

# UC Irvine

## UC Irvine Electronic Theses and Dissertations

### Title

Exploration of Deep Learning Models on Streamflow Simulations

### Permalink

<https://escholarship.org/uc/item/2vv5n2fb>

### Author

Li, Jinyang

### Publication Date

2021

Peer reviewed|Thesis/dissertation

UNIVERSITY OF CALIFORNIA,  
IRVINE

Exploration of Deep Learning Models on Streamflow Simulations

THESIS

submitted in partial satisfaction of the requirements  
for the degree of

MASTER OF SCIENCE

in Civil and Environmental Engineering

by

Jinyang Li

Committee Members:  
Professor Kuolin Hsu, Chair  
Professor Soroosh Sorooshian  
Professor Russell Detwiler

2021



# TABLE OF CONTENTS

<b>LIST OF FIGURES</b> .....	iv
<b>LIST OF TABLES</b> .....	v
<b>ACKNOWLEDGMENTS</b> .....	vi
<b>ABSTRACT</b> .....	viii
<b>1 Introduction</b> .....	1
1.1 Streamflow Simulation .....	1
1.2 Research Motivation and Objectives .....	3
1.3 Organization of the Thesis .....	4
<b>2 Background</b> .....	5
2.1 Overview of Streamflow Simulation Strategies .....	5
2.1.1 Physically-based Method .....	5
2.1.2 Data-driven Method .....	8
2.2 Overview of Deep Learning .....	11
<b>3 Exploration of Deep Learning Models on Streamflow Simulations</b> .....	14
3.1 Data and Materials .....	14
3.1.1 Study Area .....	14
3.1.2 Gauged Precipitation and Soil Moisture .....	14
3.1.3 Gauged Streamflow .....	15
3.1.4 ParFlow-CLM Model Products .....	16
3.1.5 Data Pre-processing .....	17
3.2 Methodology .....	19
3.2.1 ANN .....	19
3.2.2 RNN .....	20
3.2.3 LSTM .....	21
3.2.4 U-net .....	23
3.3 Model Development .....	25

3.3.1 Training and Evaluation Methods .....	25
3.3.2 Implementation.....	26
3.3 Results and Discussion.....	27
3.3.1 Performance of ANN, RNN and LSTM models on Different Lead Time .....	27
3.3.2 Performance of ANN, RNN and LSTM Models on Different Sites. ....	29
3.3.3 Performance of the U-net Model on Learning From the ParFlow-CLM Model. .....	33
<b>4 Summary and Conclusions.....</b>	<b>37</b>
4.1 Conclusions .....	37
4.2 Deficiency and future work.....	38
<b>REFERENCE.....</b>	<b>39</b>

## LIST OF FIGURES

Figure 2-1: Schematic of the ParFlow-CLM model. ....	6
Figure 3-1: The locations of ARS sites in the Little Washita watershed. ....	15
Figure 3-2: The locations of USGS streamflow sites in the Little Washita watershed. ....	16
Figure 3-3: The typical structure of the ANN model. ....	20
Figure 3-4: The structure of the RNN model. ....	21
Figure 3-5: The structure of a LSTM module. ....	22
Figure 3-6: The structure of the U-net model. ....	25
Figure 3-7: Pearson correlation coefficients ( $r$ ) of LSTM, ANN and RNN models with different lead time on validation dataset at the Ninnekah site. ....	27
Figure 3-8: Pearson correlation coefficients ( $r$ ) of LSTM, ANN and RNN models with different lead time on training dataset at the Ninnekah site. ....	28
Figure 3-9: Observations and ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Ninnekah site. ....	30
Figure 3-10: Observations and ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Cement site. ....	32
Figure 3-11: Observations and ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Cyril site. ....	33
Figure 3-12: Performance of the U-net model. ....	34
Figure 3-13: Simulations of the ParFlow-CLM and the U-net model for validation on the grid (9,37). ....	35
Figure 3-14: Simulations of the ParFlow-CLM and the U-net model for validation on the grid (26, 39). ....	35

## LIST OF TABLES

Table 3-1: The Pearson correlation coefficient ( $r$ ) and RMSE at three sites. ....	31
--	----

## **ACKNOWLEDGMENTS**

I would like to sincerely thank my advisor Professor Kuolin Hsu for his support, encouragement and care throughout my Master time. Professor Hsu encouraged me to enjoy the process to make things right, because it is beneficial to find the truth while writing a paper is just summarizing something. I was also astonished by his broad knowledge in different areas of science and always being ready to embrace new technologies, which inspired me to work hard and be ready for incoming challenges. I would also like to thank my committee members, Professor Soroosh Sorooshian and Professor Russell Detwiler, for kindly spending their time and giving me valuable suggestions and evaluations on my Thesis. I benefited and broadened my knowledge on hydrology from their instructions on Hydrologic Systems and Groundwater Hydrology.

