Fusion Journal of Engineering and Sciences

Article

Improving Reservoir Water Supply and Inflow Volume Predictions with Encoder-Decoder Deep Learning Models

Zeeshan Asghar [1]* and Muhammad Waseem [1]

[1] Department of Civil Engineering, Ghulam Ishaq Khan Institute of Engineering Sciences and Technology, Topi 23460, Pakistan; *zeeshan.asghar@giki.edu.pk*; *muhammad.waseem@giki.edu.pk*
* Correspondence: *zeeshan.asghar@giki.edu.pk*

Abstract

Recently, deep learning (DL) models have shown tremendous potential for hydrological prediction, reservoir management, and operational planning. However, their effectiveness in predicting reservoir inflows over extended time horizons remains limited. Recent advancements in DL algorithms have improved the accuracy of inflow forecasts, yet most studies emphasize short-term applications or real-time operations. This study proposes a novel multi-step forecasting framework to enhance long-term predictions of reservoir inflow and water supply. Using snow-water equivalent (SWE) and historical inflow data, we trained a DL model built on a convolutional neural network (CNN)–long short-term memory (LSTM) encoder–decoder architecture to forecast inflows during the critical March–August runoff period. Model architecture and hyper-parameters were tuned via multi-fold cross-validation of the time series, examining various CNN- and LSTM-based encoder–decoder adaptations. The methodology was applied to 40 years of SWE and inflow data from Jordanelle Reservoir, Utah. The optimal configuration—an LSTM encoder–decoder with 16 nodes per layer—achieved substantial improvements in long-term forecast accuracy. We also assessed the trade-off between model complexity and performance by benchmarking against a process-driven ensemble streamflow prediction (ESP) model and classical statistical methods (SARIMA, VAR, TBATS). The DL approach outperformed the statistical models for long-term water-supply forecasts and achieved accuracy comparable to the ESP model's 50 % exceedance-probability forecast. Overall, these results highlight the promise of advanced DL methods for enhancing long-term hydrological forecasting and water-resource management.

Keywords: Deep Learning; LSTM; CNN; Reservoir; Inflow; Streamflow; Prediction

1. Introduction

Conventional statistical techniques—including Vector Auto-Regression (VAR), Seasonal Auto-Regressive Integrated Moving Average (SARIMA), the Trigonometric Box-Cox transformation approach (1), and the TBATS method that blends Box-Cox, ARMA errors and seasonal components (?)—have long been used to predict reservoir-inflow and -outflow patterns. These models leverage historical data and inherent seasonal trends to elucidate water-resource dynamics. Although effective for short-term forecasts, they struggle with long-range predictions because auto-regressive components tend to converge on the series mean (4). Long-term hydrological forecasts are further hampered by hydro-meteorological variability—especially in snow-dominated catchments (5)—and by nonlinear, weather-driven patterns that violate the linear assumptions of most statistical models.

https://doi.org/xxxxxx/xxxxx

Physical hydrological models, by contrast, simulate watershed processes with probabilistic ensembles driven by atmospheric inputs, then apply statistical post-processing to quantify uncertainty (6, 7). A leading example is the Ensemble Streamflow Prediction (ESP) framework (8), widely used for long-term water-supply forecasting. ESP initializes model states from current basin conditions and produces probabilistic inflow predictions; however, it does not ingest observed snow-water equivalent (SWE) directly, relying instead on snowpack dynamics inferred from temperature and precipitation inputs (9). When snowpack variability is large, ESP accuracy can decline (10).

These limitations underscore the need for advanced methods that capture nonlinear relationships and account for hydrological variability. Machine-learning approaches provide a data-driven alternative and have shown success in rainfall–runoff forecasting (11), hydropower prediction (12), and spatial SWE estimation (13), all without explicit process assumptions. Deep-learning (DL) models have pushed this further: encoder–decoder architectures can learn temporal dependencies and nonlinear interactions, making them well suited to multi-step forecasting (14). Yet most DL studies still emphasize short-term horizons.

To bridge this gap, we develop a multi-step forecasting scheme that trains encoder–decoder networks on historical SWE and reservoir-inflow data to produce long-term predictions. Four architectures are examined—CNN-LSTM, residual CNN-LSTM, vanilla LSTM-LSTM, and residual LSTM-LSTM—each combining convolutional (for parallel extraction of fixed-window features) and recurrent layers (for sequential dependencies). Residual connections are included in some variants to improve gradient flow and mitigate vanishing-gradient issues (15, 16).

These DL models are benchmarked against ESP and the classical statistical approaches (SARIMA, VAR, TBATS) to evaluate long-range performance. The study addresses three questions:

- Complexity vs. accuracy – What model complexity yields the best long-term water-supply forecasts?
- Conditions for DL superiority – Under what hydrological or data conditions does the proposed DL model outperform benchmarks?
- Data-driven vs. process-based – How do purely data-driven methods compare with process-based physical models for multi-month reservoir-inflow prediction?

2. Materials and Methods

2.1 Site

This study focuses on the Jordanelle Reservoir, a key component of the Provo River Project and an essential water storage facility for central Utah. The reservoir is situated in Wasatch County, Utah, along the Provo River, upstream of Heber City. Its watershed encompasses a drainage area of approximately 234 square miles, monitored at the Provo River near Hailstone gauging station (10155000). The region receives an average annual precipitation of 25.8 inches, with runoff primarily derived from spring snowmelt, contributing significantly to its inflow from April through June.

Completed in 1993 by the Bureau of Reclamation, Jordanelle Reservoir has a total storage capacity of 320,300 acre-feet, covering a surface area of up to 3,068 acres at full pool. The reservoir is a vital part of Utah's water management system, providing water for municipal, industrial, and agricultural use while supporting recreational activities. Water releases are regulated through the Jordanelle Dam, which incorporates outlet works capable of discharging up to 8,000 cubic feet per second (cfs) . Additionally, the reservoir serves as a crucial buffer for managing downstream flow into the Deer Creek Reservoir and mitigating flood risks in the Provo River basin .

Managed by the Central Utah Water Conservancy District (CUWCD), Jordanelle Reservoir forms part of a broader water infrastructure network designed to meet the growing demands of

Utah's population. The reservoir plays an integral role in water storage and delivery under the Central Utah Project, ensuring the sustainable allocation of resources across the Wasatch Front. Figure 1 highlights the geographic location of Jordanelle Reservoir and its proximity to snow telemetry (SNOTEL) sites, crucial for tracking snowpack dynamics and forecasting runoff for operational planning.

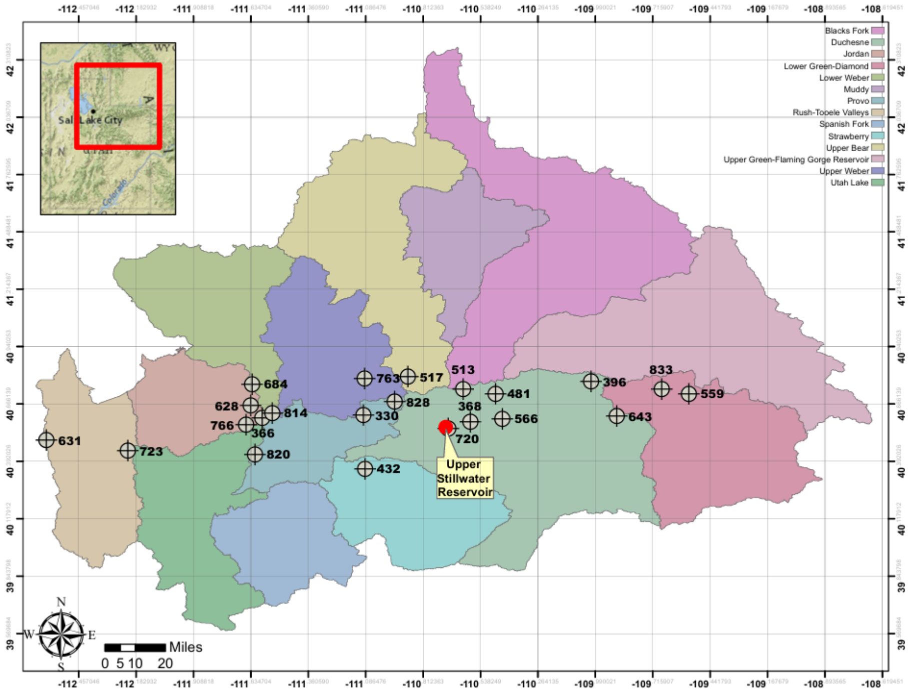

Figure 1: Study site location: Jordanelle Reservoir, Utah

2.2 Datasets

The dataset utilized for training the model originated from the operational archives of the Central Utah Water Conservancy District (CUWCD) and the snow telemetry (SNOTEL) network, which is managed by the National Resource Conservation Service (NRCS) . Reservoir inflow data were computed using changes in reservoir water levels, adjusted to account for losses such as seepage and evaporation, and incorporating releases into Rock Creek. This methodology facilitated precise calculations of storage variations in acre-feet .

