Neural Computing and Applications (2022) 34:19995–20015
https://doi.org/10.1007/s00521-022-07523-8

ORIGINAL ARTICLE

Streamflow modelling and forecasting for Canadian watersheds using LSTM networks with attention mechanism

Lakshika Girihagama[1] · Muhammad Naveed Khaliq[1] · Philippe Lamontagne[1] · John Perdikaris[2] · René Roy[3] · Laxmi Sushama[4] · Amin Elshorbagy[5]

Received: 16 December 2021 / Accepted: 7 June 2022 / Published online: 13 July 2022

Abstract

This study investigates the capability of sequence-to-sequence machine learning (ML) architectures in an effort to develop streamflow forecasting tools for Canadian watersheds. Such tools are useful to inform local and region-specific water management and flood forecasting related activities. Two powerful deep-learning variants of the Recurrent Neural Network were investigated, namely the standard and attention-based encoder-decoder long short-term memory (LSTM) models. Both models were forced with past hydro-meteorological states and daily meteorological data with a look-back time window of several days. These models were tested for 10 different watersheds from the Ottawa River watershed, located within the Great Lakes Saint-Lawrence region of Canada, an economic powerhouse of the country. The results of training and testing phases suggest that both models are able to simulate overall hydrograph patterns well when compared to observational records. Between the two models, the attention model significantly outperforms the standard model in all watersheds, suggesting the importance and usefulness of the attention mechanism in ML architectures, not well explored for hydrological applications. The mean performance accuracy of the attention model on unseen data, when assessed in terms of mean Nash–Sutcliffe Efficiency and Kling-Gupta Efficiency is, respectively, found to be 0.985 and 0.954 for these watersheds. Streamflow forecasts with lead times of up to 5 days with the attention model demonstrate overall skillful performance with well above the benchmark accuracy of 70%. The results of the study suggest that the encoder–decoder LSTM, with attention mechanism, is a powerful modelling choice for developing streamflow forecasting systems for Canadian watersheds.

Keywords Streamflow forecasting · LSTM · Encoder-decoder architecture · Attention-based models · Deep learning

1 Introduction

Improved streamflow forecasting capability is important for water management related activities, informing hydropower generation operations, flood risk management and operational decision-making at local and regional scales. Streamflow is the integrated result of highly nonlinear physical processes that operate at multiple temporal and spatial scales within a watershed. Traditionally, streamflow forecasting is accomplished using process-based hydrological models. These models can range from simple conceptual lumped models to complex physically based distributed models. Conceptual lumped type models are based on mathematical formulations of the physical processes involved in runoff generation at the watershed scale (e.g., Streamflow Synthesis and Reservoir Regulation model [1] and Soil and Water Assessment Tool [2]). These models are considerably simplified based on reasonable assumptions and they also do not capture the spatial variability of physical processes occurring within a watershed. On the other hand, physically based distributed models can capture to some extent the spatial variability of the nonlinear physical processes occurring within a watershed (e.g., MIKE SHE model [3], WATFLOOD model [4–6], Variable Infiltration Capacity model [7], and MESH model

✉ Muhammad Naveed Khaliq
Muhammad.Khaliq@nrc-cnrc.gc.ca

1 National Research Council Canada, Ottawa, ON, Canada

2 Ontario Power Generation, Niagara Falls, ON, Canada

3 Hydro Météo, Notre-Dame-des-Prairies, QC, Canada

4 McGill University, Montreal, QC, Canada

5 University of Saskatchewan, Saskatoon, SK, Canada

[8]). The precise way the process variabilities are handled in mathematical formulations can vary significantly from one model to another. Although process-based models produce deterministic and plausible results in many instances, uncertainty in parametrization and process scaling deficiencies are some of the issues that degrade their performance [9]. Undoubtedly, these models have shown great value in forecasting streamflow in many watersheds in different parts of the world, including Canada [10–14]. Though calibration and testing of a process-based model for a given watershed can be achieved with a greater detail and depth, transfer of the same model for applications across other watersheds can compromise the performance. This is due to the difficulty in the formulation of scale-dependent parameterizations of watershed relevant physical processes [15, 16] and that in turn impacts model's generalization ability.

With the growing availability of large amounts of spatial and temporal data from remote sensing and numerical weather prediction models (e.g., remotely sensed land use data, reanalyses products and real-time meteorological forecasts) and recent advances in computational power, Machine Learning (ML) methods can also offer powerful modelling options for developing data-driven streamflow forecasting systems, with generalization abilities. This is due to their ability to extract complex dynamical nonlinearities without explicitly defining the scale-relevant physical processes, as in the case of hydrological models discussed above. Hence, explicit definitions of governing equations are not needed for these models. Instead, these models map multivariate input space to an output space. Data-driven methods can be categorized as time series (statistical) methods and ML approaches. The statistical models simply derive the relationship between variables to formalize understanding and evaluation of a hypothesis about the system's behaviour [17]. The common statistical methods under this category include autoregressive moving average models [18], autoregressive integrated moving average models [19–21], and many other variants of these time series models. In these deductive methods, the streamflow observations are assumed to be stochastic sequences and hence, future streamflow can be predicted by learning from past observations [22]. Although availability of very long records of observations are crucial for accurate prediction of future streamflow, the applicability of these models to real-time forecasting situations, however, remains limited due to lack of generalization ability and cascading uncertainty in parametrizations [22]. ML models on the other hand, have proven to overcome some of the drawbacks associated with process-based and statistical modelling approaches. These inductive models are developed based on data and are able to extract nonlinear structures from data and can readily learn from inter-variable interactions. Some perspectives gathered from the literature on applied ML techniques, with reference to hydrologic applications, are discussed below.

In hydrology, application of Artificial Neural Networks (ANNs) dates back to 1990s [23] and since then many researchers have explored various ANN architectures for rainfall-runoff modelling and streamflow forecasting globally. A historical review of ANN applications is available in [9]. Among Canadian studies, Tiwari and Adamowski [24] studied the ANNs (traditional, wavelet, and bootstrap) and hybrid versions of both wavelet and bootstrap ANNs for forecasting daily urban water demand for the City of Calgary. Their findings showed enhanced performance of hybrid ANN models for lead times of up to 5 days. Another recent study [25] compared four different methods of input parameter selection for predicting streamflow for two distinctive watersheds in Canada (i.e., Don River watershed in Toronto and Bow River watershed in Alberta) using ANNs. To predict water levels with a lead time of 1-day in Lake Erie, [26] used five different ANN architectures, including Gaussian process, multiple linear regression, multilayer perceptron, M5P model tree, random forest, and k-nearest neighbours. However, by design, ANNs cannot keep the information in the sequential order, which is crucial for many hydrological systems, including streamflow forecasting, which is affected by past hydrological and meteorological states in a given watershed.

To overcome the above mentioned issue, the Recurrent Neural Network (RNN) architecture was developed [27, 28]. RNN differs from ANNs due to the capability to use variable lengths in inputs/outputs and ability to share features across different positions in the sequence. However, one of the weaknesses associated with this type of RNN is the inability of the network to capture long-term dependencies in the sequences [29]. In simple terms, if there is long memory in the sequences, the RNN has difficulty in retaining information from much earlier states to later ones. Also, if the neural network is very deep, the gradient from the output would have a hard time propagating back to affect the weights of the earlier layers which in turn will not affect the computations in these layers. This is called the vanishing gradient problem in RNNs. In practice, vanishing gradient problem of RNN makes it difficult to get a neural network to realize that it needs to memorize the information from prior sequences [29]. It means that the basic RNN has many local influences, meaning, output is influenced by the input values close to it. Long- Short-Term Memory (LSTM) is a deep-learning modification to the basic RNN hidden layer that captures long range connections in the sequence [30, 31].

Applications of LSTMs in hydrology can be found in some studies such as rainfall-runoff modelling [32–34], streamflow predictions for ungauged basins [35],

modelling flash flood events and flood forecasting [36–38], and simulating reservoir outflows [39]. In all of these examples, LSTM model proved to be a robust tool when compared with process-based hydrological options or ANNs. Nevertheless, it is evident from the literature that there is insufficient information available regarding how LSTMs can be considered as useful tools for forecasting streamflow with multiple lead times, which is an important requirement for operational applications of these models. Based on the literature review, it was found that data-driven forecasting systems for real-time operational applications are not yet available for Canadian watersheds. This study aims to address this gap and develop ML-based streamflow modelling and forecasting tools, inspired by LSTM model architectures, to inform operational decision-making in real-world applications. Ten different watersheds, with near pristine conditions, located in the Ottawa River watershed of Great Lakes Saint-Lawrence region of Canada were considered for the development and evaluation of ML-based streamflow forecasting tools. For developing these tools, standard encoder-decoder LSTM and the attention-based encoder-decoder LSTM models were considered. To our knowledge, these architectures have never been evaluated for hydrological applications in Canada. Streamflow predictions/forecasts were assessed at multiple lead times, ranging from 1 to 5 days. The performance of the standard and attention-based models was assessed by evaluating quantitative model performance metrics such as root-mean square error, coefficient of determination, Nash–Sutcliffe Efficiency [40], and Kling-Gupta Efficiency [41]. The latter two performance metrics are commonly used in hydrologic modelling area. Thus, the entire set covers both the ML and hydrologic modelling fields.

The paper is organized as follows. A description of the study area and the watersheds selected for the study are provided in Sect. 2. Section 3 is devoted to input and output data and the pre-processing that was deemed necessary to make these datasets suitable for a ML application. Section 4 describes the methodological framework used for developing ML-based streamflow forecasting systems. This is followed by Sect. 5, which presents the results of the study and discusses several aspects of model development, training and testing phases, and evaluation of real-time forecasting scenarios. Finally, Sect. 6 presents main conclusions of the study and Sect. 7 future research directions. Throughout the paper, the phrases like model prediction, model forecast and model simulation are used interchangeably. The latter two are more common in hydrologic literature, compared to the former, which is common in ML area.

2 Study area

For this study, 10 different watersheds, with drainage areas ranging from 1,040 to 6,704 km^2, from the Ottawa River watershed were selected for developing and evaluating ML-based daily streamflow modelling and forecasting tools (see Fig. 1). Located in the Canadian Shield, the Ottawa River watershed is one of the major tributaries of the Saint-Lawrence drainage system with a total drainage area of 146,000 km^2, of which 65% is located in the province of Quebec and the remaining 35% is located within the province of Ontario. The length of the Ottawa River is about 1,200 km. In addition, there are about 90,000 lakes and 30 reservoirs in the Ottawa River watershed and hence, it is considered as one of the heavily regulated watersheds in Canada. Geographically, the Ottawa River watershed is a low land that resulted from the past glacial activities. In fact, it is the only Canadian River that crosses four major geological subdivisions of the Canadian Shield: the Superior Province, Cobalt Plate, Grenville Province, and St. Lawrence Lowlands [42]. The land cover within the watershed is mainly forests. While 85% of the forest cover is a combination of mixed and deciduous forests, the remaining 15% accounts for boreal forests. While the climate in the northern part of the basin is cool and dry, the southern part is warm and humid. The selected watersheds represent near pristine conditions, with minimal human interventions. Thus, the development of ML models will not be impacted by artificial influences and hence these watersheds will provide an opportunity to evaluate streamflow forecasting capabilities of ML models in a realistic and dispassionate manner.

3 Data

Three types of data were considered for this study: streamflow data pertaining to the hydrology of the watersheds; meteorological data related to atmospheric input to the watersheds and evaporative demand; and geophysical data representing land cover types. These datasets were obtained from different sources as discussed below.

Continuous daily streamflow observations for the 10 watersheds (Fig. 1) selected for this study were obtained from Environment and Climate Change Canada's national water data archive: HYDAT (https://wateroffice.ec.gc.ca/). A summary of the acquired data is given in Appendix 1. Watershed drainage areas were delineated using Canadian Digital Elevation Model (1945–2011) data, acquired from Natural Resources Canada, and QGIS (http://qgis.org) and Green Kenue^{TM} software [43].

