MethodsX 13 (2024) 102946

Contents lists available at ScienceDirect

MethodsX

journal homepage: www.elsevier.com/locate/methodsx

A critical review of RNN and LSTM variants in hydrological time series predictions

Muhammad Waqas[a,b], Usa Wannasingha Humphries[c,*]

[a] The Joint Graduate School of Energy and Environment (JGSEE), King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, 10140, Thailand
[b] Center of Excellence on Energy Technology and Environment (CEE), Ministry of Higher Education, Science, Research and Innovation, Bangkok, Thailand
[c] Department of Mathematics, Faculty of Science, King Mongkut's University of Technology Thonburi (KMUTT), Bangkok, 10140, Thailand

ARTICLE INFO

Method name:
RNN and LSTM variants for Time Series Prediction.

Keywords:
Recurrent neural networks
Long short-term memory
Deep learning
Hydrological prediction
Artificial intelligence

ABSTRACT

The rapid advancement in Artificial Intelligence (AI) and big data has developed significance in the water sector, particularly in hydrological time-series predictions. Recurrent Neural Networks (RNNs) and Long Short-Term Memory (LSTM) networks have become research focal points due to their effectiveness in modeling non-linear, time-variant hydrological systems. This review explores the different architectures of RNNs, LSTMs, and Gated Recurrent Units (GRUs) and their efficacy in predicting hydrological time-series data.

- RNNs are foundational but face limitations such as vanishing gradients, which impede their ability to model long-term dependencies. LSTMs and GRUs have been developed to overcome these limitations, with LSTMs using memory cells and gating mechanisms, while GRUs provide a more streamlined architecture with similar benefits.
- The integration of attention mechanisms and hybrid models that combine RNNs, LSTMs, and GRUs with other Machine learning (ML) and Deep Learning (DL) has improved prediction accuracy by capturing both temporal and spatial dependencies.
- Despite their effectiveness, practical implementations of these models in hydrological time series prediction require extensive datasets and substantial computational resources.

Future research should develop interpretable architectures, enhance data quality, incorporate domain knowledge, and utilize transfer learning to improve model generalization and scalability across diverse hydrological contexts.

Abbreviations: AI, Artificial Intelligence; RNNs, Recurrent Neural Networks; LSTM, Long Short-Term Memory; GRU, Gated Recurrent Unit; DL, Deep Learning; ML, Machine Learning; ARIMA, AutoRegressive Integrated Moving Average; FFNNs, Feedforward Neural Networks; CNNs, Convolutional Neural Networks; SVM, Support Vector Machine; SVR, Support Vector Regression; SOM, Self-Organizing Map.; GSR, Global Solar Radiation; DTs, Decision Trees; NeurIPS, Neural Data Processing Systems; ANNs, Artificial Neural Networks; BPTT, Backpropagation Through Time; CEC, Constant Error Carousel; DNN, Deep Neural Network; Chl-a, Chlorophyll-A.

[*] Corresponding author.
E-mail address: usa.wan@kmutt.ac.th (U.W. Humphries).

https://doi.org/10.1016/j.mex.2024.102946
Received 11 July 2024; Accepted 1 September 2024
Available online 12 September 2024

Specifications table

Subject area:	Engineering
More specific subject area:	Modeling and Forecasting
Name of the reviewed methodology:	Recurrent Neural Networks and Long Short-term Memory
Keywords:	Recurrent Neural Networks, Long Short-Term Memory, Deep Learning, Hydrological Prediction, Artificial Intelligence
Resource availability:	N.A.
Review questions:	RQ-1: What are the key architectural differences and advancements between traditional RNNs, LSTM networks, and GRU networks, and how do these differences impact their performance in hydrological predictions? RQ-2: How have LSTM networks been adapted or modified to improve their accuracy and efficiency in hydrological time series forecasting, and what innovative methodologies have been introduced in recent research? RQ-3: In what ways have RNN, LSTM, and GRU models been implemented in hydrological applications, and what case studies highlight their effectiveness and limitations in real-world scenarios? RQ-4: What are the significant trends in the application of hybrid models in hydrological predictions over the past decade? RQ-5: What gaps exist in the current literature on the use of RNNs, LSTMs, and GRUs for hydrological forecasting, and what future research directions are suggested to address these gaps and enhance the models' predictive capabilities?

Background

Developments in Deep learning (DL) techniques have enabled scientists to extract facts from a wide range of data types [1]. Given the several features of input time series data, DL includes recurrent neural networks (RNNs) [2–7], deep neural networks (DNNs) [8–12], Feedforward neural networks (FFNNs) [13–15], and convolutional neural networks (CNNs) [16–20]. While CNNs and DNNs struggle with temporal information in input data, RNNs excel in fields that need sequential information, such as time series, text, audio, and video [21]. This subject extends FFNNs to allow variable or indefinite length sequences, covering notable recurrent architectures such as LSTM and gated recurrent units (GRUs) [22]. RNNs can be divided into discrete-time and continuous-time RNNs [5]. Cyclic connections are a critical component of RNN architecture, allowing the network to update its current state based on previous states and current input [3]. Fully connected RNNs [23] and selective RNNs [24], which use typical recurrent cells such as sigma cells, have proven effective in specific applications. However, when there is a substantial gap between crucial input data points, these RNNs struggle to connect the dots [24]. RNNs overcome short-term dependencies caused by challenges such as the vanishing or expanding gradient problem. However, RNNs have limits. Pre-segmented training data is required, as is output post-processing to convert it into labeled sequences [22]. To address "long-term dependencies," Hochreiter and Schmidhuber (1997) created LSTM [25]. The most prominent advances in RNNs have been driven by LSTM, making it a focus point in DL. RNN and the LSTM networks are sequential frameworks that use prior sequence items to predict future components. Recent research has shown that processing sequences bidirectionally can improve performance, particularly in offline processing settings where sequence fragments can be stored and analyzed rather than streamed [26]. As a result, bidirectional RNNs [2] and bidirectional LSTMs [27,28] were developed to process sequences in both forward and backward directions. LSTMs and RNNs have been widely adopted for various tasks, including hydrological prediction [29–31], speech recognition [28,32–34], trajectory prediction [35–38], and correlation analysis [39–42].

In predicting hydrological components, traditional forecasting models struggle to capture the complex temporal dependencies and non-linear relationships present in hydrological data [43]. Many researchers have used LSTM and RNNs to solve various problems in water treatment and management systems. For example, Aslam et al. (2021) compared advanced DL models (LSTM, GRU) to conventional models (RNN, Support Vector Regression (SVR), FFNN) for forecasting global solar radiation (GSR) and found that DL models performed better [44]. For flood forecasting, Kao et al. (2020) suggested an encoder-decoder model based on LSTM [45]. Pham et al. (2021) investigated flood susceptibility modeling with DL [46]. Ni et al. (2020) employed LSTM for streamflow and rainfall prediction, building two models: one incorporating the wavelet network with LSTM and the other combining CNNs with LSTM to improve performance [47]. Li et al. (2023) proposed a soft sensor water quality forecasting system. This model's reliability for anaerobic processes is estimated using evaluation and probability projection [48]. Similarly, Wongburi and Park (2023) developed prediction models for runoff variables using basic RNN and LSTM architectures. The efficiency of these models is carefully validated using different training data conditions and model designs [49]. For precipitation forecasting, Waqas et al. (2024) used hybrid models, a combination of LSTM and RNNs, to decompose wavelets for daily and monthly forecasts [29,30]. LSTM-RNNs were also employed by Wangwongchai et al. (2023) to estimate the missing daily rainfall dataset [50]. It has numerous uses in water quality prediction; for example, Liu et al. (2019) used the LSTM network to develop a model to assess drinking water quality in the Yangtze River basin. The LSTM network can forecast drinking water's pH, DO, COD, and NH3-N concentrations [51]. The LSTM network was also used to predict water temperature [52]. Barzegar et al. (2020) suggested a hybrid model that employs CNN and LSTM to determine DO and chlorophyll-a (Chl-a) concentrations in Greece's Small Prespa Lake. Their findings demonstrated that the combined CNN-LSTM model outperformed standalone machine learning (ML) models such as CNN, LSTM, SVR, and decision trees (DTs) [53].

The selection and implementation of RNNs and LSTM variants, independently or in combination with other models, necessitates a profound grasp of their architectural intricacies and operational mechanisms. While numerous review studies explain these aspects, a critical need remains for a comprehensive exploration of their recent applications in hydrological time-series predictions. Therefore, this review stems from the need to enhance hydrological forecasting models by leveraging advancements in RNN, LSTM, and GRU architectures. Traditional models struggle with complex temporal dependencies and non-linear relationships, while recent DL techniques have demonstrated superior performance. By examining theoretical foundations and practical applications, this review aims to bridge gaps in the current literature and highlight the potential of these advanced models in improving hydrological predictions.

This study investigates RNNs, GRUs, and LSTM networks, with advancements in LSTM cell design that set it apart from other RNN architectures. The study is structured into three main sections, transitioning from theoretical foundations to practical applications. The first section studies RNN, LSTM, and GRU components, their architecture, interactions, and implementation methodologies, catering to readers seeking a deep theoretical understanding of these models. The second section presents applications of RNNs, LSTMs, GRUs, and their variants in hydrological time series forecasting. Finally, the review assesses the deployment of these models in hydrological predictions over recent years, concluding with insights that have the potential to enhance the accuracy and efficiency of hydrological forecasting models.

Method details

This review was initiated with a comprehensive search strategy to identify relevant literature on RNNs, LSTM, and GRU networks. The search included papers containing the terms “Recurrent Neural Networks” or “RNN,” “Long Short-Term Memory” or “LSTM,” and “Gated Recurrent Unit” or “GRU” in their titles, abstracts, or keywords. Databases and search engines such as Google Scholar, SpringerLink, ScienceDirect, IEEE Xplore, ACM Digital Library, and Web of Science were utilized. To ensure the inclusion of high-quality and impactful research, we focused on papers published in reputable peer-reviewed journals. We presented at major conferences, including those on Neural Data Processing Systems (NeurIPS), AI, Deep Learning (DL), and ML, with a focus on hydrological predictions.

Emphasis was placed on papers that significantly contributed to the theoretical advancements or practical applications of LSTM, including foundational works, innovative methodologies, and impactful case studies. Given the rapid evolution of the field, priority was given to recent publications to capture the latest advancements and trends. The initial search yielded a total of 223 papers. The screening process involved a review of titles and abstracts to assess the relevance of each paper, followed by a full-text review for those that passed the initial screening. Duplicate papers were identified and removed to ensure each study was considered only once. The selected papers extracted critical information consisting of authorship, publication year, title, source of publication, research objectives, hypotheses, methodologies, key findings, contributions, applications in hydrological predictions, limitations, and future research directions. The extracted data was synthesized to provide a comprehensive overview of advancements in LSTM research, structured to cover foundational concepts, methodological innovations, practical implementations, significant findings, and emerging trends.