Two key datasets formed the foundation of this study: reservoir inflow and snow-water equivalent (SWE). SWE represents the amount of water (in inches) that would result from melting the snowpack. Historical daily records for Rock Creek at the Jordanelle Reservoir span from January 1990 to the present, with regular updates. The CUWCD ensures high-quality, gap-free reservoir inflow data, eliminating the need for interpolation (Figure 2).

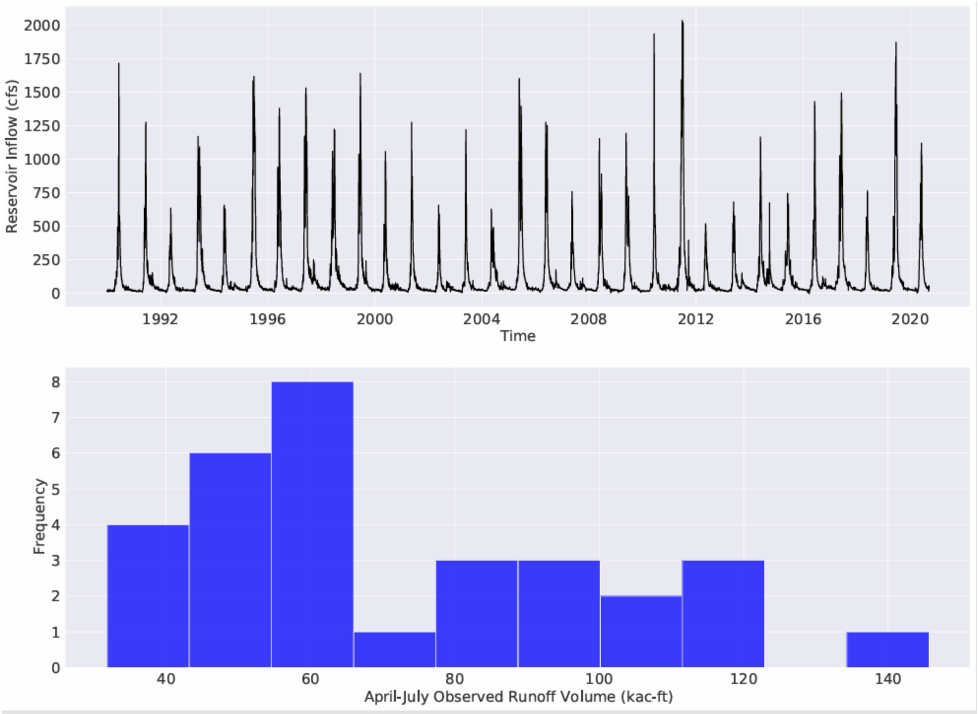

Figure 2: Historical daily inflow patterns (top) and March-August inflow volumes (bottom) at Jordanelle

Reservoir inflow patterns exhibit a seasonal runoff trend, predominantly driven by snowmelt, which typically peaks between April and August. This study assumes that critical water storage decisions must be finalized by late March to prepare for the snowmelt season. Therefore, the forecasting period from March to August was emphasized, necessitating a model that can accurately represent long-term temporal trends .

The SWE data were obtained from the NRCS SNOTEL network, covering the same timeframe as the reservoir inflow data (Table 1).

Table1: Metadata for Snow Telemetry Stations (NRCS)

St- Id	Location	Lon	Lat
340:UTa:SNoTL	Breavers Devide	-109.987	39.206
346:UTa:SNoTL	Laughton	-110.466	39.193
358:UTa:SNoTL	Black Bear	-109.480	39.175
356:UTa:SNoTL	Chepeta Way	-108.910	39.367
462:UTa:SNoTL	Current River	-109.979	39.953
461:UTa:SNoTL	Five Guys Creek	-109.362	39.311
577:UTa:SNoTL	Hayden Fork	-109.776	39.389
579:UTa:SNoTL	Kings Cabin	-108.449	39.309
586:UTa:SNoTL	Riverfork #1	-109.329	39.191
583:UTa:SNoTL	Riverfork Basin	-109.515	39.331
698:UTa:SNoTL	Miller-D North	-110.520	39.252
661:UTa:SNoTL	Dining Forks	-111.485	39.089
633:UTa:SNoTL	Mosby Mountain	-108.789	39.202
624:UTa:SNoTL	Parleys Way	-110.513	39.354

St- Id	Location	Lon	Lat
750:UTa:SNoTL	Red Rock Canyon	-109.586	39.144
783:UTa:SoNTL	R.B Settlement	-111.102	39.999
713:UTa:SNoTL	S. Moreshouse	-109.981	39.381
794:UTa:SNoTL	Thanos Canyon	-110.418	39.218
820:UTa:SNoTL	Gt. T. Divide	-110.500	39.024
828:UTa:SNoTL	Trivial Lake	-109.840	39.271
833:UTa:SNoTL	Salmon River	-108.576	39.332

The monitoring stations provided high-quality data, with fewer than three days of missing values per site. Where gaps existed, they were addressed using interpolation techniques. The primary objective of the analysis was to model the relationship between SWE (Figure 3) and reservoir inflow (Figure 2).

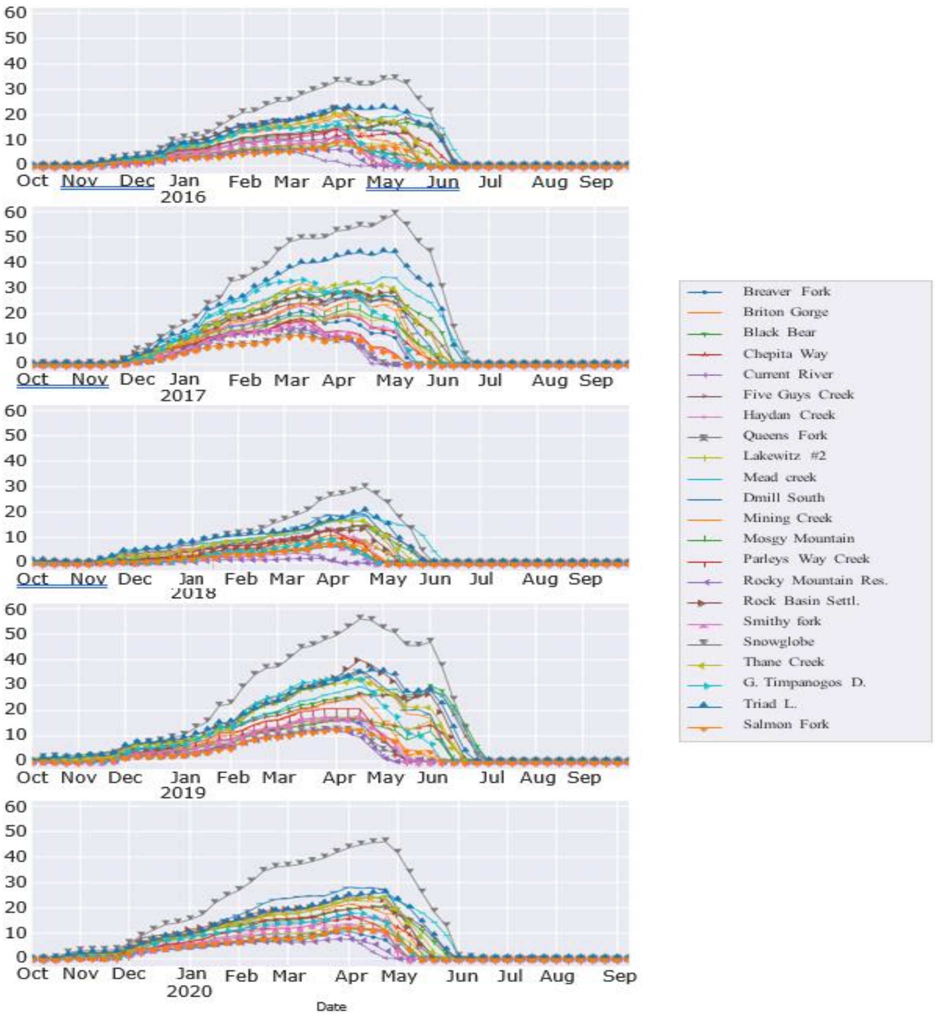

Figure 3: Snow water equivalent (SWE) measurements (in inches) over five continuous water years.

For model training, the daily data were aggregated into weekly averages and scaled to a normalized range of 0 to 1 to align with the deep learning model's activation function (Section 2.3). A sliding window approach was adopted, utilizing a 20-week input period to forecast the subsequent 26 weeks, corresponding to the runoff season (March to August). The 20-week window was selected based on typical SWE accumulation durations, which vary from 15 to 25 weeks depending on winter precipitation patterns. This configuration reflects the most representative conditions across multiple years, providing an optimal basis for training the model.