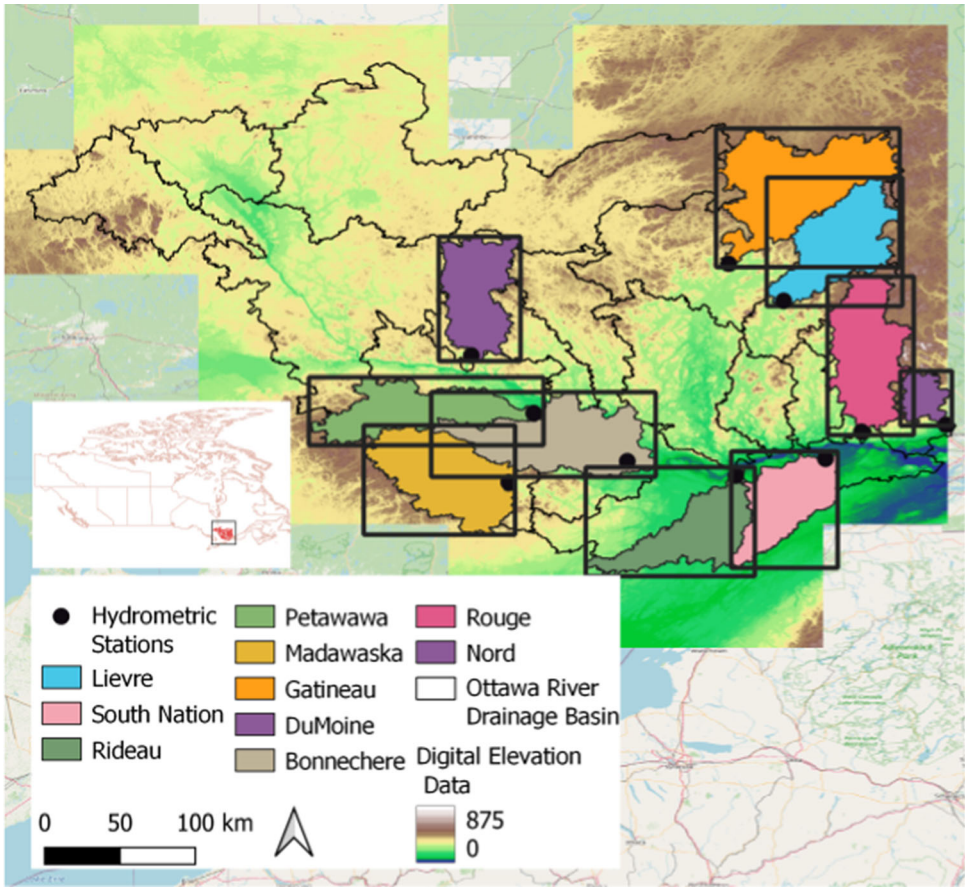

Fig. 1 Study domain, showing locations of selected watersheds and the corresponding streamflow gauging stations, extracted from the HYDAT database of Environment and Climate Change Canada. Digital elevation data *Source*: Canadian Digital Elevation Model, 1945–2011

Meteorological and geophysical data were sourced from ERA5-Land, which is a reanalysis gridded dataset with a spatial resolution of 9 km and consists of variables that describe energy and water cycle over land [44]. This dataset provides a uniform gridded data source for all watersheds and avoids lack of data related issues and is considered to be quite suitable for model development and testing purposes [44]. While ERA5-Land hourly data records which contain temporal variability of land variables were available from January 1950 to present, the temporal range of data for each watershed was different due to the limitations imposed by the availability of streamflow data for the selected watersheds. For the purpose of this study, gridded data were area averaged for all watersheds. The variables used in this study are air temperature measured at 2 m from the surface, east–west wind velocity measured at 10 m from the surface, north–south wind velocity measured at 10 m from the surface, total precipitation (in the form of liquid and frozen water), atmospheric surface pressure, leaf area index (the ratio of one half of total leaf area per unit horizontal ground surface) for high vegetation type, snow water equivalent (cumulative depth), and volumetric soil water content for the upper most (0–7 cm) soil layer (Fig. 2). Except for total precipitation, daily averages were obtained for all variables. For precipitation, accumulated daily precipitation was derived from the hourly data. Temporal plots of daily climatological streamflow, meteorological variables, and vegetation indices, shown in Fig. 2 for selected watersheds, clearly exhibit a seasonal pattern, which is compatible with the physical understanding of the integrated land-river-atmospheric systems. For instance, peak river flows occur during spring as a result of snowmelt or combined snowmelt and rain-on-snow events, which is typical to Canadian conditions, and low flows occur during dry summer or during winter freeze-ups. Temperature and evaporation have maximum values in summer. Similar patterns were noted for other watersheds, when plots similar to Fig. 2 were developed for the same variables. Such parallel plots help uncover structural features of data which are important for developing ML-based forecasting tools.

3.1 Data pre-processing

Pre-processing of input variables/features is key to ML model development and in attaining high performance and accuracy. The steps involved in data pre-processing includes splitting available data into training, validation,

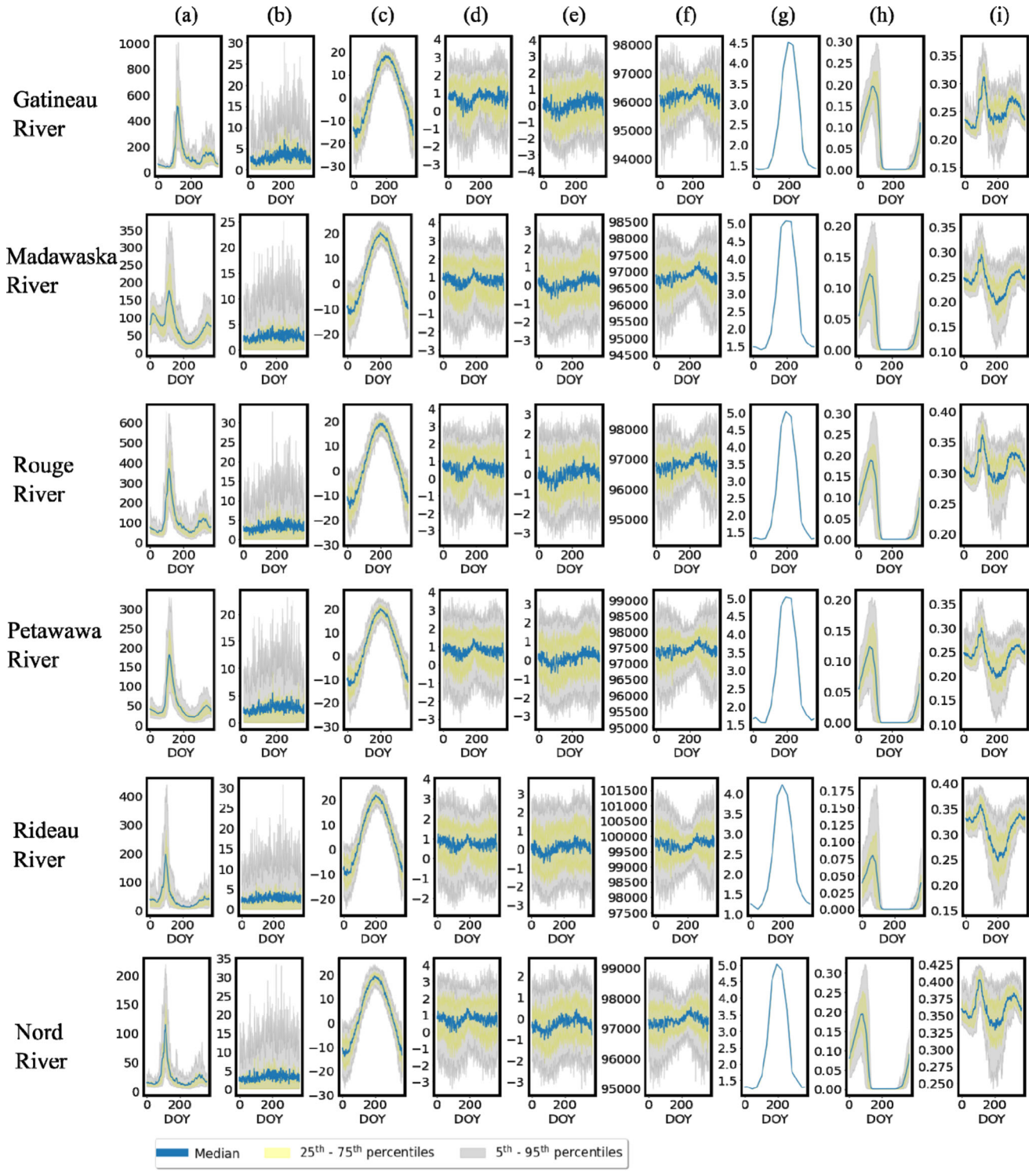

Fig. 2 Daily climatological plots of hydrological, meteorological and land data [columns (a) to (i)] used in the study for six selected watersheds (Gatineau, Madawaska, Rouge, Petawawa, Rideau, and Nord). **a** Streamflow [m^3/s] measured at the outlet, **b** accumulated daily total precipitation [mm], **c** temperature [°C] measured at 2 m from the surface, **d** east–west wind velocity component [m/s] measured at 10 m above the surface, **e** north–south wind velocity component [m/s] measured at 10 m from the surface, **f** surface atmospheric pressure [Pa], **g** leaf area index-high vegetation [.], **h** snow water equivalent [m], and **i** volumetric soil water content [.] measured for the 0–7 cm top soil layer. DOY means day of the year

and testing portions; feature selection; feature engineering if deemed necessary; and scaling of data. Some of these steps and various decisions taken to prepare the data for model development and evaluation are elaborated below.

3.1.1 Data splitting

An important aspect of data pre-processing for ML model development is splitting data into training, validation, and test sets. This data partitioning is done to ensure a generalized model. The training set is used as a sample dataset for model fitting. The validation set is used for an unbiased evaluation of the model, fitted on the training dataset, and re-training/stopping purposes. Tuning of hyperparameters of the trained model normally occurs during the validation stage. The evaluation of the model becomes biased when validation dataset is incorporated in the model. This can cause data leakage. Hence, a third sample of unseen test set is held out and is used only at the end of the model training phase. Thus, the test data provides final sanity check before putting the model in operational use. For this study, data were split by consecutive dates to preserve the temporal order of the sequences and to avoid look-ahead bias and data leakage. Additional information on training, validation, and testing data splits used in this study is given in Appendix 2.

3.1.2 Missing data

Only a small portion ($< 0.1\%$) of streamflow data were missing in some of the watersheds mainly for low-to-medium flows. Hence, imputation of those missing values was obtained to preserve continuity in the data. To fill these values, several imputation techniques were examined. The tested imputers were mean, median, decision tree, *k*-nearest neighbours (using $k = 14$ and 30, where k is the number of nearby samples), and extra tree (with $n = 7$, 14, 20, 25, and 30; n is the number of trees in the model). Each imputer was iterated for each watershed to select the best estimator with a minimum error between the imputed vs. the actual.

3.1.3 Feature selection

The feature selection is one of the core processes in ML because it significantly impacts the performance of the desired model. In the context of streamflow forecasting, this process is designed to determine the set of variables that are physically most meaningful and are related to the target variable (i.e., streamflow). The runoff generation within a watershed is in fact a physical phenomenon and therefore physical relevance is an important aspect. This step is also useful in eliminating irrelevant features that do not affect the target variable of interest. In a nutshell, this step helps avoid overfitting, reduce training time, enhance model credibility, and improve model accuracy. Among many techniques, inter-variable correlations of input variables show how various variables are related to each other and with the target variable as well. Apart from that, correlations of various pairs of variables are also very insightful. For instance, from a highly correlated pair, only one variable can be retained in order to obtain a robust set of variables and that helps to develop a credible model. ML, after all, is a data-driven modelling framework and the trained model will be as good as the data it is trained on. Following recommendations from [45] and [46], Variance Inflation Factor (VIF) analysis with a threshold of 5.0 was used to investigate multi-collinearity issues and eliminating redundant variables. We understand that there are several other methods available for feature selection, inter-variable correlation and VIF analyses used here might not be the best methods in all cases.

3.1.4 Feature engineering

The distribution of daily streamflow sequences shows a typical exponential tail behavior due to lower number of high flow values compared to the rest. Scaling of such data using minimum and maximum values of the sequence will result in the majority of scaled streamflow values falling in the lower end of the scaling range. This will lead to an imbalanced distribution of the target variable. In such situations, transformation techniques are often used such as log transformation [47, 48] or more generalized Box-Cox transformation [49]. Transformation of streamflow values helps the ML model obtain a balanced target variable, a tenable feature that helps the ML model learn faster. After some experimentation, log-transformation was deemed appropriate for this study. Thus, log-transformed streamflow values were used for ML model development.