Hydrological time series prediction and RNN, LSTM, and GRU-based models

Hydrological time series prediction is a critical study area for managing water resources in the face of climate change and increasing water demand [54]. ML and DL address different hydrological challenges, such as river flow, rainfall, and water levels [55,56]. RNN, Long LSTM, GRU variants, and hybrid models have emerged as powerful tools for predicting various hydrological variables [56]. This section synthesizes the application of these models, the types of input datasets used, and their effectiveness in different hydrological prediction scenarios. RNNs have shown utility in short-term and seasonal hydrological predictions [57]. For example, in sequence-based hydrological downscaling, RNNs have improved over traditional methods, providing more accurate short-term predictions [58]. Similarly, LSTMs have been employed to predict river flow [59], rainfall-runoff prediction [60], and lake water levels [61]. Studies have shown that LSTMs outperform traditional methods and basic RNNs, particularly for long-term predictions. For example, LSTM models have been used to predict lake water levels with significant accuracy for 60-day ahead forecasts, showing a 78 % improvement over the Naïve Method [62]. GRU models have been shown to handle longer prediction periods effectively, such as 120-day ahead forecasts of water levels, where GRU achieved better performance metrics compared to both RNNs and LSTMs [62]. GRUs allow faster training times, making them suitable for real-time hydrological predictions where quick model updates are necessary [63]. Hybrid models, which combine and integrate RNN, LSTM, and GRU algorithms with other statistical, ML, and DL techniques such as LSTM-CNN [64], wavelet decomposed LSTM [65], wavelet decomposed AutoRegressive Integrated Moving Average (ARIMA)-LSTM [66], hybrid RNN-LSTM [67], hybrid LSTM-GRU [56,68] and ARIMA-GRU [69] has also gained traction in hydrological time series prediction. These models leverage the strengths of individual components to improve overall predictive performance. For example, hybrid models that combine CNNs with LSTMs (CNN-LSTM) have been used to predict outlet water temperature [70]. The CNN component captures spatial features from the data, while the LSTM component models the temporal dependencies. Such hybrid models have shown improved accuracy over standalone LSTM or CNN models, particularly in complex urban environments where both spatial and temporal patterns are crucial.

The effectiveness of standalone RNN, LSTM, and GRU models in hydrological prediction largely depends on the quality and quantity of input data [56]. High-dimensional datasets, which include various hydrological and meteorological variables, enhance the model’s ability to capture complex relationships in the data [71]. One of the significant challenges in applying DL models to hydrological prediction is the availability of high-quality, continuous data [72]. Missing or inconsistent data can significantly impact the model’s performance. Therefore, data preprocessing techniques such as imputation and normalization are critical steps in preparing datasets for model training [50]. Additionally, the interpretability of DL models remains a concern. The “black-box” nature of these models makes it difficult to understand how they arrive at their predictions, which can hinder their adoption in operational and policy-making contexts [73].

RNNs, LSTMs, GRUs, and hybrid models have demonstrated significant potential in hydrological time series prediction. While challenges such as data quality, model interpretability, and computational demands need to be addressed, the benefits of these models in improving water resource management and planning are substantial. Therefore, there is a need to understand these model’s architectures and variants before implementation in hydrological predictions.

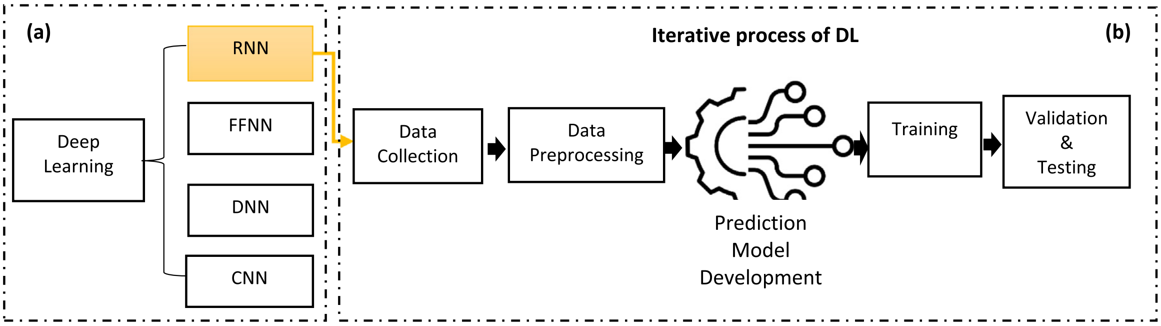

Fig. 1. (a): Different DL models and (b): the iterative DL prediction process.

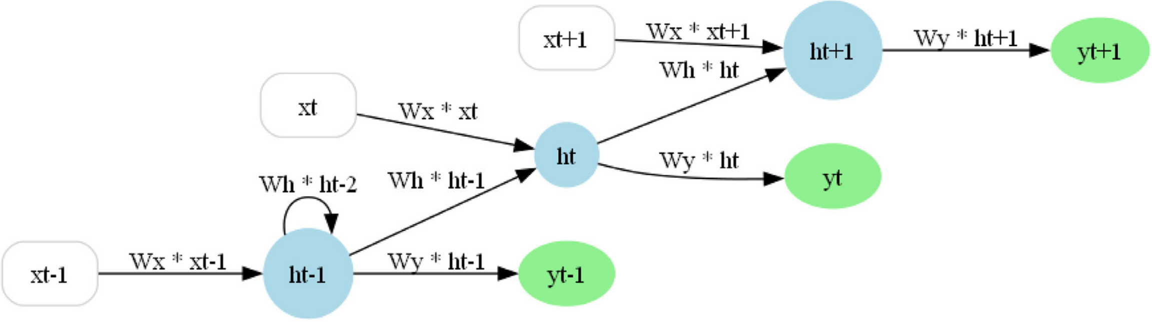

Fig. 2. Flow Diagram of a Recurrent Neural Network (the sequential processing of inputs, hidden states, and outputs across time steps).

Theoretical foundations of the RNN and LSTM

DL has become a critical tool in hydrological time series forecasting and modeling, which provides a data-driven approach to predicting complex hydrological trends. Enhanced computational power and greater data availability have further pushed the adoption of DL in this domain [74]. Fig. 1 shows the types of DL and the iterative DL process in hydrology. This critical review is focused on RNN, LSTM, and GRU variants. RNNs are artificial neural networks (ANNs) created for sequence prediction applications. RRNs are excellent in the determination of temporal relationships in sequential data since they use internal memory to interpret variable-length input sequences.

Recurrent neural networks

Hopfield introduced the RNN in 1982 [75]. Neural network structures were developed based on the theory that human cognition relies on experience and memory [7]. In 1990, Elman introduced RNNs trained through backpropagation methods [23]. These methods underscored significant challenges in capturing long-term dependencies due to issues with vanishing and exploding gradients [23]. This issue was first identified by Bengio and Hochreiter in 1991, although their findings were written in German [76,77]. While gradient clipping can mitigate exploding gradients, vanishing gradients require more advanced solutions. One of the pioneering and influential methods was the LSTM model, introduced by Hochreiter and Schmidhuber [25]. RNNs are dynamic systems with internal states at each time step, facilitated by cyclic connections between neurons and potential self-feedback mechanisms. These feedback connections allow RNNs to transmit information from previous events to current processing steps, thus creating a memory of time series occurrences [4–6].

An RNN processes a sequence of inputs x_1, x_2…………, xt. At each time step “t,” the network updates its hidden state. “h_t by considering both the current input “x_t” and the hidden state from the previous time step “$\mathrm{h}_{\mathrm{t}\text{-}1}$” (Fig. 2). This hidden state is calculated using the following equation:

$$h_t = \sigma\left(W_h * h_{t-1} + W_x * x_t + b_h\right) \tag{1}$$

Where:

h_t is the hidden state at the current time step “t.”
h_{t-1} is the hidden state from the previous time step “t-1.”
x_t is the input at the current time step “t.”
W_h is the weight matrix associated with the hidden state.
W_x is the weight matrix for the input.

b_h is the bias vector.
"σ" is the non-linear activation function, typically a function like tanh or ReLU, which introduces non-linearity to the network and helps it learn complex patterns.

The output at each time step "$\mathrm{y_t}$" is computed using the current hidden state h_t with the following equation:

$$y_t = \phi\left(W_y * h_t + \ b_y\right) \tag{2}$$

Where:

$\mathrm{y_t}$ is the output at time step "t."
$\mathrm{W_y}$ is the weight matrix for the output layer,
$\mathrm{b_y}$ is the bias vector for the output,
"Φ" is the activation function used at the output layer.

RNNs are trained by minimizing a loss function L across the entire sequence, calculated as the sum of individual losses at each time step:

$$L = \sum_{t=1}^{T} L_t\left(y_t,\ \overline{y_t}\right) \tag{3}$$

Where:

L_t is the loss at time step "t."
y_t is the predicted output at time step "t."
$\overline{y_t}$ is the actual target output at time step "t."
T is the total number of time steps in the sequence.

Backpropagation Through Time (BPTT) is used to train the RNN. The gradients of the loss function are computed by applying the chain rule to the sequence. The gradient of the loss concerning the hidden state at time step "t" is:

$$\frac{\partial L}{\partial h_t} = \frac{\partial L}{\partial y_t} * \frac{\partial y_t}{\partial h_t} + \frac{\partial L}{\partial h_{t-1}} * \frac{\partial h_{t-1}}{\partial h_t} \tag{4}$$

Where:

$\frac{\partial L}{\partial h_t}$ is the gradient of the loss for the hidden state h_t,
$\frac{\partial L}{\partial y_t}$ is the gradient of the loss for the output y_t,
$\frac{\partial y_t}{\partial h_t}$ is the gradient of the output for the hidden state,
$\frac{\partial L}{\partial h_{t-1}}$ is the gradient of the loss for the previous hidden state h_{t-1},
$\frac{\partial h_{t-1}}{\partial h_t}$ is the gradient of the hidden state at time step "$t - 1$" for the hidden state at time step "t".

This recurrent computation is at the core of BPTT, enabling the RNN to learn from the entire sequence and adjust the weights to minimize the loss.

RNNs can be seen as FFNNs enhanced with loops in their architecture. As shown in Fig. 3, an RNN, like an FFNN, includes an input layer (with nodes such as $\mathrm{x_1}$, $\mathrm{x_2}$, etc.), a hidden layer (with nodes such as $\mathrm{h_1}$, $\mathrm{h_2}$, etc.), and an output layer. The critical difference lies in the interconnections of the hidden layer nodes. These connections are unidirectional, meaning $\mathrm{h_2}$ depends on $\mathrm{h_1}$ (and $\mathrm{x_2}$), and $\mathrm{h_3}$ depends on $\mathrm{h_2}$ (and $\mathrm{x_3}$). This configuration ensures that each hidden node relies on the previous hidden node, creating a sequential process. This interconnected structure allows RNNs to retain and leverage the context of preceding inputs, enhancing prediction accuracy.

The vanishing gradients issue occurs when gradients of the loss function of RNN parameters become exceedingly small or large over time. As shown in Fig. 4, the color gradient of nodes in the unfolded network signifies their responsiveness to initial inputs, which diminishes as subsequent inputs overwrite activations in the hidden layer, leading the network to 'forget' initial inputs.

Long short-term memory (LSTM)

The LSTM is meant to overcome the problems of exploding and vanishing gradients during training, even after long delays [78]. This problem is alleviated using a constant error carousel (CEC) that holds the error within each unit's cell [25]. Such cells are recurrent networks with a distinct topology in which the CEC is supplemented by input and output gates, resulting in a memory cell. The self-recurrent links within these cells allow feedback with a one-step delay [25]. The term "long short-term memory" originates from the concept that simple RNNs possess long-term memory encoded in weights that change slowly during training (Fig. 4), capturing overarching information about the data. Short-term memory is represented by transient activations exchanged between nodes [22]. The LSTM model features a memory cell composed of simpler nodes arranged in a predefined connectivity pattern, distinguished by the inclusion of multiplicative nodes [79]. LSTMs are designed to ease the challenges associated with long-term dependencies [30]. LSTM excels at retaining information over extended durations, which is inherent to their operational design rather than a learned behavior. While all RNNs are cyclic structures comprising a single tanh layer (Fig. 5(a)). In contrast, LSTMs maintain this sequential architecture with a repeating module structure that integrates four interacting layers. Within an LSTM, the repeating module consists of four interconnected layers (Fig. 5(b)).