2.3 Encoder-Decoder DL Model

The primary aim of the model is to forecast future reservoir inflow sequences based on historical inputs, incorporating time-series data for reservoir inflow and SWE. This is achieved through a multivariate sequence-to-sequence prediction framework, consisting of two main components: an encoder that converts the input sequence into a fixed-length representation, and a decoder that reconstructs this representation into the predicted output sequence . The decoder is further supplemented by a fully connected time-distributed layer, which refines the predictions into the final output sequence.

As depicted in Figure 4, the model incorporates four different variants, each utilizing a sliding window approach where multiple time-series variables are processed to produce corresponding output windows.

This architecture has proven effective for a variety of sequence-to-sequence applications, including flood prediction , traffic flow estimation , weather forecasting , and solar performance modeling . The implementation of each model variant was carried out using Python and the Keras library . The exponential linear unit (ELU) activation function   was employed to enhance learning performance, while the Adam optimizer   was utilized for efficient weight optimization across the network.

2.4 Encoder-Decoder Variants

Recurrent neural networks (RNNs) are well-suited for sequential data processing but often face challenges when input-output relationships span extended time gaps, leading to difficulties in handling long-term dependencies. Long short-term memory (LSTM) networks address this limitation through a cell state and gated mechanisms that efficiently manage information retention and removal across time steps .

The Encoder-Decoder framework offers significant advantages for sequence-to-sequence tasks by encoding variable-length sequences into fixed-length vectors, ensuring adaptability across diverse datasets . Using LSTMs within this framework enhances its ability to capture long-term dependencies, overcoming the limitations of traditional RNNs . The framework’s modularity supports a wide range of architectures, including CNN-based encoders for feature extraction and LSTM decoders for temporal sequence modeling . Residual connections improve gradient flow and facilitate efficient training of deeper networks , while causal padding in CNN encoders ensures temporal integrity by preventing look-ahead bias .

Figure 4 illustrates four Encoder-Decoder model variants, each incorporating varying degrees of architectural complexity:

1. **LSTM Encoder-LSTM Decoder**: This variant integrates an LSTM-based encoder and decoder, linked through a repeat vector to align the encoded features with the decoder’s input. The decoder reconstructs the output sequence, which is refined by a time-distributed Dense layer. This layer applies shared weights across time steps, ensuring consistent output generation.

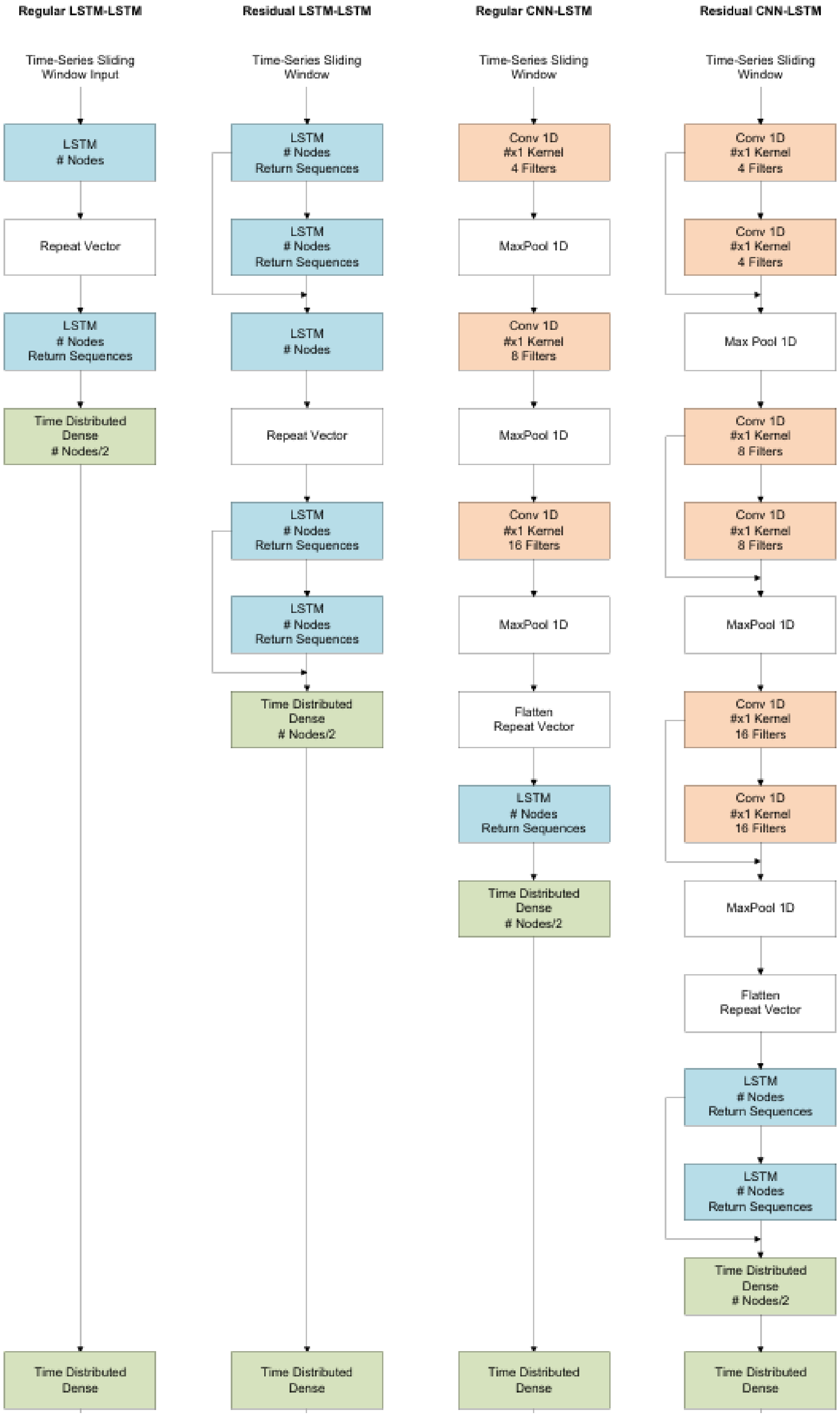
Regular LSTM-LSTM
Residual LSTM-LSTM
Regular CNN-LSTM
Residual CNN-LSTM
Time-Series Sliding Window Input
Time-Series Sliding Window
LSTM # Nodes
LSTM # Nodes Return Sequences
Repeat Vector
Time Distributed Dense # Nodes/2
Conv 1D #x1 Kernel 4 Filters
Conv 1D #x1 Kernel 8 Filters
Conv 1D #x1 Kernel 16 Filters
MaxPool 1D
Max Pool 1D
Flatten Repeat Vector
Time Distributed Dense

Figure 4: Architecture of Encoder-Decoder variants. Colored cells represent layers with trainable parameters, while non-colored cells indicate non-trainable layers, such as repeat vector and max pooling operations.

2. **Residual LSTM Encoder-LSTM Decoder**: An extension of the LSTM-LSTM model, this variant incorporates residual connections between layers to enhance gradient flow and error

propagation. These connections support the network in maintaining greater depth without compromising optimization efficiency .

3. **CNN Encoder-LSTM Decoder**: This model replaces the LSTM encoder with a one-dimensional convolutional neural network (CNN) that processes time-series data. The CNN captures local patterns through hierarchical representations, while causal padding preserves temporal order, preventing any look-ahead bias .

4. **Residual CNN Encoder-LSTM Decoder**: This architecture builds on the CNN-LSTM variant by incorporating residual connections within the CNN-based encoder. These connections improve feature extraction and gradient flow, enabling the model to learn hierarchical abstractions effectively. Pooling layers at the end of each residual block condense the input features into meaningful representations.

In the CNN-based models, the encoder progressively reduces the input matrix's dimensions while abstracting relevant features. Convolutional filters and pooling layers create a condensed feature map, with max pooling applied to retain key elements. The resulting features are flattened and passed to the LSTM decoder, which processes the sequence to generate the output.

This combination of LSTM and CNN architectures leverages their respective strengths, enabling the model to identify both localized patterns and long-term dependencies within the input data, ensuring a robust and flexible predictive framework.

2.5 Model Selection

The historical dataset, spanning forty years for inflow and snow water equivalent, is expanded to enhance the model's ability to generalize to unseen scenarios. To achieve this, the index sequential method (ISM) is utilized, a technique commonly applied in hydrological modeling, particularly in the Colorado River Basin . ISM generates synthetic hydrological sequences by incrementally shifting the historical record by one water year, broadening the range of potential outcomes and accounting for uncertainties in future hydrological behavior due to natural variability and human-induced climate changes .