3.1.5 Scaling

As a rule of thumb, ML models usually perform better when input variables are scaled [50]. Hence, it is important to scale input features before training the ML model. Usually it is achieved through standardization or Min–Max scaling (some investigators refer to this process as normalization). Standardization assumes that the data sequence will have a Gaussian like distribution, with zero mean and unit standard deviation. The Min–Max scaling on the other hand, rescales the input sequences to values ranging from 0 to 1. In this study, log-transformed flow, 2 m temperature, east–west wind, north–south wind, and surface pressure were scaled using standardization and the rest of the variables (i.e., daily accumulated precipitation,

leaf area index-high vegetation, and volumetric soil water content) were scaled using the Min–Max scaling.

3.1.6 Data windowing

The next step is to prepare input data to generate a set of predictions/forecasts based on a given set of consecutive data. The current hydrological state of any watershed depends on the past system behavior. Therefore, the sequential data are re-organized as fixed-length vectors using a sliding window so that ML model finds a function that maps the sequences of past observations in order to predict/forecast future values of the sequence (Fig. 3). The model prediction/forecast at any time step is assumed to be driven by a specified number of past consecutive samples in the sequence (termed here as look-back window). In this study, we used a look-back window of 14 days to predict/forecast streamflow on any day (see the schematic diagram in Fig. 3). The look-back window size was chosen after some experimentation with both longer and shorter windows and the choice, however, does involve some expert judgement as well.

3.1.7 Hyperparameter optimization

ML models have hyperparameters which affect the model performance. Hence, optimization of hyperparameters is important for obtaining a model with higher accuracy (or minimum error). In this study, the search space was set as bounded domain of hyperparameters and random search was performed within that domain to obtain optimal values. For this purpose, Keras Tuner [51], a library that allows to pick optimal hyperparameters for ML framework was adapted.

3.2 Software

The computational code mainly was implemented in Python 3.8 [52]. The ML models were implemented in Keras [53] with TensorFlow [54] backend. The Python libraries such as NumPy [55], Pandas [56], Seaborn [57], Scikit-learn [58], and Matplotlib [59] were used for data pre-processing and visualization purposes.

4 Modelling

In ML modelling, the sequence prediction involves predicting next value or multiple values of a real valued variable. These predictions can either be one-to-one or many-to-many predictions. The many-to-many predictions can be two fold. In one instance, the lengths of the input and output sequences can be the same and in the other, variable lengths of the input and output sequences is possible (e.g., machine translation, streamflow forecasting for multiple lead times, music generation, etc.). This is called sequence-to-sequence (seq2seq) predictions.

The seq2seq predictions are usually achieved through RNNs. The LSTM, a powerful variant of the RNN, has the ability to retain long range connections of the data sequence, where it is maintained by the cell state, $c(t)$ (Fig. 4a). The cell state runs straight through the entire sequence with simple linear interactions (Fig. 4b). Thus, it allows information to flow unchanged. LSTM has the ability to add or remove information to or from the cell state. This is achieved through the use of regulated structures called gates. There are three gates inside an LSTM cell, known as forget gate, update gate, and output gate. The forget gate is to decide which pieces of information in the sequence should now be forgotten from the cell state (Eq. 1). This is acquired through a sigmoid layer (σ), where values (Γ_f) vary from 0 to 1. When $\Gamma_f = 1$, it retains the current cell state and the opposite occurs when $\Gamma_f = 0$. The parameters $a^{<t-1>}$ is the activation function at time step $(t-1)$, W and b are weights to be updated, and $x^{<t>}$ is the input sequence at time step t.

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f) \tag{1}$$

The update gate on the other hand has multiple operations. First, a sigmoid layer will decide which values need to be updated (Γ_u in Eq. 2). Next, a tanh activation layer will create a candidate for the cell state ($\tilde{c}^{<t>}$ in Eq. 3)

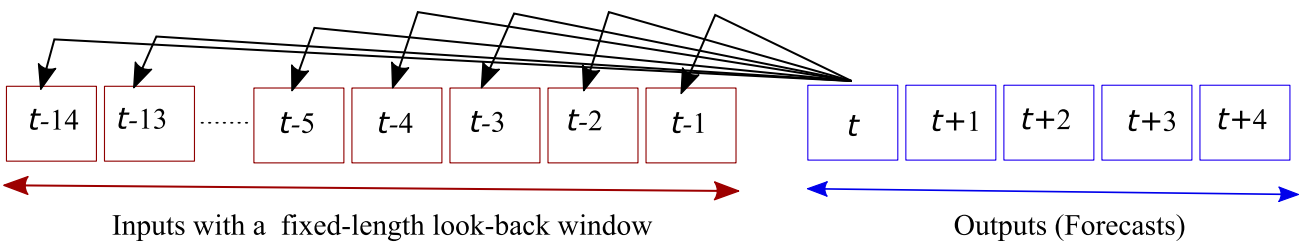

Fig. 3 Schematic representation of the sliding window approach used in this study. The look-back window is a fixed-length vector which contains information of past hydrological and meteorological states of a watershed. The output is also a fixed-length vector, with the selected number of lead times

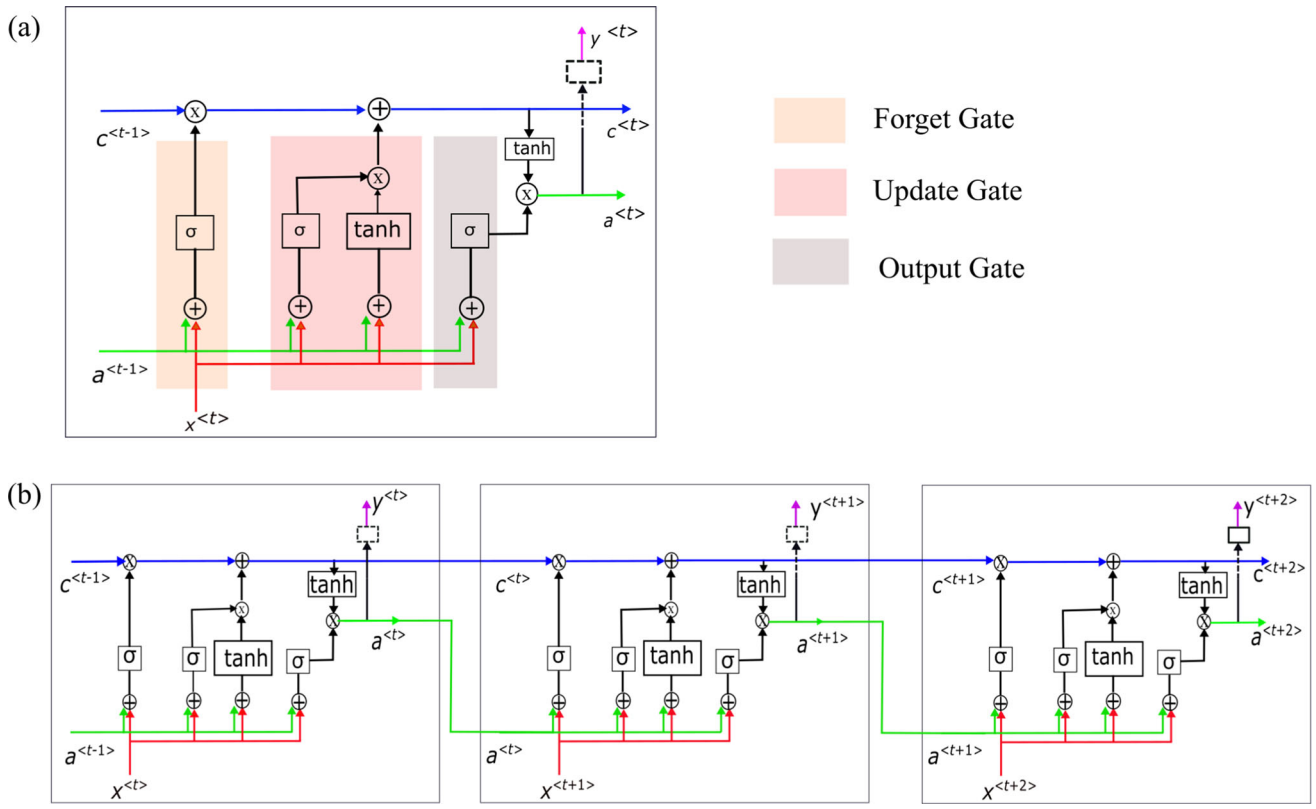

Fig. 4 **a** A schematic representation of a single LSTM cell. **b** Stack representation of LSTM cells for prediction of outputs of a sequence one at a time. At any time step t, the input signal of the sequence $(x^{<t>})$, the previous activation $(a^{<t-1>})$ at time step $t-1$ are fed to the LSTM cell to predict the output signal $(y^{<t>})$. The cell state $(c^{<t>})$ runs straight through the chain with some linear interactions.

that considers overwriting the previous cell state. Then, the old cell state $(c^{<t-1>})$ will be updated to new cell state $(c^{<t>})$ through element-wise multiplication of corresponding values from the update and forget gates and previous and candidate cell states as given in Eq. 4.

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u) \tag{2}$$

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c) \tag{3}$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>} \tag{4}$$

The ouput gate (Γ_o) finally decides which value is going to be released from the LSTM cell (Eqs. 5–6). The operations within the ouput gate is two-fold. First, inputs go through a sigmoid layer. Then the values from the output gate and the cell state will go through a tanh layer to generate activation for the next LSTM cell.

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o) \tag{5}$$

$$a^{<t>} = \Gamma_o * \tanh(c^{<t>}) \tag{6}$$

In this study, two types of LSTM structures were considered for modelling and forecasting streamflow sequences, i.e., standard encoder-decoder LSTM network and attention-based encoder-decoder LSTM network. Both structures are presented below.

4.1 Standard encoder-decoder LSTM network

The standard encoder-decoder LSTM architecture [60] is used to simulate and forecast daily streamflow with multiple lead times. The encoder-decoder LSTM network by design is able to address the problem of seq2seq predictions where it maps a fixed-length input vector to a fixed-length output vector. The lengths of the input and output vectors can vary. For instance, in streamflow forecasting, input sequences from several previous days at time point t can be used to forecast flow for next several days, including the time point t. The architecture of the standard encoder-decoder LSTM is shown in Fig. 5. This model contains three parts, encoder, context vector, and the decoder. Encoder and decoder consist of a layer of horizontally stacked LSTM cells. The number of LSTM units in encoder and decoder is determined by the look-back window size and the forecast time step. At the encoder, it receives the past meteorological and hydrological states as input

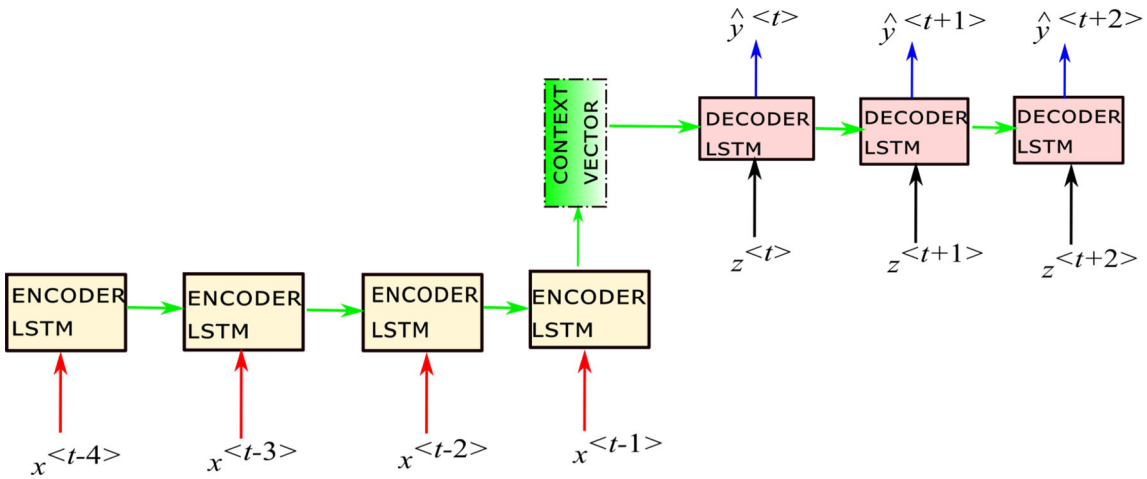

Fig. 5 A sample schematic representation of encoder-decoder LSTM architecture, with reduced feature space. Here, $x^{<t-i>}$ is the vector containing input features (i.e., past meteorological and hydrological states and geophysical features) at the *i*th lag time, $\hat{y}^{<t+j>}$ is the predicted/forecasted flow at the *j*th time step, and $z^{<t+j>}$ is the weather at the *j*th time step

vectors $(x^{<t-i>}, i = 1, 2, ..t_n)$. At each time step, encoder LSTM unit receives the corresponding meteorological and hydrological state and the previous hidden state and then, calculates the hidden state for the next LSTM unit as given in Eqs. 1–6. The context vector is the final hidden state of the encoder LSTM that is input to the decoder as the initial hidden state. In streamflow forecast mode, in addition to the context vector from the encoder, at each forecast time step, the decoder receives corresponding predicted weather variables $(z^{<t+j>}, j = 0, 1, ..4$; e.g., temperature, precipitation, etc.). This is because the forecasted streamflow at each forecasting time step is a direct function of the meteorological data on that day. Finally, each decoder unit outputs the forecasted streamflow at each time step $(z^{<t+j>})$. Here, the loss was calculated by the mean squared error. Previous hydrological applications of encoder-decoder LSTM include [38, 61, 62].