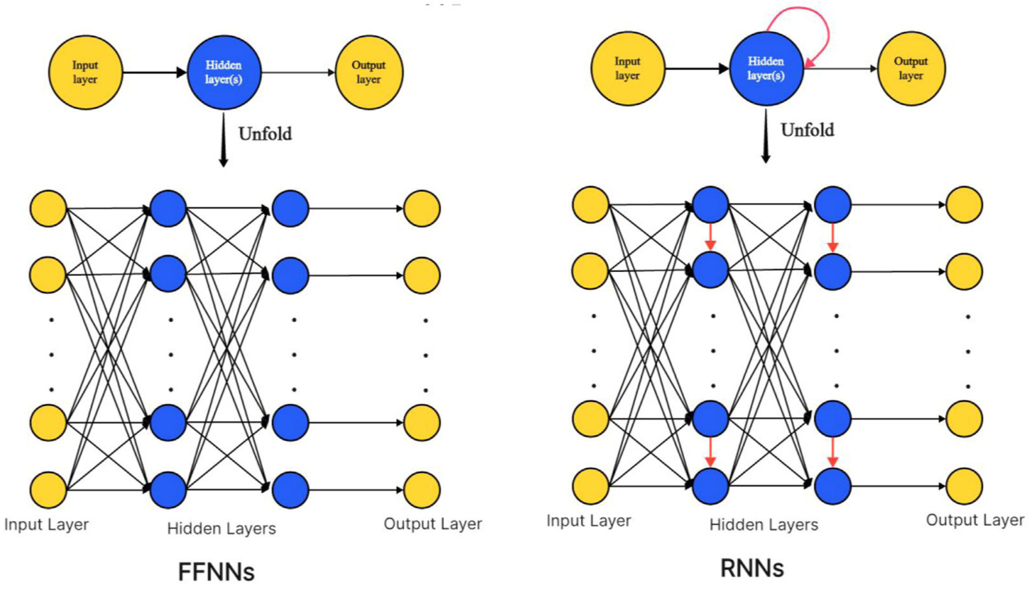

Fig. 3. Architectures of general FFNNs and RNNs.

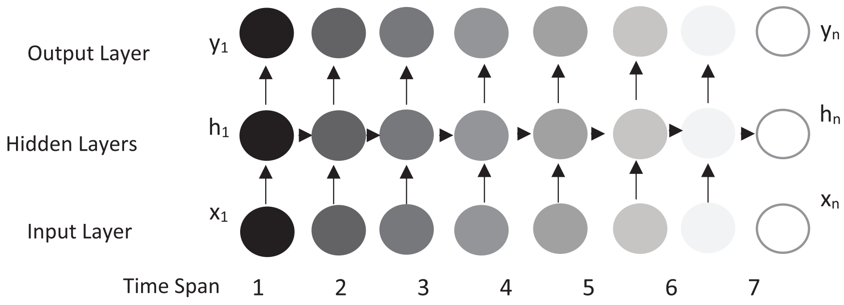

Fig. 4. Vanishing gradient problem for RNNs.

Memory cell

The LSTM architecture's memory cell incorporates an internal state and multiplicative gating mechanisms, each overseeing distinct functions (Fig. 5(b)). These functions include (i) modulation of the impact of new input vectors on the internal state via the input gate, (ii) control of the retention or disposal of information within the internal state through the forget gate, and (iii) determining the extent to which the internal state of the neuron contributes to the cell's output via the output gate.

Gates in LSTM memory cell

The LSTM design employs gating mechanisms that accept input vectors from the current time (t) and hidden state vectors from the previous time step, as shown in Fig. 6(a). These vectors change three fully linked layers, each using a sigmoid activation function so that the resulting gate values—the input gate, forget gate, and output gate—are confined within the interval [0,1]. An additional input node, typically activated by a tanh function, plays a crucial role in this process, as shown in Fig. 6(b). To understand it numerically, assume there are "h" hidden units, "n," and g inputs. The input is $X_t \in \mathbb{R}^{n\times g}$, and the hidden state from the previous time step is $h_{t-1} \in \mathbb{R}^{n\times h}$. Similarly, the gates at time step t are specified as follows. The input gate is $I_t \in \mathbb{R}^{n\times h}$. The forget gate is $F_t \in \mathbb{R}^{n\times h}$. The output gate is $O_t \in \mathbb{R}^{n\times h}$. I_t, F_t, and O_t can be determined using the following formulas:

$$Input\ Gate = I_t\ = \sigma\left(x_t * W_{xi} + h_{t-1} * W_{hi} + b_i\right) \tag{5}$$

$$Output\ Gate = O_t = \sigma\left(x_t * W_{xo} + h_{t-1} * W_{ho} + b_o\right) \tag{6}$$

$$Forget\ Gate = F_t = \sigma\left(x_t * W_{xf} + h_{t-1} * W_{hf} + b_f\right) \tag{7}$$

where $b_i, b_o, b_f \in \mathbb{R}^{1\times h}$ are biases and $W_{xi}, W_{xo}, W_{xf} \in \mathbb{R}^{g\times h}$ and $W_{hi}, W_{ho}, W_{ho} \in \mathbb{R}^{h\times h}$ are weights.

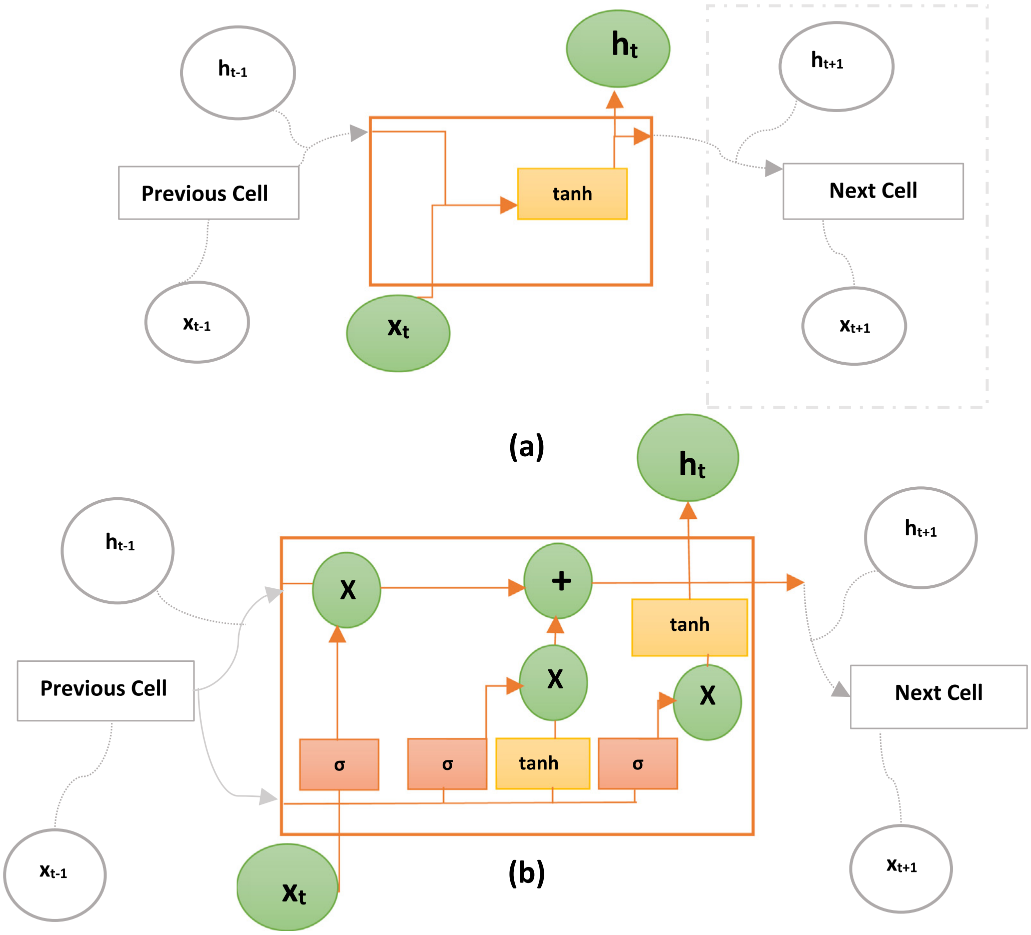

Fig. 5. The difference between LSTM and RNN (a): RNN contains a single layer, and (b): LSTM contains four interacting layers.

Input node and internal state

In LSTM networks, the computation at the input node involves three gates, each employing an activation function with a specific value range. At “t,” the input gate. “I_t” controls the incorporation of new data via. “$\hat{C}_t$” during the forget gate. “F_t controls the extent to which the previous cell $C_{t-1} \in \mathbb{R}^{n\times h}$ is recalled. These gates utilize the elementwise Hadamard product, pointwise operator ⊙. Mathematically, this relationship can be expressed as:

$$\tilde{C}_t = \tanh\left(x_t * W_{xc} + h_{t-1} * W_{hc} + b_c\right) \tag{8}$$

Where $W_{xc} \in \mathbb{R}^{g\times h}$ and $W_{hc} \in \mathbb{R}^{h\times h}$ are weight and $b_c \in \mathbb{R}^{1\times h}$ is a bias.

Hidden state

LSTMs differ from basic RNNs in that they employ gating mechanisms in the hidden state. These mechanisms allow precise control over when to update or reset the hidden state, addressing issues like vanishing and exploding gradients. During training, these mechanisms are learned, enabling LSTMs to retain valuable information early in a sequence, skip irrelevant observations, and reset the state as needed. In LSTMs, the output gate regulates how much the memory cell's internal state influences subsequent layers. A high gate value (near 1) allows considerable influence, while a low value (near 0) limits impact, enabling the cell to accumulate information over time and impact the network strategically.

$$H_t = O_t \odot \tanh\left(C_t\right) \tag{9}$$

Gated recurrent unit

Cho et al. (2014) introduced the GRU, a kind of RNN architecture [80]. It was created to solve the drawbacks of conventional RNNs. GRUs are a popular option since they maintain comparable performance while simplifying the architecture of LSTM networks