My sincerest gratitude goes to my colleagues in Professor Hsu' lab, who were always supportive and caring. I would like to thank Ai-Ling, who acted like a big sister and gave me support when I needed it and arranged everything nicely. I would also like to thank Dr. Baoxiang Pan, who always gave me insightful comments on my research, encouraged me to think and explore more.

I would also like to thank all my friends in Irvine and around the world. You guys made me feel like I am unique and brought happiness to my life. I would like to thank my roommates Jiajian Luo and Rui Han, who cultivated me a warm place in Irvine. My special thankfulness goes to my mentors in ISG, Willis Lin and Larry Tan, who provided me consistent support and care, and made me feel secure in unfamiliar places.

My deepest gratitude goes to my family, and I would not be here with their support. To my parents, Zhuolong Li and Pinglan Xie, I am really thankful for your unconditional love and care, and fully respect and support every decision I make. To my best friends in the world, Jiarui Li and Bin Xie, who have always been here that I can share my happiness and sorrowfulness with.

Partial financial support of this study is made available from the National Institutes of Health (U19 AI129326) and Department of Energy (DOE Prime Award DEIA0000018).

# **ABSTRACT**

Exploration of Deep Learning Models on Streamflow Simulations

By

Jinyang Li

Master of Science in Civil and Environmental Engineering

University of California, Irvine, 2021

Professor Kuolin Hsu, Chair

Streamflow simulation and forecasting is an important approach for water resources management and flood mitigation, and both physically-based and data-driven models have been used to predict the streamflow at daily and hourly scale. Although physically-based models can help understand the underlying mechanisms of the hydrological process based on mathematical and physical equations, the performance of the models largely depends on the availability and quality of the spatial information. The computation cost also limits its application in the fast, reliable and accurate streamflow simulation and forecasting. In contrast, data-driven models, especially deep learning models, are capable of learning the nonlinear relationships between the inputs and outputs without explicitly referring to the physical process. In addition, deep learning techniques, powered by graphics processing units (GPUs) can allow researchers to train models 10 or 20 times faster, which is more promising for inputs-outputs simulation.

Currently, the deep learning models rely heavily on prior streamflow information to make predictions, while the hydrological observations are limited in the real world applications. In this Thesis, the effectiveness of deep learning models on streamflow simulation without using the previous streamflow information was explored. In specific, the three deep learning models the ANN, RNN and LSTM models were used to simulate the streamflow at multiple lead time and different sites using the precipitation and soil moisture. In addition, the effectiveness of deep learning models on learning from a physically-based model to generate pixel-to-pixel streamflow mapping has also been tested.

All three deep learning models can achieve reliable and accurate streamflow simulations at the lead time of 1-5 h at the Ninnekah site with Pearson correlation coefficient ( $r$ ) over 0.9 for all three models. Based on the prediction of the peak and low flow on different sites, the LSTM model shows more consistent and reliable streamflow simulation results and can work better in different basin scales. For the U-net model, it can learn the patterns of ParFlow-CLM model well and achieves average  $r$  of 0.770 and 0.820 in the whole domain for validation and calibration, respectively. However, the models have limitations in predicting the peak due to the feature of the imbalanced data.

# Chapter 1

## 1 Introduction

### 1.1 Streamflow Simulation

Streamflow simulation and forecasting is crucial for water resources management and flood control (Tongal et al., 2018; Noori & Kalin., 2016). Precipitation is taken as a major component and determined the occurrence of drought or flood event in hydrological process (Hu et al., 2018), but the relationship between precipitation and streamflow is highly nonlinear and complex because of the spatiotemporal variability of watershed characteristics and heterogeneity in precipitation (Moradkhani et al., 2004). In addition, streamflow is also driven by other factors including groundwater, snowmelt and evaporation (Haiyan el at., 2018). Currently, physically-based models and data-driven models have been developed to predict streamflow at daily or hourly scale. Physically-based models work by representing physical processes and boundary conditions, and solving the complex mathematical equations behind it, while data-driven models learn the underlying relationships between inputs and outputs without explicitly referencing the physical process (Noori & Kalin., 2016).