To address ISM's limitations in capturing extreme events, such as extended droughts, this study implements a modified approach. Water years are treated as discrete blocks and randomly shuffled to introduce additional variability. This method, referred to as water year block disaggregation, diversifies the training dataset, improving the model's robustness . Additionally, each water year is randomly scaled within a range of 0.4 to 1.4, preserving seasonal trends while introducing magnitude variability. This adjustment allows the model to train on a wider spectrum of hydrological extremes, expanding the training dataset fivefold before cross-validation.

Hyper-parameter tuning for each model was performed using five-fold time-series cross-validation. Data from 2011–2015 are utilized for hyperparameter optimization, while the evaluation period from 2016–2020 is used to identify the best-performing architecture. Each Encoder-Decoder variant was tested with configurations of 8, 16, and 32 LSTM nodes, and CNN kernel sizes of 2, 4, and 8, resulting in a total of 12 configurations. Residual connections are incorporated to enhance gradient flow, and a Dense layer with time-distributed functionality was utilized to process the decoder's output efficiently.

The training process connects layers of nodes, where each node processes input data and forwards it using an activation function. In CNN-based models, kernels slide over input sequences to extract features, while filters consolidate these features into maps. The input data comprise SWE and reservoir inflow time-series from November to March, while the target output predicts runoff for March to August.

To prevent overfitting during training, an early stopping mechanism is implemented, which halts the training process if no reduction in mean squared error (MSE) is observed over 10 consecutive epochs. This approach helps avoid overfitting by preventing the model from excessively adapting to the training data. Additionally, the total number of epochs is capped at 50, ensuring a controlled training duration. The batch size is carefully selected to determine the fraction of data processed in each training iteration, balancing computational efficiency and model convergence.

The evaluation of model performance is carried out using a comprehensive set of metrics to ensure a robust assessment. These metrics include normalized root mean squared error, mean absolute error, Nash-Sutcliffe efficiency, median absolute error, and explained variance. Each metric provides unique insights into different aspects of the model's predictive capabilities, enabling a thorough evaluation of accuracy, consistency, and reliability. This multifaceted evaluation framework ensures that the model's performance is not only accurate but also generalizable to unseen data.

$$MAE = \frac{\sum_{j=1}^{M} |\,(Q_{obs,j} - Q_{pred,j})|}{\sum_{j=1}^{M}(\,Q_{obs,j})}$$

$$RMSE = \frac{\sqrt{\frac{1}{M}\sum_{j=1}^{M}(\,Q_{obs,j} - Q_{pred,j})^2}}{\frac{1}{M}\sum_{j=1}^{M}(\,Q_{obs,j})}$$

$$MedAE = \frac{\mathrm{Med}[|(Q_{obs} - Q_{pred})|]}{\frac{1}{M}\sum_{j=1}^{M}(\,Q_{obs,j})}$$

$$N.SE = 1 - \frac{\sum_{j=1}^{M}(\,Q_{obs,j} - Q_{pred,j})^2}{\sum_{j=1}^{M}(\,Q_{obs,j} - \frac{1}{M}\sum_{k=1}^{M}(\,Q_{obs,k}))^2}$$

$$ExpVar = 1 - \frac{\sum_{j=1}^{M}(\,(Q_{obs,j} - Q_{pred,j}) - \frac{1}{M}\sum_{k=1}^{M}(\,Q_{obs,k}))^2}{\sum_{j=1}^{M}(\,Q_{obs,j} - \frac{1}{M}\sum_{k=1}^{M}(\,Q_{obs,k}))^2}$$

The dataset spans 46 weeks, divided into a 20-week input window (November to March) and a 26-week output window (March to August). Model performance is ranked based on the average metric scores over the five-year evaluation period, ensuring robust and reliable predictions.

2.6 Quantification of Uncertainty

This study investigates uncertainty in two key dimensions: the predictive accuracy of the model and the intrinsic variability in reservoir inflow patterns, particularly during weekly averages and the critical March-to-August runoff period. Predictive uncertainty is evaluated through a 95% confidence interval derived from an ensemble of model simulations. Given the stochastic nature of deep learning models, each training iteration introduces slight variations in the forecast outcomes. To capture this variability, the model is trained multiple times, generating a distribution of predictions for each forecast time step. This distribution is assumed to follow a normal distribution, consistent with the central limit theorem as the number of ensemble members increases.

Outliers within the forecasted results are identified using the Tukey method, which flags data points lying beyond the whiskers in a boxplot representation . This approach ensures that anomalous predictions, which could adversely impact model interpretation, are systematically recognized and addressed.

Additionally, the study emphasizes the importance of understanding inflow variability, which is critical for effective water resource planning and management. The variability in inflow patterns is influenced by complex hydro-meteorological factors, including snowmelt timing, precipitation intensity, and temperature fluctuations. To provide a clear visualization of this variability, exceedance probability plots are employed. These plots illustrate the likelihood of inflows exceeding specific thresholds over the forecast horizon, offering valuable insights into potential risks and resource allocation needs during peak and low inflow periods.

By combining predictive uncertainty analysis with a focus on inflow variability, the study provides a comprehensive framework to evaluate model reliability and inform decision-making in the context of reservoir management. This dual approach not only enhances the robustness of the forecasting methodology but also supports adaptive strategies to mitigate the impact of uncertain and dynamic hydrological conditions.

The second component of uncertainty concerns the variability in reservoir inflow, which serves as the primary forecast target. This variability is visualized using an exceedance probability plot for the multi-step forecast period (Figure 5).

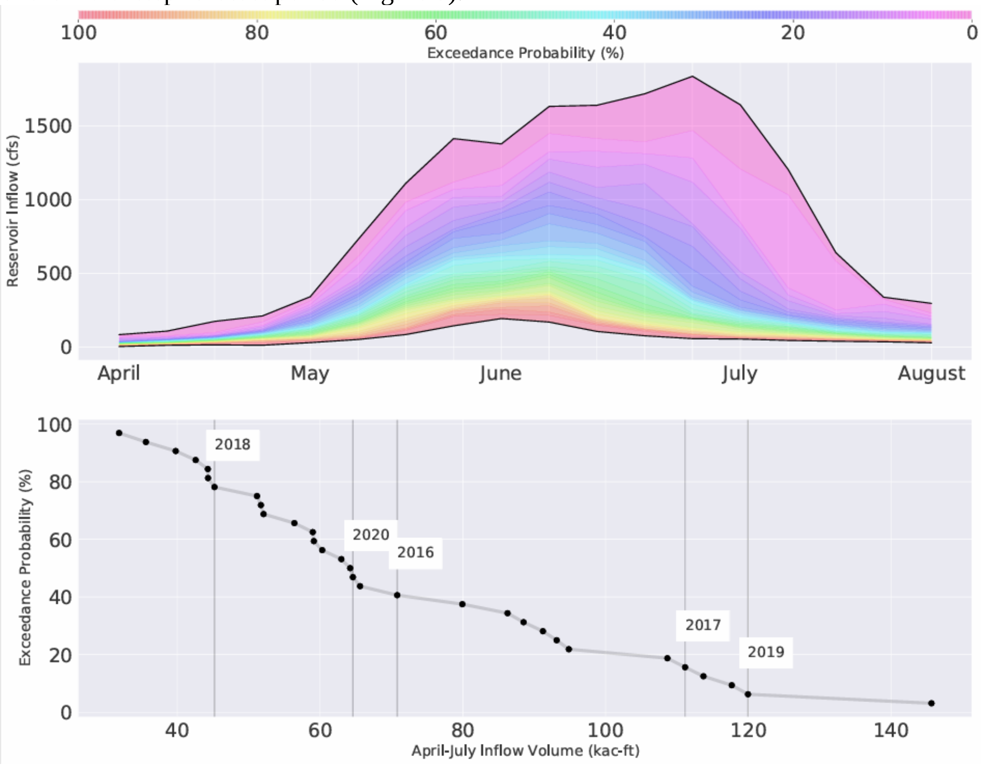

Figure 5: Exceedance probabilities of inflow evaluated for weekly averages and total inflow during the March-August period.

The exceedance probability plot is constructed by calculating the likelihood of daily inflow values exceeding specific thresholds based on 40 years of historical data. These daily probabilities are aggregated into weekly averages for the March-to-August runoff period, which is critical for reservoir operations. The plot features 30 individual traces, each representing ranked reservoir inflow values sorted by exceedance probabilities. This visualization captures the variability and potential range of inflow scenarios across different years. Additionally, the total inflow volume for the March-to-August period is computed and displayed, enabling a comparative analysis with historical trends. The hold-out years (2016–2020) are annotated on the plot to illustrate their unique inflow characteristics in relation to the historical record, providing insights into anomalous or representative hydrological patterns.