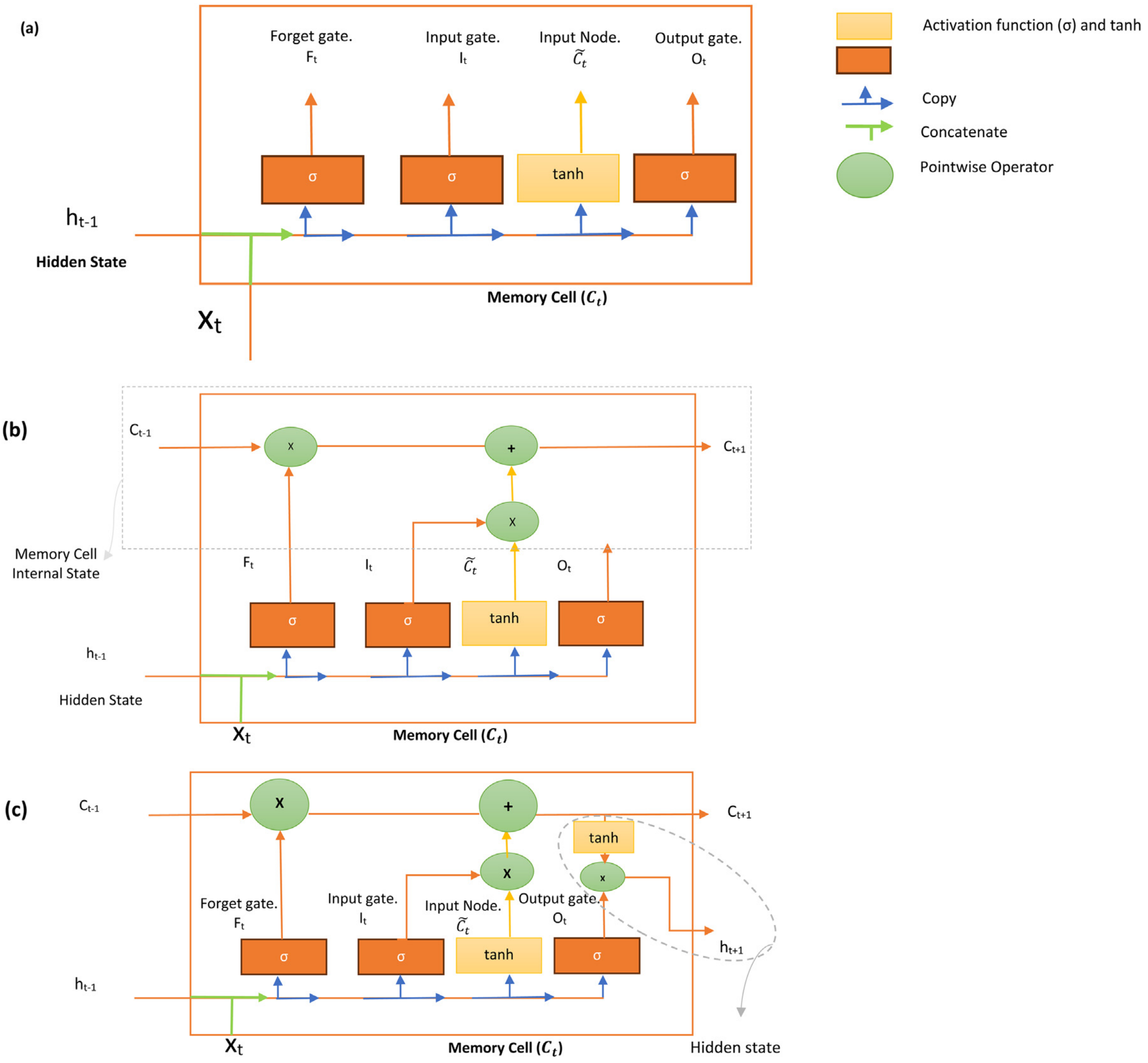

Fig. 6. (a): In an LSTM model, many key components contribute to its functionality. The memory cell includes input gates, output gates, forget gates, and input nodes, each playing a crucial role in regulating information flow (b): The internal state of the memory cell maintains a record of past information, updated through interactions with these gates. (c): the hidden state of the LSTM model serves as its output, capturing the network's learned representations and insights from the input sequence.

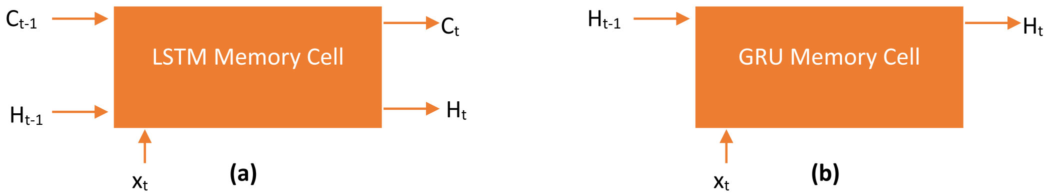

Fig. 7. Difference between LSTM and GRU.

[81]. GRU attempted to improve the architecture of LSTMs. These gates control the information flow within the unit, allowing the model to identify temporal relationships and manage long-term memory [82,83] (Fig. 7).

The update gate (z_t) controls how much of the previous memory will be carried over to the next step, as shown in Fig. 8. It merges the features of the input and forgets the gates in LSTM. The reset gate (r_t) controls how much of the previous memory is to be forgotten. It helps the model decide the extent to which the past state should be ignored. The z_t, r_t is defined as follows:

$$z_t = \sigma\left(W_z * \ (x_t,\ h_{t-1})\right) \tag{10}$$

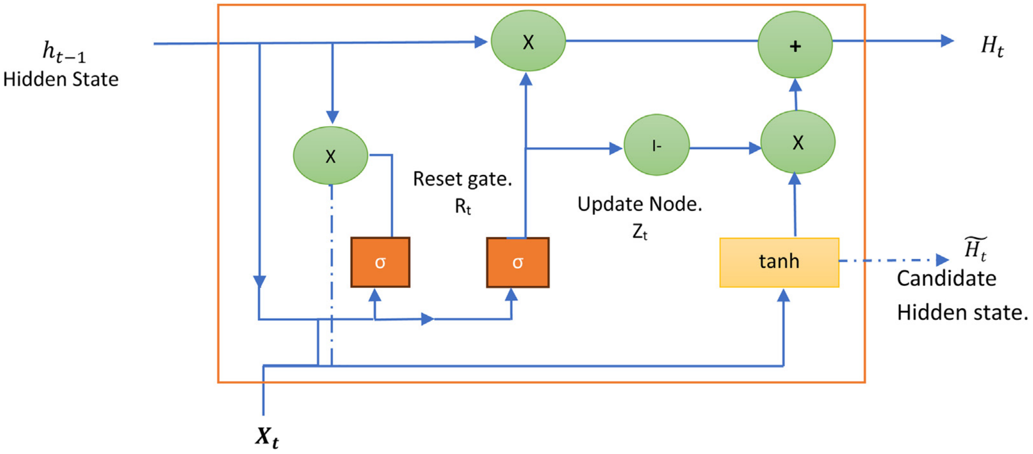

Fig. 8. Architecture of Gated Recurrent Unit.

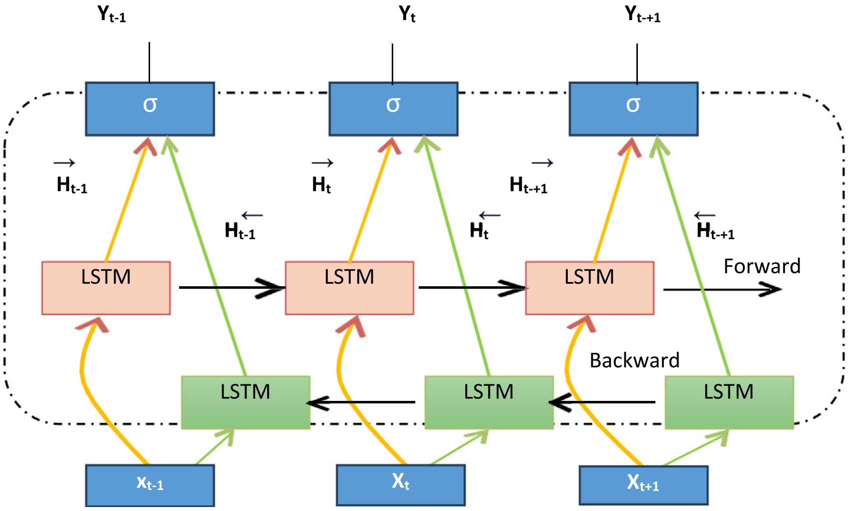

Fig. 9. The architecture of Bi-LSTM.

Here, W_z denotes the weight matrix at (z_t), $(h_{t-1},\ x_t)$ is the concatenation of the prior hidden state h_{t-1} with the current input x_t, where σ is the sigmoid activation function.

$$r_t = \sigma\left(W_r * \ \left(h_{t-1},\ x_t\right)\right) \tag{11}$$

W_r is the weight matrix at (z_t).

The current memory $(\tilde{h}_t)$ represents a possible hidden state that includes the reset gate. It mixes the new input and the previous concealed state.

$$\tilde{h}_t = \tanh\left(W * \ \left(r_t * \ h_{t-1},\ x_t\right)\right) \tag{12}$$

W is the weight matrix, and tanh is the hyperbolic tangent activation function.

The final hidden state (h_t) is a mixture of the prior hidden state h_{t-1} and the current memory content $\tilde{h}_t$, operated by the update gate.

$$h_t = \left(1 - z_t\right) * h_{t-1} + \ z_t \ * \ \tilde{h}_t \tag{13}$$

Bidirectional LSTM

In a basic LSTM, the hidden outputs are passed between LSTM layers to both neighboring LSTM cells and used as inputs for the subsequent LSTM cell [22,74]. A Bidirectional LSTM architecture enhances this by allowing information to flow forward and backward (Fig. 9). In a Bidirectional LSTM, the forward flow captures system variations, while the backward flow helps to smooth

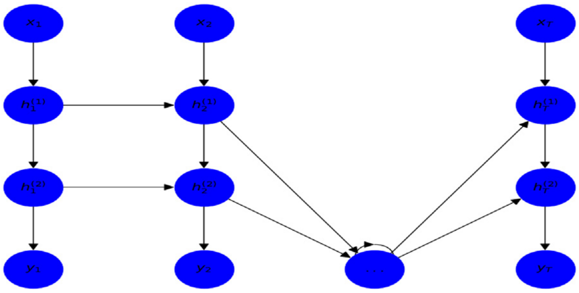

Fig. 10. The architecture of stacked LSTM.

the predictions [84]. The outputs from the forward and backward paths are then concatenated. The output y_i is obtained by combining the results from both directions. The governing equations for a Bidirectional LSTM are as follows:

$$H_{i1} = f\ \left(U_1 *\ x_i + W_1 *\ h_{i-1}\right) \tag{14}$$

$$H_{i2} = f\ \left(U_2 *\ x_i + W_2 *\ h_{i-1}\right) \tag{15}$$

$$y_i = \sigma\left(V\ *\ \left[h_{i1}\ *\ h_{i2}\right]\right) \tag{16}$$

Here, $[h_{i1}\ *\ h_{i2}\ ]$ is a hidden state at time step iii for the forward LSTM layer, and “f” is the activation function, such as ReLU or tanh, applied to the weighted sums. U_1 *and* U_2 are weight matrices for the input-to-hidden connections in the forward LSTM layer. W_1 *and* W_2 are weight matrix for the hidden-to-hidden connections in the forward LSTM layer and V weight matrix used to transform the concatenated hidden states from the forward and backward LSTM layers.

Stacked LSTM

Stacked LSTM networks are composed of multiple LSTM layers stacked on top of one another, which is the model used to detect more complicated patterns and connections in the data [85,86]. Each LSTM layer evaluates data sequences and feeds its output into the next layer[85]. In a stacked LSTM network, the hidden state output of one LSTM layer is used as input for the next LSTM layer, as shown in Fig. 10.

Mathematically, the first LSTM layer:

$$f_t^1\ =\ \sigma\ \left(\left(W_f^1\right) * \ \left(h_{t-1}^1,\ x_t\right) + b_f^1\right) \tag{17}$$

$$i_t^1 = \sigma\ \left(\left(W_i^1\right)\ *\ \ \left(h_{t-1}^1,\ x_t\right) + b_i^1\right) \tag{18}$$

$$\tilde{C}_t^1\ = \tanh\left(\left(W_c^1\right)\ *\ \ \left(h_{t-1}^1,\ x_t\right) + b_c^1\right) \tag{19}$$

$$C_t^1\ = f_t^1\ \left(c_{t-1}^1,\ i_t\right) + b_c^1\ *\ \ \tilde{C}_t^1 \tag{20}$$

$$o_t^1\ = \sigma\ \left(\left(W_o^1\right)\ *\ \ \left(h_{t-1}^1,\ x_t\right) + b_o^1\right) \tag{21}$$

$$h_t^1\ = o_t^1\ * \tanh\left(C_t^1\right) \tag{22}$$

Mathematically, the second LSTM layer:

$$f_t^2 = \sigma\ \left(\left(W_f^2\right)\ *\ \ \left(h_{t-1}^2,\ h_t^1\right) + b_f^2\right) \tag{23}$$

$$i_t^2 = \sigma\ \left(\left(W_i^2\right)\ *\ \ \left(h_{t-1}^2,\ h_t^1\right) + b_i^2\right) \tag{24}$$

$$\tilde{C}_t^2\ = \tanh\left(\left(W_c^2\right)\ *\ \ \left(h_{t-1}^2,\ h_t^1\right) + b_c^2\right) \tag{25}$$

$$C_t^2\ = f_t^2\ \left(c_{t-1}^2,\ i_t^2\right) + b_c^1\ *\ \ \tilde{C}_t^1 \tag{26}$$

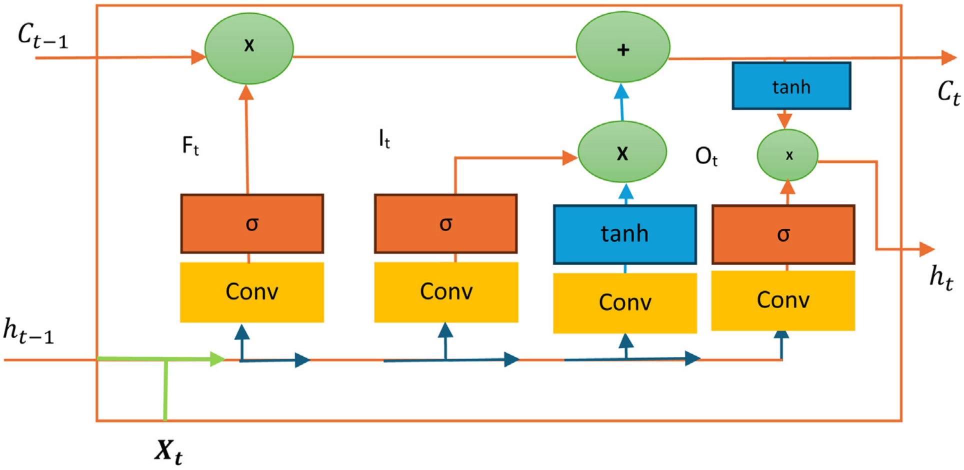

Fig. 11. The architecture of Convolutional LSTM memory cell.

$$o_t^2 = \sigma\left(\left(W_o^2\right) * \left(h_{t-1}^2, x_t\right) + b_o^2\right) \tag{27}$$

$$h_t^2 = o_t^2 * \tanh\left(C_t^2\right) \tag{28}$$

Output Layer:

$$y_t = softmax\left(V.\, h_t^2 + b_y\right) \tag{29}$$

Where, h_t^1, h_t^2 are the hidden states of the first and second LSTM layers, respectively, at time step t. C_t^1. C_t^2 are the cell states of the first and second LSTM layers, respectively, at time step t. W and V are weight matrices.