Although physically-based models can help understand the underlying mechanisms of hydrological process, the performance of the models largely depends on the availability and quality of the spatial information on model parameters and there is no firm evidence that those structures and parameters correspond to the hydrologic reality (Clark et al., 2017). In addition, physically-based models also struggle with trade-offs between

complexity of physical realism, ensemble and scale size, and simulation time period due to the available computing resources. In contrast, data-driven models obtain the relationships between the inputs and outputs directly from the data like streamflow and relevant hydrological inputs instead of simulating the internal hydrological processes (Liu et al., 2020). For example, Hsu et al. (1995) developed a combination of linear least squares and multi-start simplex (LLSSIM) method to estimate and update the parameters of Artificial Neural Network (ANN), which is a popular data-driven method in water resources, and simulated the nonlinear behaviors of rainfall-runoff process. Other studies also proved the success of ANN model in streamflow simulation (Halff et al., 1993; Zhu et al., 1994; Karunanithi et al., 1994; Shamseldin et al., 1997).

In addition to ANN, other types of data-driven model that have recurrent units in their structure, such as Recurrent Neural network (RNN), Gated recurrent unit (GRU) and Long Short-Term Memory (LSTM), have been applied in the streamflow simulation and forecasting. Since the response of the hydrological system not just depends on the input but also the current state, the use of recurrent type of data-driven models that provide dynamic internal feedback in the system outperform the ANN in streamflow simulation and have been proved by researchers (Anmala et al., 2000; Kumar et al., 2005; Kratzert et al., 2018; Liu et al., 2020; Gao et al., 2020).

Regardless of types of model for streamflow simulation and forecasting, we only have limited hydrological observations and only get part of the picture of the whole scenarios, especially for ungauged basins. Therefore, the above-mentioned data-driven models are usually lumped. In other words, they take each watershed as a unit and represent the averages of input and output variables for the entire watershed, and only can predict the

streamflow at limited points.

How to take advantage of limited hydrological observations to provide accurate, reliable predictions remains to be explored. Recent study has shown that the trained recurrent type of data-driven models can be transferred and applied to predict the streamflow of other different watersheds, and can even achieve better simulation results than the calibrated physically-based models (Kratzert et al., 2019). This evidence suggests that we can train a learning model at a watershed with relatively sufficient data to learn the hydrological response behavior. With careful fine-tuning using few observational data, we may fit this model to different catchments of similar properties. On the other hand, if it is an ungauged basin and there is no observed data for model training, we may pre-train a learning machine using potentially unlimited synthetic data generated by process-based hydrological models.

## 1.2 Research Motivation and Objectives

Although the simulation and forecasting of streamflow has been developed, the application of using limited hydrological observations without using prior streamflow information remains to be explored. In this thesis, the effectiveness of deep learning models on streamflow simulation without using the previous streamflow information was explored. In addition, the abilities of deep learning models on learning from a physically-based model to generate pixel-to-pixel streamflow mapping has also been tested.

Specifically, this thesis address the following questions:

- 1) Can deep learning models to achieve accurate and reliable streamflow simulation without using prior streamflow information?

- 2) What is the performance of the ANN, RNN and LSTM models on different lead time and sites?
- 3) Can a fully connected deep learning model learn the physical process and mechanisms from the ParFlow-CLM mode and generate pixel-to-pixel streamflow mapping?

### 1.3 Organization of the Thesis

Chapter 2 provides background information on the physically-based and data-driven methods, specifically deep learning models on streamflow simulation and forecasting, and both the strength and weakness of these methods are discussed. Then the development of deep learning models, how these models work and how to use deep learning techniques to enhance current streamflow simulation are discussed. Chapter 3 explores the application of deep learning in streamflow simulation where ANN, RNN, LSTM and U-net models are developed. Conclusions and future work are drawn in Chapter 4.

## Chapter 2

### 2 Background

#### 2.1 Overview of Streamflow Simulation Strategies

##### 2.1.1 Physically-based Method

In this section, a brief overview of important previous physically-based methods for streamflow simulation is given, and it reviews details of the state-of-the-art physically-based model ParFlow-CLM, since its input and simulated results are used as training data for the U-net model in this research.

Water flow in channels comes from the surface runoff and groundwater, which can be simulated by the Land surface models (LSMs) and groundwater models, respectively. However, Traditional LSMs do not treat the LSM boundary in a fully-process-based fashion, and assume zeros' flux or set zeros' value equal to the moisture content, which ignores the processes that can change surface fluxes and runoff (Maxwell & Miller., 2005). On the contrary, groundwater models (GWMs) often overly simplify upper boundary conditions that ignore soil heating, runoff, snow and root-zone uptake when dealing with saturated and unsaturated water flow (Maxwell & Miller., 2005). To compensate for the shortcomings of LSM and GWMs, the state-of-the-art model ParFlow-CLM coupled the aforementioned two models into one single column model.