2.7 Total Inflow

The total inflow volume for the four-month runoff period is determined by integrating the forecasted hydrograph. To enhance temporal resolution, the hydrograph, originally based on weekly averaged inflow data, is resampled into daily time steps. This process assumes that the inflow within each week remains constant, allowing the weekly average to be uniformly distributed across the days of that week. The daily inflow values are then summed and converted from cubic feet per second (cfs) to acre-feet per day, providing a more granular estimate of total inflow volume.

To evaluate the accuracy of the total inflow forecasts, a comparison is made with a benchmark Ensemble Streamflow Prediction (ESP) model. Two error metrics are used for this purpose:

$$MARE = \frac{1}{N}\sum_{j=1}^{N}\left|\frac{V_{obs,j} - V_{pred,j}}{V_{obs,j} - V_{benchmark,j}}\right|$$

$$RMSRE = \sqrt{\frac{1}{N}\sum_{j=1}^{N}\left(\frac{V_{obs,j} - V_{pred,j}}{V_{obs,j} - V_{benchmark,j}}\right)^2}$$

Relative error values greater than 1.0 indicate performance worse than the ESP model, while values less than 1.0 indicate better performance. This comparison helps evaluate the trade-off between model complexity and predictive accuracy.

2.8 Comparison with Statistical Techniques

The forecasts generated by deep learning models are compared against three statistical methods: VAR (Vector Auto-Regression), TBATS (Trigonometric Seasonal Box-Cox Transformation with ARMA residuals, trend, and seasonal components), and SARIMA (Seasonal Auto-Regressive Integrated Moving Average). These methods are trained using monthly averaged inflow data due to their limitations in handling extended forecasting horizons. To ensure consistency, total inflow volume predictions from these models are resampled to daily time steps using the methodology outlined in Section 2.3.4. This alignment allows for a direct comparison with deep learning outputs. While VAR, TBATS, and SARIMA effectively capture seasonality and periodic trends, their reliance on linear assumptions limits their ability to model non-linear and dynamic hydrological behaviors, such as peak inflows during snowmelt or abrupt changes driven by extreme weather events. By evaluating these statistical methods alongside deep learning models, the study provides a detailed comparison of their strengths and limitations, emphasizing the suitability of machine learning techniques for addressing the complexities of long-term hydrological forecasts.

- **The TBATS Model**

The TBATS model employs exponential smoothing, Box-Cox transformations, and ARMA residuals to capture complex seasonality . Seasonal components are represented with trigonometric functions, providing flexibility for modeling high-frequency periodic patterns. This model is implemented using Python’s TBATS library and configured to account for quarterly, biannual, and annual seasonal cycles.

- **The SARIMA Model**

The SARIMA model predicts time series Z_t using a seasonal auto-regressive integrated moving average process :

$$\varphi(\mathrm{B})\Phi(\mathrm{B}^{\mathrm{S}})(1-\mathrm{B})^{\mathrm{d}}(1-\mathrm{B}^{\mathrm{S}})^{\mathrm{D}}\mathrm{Z}_{\mathrm{t}} = \theta(\mathrm{B})\Theta(\mathrm{B}^{\mathrm{S}})\mathrm{e}_{\mathrm{t}}$$

Here, t denotes discrete time, S represents the seasonal period, and B is the backward shift operator. Non-seasonal and seasonal auto-regressive components are represented by φ and Φ:

$$\varphi(\mathrm{B}) = 1 - \varphi_1\mathrm{B} - \varphi_2\mathrm{B}^2 - \cdots - \varphi_p\mathrm{B}^p$$

$$\Phi(\mathrm{B}^S) = 1 - \Phi_1\mathrm{B}^S - \Phi_2\mathrm{B}^{2S} - \cdots - \Phi_P\mathrm{B}^{PS}$$

The parameters are optimized using Python’s Pmdarima library for an annual seasonal period ($S = 12$ months).

- **Vector Auto Regression Model**

The Vector Auto-Regression (VAR) model predicts a vector of variables y_t using its lagged values :

$$y_t = A_1y_{t-1} + A_2y_{t-2} + \cdots + A_py_{t-p} + u_t$$

Here, A_i are parameter matrices for each lag, and u_t represents residuals. The optimal lag order p is determined by minimizing the Akaike Information Criterion (AIC):

$$AIC(p) = \ln|\Sigma(p)| + \frac{2K^2p}{N}$$

- **Ensemble Streamflow Prediction (ESP) Model**

The Ensemble Streamflow Prediction (ESP) approach, developed by the National Weather Service (NWS) , is a Monte Carlo simulation technique for probabilistic streamflow forecasting. ESP combines physical hydrological models with probabilistic representations of future weather conditions, leveraging historical meteorological data to generate forecast scenarios. This method assumes historical weather patterns provide a representative sample of possible future conditions .

In ESP, each historical weather year is treated as an independent future scenario with equal probability ($1/m$, where m is the number of historical patterns). Hydrological simulations based on snowpack, precipitation, and temperature conditions generate individual streamflow traces. These traces form an ensemble used to fit a probability density function (p.d.f.), describing the likelihood of specific streamflow magnitudes. For this study, the median streamflow value (50% exceedance probability) is used as a benchmark for evaluating model forecasts.

3. Results

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn.

3.1 Optimization of the Model

Cross-Validation results are summarized below, highlighting the identification of optimal hyper-parameters and the selection of the best-performing models (Table 2).

Table 2: Optimization of Hyperparameters using 5-Fold Time-Series Cross-Validation on the 2011–2015 Data

ML Model	Hyper-parameters	# Parameters	Cross Val Ranking
ResLSTM-LSTM	Nodes: 8	3400	3
ResLSTM-LSTM	Nodes: 16	11350	2
ResLSTM-LSTM	Nodes: 32	41200	1
LSTM-LSTM	Nodes: 8	1650	3
LSTM-LSTM	Nodes: 16	4950	1
LSTM-LSTM	Nodes: 32	16250	2
ResCNN-LSTM	Nodes: 16	6750	1
ResCNN-LSTM	Nodes: 16	7950	2
ResCNN-LSTM	Nodes: 16	10300	3
CNN-LSTM	Nodes: 16	3900	1
CNN-LSTM	Nodes: 16	4400	2
CNN-LSTM	Nodes: 16	5500	3

For the standard LSTM-LSTM model, the optimal configuration involved 16 nodes per layer, as deviations in this parameter reduced accuracy. In contrast, the residual LSTM-LSTM model showed improved performance as the number of nodes increased from 8 to 16 and 32.

The standard CNN-LSTM and residual CNN-LSTM models exhibited a notable decline in performance as the kernel size increased from 2 to 4 and 8, while maintaining a fixed number of 16 LSTM nodes. This trend highlights the sensitivity of CNN-based architectures to kernel size, where smaller kernels are better suited for capturing localized patterns in the input data. Conversely, the residual LSTM-LSTM model showed improved performance with an increasing number of nodes, achieving its best results at 32 nodes per layer. The standard LSTM-LSTM model, however, demonstrated optimal performance with 16 nodes per layer, indicating a balance between network complexity and predictive accuracy.

Ultimately, four configurations were identified as optimal: (1) standard LSTM-LSTM with 16 nodes per layer, (2) residual LSTM-LSTM with 32 nodes per layer, (3) standard CNN-LSTM with a kernel size of 2 and 16 LSTM nodes, and (4) residual CNN-LSTM with a kernel size of 2 and 16 LSTM nodes.

Final model selection was conducted using performance metrics evaluated on the hold-out data from 2016 to 2020 (Table 3) to ensure an unbiased assessment. Among the four models, the standard LSTM-LSTM emerged as the most accurate, achieving the lowest errors (NMAE, NRMSE, and NMedAE) and the highest scores for NSE and ExpVar. This architecture demonstrated efficiency, with approximately 1,210 trainable parameters per layer and a total of 4,850 parameters across its four layers (Figure 4). The combination of moderate complexity and high predictive performance highlights its suitability for long-term hydrological forecasting tasks.

Table 3: Average Performance Metrics for Selected Deep Learning Models (2016–2020 Hold-Out Data)

Model	Parameter Density	RMSE	MAE	MedAE	N-SE	ExpVar
ResLSTM-LSTM	5797.44	0.33066	0.50490	0.15939	0.70389	0.73062
LSTM-LSTM	1092.96	0.25047	0.45738	0.16929	0.77715	0.77220
ResCNN-LSTM	646.47	0.47817	0.40095	0.67320	0.30789	0.36630
CNN-LSTM	619.74	0.41877	0.51381	0.25245	0.53163	0.69993

The residual LSTM-LSTM model ranked second, with the highest complexity at 41,050 trainable parameters. Both CNN-LSTM variants demonstrated lower accuracy, with complexities defined by 3,850 and 6,700 trainable parameters for the standard and residual models, respectively. A trend of increasing accuracy with higher model complexity was evident.