Convolutional LSTM

Convolutional LSTM (ConvLSTM) networks enhance the regular LSTM [87] by introducing convolutional elements into the LSTM design to handle spatiotemporal data [88], as shown in Fig. 11. ConvLSTM excels in spatially organized data applications like precipitation forecasting [89] and hydrological modeling [64]. ConvLSTM replaces LSTM's fully linked operations with convolutional operations. It enables the network to capture spatial and temporal dependencies in the data [88]. The ConvLSTM architecture can be stated as follows:

Input Gate:

$$i_t = \sigma\left(W_{xi} * X_t + W_{hi} * H_{t-1} + W_{ci} * C_{t-1} + b_i\right) \tag{30}$$

Forget Gate:

$$f_t = \sigma\left(W_{xf} * X_t + W_{hf} * H_{t-1} + W_{cf} * C_{t-1} + b_f\right) \tag{31}$$

Cell State:

$$C_t = f_t * C_{t-1} + i_t * \tanh\left(W_{xc} * X_t + W_{hc} * H_{t-1} + * b_c\right) \tag{32}$$

Output Gate:

$$O_t = \sigma\left(W_{xo} * X_t + W_{ho} * H_{t-1} + W_{co} * C_{t-1} + b_o\right) \tag{33}$$

Hidden State:

$$H_t = o_t * \tanh C_t \tag{34}$$

Table 1
Overview of implementation of different hybrid LSTM models in time series prediction.

Hybrid Model	Purpose	Country/Region	Year	Study Findings	Limitations	Sources
LSTM-CNN	Rainfall Prediction	India	2022	Improved accuracy in capturing spatial features and temporal dependencies	High computational cost	[90]
LSTM-SVR	Streamflow forecasting	Malaysia	2023	Enhanced prediction accuracy by combining LSTM's temporal modeling with SVR's non-linear relationship handling	Sensitivity to hyperparameters	[64]
Wavelet decomposed STM	Precipitation Forecasting	Thailand	2024	Enhanced prediction accuracy by decomposed wavelet LSTM and well-captured non-linear relationship handling	May require more data for training	[29]
LSTM-SOM	Anomaly detection	Japan	2019	Improved anomaly detection using SOM to cluster input features before feeding into LSTM	Dependency on the quality of input data	[91]
LSTM-GRU	Short-term runoff prediction	China	2020	Enhanced computational efficiency compared to traditional LSTM while maintaining prediction accuracy.	It requires more data for training compared to simpler models	[92]
LSTM-DNN	Streamflow Prediction	China	2023	Improved generalization and robustness in predicting streamflow using deep neural networks	Sensitivity to architecture design choices	[93]
LSTM-Attention	Multi-step ahead prediction	South Korea	2023	Enhanced performance in multi-step ahead streamflow forecasting by attending to relevant input features	Complexity in tuning attention mechanism parameters	[94]
LSTM-Transformer	Time series power forecasting	China	2023	Combined advantages of LSTM's sequential modeling and Transformer's self-attention mechanism for improved accuracy	Higher computational cost and memory requirements	[95]
LSTM-SVM	Water quality prediction	Algeria	2022	Enhanced prediction accuracy for water quality parameters by integrating SVM with LSTM	Sensitivity to kernel and regularization parameters	[96]
LSTM-RNN	Extreme event prediction	China	2021	Improved prediction of extreme hydrological events using a recursive neural network in conjunction with LSTM	Limited interpretability of complex model structures	[97,98]

Hybrid models in time series prediction

Many hybrid LSTM models have been implemented for time series prediction in recent years, as mentioned in Table 1. The LSTM-CNN model, applied for rainfall prediction in India in 2022, improved accuracy by capturing spatial features and temporal dependencies, though at the cost of high computational requirements [90]. Similarly, the LSTM-SVR model, used for streamflow forecasting in Malaysia in 2023, enhanced prediction accuracy by combining LSTM's temporal modeling capabilities with SVR's handling of non-linear relationships despite its sensitivity to hyperparameter settings [64]. The wavelet decomposed LSTM model for precipitation forecasting in Thailand achieved higher accuracy than other models by capturing non-linear relationships through wavelet decomposition, while it requires more data for practical training [29]. In Japan, the LSTM-SOM model, developed for anomaly detection in 2019, showed improved performance by clustering input features with SOM before feeding them into LSTM, but dependency on input data quality was a limitation [91]. The LSTM-GRU hybrid model, designed for short-term runoff prediction in China in 2020, enhanced computational efficiency and prediction accuracy compared to traditional LSTM models. It necessitated more training data than simpler models [92]. For streamflow prediction in China in 2023, the LSTM-DNN model demonstrated improved generalization and robustness by integrating deep neural networks with LSTM. It was sensitive to the design of its architecture [93]. The LSTM-Attention model applied for multi-step ahead prediction in South Korea in 2023 advanced from an attention mechanism that focused on relevant input features, enhancing forecasting performance, and complexity required careful tuning of attention parameters [94]. The LSTM-Transformer model, used for time series power forecasting in China in 2023, combined LSTM's sequential modeling with the Transformer's self-attention mechanism to improve accuracy but at a higher computational and memory cost [95]. In Algeria in 2022, the LSTM- Support Vector Machine (SVM) model for water quality prediction had good accuracy by integrating SVM with LSTM, though it was sensitive to kernel and regularization parameter choices [96]. Lastly, the LSTM-RNN model for extreme event prediction in China in 2021 improved forecasts of extreme hydrological events by combining RNN with LSTM, yet its complex structure limited interpretability [97,98].

This comparative analysis evaluates the efficacy of RNNs, LSTM, and GRUs in hydrological forecasting. LSTM networks have superior performance over traditional RNNs and GRUs due to their advanced memory cells and gating mechanisms, which enhance their ability to capture long-term dependencies essential for hydrological predictions [93]. These attributes enable LSTMs to manage complex temporal relationships effectively, providing robust forecasts for streamflow and rainfall. In contrast, GRUs, although simpler by integrating input and forget gates into a single update gate, offer a balance between computational efficiency and performance, making them suitable for scenarios with limited resources [94]. Despite their efficiency, GRUs slightly lag LSTMs in capturing intricate dependencies. Traditional RNNs, constrained by vanishing and exploding gradient issues, are less effective for long-term forecasting but remain helpful for short-term predictions due to their simple structure and faster training times. LSTMs have been widely adopted in hydrological applications, including precipitation forecasting and flood prediction, owing to their capacity for handling long-term dependencies [29]. GRUs also show promise in hydrological predictions, providing a viable alternative to LSTMs in resource-constrained environments. RNNs, while less effective for long-term tasks, are valuable for short-term forecasts and baseline comparisons [99].

RNNs, LSTMs, and GRUs, as well as their derivatives, have been applied to a different hydrological problem (rainfall-runoff prediction); their advantages and shortcomings are shown in Table 2. RNNs have demonstrated efficacy in capturing temporal dependencies across various regions, including India and Taiwan, where they have been used for monthly and daily rainfall-runoff modeling. Despite their strength in short-term predictions and managing noisy data, RNNs are limited by the vanishing gradient problem, which hampers their ability to capture long-term dependencies, as observed in applications within the Bardha Watershed and North Carolina (e.g., [100,101]). LSTM networks, an advanced variant of RNNs, have been applied extensively across different geographical regions, including the UK and the USA, and are notably effective in addressing the vanishing gradient problem. LSTMs are proficient in capturing long-term dependencies and managing complex temporal relationships, making them suitable for diverse hydrological time series applications. However, their computational intensity and sensitivity to hyperparameters are significant drawbacks, as seen in studies involving watersheds in Iowa and the Brays Bayou watershed, Texas ([60,102]). GRU networks, a more simplified variant, have shown promise in short-term runoff prediction with faster training times and comparable performance to LSTM models. GRUs are particularly effective in real-time applications and managing high-dimensional data, as demonstrated in regions like Southeast China and Bhutan. However, their simplified architecture may struggle with capturing long-term dependencies, and their effectiveness diminishes with smaller datasets, as highlighted in studies conducted in the Fujian Province and the Wei River Basin, Shaanxi, China ([92,103]). The effectiveness of these networks varies depending on the application context, with each model exhibiting specific advantages and limitations that must be considered for optimal hydrological modeling outcomes.

Computational efficiency and resource consumption in hydrological prediction

The computational efficiency and resource consumption of RNNs, LSTMs, and GRUs in hydrological forecasting vary when dealing with large-scale datasets. Traditional RNNs, due to their simpler architecture, tend to have lower computational requirements and faster training times, making them suitable for smaller datasets or short-term predictions [79]. However, their inability to capture long-term dependencies efficiently limits their applicability in large-scale, complex hydrological forecasting [22].

LSTMs, equipped with advanced memory cells and gating mechanisms, excel in capturing long-term dependencies and non-linear relationships, but this comes at a high computational [78]. Their training times are longer and require substantial computational resources in large-scale applications [67]. LSTMs also demand more memory due to their complex architecture, which can be a limiting factor when working with extensive datasets [42].

GRUs offer a balance between LSTMs and RNNs, providing improved computational efficiency while still retaining some ability to capture longer dependencies [81]. GRUs have fewer parameters than LSTMs, which results in faster training times and lower resource consumption [113]. However, GRUs may struggle with very long-term dependencies and often require larger datasets for optimal performance. GRUs offer a middle ground for hydrological forecasting with more favorable computational efficiency than LSTMs [114].

Effectiveness to outliers, noisy data, and missing data

RNNs, LSTMs, and GRUs exhibit varying levels of robustness to outliers, noisy data, and missing data, which are common in time series datasets. RNNs tend to struggle the most with these issues due to their simple architecture. While they can capture temporal dependencies, RNNs are vulnerable to outliers, which can distort the learning process and lead to inaccurate predictions [22]. Their performance also decreases in the presence of noisy data, as they lack advanced mechanisms to filter out irrelevant variations [115]. Missing data further complicates RNN. It needs imputation techniques or data preprocessing before training to avoid model failure [116].