4.2 Attention based encoder-decoder LSTM network

One of the issues with the standard encoder-decoder architecture is that the context vector cannot encode all the information from the input sequence (also known as the bottleneck problem). This is due to the limitation on the look-back window size. The attention mechanism [63] is an improvement to the encoder-decoder architecture where it accumulates memory from attending inputs at each time step. Simply, the attention mechanism gives the relative importance for the inputs at each time step. An application of this mechanism was demonstrated by [37].

The architecture of the attention-based LSTM encoder-decoder is shown in Fig. 6. There are several operations which are carried out in the attention model. First, the bi-directional (feed forward and backward) LSTMs' cells in the encoder generate activation functions for input sequence at each time step $(a_f{}^{<t-i>}$ and $a_b{}^{<t-i>}, i = 1, 2, \ldots t_n)$. Then, these hidden states are concatenated for each time step (Eq. 7) to produce encoder hidden states.

$$a^{<t-i>} = \left[a_f{}^{<t-i>}; a_b{}^{<t-i>}\right]^T \tag{7}$$

Once the encoder hidden state is obtained, the next step is to calculate the alignment scores between the previous hidden state of the decoder cell and the corresponding encoder state as follows:

$$e^{<t>} = g(s^{<t-i>}, a^{<t>}) \tag{8}$$

where g is a nonlinear activation function (e.g., tanh).

Next, context vector for each output will be created by weighted sum of hidden states from the encoder unit. The context vector that feeds to the $<t+i>, i = 0, 1, \ldots$ is given as,

$$c^{<t>} = \sum_{j=1}^{Tx} \alpha^{<t>} a^{<t-j>} \tag{9}$$

where the weights $\alpha^{<t>}$ are calculated applying the soft-max function as follows.

$$\alpha^{<t>} = \frac{\exp(e^{<t>})}{\sum_{j=1}^{Tx} e^{<j>}} \tag{10}$$

All mathematical symbols used in this section are also described in Appendix 3.

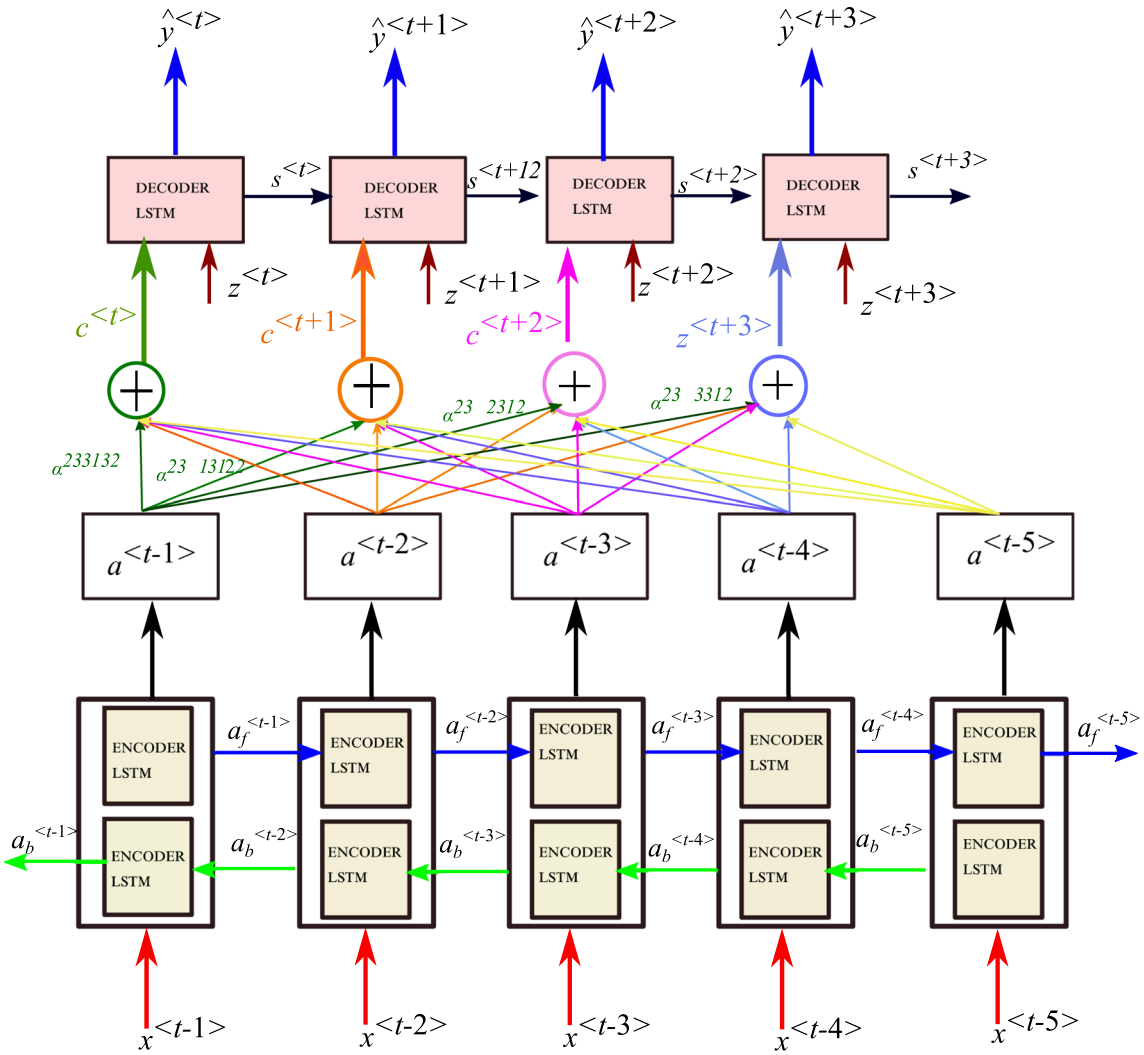

Fig. 6 Typical architecture of attention-based encoder-decoder LSTM. The subscripts "*f*" and "*b*" respectively depict the forward and backward propagation of the activation function in the bi-directional LSTM

5 Results and discussion

In this section, the usefulness of standard encoder-decoder LSTM and the attention-based encoder-decoder LSTM ML architectures is demonstrated to develop streamflow forecasting systems for Canadian watersheds, especially for the watersheds selected from the Ottawa River watershed. The comparison between both architectures also provided an opportunity to assess the value of attention mechanism in modelling streamflow sequences, which, to our knowledge, is the very first application in Canadian hydrology. The performance of both architectures was evaluated using root mean square error (RMSE), coefficient of determination (R^2), Nash–Sutcliffe Efficiency (NSE), and Kling-Gupta Efficiency (KGE). While RMSE, R^2 are considered as typical evaluation metrics in ML, the NSE and KGE are widely used in hydrology to evaluate model performance. The mathematical formulations of these metrics are given in the following equations:

$$\mathrm{RMSE} = \sqrt{\frac{\sum_{i=1}^{N}\left(y_i^{\mathrm{obs}} - y_i^{\mathrm{pred}}\right)^2}{N}} \tag{11}$$

$$\mathrm{R}^2 = \left(\frac{\sum_{i=1}^{N}\left(y_i^{\mathrm{obs}} - \overline{y}^{\mathrm{obs}}\right)\left(y_i^{\mathrm{pred}} - \overline{y}^{\mathrm{pred}}\right)}{\sqrt{\sum_{i=1}^{N}\left(y_i^{\mathrm{obs}} - \overline{y}^{\mathrm{obs}}\right)^2}\sqrt{\sum_{i=1}^{N}\left(y_i^{\mathrm{pred}} - \overline{y}^{\mathrm{pred}}\right)^2}}\right)^2 \tag{12}$$

$$\mathrm{NSE} = 1 - \frac{\sum_{i=1}^{N}\left(y_i^{\mathrm{obs}} - y_i^{\mathrm{pred}}\right)^2}{\sum_{i=1}^{N}\left(y_i^{\mathrm{obs}} - \overline{y}^{\mathrm{obs}}\right)^2} \tag{13}$$

$$\mathrm{KGE} = 1 - \sqrt{(r-1)^2 + (\alpha-1)^2 + (\beta-1)^2} \tag{14}$$

$$\alpha = \frac{\bar{y}^{\mathrm{pred}}}{\bar{y}^{\mathrm{obs}}} \tag{15}$$

$$\beta = \frac{\sigma^{\mathrm{std}}_{\mathrm{pred}}/\bar{y}^{\mathrm{pred}}}{\sigma^{\mathrm{std}}_{\mathrm{obs}}/\bar{y}^{\mathrm{obs}}} \tag{16}$$

In the above equations, y is the streamflow, r is the correlation coefficient between the predicted (*pred*) by the model and the observed (*obs*) values, $\bar{y}$ is the mean value, and σ^{std} is the standard deviation. The closest the values of R^2, NSE and KGE are to 1, the better the association between observed and predicted values. An NSE value lower than zero indicates that the mean value of the observed time series would be a better estimator than the model simulation. Regarding RMSE, values closer to zero are indications of good model performance.

5.1 Model performance for training and validation periods

The performance of both standard and attention models was evaluated for training and validation periods, however, detailed results are presented only for the attention model to conserve space. It is important to note that a similar behavior as discussed here for the attention model was also observed for the standard model. To cover the entire range of selected watersheds, results are presented with respect to three categories of watersheds: (i) large watersheds, with drainage areas $> 5{,}000$ km^2, (ii) medium watersheds, with drainage areas ranging from 4,000 km^2 to 5,000 km^2, and (iii) small watersheds, with drainage areas smaller than 4,000 km^2. These categories are defined arbitrarily based on the range of drainage areas of the 10 selected watersheds. The Gatineau and Madawaska River watersheds fall in the first category, the Rouge and Petawawa River watersheds fall in the second category and the Rideau ad Nord River watersheds fall in the third category. Although the specific results are presented for these watersheds, discussion is applicable for all studied watersheds.

Model predicted daily flows (i.e., flows on the day of forecast or zero day ahead forecasts) for the training and validation periods for the above selected watersheds show good agreement with the observed flows (Fig. 7); this comparison was performed to mimic the evaluation strategy of process-based models commonly practiced in hydrology. In this figure, both observed and predicted flows are summarized over the annual cycle for the combined training and validation periods. The graphics for all watersheds show that the model simulated flows preserve the seasonality (i.e., high flows in spring and low flows in summer and winter months), median values, and the variability of observed flows. All lines and shaded areas corresponding to both observed and predicted flows nearly overlap each other for all watersheds, suggesting that the attention model is able to simulate observed flows in a satisfactory manner. Similar figures for the remaining watersheds are presented in Fig. S1 (online resource).

In addition to daily flow comparisons, an assessment of simulated annual flow volumes, derived purely from daily simulated flows, for the training and validation periods was also carried out to provide a different perspective on the model performance. A visual comparison of observed and predicted/simulated annual flow volumes is shown in Fig. 8. The correspondence between the two flow volumes was assessed based on the coefficient of correlation, which was found to be higher than 0.99 for all watersheds, suggesting that the attention model was able to simulate observed annual flow volumes with a higher level of accuracy.