The groundwater model uses the ParFlow, which was developed by Ashby and Falgout (1996). It uses the Richards equation to compute three-dimensional variably saturated groundwater flow, while applying the two-dimensional kinematic wave and Manning's

equations to represent overland flow [Kollet & Maxwell., 2006; Kuffour et al., 2020]. The CLM model uses the version developed by Dai (2003), which computes temperature and water balance including canopy water storage, snow water, soil moisture and runoff [Dai et al., 2003].

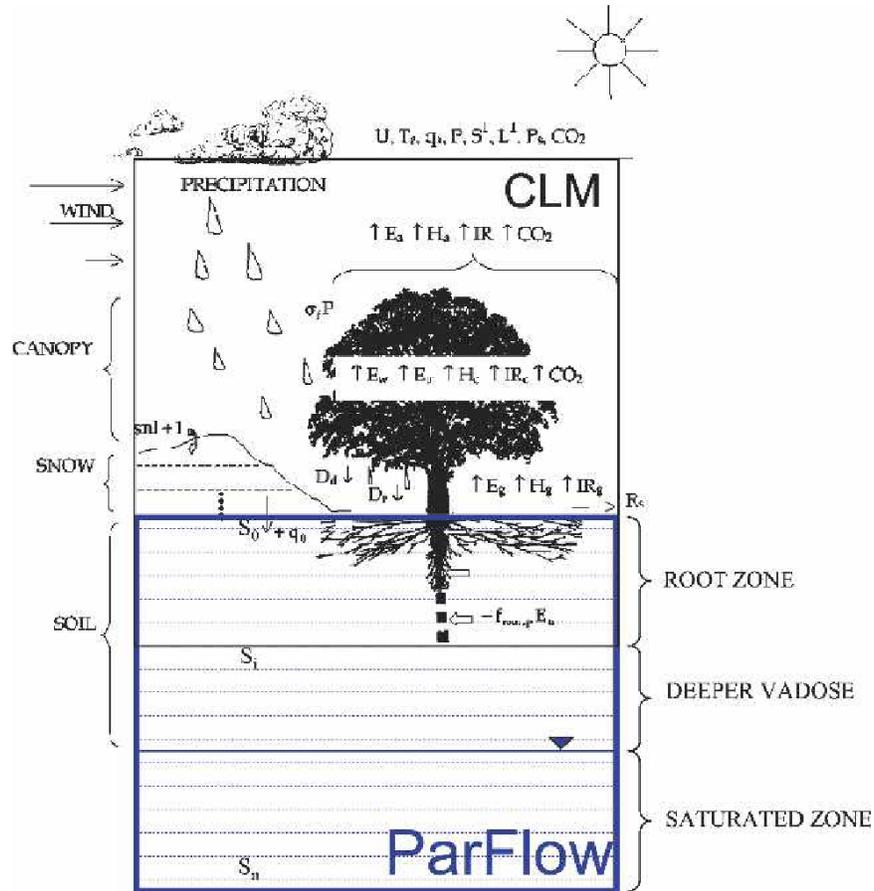


Figure 2-1: Schematic of the ParFlow-CLM model [Maxwell & Miller., 2005]. The lower box is represented by the ParFlow model, and the upper box is calculated by the CLM model. Two models overlap and communicate with each other at the root zone.

The two models communicate via the shallow soil layers where they are the first soil layer below the ground for CLM and the uppermost cell layer in ParFlow [Maxwell & Miller.,

2005], as shown in Figure 2-1. In this way, the soil moisture simulated by ParFlow is passed into CLM and replaces the soil moisture module of CLM. Then the fluxes, such as infiltration, evaporation and root uptake, generated by CLM, are delivered to ParFlow at each time step [Maxwell & Miller., 2005; Kollet & Maxwell., 2008]. Additionally, the dynamic groundwater may have transient, three-dimensional components, not just limited to one and two dimensional interactions between surface and subsurface [Kollet & Maxwell., 2008]. For example, the movement speed of water is determined by the hydraulic gradient of the aquifer, and then the water would move from high hydraulic head to the relatively low hydraulic head. Therefore, a change in water table due to the recharging from precipitation will cause a response throughout the system, and will influence the groundwater behavior and energy balance in discharge areas [Kollet & Maxwell., 2008]. ParFlow-CLM achieves this by replacing the integrated, fully distributed overland flow simulation in ParFlow. Comparison of runoff between the uncoupled and coupled models shows a better groundwater representation with coupled land surface schemes (Maxwell & Miller., 2005).