LSTM networks, in contrast, are more resilient to outliers and noise due to their memory cells and gating mechanisms, which allow them to selectively retain relevant information and filter out less critical data points. This capacity makes LSTMs more effective at handling noisy datasets commonly found in hydrological applications, such as rainfall-runoff modeling and streamflow forecasting [67]. LSTMs are also better equipped to manage missing data due to their ability to maintain context over long sequences, making them less reliant on continuous data [50]. However, even with their robustness, LSTMs can be sensitive to extreme outliers, which still necessitate preprocessing techniques such as outlier removal or normalization to maintain performance [43].

GRUs, like LSTMs, exhibit a higher tolerance to noise and missing data compared to traditional RNNs. Their simpler architecture, while faster and more computationally efficient, includes mechanisms to retain essential information, which helps them maintain stability in the presence of data inconsistencies [117]. GRUs handle noisy data effectively in applications like drought prediction and runoff modeling [101] but are less effective than LSTMs when managing complex temporal relationships in particularly noisy or incomplete datasets [113].

Impact of hyperparameter tuning on prediction performance

Hyperparameter tuning is critical in determining the prediction performance of RNNs, LSTMs, and GRUs [118]. The selection of appropriate hyperparameters, such as the number of layers, the size of hidden states, learning rate, batch size, and dropout rate, can

Table 2
RNN, LSTM, GRU variants in hydrological (Rainfall-runoff) times series predictions.

Network Type	Application	Input Data	Output	Study Region	Effectiveness	Advantages	Disadvantages	Source
RNN	Monthly Rainfall-runoff modeling	Historical monthly rainfall and runoff data	Predicted runoff	Bardha Watershed, India	RNN-based model outperformed in both training and testing periods.	Captures temporal dependencies in data	vanishing gradient problem	[100]
	Rainfall-runoff	Historical rainfall data	Predicted runoff	Five small Amazon Basins	RNN-based models are capable of spatial and temporal changes that influence long-term simulations	Suitable for short-term predictions	High requirement for historical data	[104]
	Rainfall-runoff	Gridded and Meteorological Historical rainfall and runoff data	Predicted daily runoff	North Carolina, USA	RNN models demonstrated the superiority of LSTM and GRU.	Effective in capturing seasonal variations	Limited ability to capture long-term dependencies	[101]
	Reconstruction of rainfall-runoff processes	Historical rainfall data	Predicted runoff	Taiwan's Wu-Tu watershed	RNN-based models are appropriate for the reconstruction of rainfall-runoff processes	Timely predictions, managing noisy data	Requires significant data preprocessing	[105]
	Rainfall-runoff	Historical rainfall and runoff data	Predicted runoff	Daning and Lushui River Basins, China	The proposed showed promising results in both river basins	Robust against data irregularities	Susceptible to outliers	[106]
LSTM	Rainfall-runoff	Historical rainfall and runoff data	Predicted runoff	Two watersheds in the State of Iowa, USA	The proposed model is effective for short-term flood forecast applications	Addresses vanishing gradient problem	Computationally intensive, requires careful tuning of hyperparameters	[60]
	Rainfall-runoff	Grided and meteorological observations and streamflow records and evaporation	Predicted runoff	241 catchments in the UK	Findings showed the potential of the LSTM for hydrological modeling applications.	Effective in capturing long-term dependencies	High computational cost	[107]
	Rainfall-runoff	CAMELS and meteorological Historical datasets	Predicted runoff	Various River basins in the UK	The proposed model improved the performance of daily runoff prediction and multistep ahead prediction	Manages complex temporal relationships	Vulnerable to overfitting	[108]
	Rainfall-runoff	Historical rainfall data	Predicted runoff	Brays Bayou watershed, Texas	LSTM model improved the outputs	Suitable for various hydrological time series applications	Sensitivity to initial conditions	[102]
	Rainfall-runoff Simulation	Hourly Historical rainfall and runoff data	Predicted runoff	3 water basins from different provinces of China	The proposed LSTM model showed better simulation and was more intelligent than the ANN model	Effective in learning non-linear relationships	Requires significant data preprocessing	[109]
GRU	Short-term runoff prediction	Hourly Historical precipitation and runoff dataset	Predicted runoff	Fujian Province, Southeast China	GRU performs better than LSTM in some cases.	Simplified architecture compared to LSTM	May struggle with long-term dependencies	[92]
	Rainfall-runoff prediction	Historical rainfall data	Predicted runoff	Simtokha, Bhutan	The proposed model outperformed LSTM	Faster training times	Less effective with small datasets	[110]
	Runoff Prediction	Historical meteorological and runoff data	Predicted runoff	Bailong River watershed, China	The developed model has better predictivity and adaptability.	Robust performance in real-time applications	Limited interpretability	[103]
	Runoff Prediction	Historical rainfall data	Predicted runoff	Wei River Basin, Shaanxi, China	The changes in this study enhanced the performance of the GRU runoff forecasting model.	Comparable performance to LSTM	Requires more data for training	[111]
	Runoff Estimation	Historical rainfall and runoff data	Predicted runoff	India	The result suggests that GRU's result is at par with its compatriot LSTM model	Manages high-dimensional data effectively	Limited by network depth and complexity	[112]

impact a model's ability to capture temporal dependencies and generalize to unseen data [119,120]. Poorly tuned hyperparameters can result in underfitting or overfitting, where the model either fails to capture essential patterns or fits too closely to the training data, leading to poor generalization [121].

For RNNs, hyperparameter tuning is more straightforward compared to LSTMs and GRUs [122]. However, due to their susceptibility to the vanishing gradient problem, tuning learning rates and the number of hidden units are critical to stability and convergence. In the case of LSTMs, the challenge of tuning hyperparameters increases due to their complex architecture. The number of memory cells, the size of the hidden state, and the learning rate must be carefully optimized to balance performance and computational cost [65]. LSTMs are sensitive to the choice of hyperparameters, and improper tuning can exacerbate issues such as overfitting or instability during training [30].

GRUs, with their simple structure, are easier to tune than LSTMs but still require careful optimization of parameters such as the learning rate, number of layers, and batch size [123]. Due to their faster training times and reduced computational demands, GRUs are often more forgiving when it comes to hyperparameter selection [123,124]. However, as with LSTMs, suboptimal tuning can hinder the model's ability to capture long-term dependencies and lead to reduced prediction accuracy [78].

RNNs, LSTMs, and GRUs differ in their robustness to noisy, outlier-laden, or missing data. Their performance is highly dependent on effective hyperparameter tuning. The right combination of hyperparameters is crucial for unlocking the full potential of these models, particularly for complex hydrological forecasting tasks.

Potential limitations and factors affecting hydrological time series prediction

The vanishing gradient problem of RNNs delays the ability to capture long-term dependencies, which results in low accuracy in long-term predictions [5]. LSTMs address these issues but are computationally intensive and prone to overfitting with small datasets [64]. GRUs offer a balance but may struggle with complex, long-term dependencies and require larger datasets for optimal performance [117]. Familiar sources of error in these models include inaccurate input data, insufficient training data, model overfitting, and sensitivity to hyperparameter settings [120]. Solutions to these limitations involve using regularization techniques like dropout to reduce overfitting, employing data augmentation to enhance training sets, and optimizing hyperparameters through cross-validation [120]. Factors affecting hydrological time series prediction include the quality of input data [125] (e.g., historical rainfall or streamflow [30]), data preprocessing methods, model architecture, and the length of temporal sequences. External factors such as climate variability, land-use changes, and human interventions (e.g., data collection and handling [50]) also play a crucial role [126]. Tackling these factors requires incorporating diverse data sources, employing advanced preprocessing techniques (e.g., wavelet decomposition [65]), and utilizing ensemble methods to enhance robustness in hydrological forecasting.

Answers to RQs

RQ-1: What are the key architectural differences and advancements between traditional RNNs, LSTM networks, and GRU networks, and how do these differences impact their performance in hydrological predictions?

Traditional RNNs had a simple architecture consisting of a single recurrent layer that fed its output back into itself, allowing them to capture temporal dependencies. However, they suffered from the vanishing gradient problem, which hindered their ability to learn long-term dependencies. LSTM networks addressed this issue with a more complex architecture that included memory cells and gates (input, output, and forget) to regulate the flow of information, making them more effective at capturing long-term dependencies. GRU networks, a variant of LSTMs, simplified this architecture by combining the forget and input gates into a single update gate and merging the cell state and hidden state. These advancements enhanced the performance of LSTMs and GRUs in hydrological predictions by improving their ability to model complex temporal patterns and handle long-term dependencies more efficiently.

RQ-2: How have LSTM networks been adapted or modified to improve their accuracy and efficiency in hydrological time series forecasting, and what innovative methodologies have been introduced in recent research?

LSTM networks have been adapted in various ways to enhance their accuracy and efficiency in hydrological time series forecasting. One common approach is the integration of attention mechanisms, which allow the model to focus on relevant parts of the input sequence, thereby improving prediction accuracy. Hybrid models combining LSTMs with CNNs have been introduced to capture both spatial and temporal dependencies in hydrological data. Wavelet transform has been employed to decompose the time series into different frequency components before feeding them into LSTM networks, enhancing the model's ability to capture multi-scale features. Recent research also explores the use of encoder-decoder architectures and ensemble learning techniques to combine multiple LSTM models, leading to more robust and accurate predictions in hydrological forecasting.

RQ-3: In what ways have RNN, LSTM, and GRU models been implemented in hydrological applications, and what case studies highlight their effectiveness and limitations in real-world scenarios?

RNN, LSTM, and GRU models have been implemented in various hydrological applications, including streamflow prediction, rainfall forecasting, and water quality monitoring. For example, LSTM networks have been used to predict streamflow in river basins, demonstrating superior performance over traditional models due to their ability to capture complex temporal dependencies. In rainfall forecasting, hybrid models combining LSTMs with wavelet transforms have shown enhanced accuracy by effectively decomposing and modeling the time series data. However, these models face limitations, such as the need for large datasets for training and computational intensity. Case studies, such as the application of LSTM models in the Yangtze River basin for water quality prediction, highlight their effectiveness in capturing intricate patterns in hydrological data. They also underscore the challenges in model generalization and scalability across different hydrological settings.

RQ-4: What are the significant trends in the application of hybrid models in hydrological predictions over the past decade?

Over the past decade, significant trends in the application of hybrid models for hydrological predictions have emerged, focusing on integrating various machine learning techniques to enhance prediction accuracy. A prominent trend is the combination of LSTM networks with CNNs to capture both spatial and temporal dependencies in hydrological data, leading to improved performance in tasks like rainfall and streamflow forecasting. Another trend is the use of wavelet transforms to decompose time series data into different frequency components before feeding them into LSTM models, allowing for better multi-scale feature extraction. Ensemble learning techniques, which combine multiple models to improve robustness and accuracy, have also gained traction. These hybrid approaches have shown considerable promise in addressing the complex and non-linear nature of hydrological systems, leading to more reliable and accurate predictions.

RQ-5: What gaps exist in the current literature on the use of RNNs, LSTMs, and GRUs for hydrological forecasting, and what future research directions are suggested to address these gaps and enhance the models' predictive capabilities?

Despite significant advancements, several gaps exist in the current literature on using RNNs, LSTMs, and GRUs for hydrological forecasting. One crucial gap is the limited understanding of these models' interpretability, making it challenging to explain and trust their predictions. Additionally, there is a need for more research on the generalization capabilities of these models across different hydrological settings and regions. Data scarcity and quality also pose significant challenges, as high-quality, extensive datasets are crucial for training these models effectively. Future research should focus on developing more interpretable model architectures and enhancing data collection and preprocessing methods. Exploring the integration of domain knowledge into model training and leveraging transfer learning techniques can help improve model generalization. Addressing these gaps will enhance the predictive capabilities of RNNs, LSTMs, and GRUs in hydrological forecasting.

Conclusion

This review has examined the different variants and applications of RNN, LSTM, and GRU architectures in hydrological time-series forecasting.