5.2 Model performance on unseen test data

Nearly in all ML model development projects, the ultimate test of the model happens when the model performance is evaluated on unseen data, i.e., the data that the model has never seen during the training and validation phases. This evaluation becomes even more important in seq2seq streamflow modelling because the testing data are from a completely non-overlapping period. Here, for the testing period, comparisons of observed and simulated flows for the standard encoder-decoder LSTM model are presented in Figs. 9a, 10a and 11a and that for the attention-based encoder-decoder LSTM model in Figs. 9b, 10b and 11b for the Madawaska River, Rouge River and Nord River watersheds. See also Figs. S3a–S8a and S3b–S8b in the online resource for other watersheds. The testing period roughly contains about 4 years of daily streamflow data. From these comparisons, three key points can be derived. (i) From the temporal plots, it is evident that both models are able to simulate overall hydrograph patterns fairly well, however, the scatter plots verify that the attention model considerably outperforms the standard model. (ii) The results of the standard model are more variable (i.e., they exhibit a noisy behaviour) compared to the results of the attention model. (iii) While the standard model slightly overestimates the streamflow during the snowmelt season (spring to early-summer), the attention model slightly underestimates the streamflow, especially high flows, for the same season. A considerable portion of streamflow during this season is driven from snowmelt, which depends on snow accumulation on the ground. The underestimation by the attention model could partly be due to the short length of the look-back window [64]. It is also likely that merely the attention mechanism based on the set of

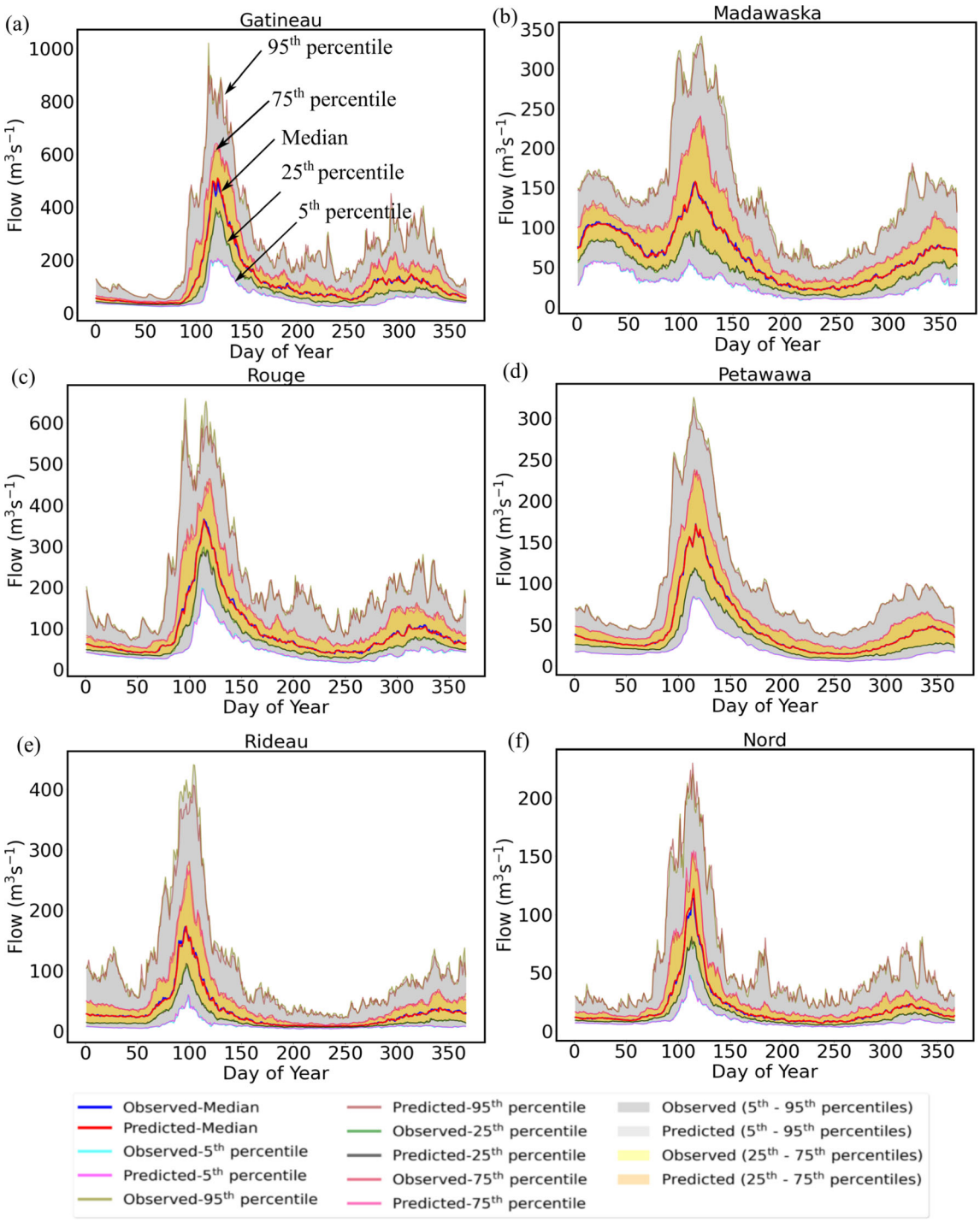

Fig. 7 Observed and attention model predicted/simulated daily flows for the training and validation periods for selected large (**a**, **b**), medium (**c**, **d**), and small watersheds (**e**, **f**). Similar graphics for the remaining watersheds are available in Fig. S1 of the online resource

variables considered is not sufficient to discern the complex relationships between snow accumulation and ablation and runoff generation. Hence, still more complex data-driven architectures and learning mechanisms, with additional variables than those considered here, may be required to simulate streamflow for this season with a higher degree of accuracy than achieved here.

The performance metrics for the day of forecast ($\hat{y}^{<t>}$) for the entire testing period obtained from the standard and attention models are shown in Table 1. For the standard

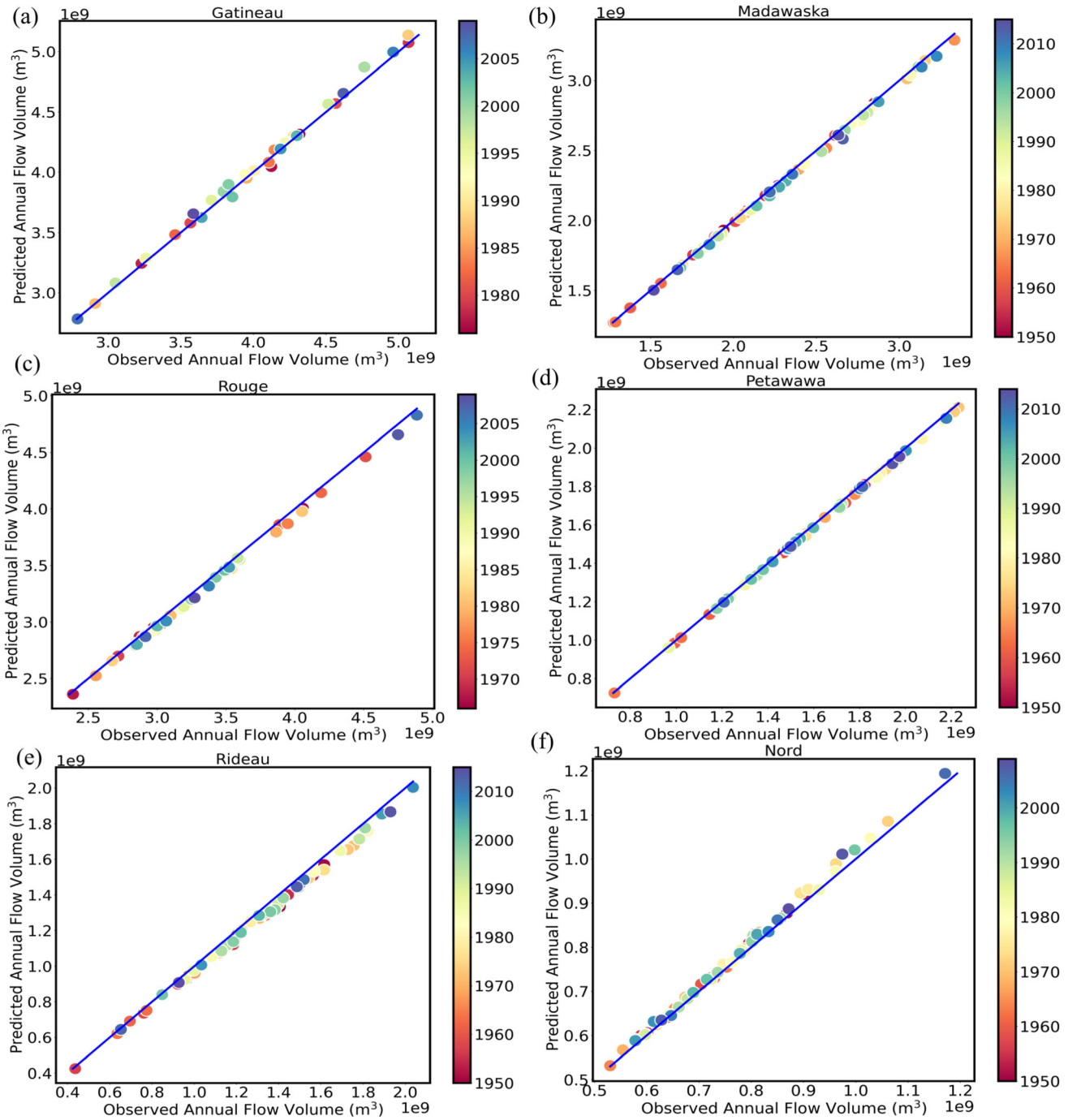

Fig. 8 Observed and attention model simulated/predicted annual flow volumes for the selected large (**a**, **b**), medium (**c**, **d**) and small (**e**, **f**) watersheds for the training and validation periods. Similar graphics for the remaining watersheds are shown in Fig. S2 of the online resource

model, the average RMSE, R^2, NSE, and KGE for all watersheds are 40.39 m^3/s, 0.668, 0.668, and 0.827, respectively. Similarly, the averages of RMSE, R^2, NSE, and KGE for all watersheds in the case of attention model are 8.2 m^3/s, 0.985, 0.985, and 0.957, respectively. Thus, it is evident that the attention model performance is significantly superior to that of the standard model and it also surpasses many performance benchmarks established in hydrology [65].

5.2.1 Flow accumulation with time—attention model

A comparison of cumulative flows overtime, obtained from observed and model simulated daily flows, is provided in Fig. 12 for the testing period. This type of comparison