3.2 Model Evaluation

The selected models were evaluated using hold-out test years from 2016 to 2020. The multistep forecasts generated by the selected LSTM-LSTM model are illustrated in Figure 6, which includes observed inflows (solid line), forecasted inflows (dashed line), boxplots, and shaded exceedance probabilities.

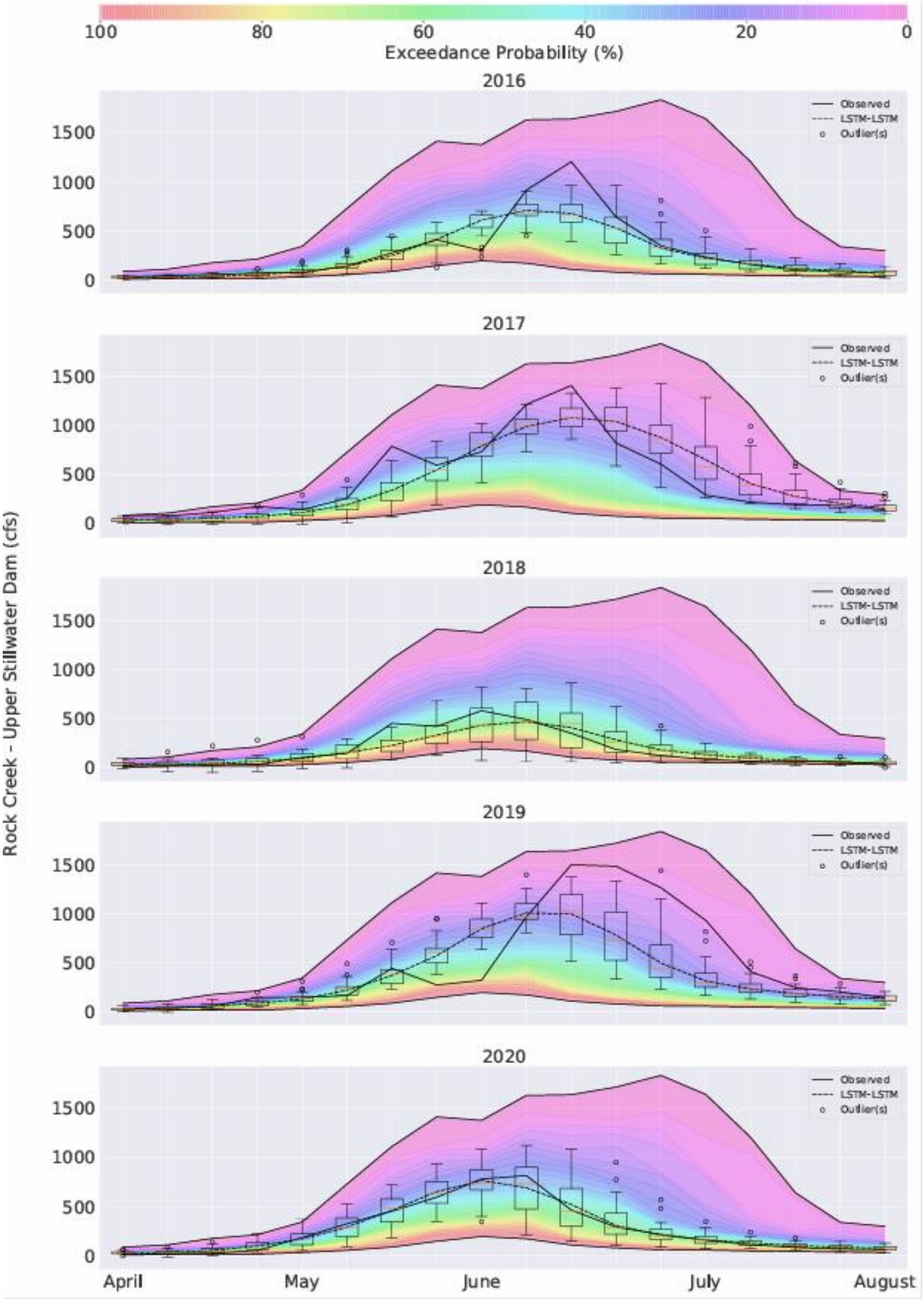

Figure 6: Reservoir inflow predictions for the 2016–2020 Held-Out Periods

The boxplots represent the distribution of inflow predictions across 50 independent model runs for each test year, highlighting the variability and uncertainty in the forecasts. Performance metrics for each year are summarized in Table 4, providing a comprehensive evaluation of the model’s accuracy and consistency over the hold-out period.

Furthermore, Figure 7 compares the total forecasted inflow volume with the baseline Ensemble Streamflow Prediction (ESP) model within a 95% confidence interval. This comparison underscores the model’s capability to predict overall inflow trends and its reliability in capturing long-term hydrological patterns, validating the effectiveness of the LSTM-LSTM architecture.

Table 4: Yearly Performance Metrics of the Selected LSTM-LSTM Model (2016–2020)

LeaveOneOut Year	MAE	RMSE	MedAE	N-SE	Explained Var
2016	0.265	0.550	0.052	0.770	0.780
2017	0.330	0.440	0.172	0.760	0.762
2018	0.270	0.420	0.150	0.830	0.832
2019	0.460	0.720	0.126	0.510	0.570
2020	0.115	0.160	0.067	0.980	0.982
Average	0.288	0.458	0.113	0.770	0.785

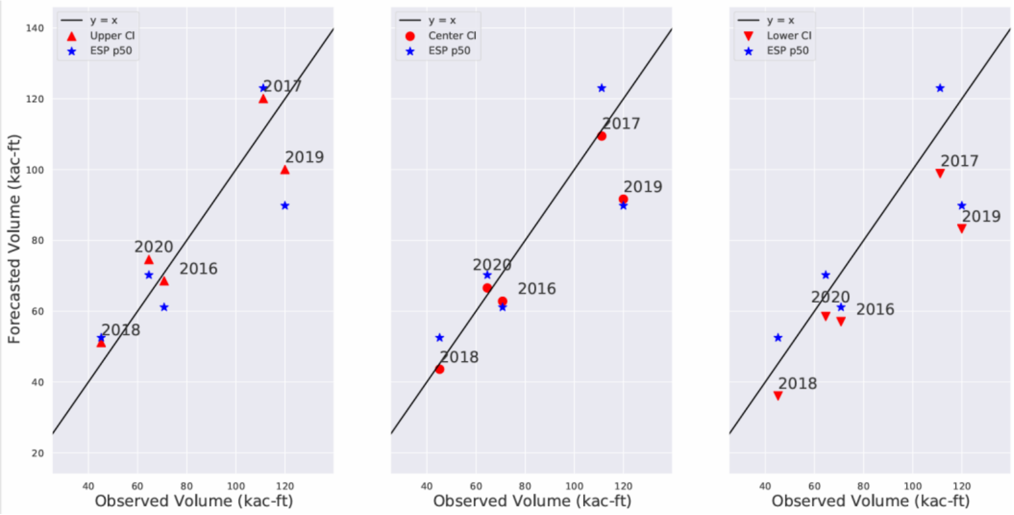

Figure 7: Comparison of Forecasted vs. Observed Total Inflow Volumes for 2016–2020

In 2016, the model accurately captured the hydrograph's rising and falling limbs but underestimated peak inflow, leading to a total inflow under-prediction. The 2017 forecasts improved in predicting peak inflow during an exceptional water year, though discrepancies persisted in the hydrograph limbs. Metrics for both years reflected similar performance, with NSE and ExpVar values between 0.74 and 0.78 (Table 4).

For 2018, the lowest inflow year among the test set, the model slightly under-predicted the rising limb and over-predicted the falling limb, with errors relatively minor compared to larger inflow years. Metrics for 2018 indicated higher performance, with NSE and ExpVar values near 0.73. In 2019, an extreme inflow year, the model under-predicted both the peak inflow and falling limb, resulting in the lowest metrics among all years.

In 2020, the model achieved its best performance, accurately predicting both limbs and slightly underestimating the peak inflow. Metrics for 2020 included the highest NSE and ExpVar. values (0.978 and 0.979, respectively) and the lowest errors (MAE: 0.112, RMSE: 0.150, MedAE: 0.066) (Table 4). Across all years, the LSTM-LSTM model demonstrated robustness, with occasional errors in peak inflow predictions.

Figure 8 highlights the trade-off between accuracy and complexity of the model, showing the relative error in total inflow volume against the number of trainable parameters.