- Traditional RNNs, although foundational in temporal sequence modeling, are hindered by the vanishing gradient problem, restricting their ability to capture long-term dependencies. LSTM networks address this limitation through advanced architecture, incorporating memory cells and gating mechanisms to enhance their capacity for modeling extended temporal dependencies. GRU networks, as a streamlined variant of LSTMs, integrate the forget and input gates into a single update gate, offering comparable performance with improved computational efficiency.
- Recent innovations, such as hybrid models combining LSTMs with other ML and DL models (e.g., CNNs), have shown improved capabilities. The use of wavelet transforms to decompose time-series data before LSTM processing has further enhanced these models' ability to capture multi-scale features. Encoder-decoder architectures and ensemble learning techniques, which aggregate multiple LSTM models, have also been explored, contributing to more robust and precise predictions.
- Despite the promising outcomes of these models in applications such as streamflow prediction, rainfall forecasting, and water quality assessment, challenges persist regarding the need for extensive training datasets and the high computational demands of these models. As the field advances, addressing issues related to model interpretability, generalization, and data quality will be critical for improving the scalability and applicability of these architectures across diverse hydrological contexts.

Future research should focus on predicting and understanding the discrepancies between real and simulated environments to enhance model accuracy and applicability. By identifying the factors that cause these differences, researchers can develop more robust and generalizable models. This approach will be crucial in refining simulation techniques, improving data quality, and ensuring that models better reflect real-world hydrological systems.

Limitations

This review focuses on architectural and methodological innovations in RNN, LSTM, and GRU variants, with limited research on practical implementation challenges. Further empirical validation across diverse hydrological settings would be necessary to substantiate the findings and ensure their generalizability.

Ethics statements

No data was used in this research.

Funding

No Funding was used in this research.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgments

The authors would like to express their gratitude to The Joint Graduate School of Energy and Environment (JGSEE), King Mongkut's University of Technology Thonburi and the Center of Excellence on Energy Technology and Environment (CEE), Ministry of Higher Education, Science (MHESI), Research and Innovation for providing us with this opportunity.

This research project is supported by King Mongkut's University of Technology Thonburi (KMUTT), Thailand Science Research and Innovation (TSRI), and the National Science, Research and Innovation Fund (NSRF) Fiscal year 2024.

References

[1] M.M. Najafabadi, et al., Deep learning applications and challenges in big data analytics, J. Big Data 2 (2015) 1–21.
[2] M. Schuster, K.K. Paliwal, Bidirectional recurrent neural networks, IEEE Transact. Signal Process. 45 (11) (1997) 2673–2681.
[3] H. Salehinejad, et al., arXiv preprint, 2017.
[4] L.R. Medsker, L. Jain, Recurrent neural networks, Des. Applic. 5 (64–67) (2001) 2.
[5] S. Grossberg, Recurrent neural networks, Scholarpedia 8 (2) (2013) 1888.
[6] A.L. Caterini, et al., Recurrent neural networks, Deep Neur. Netw. Math. Framew. (2018) 59–79.
[7] L. Medsker, L.C. Jain, Recurrent Neural networks: Design and Applications, CRC press, 1999.
[8] M. Grünig, et al., Applying deep neural networks to predict incidence and phenology of plant pests and diseases, Ecosphere 12 (10) (2021) e03791.
[9] C. Szegedy, A. Toshev, D. Erhan, Deep neural networks for object detection, Adv. Neur. Inf. Process. Syst. (2013) 26.
[10] V. Sze, et al., Efficient processing of deep neural networks: a tutorial and survey, Proceed. IEEE 105 (12) (2017) 2295–2329.
[11] W. Samek, et al., Explaining deep neural networks and beyond: a review of methods and applications, Proceed. IEEE 109 (3) (2021) 247–278.
[12] H. Larochelle, et al., Exploring strategies for training deep neural networks, J. Mach. Learn. Res. 10 (1) (2009).
[13] G. Bebis, M. Georgiopoulos, Feed-forward neural networks, IEEE Potent. 13 (4) (1994) 27–31.
[14] D. Svozil, V. Kvasnicka, J. Pospichal, Introduction to multi-layer feed-forward neural networks, Chemometr. Intell. Lab. Syst. 39 (1) (1997) 43–62.
[15] X. Glorot, Y. Bengio, Understanding the difficulty of training deep feedforward neural networks, in: Proceedings of the thirteenth international conference on artificial intelligence and statistics, 2010 JMLR Workshop and Conference Proceedings.
[16] K. O'shea, R. Nash, arXiv preprint, 2015.
[17] A. Vedaldi, K. Lenc, Matconvnet: convolutional neural networks for matlab, in: Proceedings of the 23rd ACM international conference on Multimedia, 2015.
[18] J. Gu, et al., Recent advances in convolutional neural networks, Pattern Recognit. 77 (2018) 354–377.
[19] T. Kattenborn, et al., Review on Convolutional Neural Networks (CNN) in vegetation remote sensing, ISPRS J. Photogrammetr. Remote Sens. 173 (2021) 24–49.
[20] Z. Li, et al., A survey of convolutional neural networks: analysis, applications, and prospects, IEEE Trans. Neur. Netw. Learn. Syst. 33 (12) (2021) 6999–7019.
[21] J. Yu, A. de Antonio, E. Villalba-Mora, Deep learning (CNN, RNN) applications for smart homes: a systematic review, Computers 11 (2) (2022) 26.
[22] R. DiPietro, G.D. Hager, in: *Deep learning: RNNs and LSTM*, in *Handbook of Medical Image Computing and Computer Assisted Intervention*, Elsevier, 2020, pp. 503–519.
[23] J.L. Elman, Finding structure in time, Cogn. Sci. 14 (2) (1990) 179–211.
[24] B. Šter, Selective recurrent neural network, Neural Process. Lett. 38 (2013) 1–15.
[25] S. Hochreiter, J. Schmidhuber, Long short-term memory, Neural Comput 9 (8) (1997) 1735–1780.
[26] B. Ghojogh, A. Ghodsi, arXiv preprint, 2023.
[27] A. Graves, J. Schmidhuber, Framewise phoneme classification with bidirectional LSTM and other neural network architectures, Neur. Netw. 18 (5–6) (2005) 602–610.
[28] A. Graves, N. Jaitly, A.-r. Mohamed, Hybrid speech recognition with deep bidirectional LSTM, 2013 IEEE workshop on automatic speech recognition and understanding, IEEE, 2013.
[29] M. Waqas, et al., Advancements in daily precipitation forecasting: a deep dive into daily precipitation forecasting hybrid methods in the tropical climate of Thailand, MethodsX (2024) 102757.
[30] M. Waqas, et al., Incorporating novel input variable selection method for in the different water basins of Thailand, Alexandr. Eng. J. 86 (2024) 557–576.
[31] K. Johny, M.L. Pai, S. Adarsh, A multivariate EMD-LSTM model aided with Time Dependent Intrinsic Cross-Correlation for monthly rainfall prediction, Appl. Soft. Comput. 123 (2022) 108941.
[32] R. Shashidhar, S. Patilkulkarni, S. Puneeth, Combining audio and visual speech recognition using LSTM and deep convolutional neural network, Int. J. Inform. Technol. 14 (7) (2022) 3425–3436.
[33] S. Bhaskar, T. Thasleema, LSTM model for visual speech recognition through facial expressions, Multimed. Tool. Appl. 82 (4) (2023) 5455–5472.
[34] Q. Wang, et al., A novel privacy-preserving speech recognition framework using bidirectional LSTM, J. Cloud Comput. 9 (1) (2020) 36.
[35] R. Quan, et al., Holistic LSTM for pedestrian trajectory prediction, IEEE Transact. Image Process. 30 (2021) 3229–3239.
[36] G. Xie, et al., Motion trajectory prediction based on a CNN-LSTM sequential model, Sci. China Inform. Sci. 63 (2020) 1–21.
[37] X. Song, et al., Pedestrian trajectory prediction based on deep convolutional LSTM network, IEEE Transact. Intell. Transport. Syst. 22 (6) (2020) 3285–3302.
[38] L. Rossi, et al., Vehicle trajectory prediction and generation using LSTM models and GANs, PLoS ONE 16 (7) (2021) e0253868.
[39] X. Peng, Y. Li, A fusion method based on EEMD, Pearson correlation analysis, improved LSTM, and Gaussian function-Trust region algorithm for state of health Prediction of lithium-ion Batteries, J. Electrochem. Energy Convers. Storage 19 (3) (2022) 030903.
[40] J.Q. Wang, Y. Du, J. Wang, LSTM based long-term energy consumption prediction with periodicity, Energy 197 (2020) 117197.
[41] A.K. Dubey, et al., Study and analysis of SARIMA and LSTM in forecasting time series data, Sustain. Energy Technolog. Assessm. 47 (2021) 101474.
[42] J. Wei, et al., Ultra-short-term forecasting of wind power based on multi-task learning and LSTM, Int. J. Electr. Power Energy Syst. 149 (2023) 109073.
[43] M. Waqas, et al., A deep learning perspective on meteorological droughts prediction in the Mun River Basin, Thailand, AIP Adv. 14 (8) (2024).
[44] S. Aslam, et al., A survey on deep learning methods for power load and renewable energy forecasting in smart microgrids, Renew. Sustain. Energy Rev. 144 (2021) 110992.
[45] I.-F. Kao, et al., Exploring a Long Short-Term Memory based Encoder-Decoder framework for multi-step-ahead flood forecasting, J. Hydrol. 583 (2020) 124631.
[46] B.T. Pham, et al., Can deep learning algorithms outperform benchmark machine learning algorithms in flood susceptibility modeling? J. Hydrol. 592 (2021) 125615.