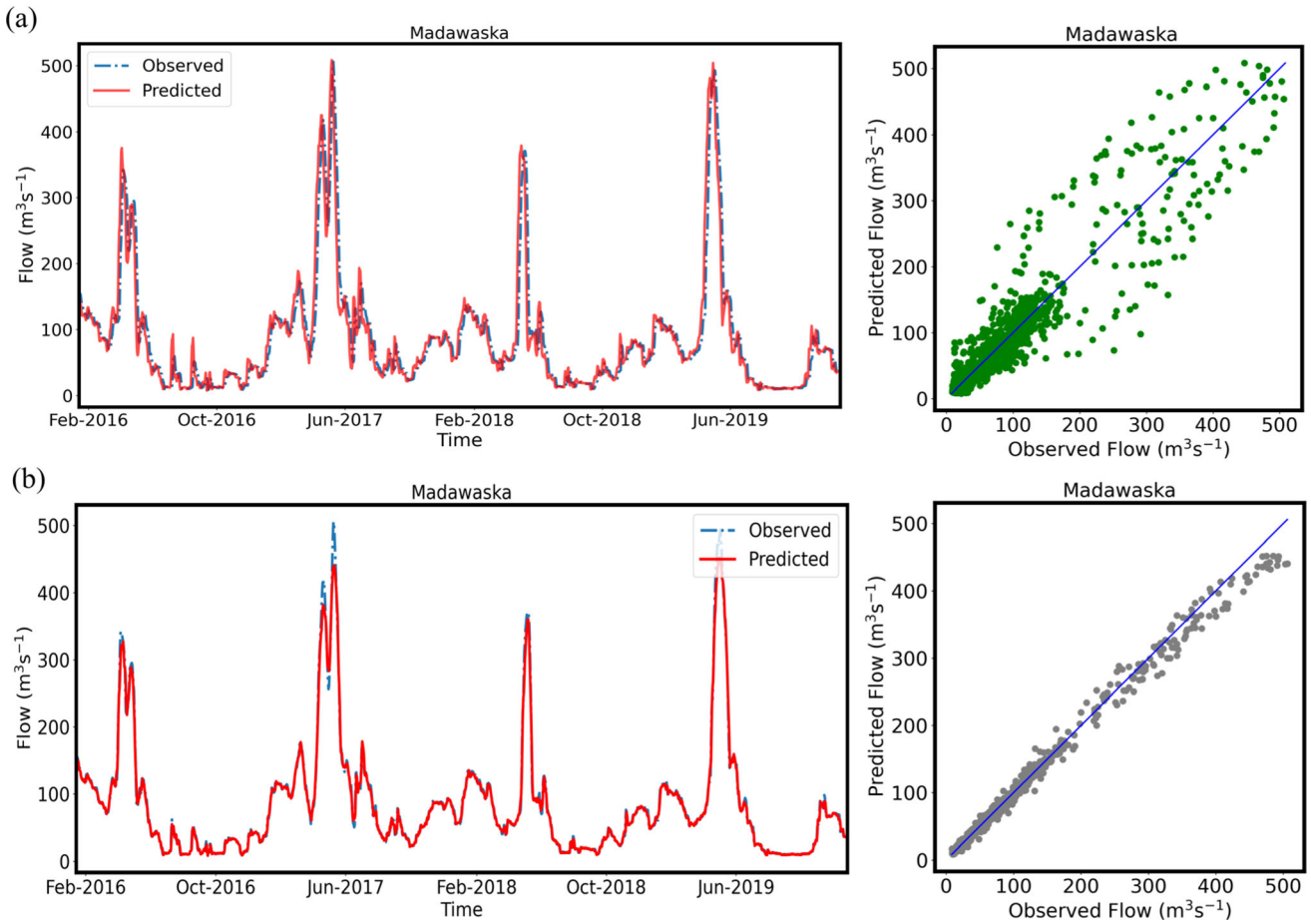

Fig. 9 Direct comparisons of observed and model simulated streamflow for the same day on unseen data (test set) for the **a** standard encoder-decoder LSTM and **b** attention-based encoder-decoder LSTM for the Madawaska River watershed

serves as a surrogate for ensuring mass conservation, which is an important aspect if the target model is adopted as a simulation tool for reservoir design and operations. The results of these comparisons show that the attention model is also able to preserve the overall mass balance in a satisfactory manner for most of the watersheds, suggesting higher level of confidence in modelling results. However, small discrepancies in these comparisons were noted for the Rideau River, Bonnechere River and South Nation River watersheds. These small discrepancies perhaps could be attributed to slight underestimation of annual peak flows by the attention model.

5.2.2 Multiple lead time future forecasts—attention model

To investigate suitability of ML models as potential forecasting tools for real-time operational applications, the performance of both models was also assessed for multiple lead times using the testing period as the testbed and assuming future meteorological inputs as real-time forecasts coming from a numerical weather prediction model (i.e., by emulating a real-time forecast scenario). The assessment results based on the NSE and KGE metrics are provided in Fig. 13 for the attention model. For all watersheds, the majority of the NSE and KGE values are well above 0.7, which is generally considered as a good model performance level [65]. In a real-time forecast scenario, the accuracy of future predictions/forecasts generally decreases as the lead time increases. This character is also visible in the results shown in Fig. 13. Although slightly inferior to the attention model, similar results were also noticed for the standard model. Nevertheless, forecasting flows/floods with lead times of up to 5 days will enable the responsible authorities to issue warnings and to take necessary actions to safeguard the public and infrastructure.

6 Conclusions

From the results presented and discussed in this paper, the following main conclusions can be drawn:

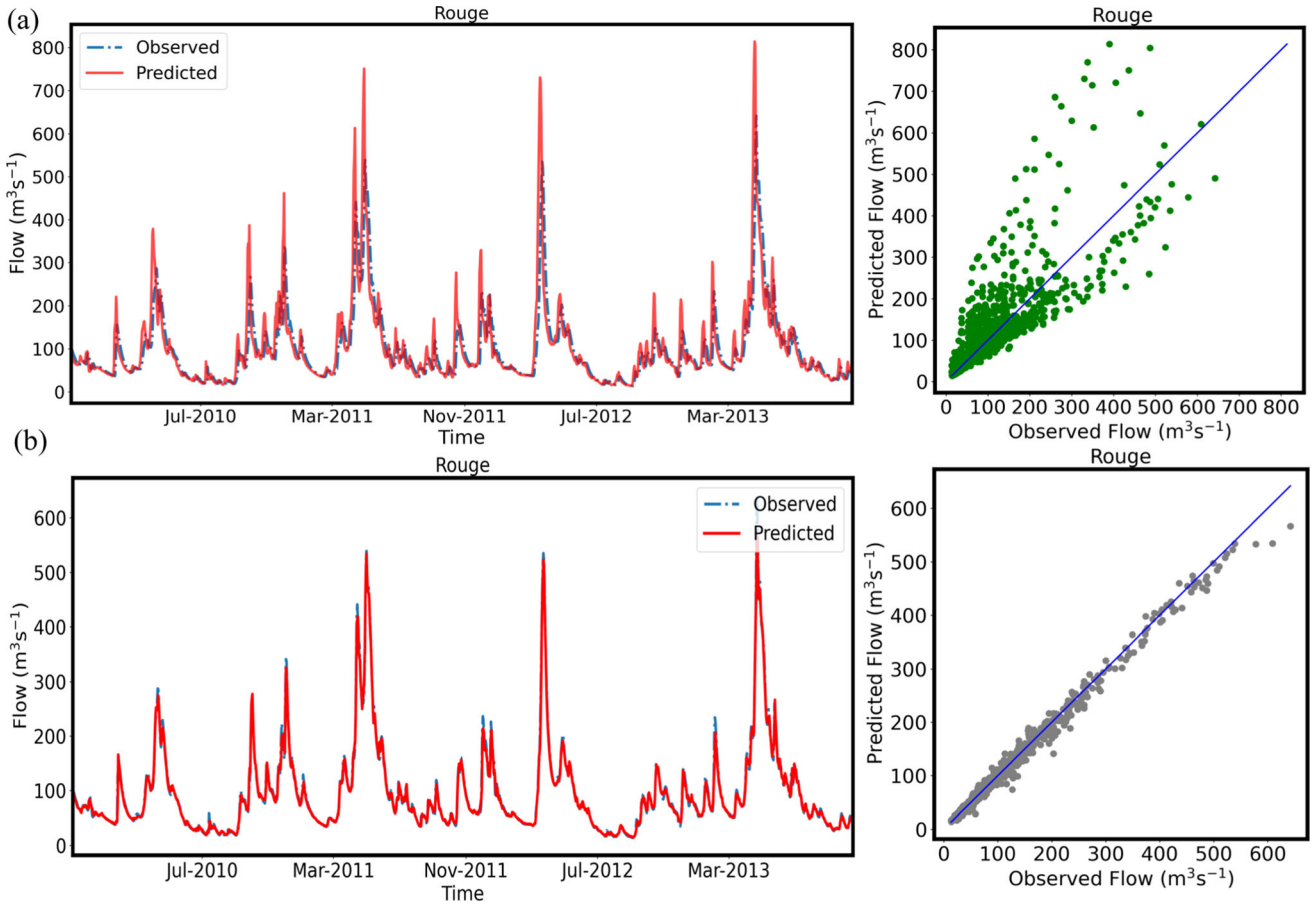

Fig. 10 Same as Fig. 9 but for Rouge River watershed

- The evaluation of standard and attention-based encoder-decoder LSTM models for the training and validation periods suggest that both models can simulate overall hydrograph shapes, annual flow accumulations, seasonal patterns, and streamflow variability across the entire year fairly well. However, in relative terms, the attention model was found to perform much better than the standard model for all studied watersheds.
- The evaluation of both models on unseen non-overlapping data suggests that the attention-based model significantly outperforms the standard model in terms of values of the performance metrics (i.e., RMSE, NSE and KGE), which were found to be considerably better than the commonly accepted benchmark values. Given such a superior level of performance, the attention model can be used with a higher level of confidence for developing real-time streamflow forecasting systems for Canadian watersheds.
- The standard and attention models were also tested for simulating streamflow in an emulated real-time forecasting mode considering multiple lead times, ranging from one to five days, on 10 different watersheds. In this regard, it can be stated that the ML tools driven by LSTM networks, compared to hydrological modelling options, can be used as reliable alternatives for developing real-time streamflow forecasting systems, including flood forecasting, in Canada and other parts of the world.
- To our knowledge, this study is the very first application of the attention mechanism in Canadian hydrology and therefore can be considered as a neat contribution to the broad literature on hydrologic forecasting and earth system science.

7 Future research and recommendations

It is hoped that through large scale applications and targeted evaluations, ML models would continue to evolve and mature as efficient data-driven solutions for real-time streamflow modelling and forecasting system developments. We intend to continue our research along the lines initiated in this paper for developing ML-based forecasting tools to be readily integrated with regional hydrological and water resources management systems. This can be

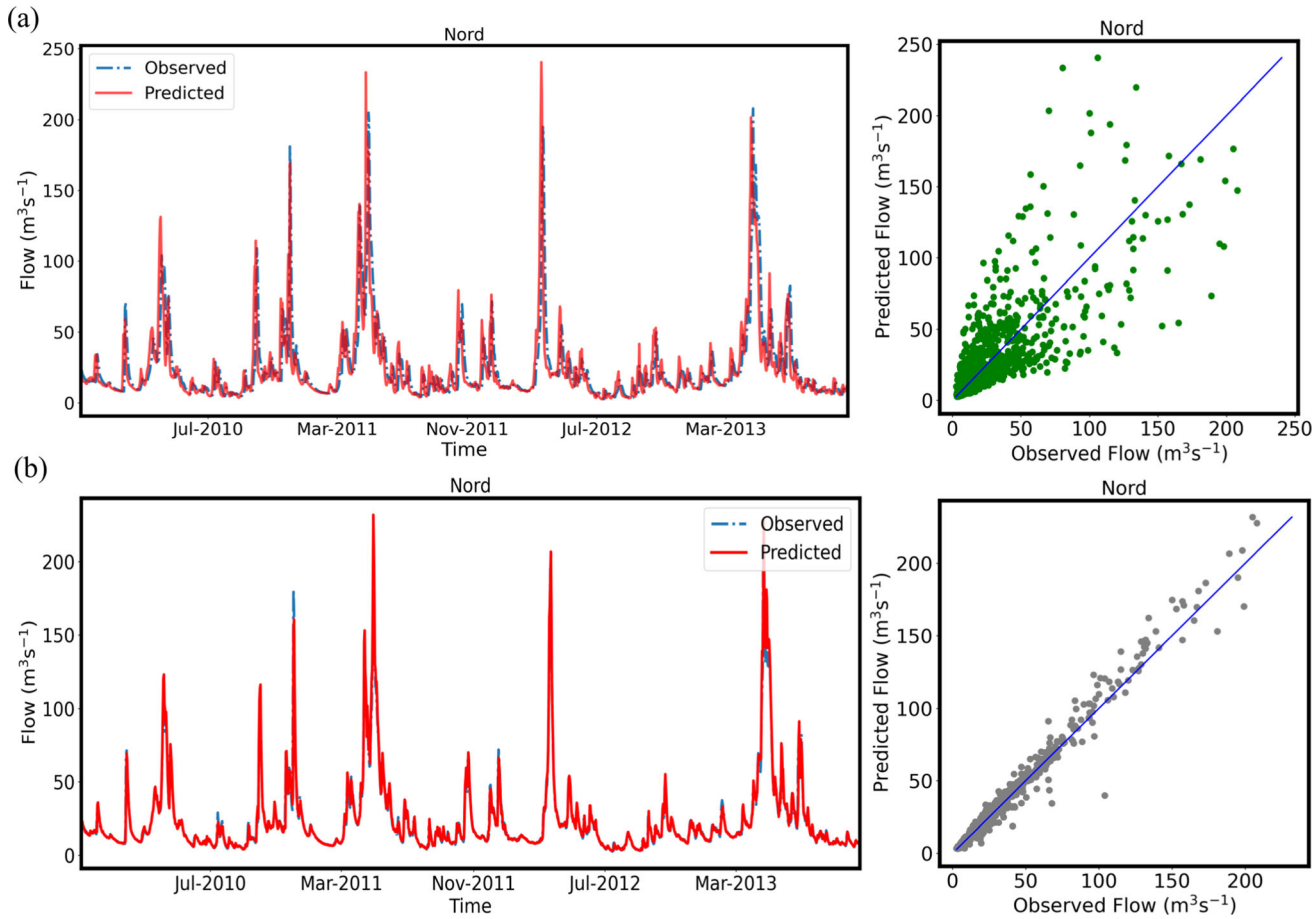

Fig. 11 Same as Fig. 9 but for Nord River watershed

Table 1 Performance metrics for the day of forecast obtained from the standard and attention models

Watershed	RMSE (m^3/s)		R^2		NSE		KGE	
	Standard	Attention	Standard	Attention	Standard	Attention	Standard	Attention
South Nation	79.4	19.5	0.543	0.968	0.455	0.967	0.733	0.941
Gatineau	74.0	13.3	0.735	0.991	0.712	0.991	0.856	0.991
Rouge	58.5	8.1	0.667	0.992	0.537	0.991	0.743	0.965
Nord	18.8	4.2	0.590	0.980	0.520	0.976	0.766	0.946
Rideau	42.5	9.3	0.688	0.984	0.611	0.981	0.796	0.937
Madawaska	33.4	8.1	0.866	0.994	0.860	0.992	0.930	0.954
Petawawa	34.7	4.6	0.783	0.996	0.720	0.995	0.821	0.963
Bonnechere	10.7	4.7	0.876	0.976	0.874	0.976	0.932	0.969
Lievre	33.3	5.9	0.656	0.989	0.580	0.987	0.792	0.944
DuMoine	18.6	4.3	0.828	0.991	0.811	0.990	0.900	0.962

achieved by embedding ML tools in existing ensemble frameworks or by developing an ensemble framework based totally on ML architectures. Of course, in real-world circumstances and applications, such decisions are generally associated with many management related decisions and regional/national priorities.

