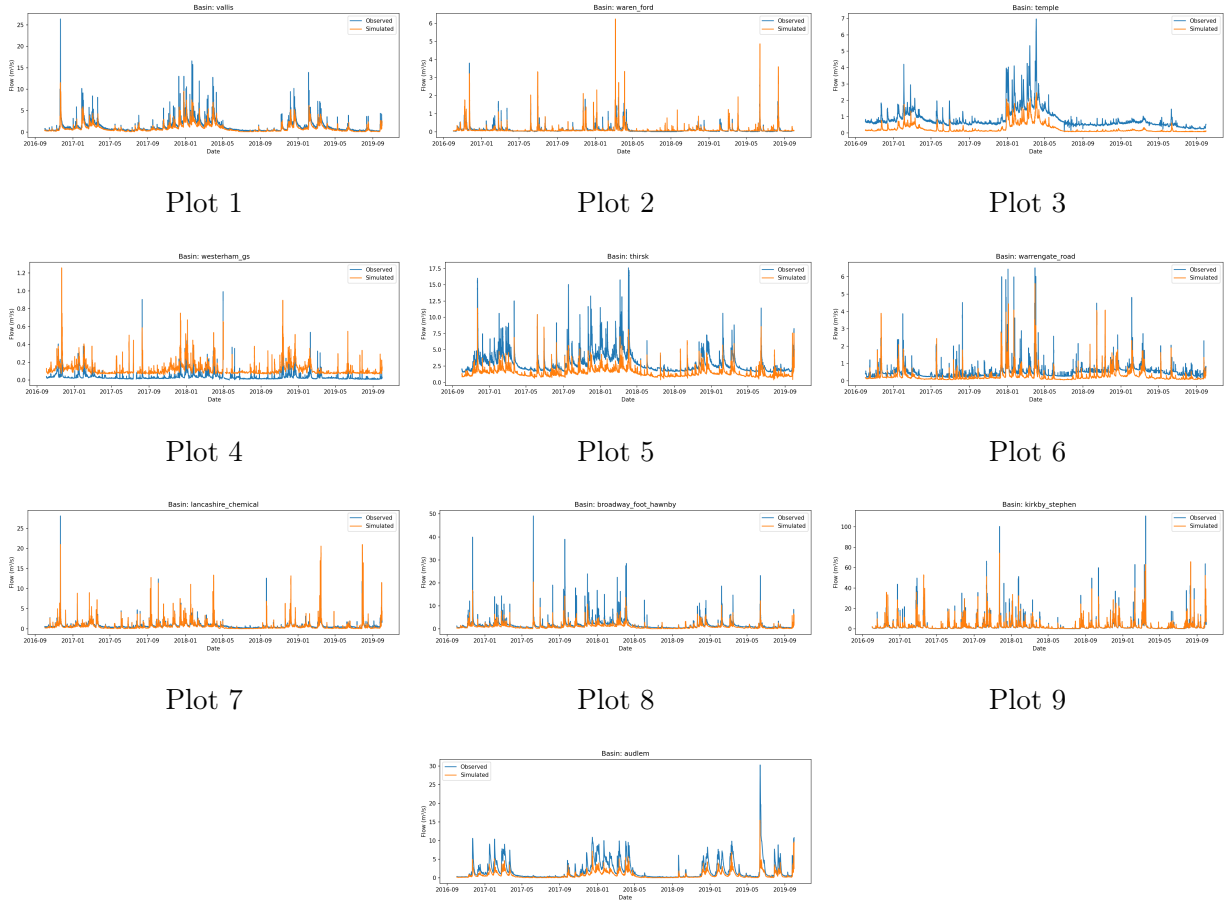


Plot 9



Plot 10

Figure A.1: These are ten plots from the Model with 300 basins (run 3). This is from the test file.



Plot 1 Plot 2 Plot 3

Plot 4 Plot 5 Plot 6

Plot 7 Plot 8 Plot 9

Plot 10

Figure A.2: These are ten plots from the Model with 256 basins (run 1). This is from the test file.

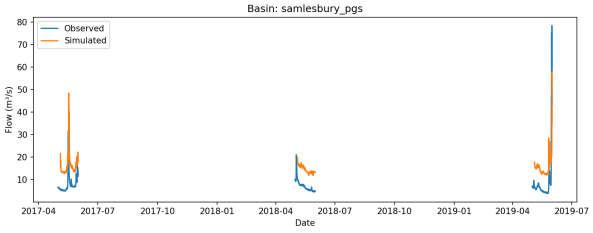


Figure A.3: Plot with missing data (Samlesbury_pgs)

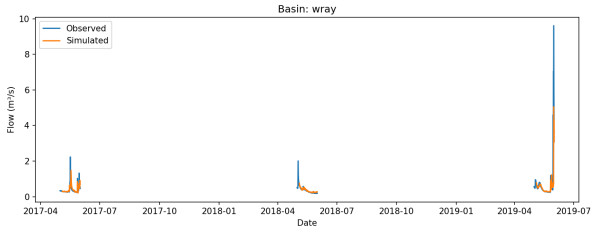


Figure A.4: Plot with missing data (Wray)