[47] L. Ni, et al., Streamflow and rainfall forecasting by two long short-term memory-based models, J. Hydrol. 583 (2020) 124296.
[48] J. Li, et al., Water quality soft-sensor prediction in anaerobic process using deep neural network optimized by Tree-structured Parzen Estimator, Front. Environ. Sci. Eng. 17 (6) (2023) 67.
[49] P. Wongburi, J.K. Park, Prediction of wastewater treatment plant effluent water quality using recurrent neural network (RNN) models, Water (Basel) 15 (19) (2023) 3325.
[50] A. Wangwongchai, et al., Imputation of missing daily rainfall data; A comparison between artificial intelligence and statistical techniques, MethodsX 11 (2023) 102459.
[51] P. Liu, et al., Analysis and prediction of water quality using LSTM deep neural networks in IoT environment, Sustainability 11 (7) (2019) 2058.
[52] Z. Hu, et al., A water quality prediction method based on the deep LSTM network considering correlation in smart mariculture, Sensors 19 (6) (2019) 1420.
[53] R. Barzegar, M.T. Aalami, J. Adamowski, Short-term water quality variable prediction using a hybrid CNN–LSTM deep learning model, Stochast. Environ. Res. Risk Assessm. 34 (2) (2020) 415–433.
[54] S.L. Zubaidi, et al., A method for predicting long-term municipal water demands under climate change, Water Resour. Manag. 34 (2020) 1265–1279.
[55] M. Sit, et al., A comprehensive review of deep learning applications in hydrology and water resources, Water Sci. Technol. 82 (12) (2020) 2635–2670.
[56] Y. Zhang, Utilizing Hybrid LSTM and GRU Models for Urban Hydrological Prediction, University of Guelph, 2024.
[57] S. Yang, et al., Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model, J. Hydrol. 579 (2019) 124229.
[58] Q. Wang, et al., Sequence-based statistical downscaling and its application to hydrologic simulations based on machine learning and big data, J. Hydrol. 586 (2020) 124875.
[59] X.-H. Le, et al., Application of long short-term memory (LSTM) neural network for flood forecasting, Water (Basel) 11 (7) (2019) 1387.
[60] Z. Xiang, J. Yan, I. Demir, A rainfall-runoff model with LSTM-based sequence-to-sequence learning, Water Resour. Res. 56 (1) (2020) e2019WR025326.
[61] Y. Liu, et al., Short term real-time rolling forecast of urban river water levels based on LSTM: a case study in Fuzhou City, China, Int. J. Environ. Res. Public Health 18 (17) (2021) 9287.
[62] S. Ozdemir, S. Ozkan Yildirim, Prediction of Water Level in Lakes by RNN-Based Deep Learning Algorithms to Preserve Sustainability in Changing Climate and Relationship to Microcystin, Sustainability 15 (22) (2023) 16008.
[63] R. Moura, et al., Predicting flood events with streaming data: a preliminary approach with GRU and ARIMA, International Conference on Optimization, Learning Algorithms and Applications, Springer, 2023.
[64] A. Dehghani, et al., Comparative evaluation of LSTM, CNN, and ConvLSTM for hourly short-term streamflow forecasting using deep learning approaches, Ecol. Inform. 75 (2023) 102119.
[65] M. Waqas, et al., Advancements in daily precipitation forecasting: a deep dive into daily precipitation forecasting hybrid methods in the tropical climate of Thailand, MethodsX 12 (2024) 102757.
[66] Z. Wang, Y. Lou, Hydrological time series forecast model based on wavelet de-noising and ARIMA-LSTM, 2019 IEEE 3rd Information Technology, Networking, Electronic and Automation Control Conference (ITNEC), IEEE, 2019.
[67] B.B. Sahoo, et al., Long short-term memory (LSTM) recurrent neural network for low-flow hydrological time series forecasting, Acta Geophys. 67 (5) (2019) 1471–1481.
[68] A.U. Muhammad, X. Li, J. Feng, Using LSTM GRU and hybrid models for streamflow forecasting, in: Machine Learning and Intelligent Communications: 4th International Conference, MLICOM 2019, Nanjing, China, August 24–25, 2019, Proceedings 4, Springer, 2019.
[69] A.R.R. Niknam, et al., Comparing ARIMA and various deep learning models for long-term water quality index forecasting in Dez River, Iran, Environ. Sci. Pollut. Res. (2024) 1–17.
[70] W. Zhang, et al., Outlet water temperature prediction of energy pile based on spatial-temporal feature extraction through CNN–LSTM hybrid model, Energy 264 (2023) 126190.
[71] A. Mazher, Visualization framework for high-dimensional spatio-temporal hydrological gridded datasets using machine-learning techniques, Water (Basel) 12 (2) (2020) 590.
[72] C. Shen, K. Lawson, Applications of deep learning in hydrology, in: Deep Learning for the Earth Sciences: A Comprehensive Approach to Remote Sensing, Climate Science, and Geosciences, 2021, pp. 283–297.
[73] S. Razavi, Deep learning, explained: fundamentals, explainability, and bridgeability to process-based modelling, Environ. Modell. Softw. 144 (2021) 105159.
[74] M. Waqas, et al., Potential of artificial intelligence-based techniques for rainfall forecasting in thailand: a comprehensive review, Water (Basel) 15 (16) (2023) 2979.
[75] J.J. Hopfield, Neural networks and physical systems with emergent collective computational abilities, Proceed. Nat. Acad. Sci. 79 (8) (1982) 2554–2558.
[76] Y. Bengio, P. Simard, P. Frasconi, Learning long-term dependencies with gradient descent is difficult, IEEE Transact. Neur. Netw. 5 (2) (1994) 157–166.
[77] S. Hochreiter, et al., Gradient flow in recurrent nets: the difficulty of learning long-term dependencies, A Field Guide to Dynamical Recurrent Neural Networks, IEEE Press In, 2001.
[78] S. Hochreiter, J. Schmidhuber, LSTM can solve hard long time lag problems, Adv. Neur. Inf. Process. Syst. 9 (1996).
[79] J. Zhao, et al., Do RNN and LSTM have long memory? International Conference on Machine Learning, PMLR, 2020.
[80] K. Cho, et al., arXiv preprint, 2014.
[81] J. Chung, et al., arXiv preprint, 2014.
[82] R. Jozefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, International conference on machine learning, PMLR, 2015.
[83] K. Greff, et al., LSTM: a search space odyssey, IEEE Trans. Neur. Netw. Learn. Syst. 28 (10) (2016) 2222–2232.
[84] Z. Cui, et al., arXiv preprint, 2018.
[85] M. Ma, et al., Predicting machine's performance record using the stacked long short-term memory (LSTM) neural networks, J. Appl. Clin. Med. Phys. 23 (3) (2022) e13558.
[86] J. Wang, B. Peng, X. Zhang, Using a stacked residual LSTM model for sentiment intensity prediction, Neurocomputing 322 (2018) 93–101.
[87] W. Luo, W. Liu, S. Gao, Remembering history with convolutional lstm for anomaly detection, 2017 IEEE International conference on multimedia and expo (ICME, IEEE, 2017.
[88] J. Su, et al., Convolutional tensor-train LSTM for spatio-temporal learning, Adv. Neur. Inf. Process. Syst. 33 (2020) 13714–13726.
[89] S. Kim, et al., arXiv preprint, 2017.
[90] D.K. Singh, A. Sahoo, A hybrid CNN–LSTM deep learning model for rainfall prediction, NeuroQuantology 20 (14) (2022) 813.
[91] T. Phiboonbanakit, et al., Unsupervised hybrid anomaly detection model for logistics fleet management systems, IET Intell. Transp. Syst. 13 (11) (2019) 1636–1648.
[92] S. Gao, et al., Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation, J. Hydrol. 589 (2020) 125188.
[93] Y. Lin, et al., Bias learning improves data driven models for streamflow prediction, J. Hydrol.: Region. Stud. 50 (2023) 101557.
[94] M. Aslam, J.-S. Kim, J. Jung, Multi-step ahead wind power forecasting based on dual-attention mechanism, Energy Rep. 9 (2023) 239–251.
[95] J. Zhu, et al., Time-series power forecasting for wind and solar energy based on the SL-transformer, Energies 16 (22) (2023) 7610.
[96] M.I. Khelil, et al., Self-organizing maps-based features selection with deep LSTM and SVM classification approaches for advanced water quality monitoring, Int. J. Intell. Eng. Syst. 15 (3) (2022).
[97] D. Yan, et al., Characteristics and prediction of extreme drought event using LSTM model in Wei River Basin, Terrest. Atmosph. Ocean. Sci. 32 (2) (2021).
[98] Y. Chen, et al., An LSTM-based neural network method of particulate pollution forecast in China, Environ. Res. Lett. 16 (4) (2021) 044006.

[99] M. Abotaleb, P.K. Dutta, Optimizing long short-term memory networks for univariate time series forecasting: a comprehensive guide, in: Hybrid Information Systems: Non-Linear Optimization Strategies with Artificial Intelligence, 2024, p. 427.
[100] P.R. Shekar, et al., A combined deep CNN-RNN network for rainfall-runoff modelling in Bardha Watershed, India, Artif. Intell. Geosci. 5 (2024) 100073.
[101] S.S. Tabas, S. Samadi, Variational Bayesian dropout with a Gaussian prior for recurrent neural networks application in rainfall–runoff modeling, Environ. Res. Lett. 17 (6) (2022) 065012.
[102] W. Li, A. Kiaghadi, C. Dawson, High temporal resolution rainfall–runoff modeling using long-short-term-memory (LSTM) networks, Neur. Comput. Applic. 33 (4) (2021) 1261–1278.
[103] Z. Yao, et al., An ensemble CNN-LSTM and GRU adaptive weighting model based improved sparrow search algorithm for predicting runoff using historical meteorological and runoff data as input, J. Hydrol. 625 (2023) 129977.
[104] L.M. de Mendonça, C.J.C. Blanco, F. de Oliveira Carvalho, Recurrent neural networks for rainfall-runoff modeling of small Amazon catchments, Model. Earth Syst. Environ. 9 (2) (2023) 2517–2531.
[105] T.y. Pan, R.y. Wang, Using recurrent neural networks to reconstruct rainfall-runoff processes, Hydrolog. Process.: Int. J. 19 (18) (2005) 3603–3619.
[106] C. Wu, K.W. Chau, Rainfall–runoff modeling using artificial neural network coupled with singular spectrum analysis, J. Hydrol. 399 (3–4) (2011) 394–409.
[107] F. Kratzert, et al., Rainfall–runoff modelling using long short-term memory (LSTM) networks, Hydrol. Earth Syst. Sci. 22 (11) (2018) 6005–6022.
[108] H. Yin, et al., Rainfall-runoff modeling using long short-term memory based step-sequence framework, J. Hydrol. 610 (2022) 127901.
[109] C. Hu, et al., Deep learning with a long short-term memory networks approach for rainfall-runoff simulation, Water (Basel) 10 (11) (2018) 1543.
[110] M. Chhetri, et al., Deep BLSTM-GRU model for monthly rainfall prediction: a case study of Simtokha, Bhutan, Remote Sens. (Basel) 12 (19) (2020) 3174.
[111] Q. Wang, et al., Impact of input filtering and architecture selection strategies on GRU runoff forecasting: a case study in the Wei River Basin, Shaanxi, China, Water (Basel) 12 (12) (2020) 3532.
[112] A. Nath, D. Barman, G. Saha, Gated Recurrent Unit: an effective tool for runoff estimation, in: Proceedings of the International Conference on Computing and Communication Systems: I3CS 2020, NEHU, Shillong, India, Springer, 2021.
[113] B.C. Mateus, et al., Comparing LSTM and GRU models to predict the condition of a pulp paper press, Energies 14 (21) (2021) 6958.
[114] K. ArunKumar, et al., Comparative analysis of Gated Recurrent Units (GRU), long Short-Term memory (LSTM) cells, autoregressive Integrated moving average (ARIMA), seasonal autoregressive Integrated moving average (SARIMA) for forecasting COVID-19 trends, Alexandr. Eng. J. 61 (10) (2022) 7585–7603.
[115] S. Hu, et al., Deep neural network for robust modulation classification under uncertain noise conditions, IEEE Transact. Vehicul. Technol. 69 (1) (2019) 564–577.
[116] S. Das, S. Datta, B.B. Chaudhuri, Handling data irregularities in classification: foundations, trends, and future challenges, Pattern Recognit. 81 (2018) 674–693.
[117] P.B. Weerakody, et al., A review of irregular time series data handling with gated recurrent neural networks, Neurocomputing 441 (2021) 161–178.
[118] N. Buslim, et al., Comparing bitcoin's prediction model using GRU, RNN, and LSTM by hyperparameter optimization grid search and random search, 2021 9th International Conference on Cyber and IT Service Management (CITSM), IEEE, 2021.
[119] C. Garbin, X. Zhu, O. Marques, *Dropout* vs. *batch normalization: an empirical study of their impact to deep learning*, Multimed. Tool. Appl. 79 (19) (2020) 12777–12815.
[120] H. Hewamalage, C. Bergmeir, K. Bandara, Recurrent neural networks for time series forecasting: current status and future directions, Int. J. Forecast. 37 (1) (2021) 388–427.
[121] G. Jaiswal, et al., Integration of hyperspectral imaging and autoencoders: benefits, applications, hyperparameter tunning and challenges, Comput. Sci. Rev. 50 (2023) 100584.
[122] M.N. Fekri, et al., Deep learning for load forecasting with smart meter data: online adaptive recurrent neural network, Appl Energy 282 (2021) 116177.
[123] Y.E. Gür, Comparative analysis of deep learning models for silver price prediction: CNN, LSTM, GRU and hybrid approach, Akdeniz İİBF Dergisi 24 (1) (2024) 1–13.
[124] H. Abbasimehr, R. Paki, Improving time series forecasting using LSTM and attention models, J. Ambient Intell. Human. Comput. 13 (1) (2022) 673–691.
[125] D. Machiwal, M.K. Jha, Time series analysis of hydrologic data for water resources planning and management: a review, J. Hydrol. Hydromech. 54 (3) (2006) 237–257.
[126] Y. Himeur, et al., Using artificial intelligence and data fusion for environmental monitoring: a review and future perspectives, Inform. Fus. 86 (2022) 44–75.