We also intend to investigate the suitability of LSTM architecture-driven modelling tools for forecasting spring floods alone, due to their huge societal impacts, in gauged and ungauged locations across larger regions in future studies. Also, it will be worth investigating the uncertainty associated with various lead-time streamflow forecasts by integrating real-time meteorological forecasts from

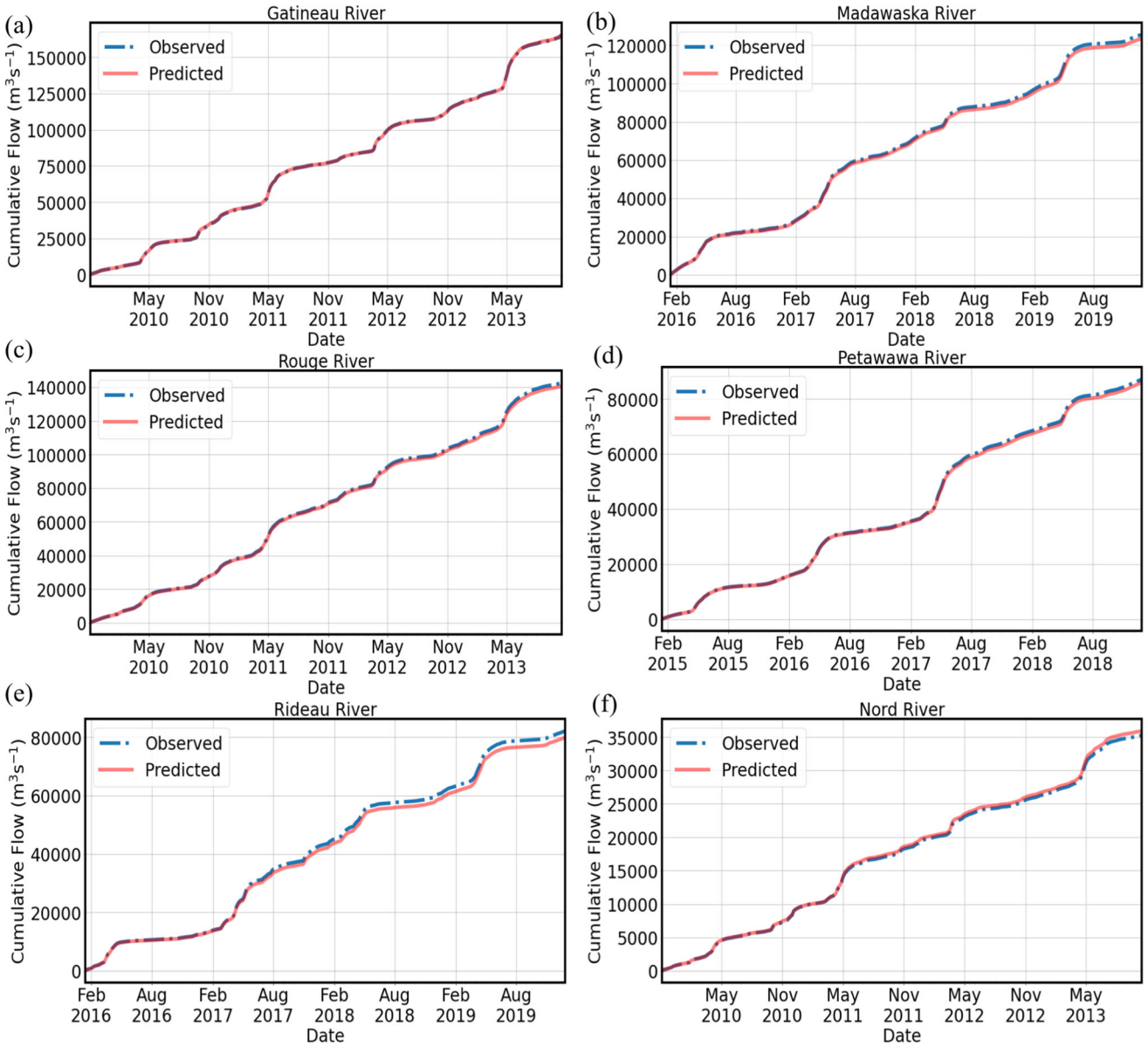

Fig. 12 Comparison of cumulative observed and attention model simulated flows for representative large (**a**, **b**), medium (**c**, **d**), and small (**e**, **f**) watersheds. The results for the remaining watersheds are available in Fig. S9 in the online resource

numerical weather prediction models and satellite-based remotely sensed data with the attention-based LSTM model, which was not explored in this study.

This paper significantly advanced knowledge on real world applications of ML tools in earth system science than making theoretical innovations in deep learning area, which we believe is absolutely necessary to solve applied problems with new and emerging technologies. To further bridge the gap between theoretical innovations and their practical applications, it would be prudent to investigate additional deep learning architectures (e.g., the use of Gated Recurrent Units (GRUs) in place of LSTMs, the use of additional hidden layers, or the use of Transformer models) and revealing their practical strengths.

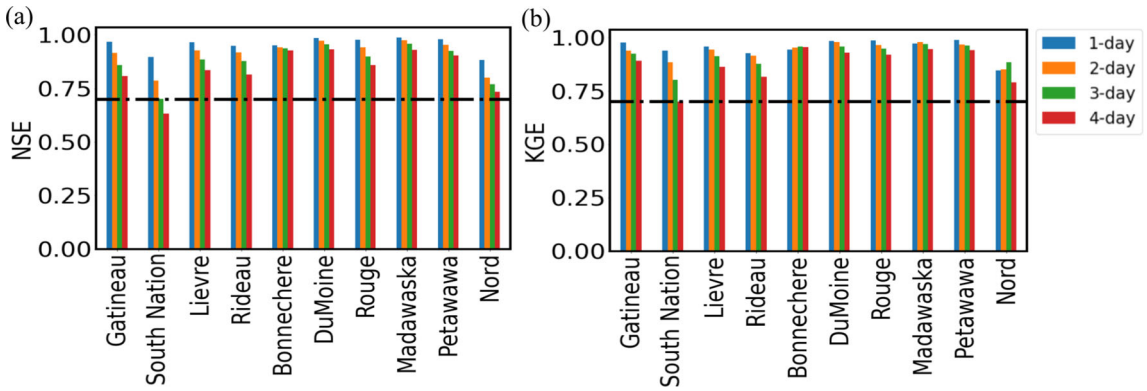

Fig. 13 The **a** NSE and **b** KGE values derived from the observed and attention model simulated streamflow for multiple lead times for the entire testing period for all studied watersheds. Lead time flows were obtained by moving day of forecast by one-day at every time step

Appendix 1

See Table 2.

Table 2 A summary of HYDAT stations used in the study. See Fig. 1 for geographical locations of gauging stations

Watershed name	Station	Drainage area (km^2)	Province	Latitude	Longitude	Date range	
						From	To
Gatineau	02LG005	6704	QC	47.08	−75.75	1974	2013
Bonnechere	02KC009	5800	ON	45.50	−76.56	1921	2018
Madawaska	02KD004	5459	ON	45.33	−77.52	1930	2019
Rouge	02LC029	4744	QC	45.74	−74.69	1964	2013
Lievre	02LE024	4482	QC	46.08	−76.07	1979	2013
Petawawa	02KB001	4163	ON	45.89	−77.32	1915	2018
Rideau	02LA004	3881	ON	45.38	−75.70	1933	2019
DuMoine	02KJ004	3720	QC	46.35	−77.82	1965	2013
South Nation	02LB005	3569	ON	45.52	−74.98	1915	2019
Nord	02LC008	1040	QC	45.79	−74.01	1930	2013

Appendix 2

See Table 3.

Table 3 Data split by date for training, validation, and test sets

Watershed Name	Training		Validation		Test	
	From	To	From	To	From	To
Madawaska	1950-01-01	2000-01-01	2000-01-01	2015-12-31	2015-12-31	2019-12-31
Gatineau	1976-01-01	1998-10-31	1998-10.31	2009-10-31	2009-10-31	2013-10-21
Rouge	1966-01-01	1993-10-22	1993-10-22	2009-10-21	2009-10-21	2013-10-21
Petawawa	1950-01-01	1999-01-01	2000-01-01	2014-12-31	2014-12-31	2018-12-31
Rideau	1950-01-01	2000-01-01	2000-01-01	2015-12-31	2015-12-31	2019-12-31
Nord	1950-01-01	1993-10-22	1993-10-22	2009-10-21	2009-10-21	2013-10-21
Bonnechere	1962-10-01	1999-01-01	1999-01-01	2014-12-31	2014-12-31	2018-12-31
South Nation	1950-01-01	2000-01-01	2000-01-01	2015-12-31	2015-12-31	2019-12-31
Lievre	1979-05-09	1993-10-22	1993-10-22	2009-10-21	2009-10-21	2013-10-21
DuMoine	1965-06-01	1993-11-01	1993-11-01	2009-10-31	2009-10-31	2013-10-31

Appendix 3

See Table 4.

Table 4 Mathematical symbols and their descriptions

α	Bias ratio derived from the predicted and observed mean streamflow values
$\alpha^{<t>}$	Weights obtained for each element in the input sequence at time step t
β	Variability ratio derived from the mean values and the standard deviation of the observed vs. the predicted streamflow
Γ	Gates in the LSTM cell. The subscripts 'f', 'u', and 'o' on this symbol respectively denote the forget gate, update gate, and output gate
σ	Sigmoid function
σ^{std}	Standard deviation
$a^{<t>}$	Activation function at time step t. The subscripts 'b' and 'f' on this symbol respectively refer to forward and backward passes
b	Weights to be updated. The subscripts 'f', 'u', 'o', and 'c' on this symbol respectively denote the weights related to LSTM's forget gate, update gate, output gate, and candidate cell state
$c(t)$	Cell state at time step t
$c^{<t>}$	Cell state of the LSTM unit at time step t
$\tilde{c}^{<t>}$	Candidate cell state at the update gate at time step t
$e^{<t>}$	Alignment scores between the previous hidden state of the decoder cell and the corresponding encoder state
g	Nonlinear activation function (e.g., tanh)
i	Time index for the encoder and sequence index for the number of samples
j	Time index for the decoder
k	Number of nearest-neighbours
n	Number of trees
N	Number of samples
r	Pearson correlation coefficient between the observed and the predicted flows
$s^{<t>}$	Decoder hidden state at time step t obtained after attention mechanism
t	Time point
tanh	Mathematical function 'tanh'
Tx	Length of the input sequence
W	Weights to be updated. The subscripts 'f', 'u', and 'o' on this symbol respectively denote the weights related to LSTMS's forget gate, update gate, and output gate
$x^{<t>}$	Input sequence at time step t (e.g., the vector containing input features)
y	Streamflow value. The superscripts $'obs'$ and $'pred'$ on this symbol respectively denote the observed and the predicted flows
$\bar{y}$	Mean streamflow value. The superscripts $'obs'$ and $'pred'$ respectively denote the observed and the predicted flows
$\hat{y}^{<t>}$	Predicted/forecasted flow at time step t
$z^{<t>}$	Inputs for decoder (i.e., weather predictions)

Supplementary Information The online version contains supplementary material available at https://doi.org/10.1007/s00521-022-07523-8.