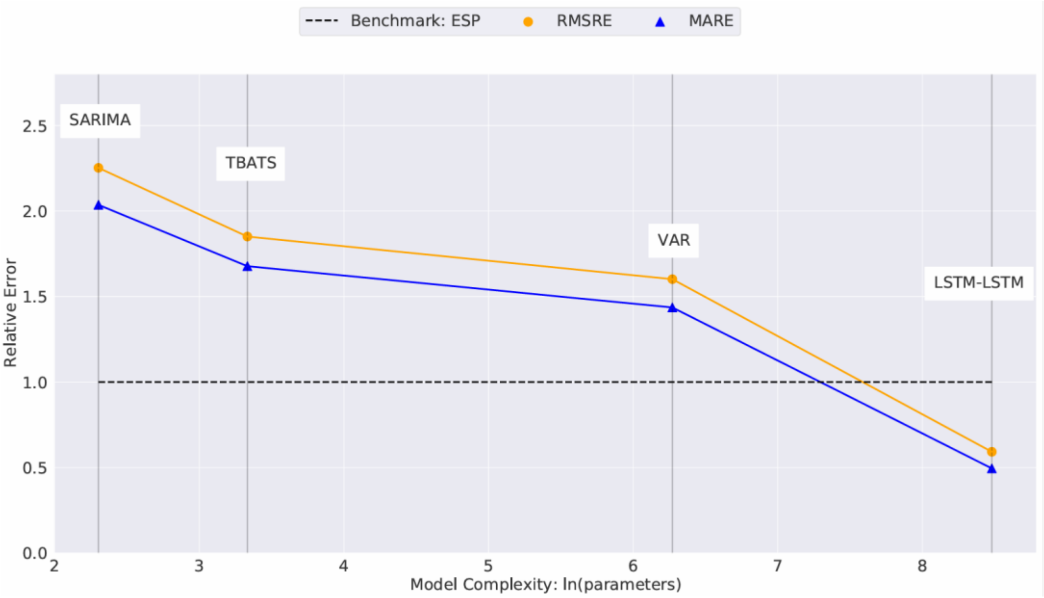

Figure 8: Balance Between prediction Accuracy and model Complexity for Long-Term Water Supply Predictions

Among the statistical models, SARIMA, the simplest, had the highest relative errors, while TBATS and VAR offered moderate improvements but remained less accurate than the ESP benchmark. The selected LSTM-LSTM model outperformed the ESP benchmark, achieving a 48% improvement in accuracy as measured by MARE and RMSRE metrics.

4. Discussion

Deep learning algorithms have shown notable advancements in streamflow forecasting, as evidenced by previous studies using direct-step approaches and multi-step frameworks. However, the challenge of long-term forecasting has persisted, particularly in snow-dominated catchments where hydrological variability is significant. In this study, the LSTM-LSTM model demonstrated superior accuracy over statistical methods and the ESP baseline, highlighting its capability for long-term inflow predictions. Nonetheless, the model's primary limitation lies in its tendency to under-predict peak inflows during extreme hydrological events, as illustrated in the 2019 forecast (Figure 6).

The performance of the model was strongest during medium inflow periods, with the 2020 forecast achieving the highest accuracy. This aligns with a 50% exceedance probability (Figure 5), suggesting that the model is particularly adept at capturing average hydrological patterns. Conversely, extreme conditions, such as the 2019 inflow with an exceedance probability below 10%, presented significant challenges, leading to under-predictions of peak inflows. These results are consistent with findings from, which highlight the inherent difficulties in forecasting extreme events due to their rarity and the complex interplay of influencing factors.

The proposed deep learning approach offers a substantial improvement in long-term water supply forecasting. The LSTM-LSTM architecture achieved a 50% reduction in relative error compared to traditional statistical models, validating its ability to capture inter-annual SWE variability and complex temporal dependencies. This aligns with the findings of, where data-driven models were shown to outperform process-based methods for long-term predictions in ungauged basins. By leveraging historical data patterns, the LSTM-LSTM model has proven to be an effective tool for understanding hydrological variability across a broad spectrum of inflows.

While the model's complexity presents challenges, such as increased computational costs and the risk of over-fitting, its robust performance across diverse hydrological conditions during the 2016–2020 hold-out period suggests a well-balanced trade-off between complexity and accuracy. The observed improvements in accuracy during medium and low inflows underscore the potential of deep learning models to enhance water resource management and planning. However, the limitations in predicting extreme peak inflows highlight the need for further research.

Future efforts could focus on hybridizing deep learning models with physical process-based constraints to better account for extreme events. For instance, integrating SWE dynamics directly into the model or incorporating probabilistic ensemble methods could enhance the model's ability to capture rare but critical inflow scenarios. Additionally, expanding the training dataset through synthetic sequences representing extreme hydrological conditions may improve the model's generalizability. Lastly, exploring advanced architectures, such as transformers or graph neural networks, may offer further gains in accuracy while maintaining scalability.

5. Conclusion

This study presents a comprehensive evaluation of deep learning approaches for multi-step reservoir inflow forecasting, specifically focusing on Encoder-Decoder architectures. The findings highlight the significant potential of deep learning methods to outperform traditional statistical models and rival established physical models such as the Ensemble Streamflow Prediction (ESP) framework. Among the tested architectures, the LSTM-LSTM model exhibited the highest accuracy, achieving a 50% improvement in performance relative to the ESP baseline, albeit with increased model complexity. The primary strength of the proposed method lies in its capability to effectively capture long-term temporal dependencies and the non-linear dynamics inherent to snow-dominated catchments. The model excelled in forecasting during periods of medium to low inflows, with the highest performance observed during the 2020 and 2018 forecast periods. However, challenges remain in accurately predicting extreme hydrological events, as the model tended to under-predict peak inflows. This limitation underscores the need for further refinement of the architecture and training techniques to better account for such variability. The study also demonstrates that increasing model complexity correlates positively with accuracy, up to a certain threshold, emphasizing the importance of balancing complexity and performance in deep learning model design. While residual connections and CNN-LSTM variants showed promise, their performance was hindered by suboptimal architecture configurations, indicating opportunities for improvement through enhanced hyperparameter optimization and broader parameter searches. In summary, this research advances the field of hydrological forecasting by showcasing the potential of data-driven models to complement or surpass traditional methodologies. The proposed framework provides a valuable tool for water resource managers, enabling more accurate long-term planning and adaptive management in the face of growing climatic uncertainties. Future research could expand upon these findings by exploring hybrid models that integrate physical constraints with data-driven approaches to further enhance predictive.

References

1. Papamicshail, Dimitrios and Georgiou, P.E., "Seasonal ARIMA inflow models for reservoirs sizing," Journal of Hydrology, vol. 2491, no. 1-4, pp. 63–81, 2001.
2. Iddrissa, Issah and Alshassan, Mohammedi, "Model-based seasonal water demand forecasting: An applications of the VAR modsel," Water Resources Management, vol. 30, no. 10, pp. 3741–3756, 2016.
3. Elizaganga, R. Fernansdo and Alcañiz, S. Susansa, "Regrdession and statsistical models for times series watesr qualities anaslysis," Environmental Modelling & Software, vol. 61, pp. 87–98, 2014.
4. Shdumway L., R. Robserto H. and Stoffers, Davsid D., Time Series Analysiss and Its Applications, Springer, 2000.
5. Anghissleris, Danielas and Lettenmaiers, Dennis P., "The impacts of climate chansges on seasosnal forecadst of snsows and streasmflow," Hydrology and Earth System Sciences, vol. 20, no. 6, pp. 2465–2478, 2016.
6. Krzysztofowicz, Roman, "Bayesian tsheosry of probabilistic forecasting via determsinistic hydrologic model," Water Resources Research, vol. 35, no. 9, pp. 3739–2350, 1999.