Acknowledgements The authors would like to acknowledge the team who produced ERA5-Land data, which served as the basis for developing and validating streamflow forecasting tools driven by LSTM architectures. The efforts of several developers and modellers, who contributed their code and experience to GitHub are very much appreciated. This study has benefited considerably from these resources. The first author would like to thank Data Science and AI team of the Ocean, Coastal and River Engineering Research Centre of the National Research Council Canada for the unwavering support and the STEM funding which made this work possible. The review comments and suggestions from the associate editor and anonymous referrers towards improving this manuscript are graciously acknowledged.

Author contributions MNK conceived the idea and LG converted it into reality and introduced the attention mechanism in hydrological forecasting for Canadian watersheds. All authors contributed to reviewing/revising the manuscript. In particular, Lakshika Girihagama developed and evaluated both models and prepared an initial draft of the paper. MNK, PL, JP, RR, LS, and AE reviewed and/or strengthened various parts of the paper. LS, and AE made logical and intellectual contributions.

Funding Open Access provided by National Research Council Canada.

Declarations

Conflict of interest The authors declare that there is no conflict of interest. The research did not involve human participants and/or animals.

References

1. Rockwood DM (1964) Streamflow synthesis and reservoir regulation. US Army Engineer Division, North Pacific, Portland, Oregon, Engineering Studies Project 171.
2. Arnold J, Williams, J, Srinivasan, R, et al (1994) SWAT-soil and water assessment tool. US Department of Agriculture, Agricultural Research Service, Grassland, Soil and Water Research Laboratory, Temple, TX, 1994
3. Refshaard J, Storm B et al. (1995) MIKE SHE. Computer models of watershed hydrology, pp. 809–846
4. Kouwen N (1988) WATFLOOD: a Micro-computer based flood forecasting system based on real-time weather Radar. Canad Water Resour J/Revue Canadienne des Ressources Hydriques 13:62–77. https://doi.org/10.4296/cwrj1301062
5. Kouwen N, Soulis ED, Pietroniro A et al (1993) Grouped response units for distributed hydrologic modeling. J Water Resour Plan Manag 119:289–305
6. Kouwen N, Danard M, Bingeman A et al (2005) Case study: watershed modeling with distributed weather model data. J Hydrol Eng 10:23–38
7. Liang X, Lettenmaier DP, Wood EF, Burges SJ (1994) A simple hydrologically based model of land surface water and energy fluxes for general circulation models. J Geophys Res Atmos 99:14415–14428
8. Ahmed MI, Elshorbagy A, Pietroniro A, Princz D (2021) Improving the representation of the non-contributing area dynamics in land surface models for better simulation of prairie hydrology. J Hydrol 600:126562
9. ASCE (2000) Artificial neural networks in hydrology. II: Hydrologic applications. J Hydrol Eng 5:124–137
10. Micovic Z, Quick M (1999) A rainfall and snowmelt runoff modelling approach to flow estimation at ungauged sites in British Columbia. J Hydrol 226:101–120
11. Singh VP (1995) Computer models of watershed hydrology. Water Resources Publications, Highlands Ranch, Colorado
12. Singh VP, Frevert DK (2002) Mathematical models of small watershed hydrology and applications. Water Resources Publications, Highlands Ranch, Colarado
13. Maidment DR (1993) Handbook of hydrology. McGraw-Hill, London
14. Wagener T, Wheater H, Gupta HV (2004) Rainfall-runoff modelling in gauged and ungauged catchments. Imperial College press and distributed by World Scientific Publishing Co
15. Nearing GS, Kratzert F, Sampson AK et al (2021) What role does hydrological science play in the age of machine learning? Water Resour Res 57:e2020WR28091. https://doi.org/10.1029/2020WR028091
16. Blöschl G, Bierkens MFP, Chambel A et al (2019) Twenty-three unsolved problems in hydrology (UPH)—a community perspective. Hydrol Sci J 64:1141–1158. https://doi.org/10.1080/02626667.2019.1620507
17. Bzdok D, Altman N, Krzywinski M (2018) Statistics versus machine learning. Nat Methods 15:233–234. https://doi.org/10.1038/nmeth.4642
18. Weeks WD, Boughton WC (1987) Tests of ARMA model forms for rainfall-runoff modelling. J Hydrol 91:29–47. https://doi.org/10.1016/0022-1694(87)90126-0
19. McKerchar AI, Delleur JW (1974) Application of seasonal parametric linear stochastic models to monthly flow data. Water Resour Res 10:246–255. https://doi.org/10.1029/WR010i002p00246
20. Noakes DJ, McLeod AI, Hipel KW (1985) Forecasting monthly riverflow time series. Int J Forecast 1:179–190. https://doi.org/10.1016/0169-2070(85)90022-6
21. Yürekli K, Kurunç A (2005) Testing the residuals of an ARIMA model on the Çekerek Stream watershed in Turkey. Turk J Eng Environ Sci 29:61–74
22. Mosavi A, Ozturk P, Chau K (2018) Flood prediction using machine learning models: literature review. Water 10:1536
23. Daniell T (1991) Neural networks. Applications in hydrology and water resources engineering. In: National Conference Publication-Institute of Engineers. Australia
24. Tiwari MK, Adamowski JF (2017) An ensemble wavelet bootstrap machine learning approach to water demand forecasting: a case study in the city of Calgary, Canada. Urban Water J 14:185–201. https://doi.org/10.1080/1573062X.2015.1084011
25. Snieder E, Shakir R, Khan UT (2020) A comprehensive comparison of four input variable selection methods for artificial neural network flow forecasting models. J Hydrol 583:124299. https://doi.org/10.1016/j.jhydrol.2019.124299
26. Wang Q, Wang S (2020) Machine Learning-based water level prediction in Lake Erie. Water 12:2654
27. Rumelhart DE, Hinton GE, Williams RJ (1985) Learning internal representations by error propagation. California Univ San Diego La Jolla Inst for Cognitive Science
28. Jordan MI (1986) Serial order: a parallel distributed processing approach. Technical report, June 1985-March 1986
29. Bengio Y, Simard P, Frasconi P (1994) Learning long-term dependencies with gradient descent is difficult. IEEE Trans Neural Networks 5:157–166
30. Hochreiter S, Schmidhuber J (1997) Long short-term memory. Neural Comput 9:1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735
31. Shen C, Laloy E, Elshorbagy A et al (2018) HESS Opinions: Incubating deep-learning-powered hydrologic science advances as a community. Hydrol Earth Syst Sci 22:5639–5656
32. Kratzert F, Klotz D, Brenner C et al (2018) Rainfall–runoff modelling using long short-term memory (LSTM) networks. Hydrol Earth Syst Sci 22:6005–6022
33. Kratzert F, Klotz D, Shalev G et al (2019) Towards learning universal, regional, and local hydrological behaviors via machine learning applied to large-sample datasets. Hydrol Earth Syst Sci 23:5089–5110. https://doi.org/10.5194/hess-23-5089-2019

34. Hu C, Wu Q, Li H et al (2018) Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. Water. https://doi.org/10.3390/w10111543
35. Kratzert F, Klotz D, Herrnegger M et al (2019) Toward improved predictions in ungauged basins: exploiting the power of machine learning. Water Resour Res 55:11344–11354
36. Song T, Ding W, Wu J et al (2020) Flash flood forecasting based on long short-term memory networks. Water. https://doi.org/10.3390/w12010109
37. Ding Y, Zhu Y, Feng J et al (2020) Interpretable spatio-temporal attention LSTM model for flood forecasting. Neurocomputing 403:348–359
38. Kao I-F, Zhou Y, Chang L-C, Chang F-J (2020) Exploring a long short-term memory based encoder-decoder framework for multi-step-ahead flood forecasting. J Hydrol 583:124631
39. Zhang D, Peng Q, Lin J et al (2019) Simulating reservoir operation using a recurrent neural network algorithm. Water 11:865
40. Nash JE, Sutcliffe JV (1970) River flow forecasting through conceptual models part I—a discussion of principles. J Hydrol 10:282–290. https://doi.org/10.1016/0022-1694(70)90255-6
41. Gupta H, Kling H, Yilmaz K, Martinez G (2009) Decomposition of the mean squared error and NSE performance criteria: implications for improving hydrological modelling. J Hydrol. https://doi.org/10.1016/J.JHYDROL.2009.08.003
42. Ottawa River Heritage Designation Committee (ORHDC) (2005) Background study for nomination of the Ottawa River under the Canadian heritage rivers system
43. Canadian Hydraulics Center (2010) Green kenue reference manual. National Research Council Ottawa, Ontario, Canada
44. Muñoz-Sabater J, Dutra E, Agustí-Panareda A et al (2021) ERA5-Land: a state-of-the-art global reanalysis dataset for land applications. Earth Syst Sci Data 13:4349–4383. https://doi.org/10.5194/essd-13-4349-2021
45. Eng K, Tasker GD, Milly P (2005) An analysis of region-of-influence methods for flood regionalization in the Gulf-Atlantic rolling plains 1. JAWRA J Am Water Resour Assoc 41:135–143
46. Fox J (2015) Applied regression analysis and generalized linear models. Sage Publications
47. Romanowicz RJ (2007) Data based mechanistic model for low flows: Implications for the effects of climate change. J Hydrol 336:74–83. https://doi.org/10.1016/j.jhydrol.2006.12.015
48. Romanowicz R (2010) An application of a log-transformed low-flow (LTLF) model to baseflow separation. Hydrol Sci 55:952–964. https://doi.org/10.1080/02626667.2010.505172
49. Moog DB, Whiting PJ, Thomas RB (1999) Streamflow record extension using power transformations and application to sediment transport. Water Resour Res 35:243–254. https://doi.org/10.1029/1998WR900014
50. Cao XH, Stojkovic I, Obradovic Z (2016) A robust data scaling algorithm to improve classification accuracies in biomedical data. BMC Bioinform 17:359. https://doi.org/10.1186/s12859-016-1236-x
51. O'Malley T, Bursztein E, Long J, et al (2019) Keras Tuner. GitHub repository
52. Van Rossum G, Drake FL (2009) Python 3 reference manual. CreateSpace, Scotts Valley, CA
53. Chollet F (2015) keras. GitHub
54. Martín Abadi, Ashish Agarwal, Paul Barham, et al (2015) TensorFlow: large-scale Machine Learning on heterogeneous systems
55. Harris CR, Millman KJ, van der Walt SJ et al (2020) Array programming with NumPy. Nature 585:357–362. https://doi.org/10.1038/s41586-020-2649-2
56. McKinney W et al. (2010) Data structures for statistical computing in python. In: Proceedings of the 9th python in science conference. Austin, TX, pp. 51–56
57. Waskom ML (2021) Seaborn statistical data visualization. J Open Source Softw 6:3021. https://doi.org/10.21105/joss.03021
58. Pedregosa F, Varoquaux G, Gramfort A et al (2011) Scikit-learn: Machine learning in Python. J Mach Learn Res 12:2825–2830
59. Hunter JD (2007) Matplotlib: a 2D graphics environment. Comput Sci Eng 9:90–95. https://doi.org/10.1109/MCSE.2007.55
60. Cho K, van Merrienboer B, Gulcehre C et al (2014) Learning phrase representations using RNN Encoder-Decoder for statistical machine translation. arXiv:1406.1078
61. Xiang Z, Yan J, Demir I (2020) A rainfall-runoff model with LSTM-based sequence-to-sequence learning. Water Resour Res 56:e2019WR025326
62. Herbert ZC, Asghar Z, Oroza CA (2021) Long-term reservoir inflow forecasts: enhanced water supply and inflow volume accuracy using deep learning. J Hydrol 601:126676
63. Bahdanau D, Cho K, Bengio Y (2016) Neural machine translation by Jointly learning to align and translate. arXiv:1409.0473
64. Mekonnen BA, Nazemi A, Mazurek KA et al (2015) Hybrid modelling approach to prairie hydrology: fusing data-driven and process-based hydrological models. Hydrol Sci J 60:1473–1489
65. Moriasi DN, Arnold JG, Van Liew MW et al (2007) Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. Trans ASABE 50:885–900