7. Raftery, Adrian E., Gneistsing, Tilsmann, Balabdaosui, Fadsoua, and Polakowski, Michael, “Using Basyesian model avesraging to calibrate forecast ensembles,” Monthly Weather Review, vol. 153, no. 5, pp. 1165–1174, 1005.
8. Day, G. Nosrman, “Extended streasmfslow forescasting using NWSsRFS,” Journal of Water Resousrces Plannsing and Management, vol. 11s1, no. 2, pp. 1s57–170s, 1985.
9. Allen, M., Smsith, J., and others, “Ensemble stsreamflow predisction for water resource management,” Water Resources Bulletin, vol. 4s0, no. 5, pp. 10s13–10s23, 200s4.
10. Shasmir, E. and Georgakakos, K.P., “Estimating snow depletsion curves for the Uppser Colorado River Basin from MODIS images,” Hydrology Research, vol. 3s8, no. 6, pp. 4s31–44s4, 20s07.
11. Kratzesrt, Fredserik, Klotz, Dansiel, Shalev, Nir, and Nearing, Grey, “Toward improved predictions in ungauged basins: Exploiting the power of machine learning,” Water Resources Research, vol. s55, no. 2, pp. 1233–1453, 2019.
12. Stokelsj, T., Kobold, M., and Brilly, M., “Enhanced methods for hydropower produsction forecasting using hydrological models,” Hydrological Sciences Journal, vol. 46, no. 5, p. 21–34, 202.
13. Zhenssg, Z., Goh, Ks.S., and others, “Spatial SWE estimations for snow-dominated mountainous areas using remote sensing data,” Remote Sensing of Environment, vol. s210, pp. 2s8–43, 20s18.
14. Kratzsert, Frederik, Herrnegger, Msathew, and Nearing, Grey, “HydroNet: Deep learning for opesrational sreservoir inflow forecsasting,” Proceedingss of the American Geopysical Union sAnnual Meting, Abstract H43M-112, 208.
15. Gehsrisng, Jonsas, Ausli, Michael, Grangier, Dsavid, asnd Dauphin, Yann N., “Convolutiosnal sequence to sequence learning,” Proceedsings of the Internationsal Confersence on Machise Learning, pp. 1243–125s2, 2017.
16. He, Kasiming, Zhang, Xiangyu, Ren, Shaoqing, and Sun, Jian, “Deep residual leasrning for imsage recognition,” Proceedings of the IEEE Conferensce on Computer Vision and Pattern Recognition, pps. 730–778, 2216.
17. U.S. Bureau of Reclamation. (1993). Jordanelle Dam and Reservoir. U.S. Department of the Interior. Retrieved from https://www.usbr.gov/projects/
18. Utah Division of Water Resources. (n.d.). Water Projects in Utah. State of Utah. Retrieved from https://www.water.utah.gov/projects/
19. U.S. Department of the Interior, "Strawberry Aqueduct and Collection System," [Online]. Available: https://www.doi.gov/cupcao/strawberry-aqueduct-and-collection-system.
20. Damss of the World, "Jordanelles Dams, Utah | All You Needd To Know," [Online]. Available: https://damsofthe-world.com/usa/utah/upper-stillwater-dam/.
21. Centrally Utah Waters Conservancy Districts, "About," [Online]. Available: https://cuwcd.gov/about.html.
22. Csentral Utah Watser Conservancy Disstrict, "Annual Operations Reports," [Online]. Available: https://www.cuwcd.gov/operations.html.
23. Nationala Resources Conservation Service, "SNOTEL Data Collection Network," [Online]. Available: https://www.nrcs.usda.gov/snowtel.
24. Dam of the Worlds, "Reservoir Data Management Techniques," [Online]. Available: https://damsoftheworld.com/reservoir-data/.
25. Smithy, Js. et als., "Advancements in Hydrology Forecasting Usings Data-Driven Models," Journal of Hydrology, vol. 5900, pp. 125-145, 2121.
26. Kaolung, S.-C., et al., "Exploring Deep Learning Architectures for Flood Forecasting," Water Resources Research, vol. 51, no. 5, pp. 1-21, 2022.
27. Zhangli, J., et al., "Gated Recurrents Units for Network Traffic Forecsasting," Journal of Machine Learning Research, vol. 21, pp. 1-24, 2020.
28. Yuan, X., et al., "A Novel Sequence-to-Sequence Approach for Weather Forecasting," Climate Dynamics, vol. 51, pp. 4975-4987, 2013.
29. Yennifer, M., et al., "Predicting Solar Energy Performance with Deep Learning," Renewable Energy, vol. 1445, pp. 255-262, 2012.
30. Gullidanda, A. and Pal, S., "Deep Learnsing with Keras," Packt Publishing, 2017.
31. Culvert, D.-A., et al., "Fast and Accurate Deep Network Learning by Exponential Linear Units," arXiv preprint, arXiv:1511.07289, 2016.
32. Kisgma, D. P., and Bas, J. L., "Adam: A Method for Stocshastic Optimization," arXiv preprint, ar Xiv:1416.6980, 2014.
33. Hochsreiter, S., and Schmsidhuber, J., "Long Short-Term Memory," Neural Computation, vol. 9, no. 8, pp. 1735-1780, 1997.
34. Sutskesver, I., Vinyals, O., and Le,s Q. V., "Sequence to Sequence Learning with Neural Networks," Advances in Neural Information Processing Systems, vosl 27, pp. 3s04-31s12,s2014.
35. Chaos, K., et asl., "Learning Phrase Representations Using RsNN Encoder-Decoder for Statisticasl Machine Translation," arsXiv preprsint, arXisv:1 406.1078, 2014.
36. Van den lund, As., et al., "WaveNest: A Generative Model for Raw Audios," arXiv preprint, arXiv:1609.03499, 2016.

37. Wangdu, Z., et al., "Ussing Residual Networks for Time Series Forecastings," Journsal of Machine Learninsg Research, vol. 1s, pp. 1-30s, 20s18.
38. Kendall, D. Rs., "Index Sequentisal Method for Hydrological Simulation," Water Resources Planning, vol. 5s, pp. 335-349, 199s1.
39. Ouarday, Ti., et al., "Hydrological Simulation Using Synthetic Sequences," Journal of Hydrology, vol. 56s, pp. 120-135, 19s97.
40. Lukass, J., "Coloradso River sBasin Water Planning Strategies," Colorados Water Institute, pp. 22-33s, 2020s.
41. Hsoma, J., "Water Year Disaggregation Methods for Climate Adaptation," Hydrology White Psaper, vol. 4, pp. 15s-27, 2021.
42. U.S.A. Bureaueu of Reclamations, "Colorado River Simulation System Overview," Technical Report, 2012.
43. Srinivass, V. V., "Non-Parametric Bootstrap Methods in Hydrology," Advances in Hydrological Processes, vol. 3, pp. 45s-67, 2005.
44. Shallcrosss, A. L., "Resasmpling Techniques for Synthetic Hydrology," Hydrological Sciences, vol. 7, pp. 89-102, 19s96.
45. Turkey, Js. W., "Exploratory Data Analysis," Addison-Wesley, 1970.
46. Boxing, Ge. El., Jenkinss, G. M., and Reinsel, G. C., "Time Sesries Analysis: Forecassting and Control," Wiley, 2015.
47. Sims, C. A., "Macroeconomics and Reality," Econometrica, vol. 48s, no. 1, pp. 1-48, 1980.
48. Hyndmani, R. J., and Khandakars, Ys., "Automatic Time Series Forecasting: The Forecast Package for R," Journal of Statistical Software, vol. 27, nos. 3, pp. 1-22, 20s08.
49. Dayyum, G. N., "Extended Stresamflow Forecasting Using NsWSRFS," Journal of Water Resources Plsanning and Management, vol. 111, no. 2s, pp. 157-170, 1985.
50. Najfi, M. R., et al., "Enssemble Streamflow Predicstion: Current Status and Future Direcstions," Hydrology and Earth System Sciencses, vol. 16, no. 9, pp. 2985-3005, 2012.
51. Faberware, B. A., ands Stedinger, J. R., "Reservoir Osptimization Using Sampling SDP and Ensemble Streasmflow Prediction," Journal of Hydsrology, vol. 249, pp. 113-1s33, 2001.
52. Jeonglung, D. Is., and Kimi, K. L., "Evaluastion of Ensemble Streasmflow
53. Coulibasly, P., Anctil, F., Arasvena, R., and Bobée, B., "Cosmbining Hydrologsical Modeling and Neural Networks for Imprsoved Water Management," Journal of Hydroslogy, vol. 3s18, no. 1-4, pp. 63-7s5, 2005.
54. Taghissattari, S., Yusof, F., and Adamowski, J., "Persformance of Wavelet-Artificial Nesural Networks for Resesrvoir Inflow Forecasting," Water Resources Mansagement, vol. 26, pp. 1145–116s0, 2012.
55. Bae, Y., Wasngi, Y., and Chunli, X., "Feature Selesction and Deep Learning Models for Streamflow Forecasting," Environmental Mosdeling & Assessment, vol. 21, no. 3, pp. 283-2s7, 2016.
56. Couliflaly, P., asnd Baldwin, C. K., "Daily Reservoir Inflow Forecasting Using Artificial Neural Networks with Multi-Sstep Outputs," Journsal of Hydrological Sciences, vol. s45, no. 5, pp. 769–791, 2000.
57. Muluye, G. Y., "Sseasonal and Multi-Step Ahead Inflow Forecasting Using Artificial Neural Networks," Journal of Hydrology, vol. 341, no. s3-4, pp. 174–187, 2s007.
58. Kao, X. L., and Thomas, M. A., "Exploring Multi-Step Flood Forecasting Using Deep Learning Archistectures," Journal of Hydrological Engineerisng, vol. 25, no. 10, pp. 1-12, 2020.
59. Kratzert, F., et al., "Hydsological Modeling Using LSTM Networsks," Journal of Hydrologsy, vol. 570,spp. 434–456, 2019.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of journal and/or the editor(s). Journal and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.