Soil Water Assessment Tool (SWAT) is another common physically-based model, which has been widely used for simulating the movement of water, sediment, nutrients and crop growth for a wide range of scales (Noori & Kalin., 2016). The strength of SWAT is that it incorporated upland and channel processes into one package, but every one of these processes simplifies the reality (Gassman et al., 2007), and current SWAT models often perform poorly where groundwater is a major contributor to streamflow, owing to the simplified groundwater dynamics in SWAT(Liu et al., 2020). Noah LSM is another popular physically-based model, which has advantages in the areas of cold season processes and

forecasting in low-level temperature and humidity (Ek et al., 2003). However, it neglects topographic effects, assumes spatially continuous soil moisture values and simplifies the parameters of surface runoff and baseflow, which makes the model's soil column wet and dry too quickly, and the simulated runoff is much larger than observations (Rosero et al., 2011).

### 2.1.2 Data-driven Method

Although physically-based models can help understand the underlying mechanisms of hydrological processes, they usually require a large number of inputs, such as hydrological data, meteorological data, topographic data, soil properties and vegetation type (Gao et al., 2020), and the performance of the model depends on the quality of the input data and parameters (Noori & Kalin., 2016). In addition, physically-based models still struggle with trade-offs between process and spatial complexity, domain and ensemble size, and the time period of the model simulation due to the high computational cost (Clark et al., 2017). Alternatively, data-driven models, like artificial neural networks (ANNs) which are well known for their capability of learning the nonlinear relationship between input variables and outputs without explicitly referring to the physical process (Noori & Kalin., 2016; Li et al., 2020), are promising methods for input-output simulation and forecasting. The advent of fast graphics processing units (GPUs), has offered thousands of times more matrix computing power than central processing units (CPUs) and has allowed researchers to train models 10 or 20 times faster (LeCun et al., 2015; Shen et al., 2018). Data-driven models powered by GPU, especially deep learning models, become increasingly popular and advances the knowledge of water resources, such as streamflow simulation (Kratzert

et al., 2018; Liu et al., 2020; Gao et al., 2020), precipitation prediction (Pan et al., 2019; Tao et al., 2018) and soil condition estimation (Fang et al., 2017; Ahmed et al., 2021).

Artificial Neuron Network (ANN) is a popular data-driven method for prediction and forecasting in the fields of water resources and environmental science (Maier & Dandy, 2000), and is capable of representing arbitrarily complex nonlinear relations between inputs and outputs of any system (Hsu et al., 1995). A typical ANN model consists of three parts: input layer, hidden layer and output layer. There are multiple nodes in each layer, and each node is connected with all the nodes in the previous layer. The input layer receives the inputs and transfers the incoming signals to the hidden layer then to the output layer to generate outputs. The sigmoidal type functions such as logistic and hyperbolic tangent functions are commonly selected as the activation function to achieve nonlinear transformation between inputs and outputs (Maier & Dandy, 2000).

Although the concept of ANN was proposed in 1943 (McCulloch & Pitts, 1943), the model was not so popular until the introduction of backpropagation for training in 1986 (Rumelhart et al., 1986; Maier & Dandy, 2000). Then, many studies applied ANNs to simulate rainfall-runoff processes (Halff et al., 1993; Zhu et al., 1994; Hsu et al., 1995; Karunanithi et al., 1994; Shamseldin et al., 1997). Although backpropagation provides gradient search strategies, the results are quite sensitive to the initial points and easily trapped in local optima, especially for high dimensional data (Hsu et al., 1995). To overcome this issue, Hsu et al (1995) used a combination of linear least squares and multi-start simplex (LLSSIM) method to reliably find the global or near-global solution to the problem in streamflow prediction. Other methods include the addition of random noise to the connection weights or inputs (Maier & Dandy, 2000). Researchers also investigated

five different activation functions on model performance in streamflow forecasting, and the results indicate that the logistic function yields the best model forecast combination performance (Shamseldin et al., 2002).

However, a drawback of the ANN model for time series analysis is that the information about the sequential order of the input is lost (Kratzert et al., 2018). The Recurrent Neural Networks (RNN) is a special feed-forward network, which has time dependencies and time delays (McCulloch & Pitts, 1943; Giles et al., 1994). While the typical ANN model needs to identify the appropriate time-delay between inputs and outputs, RNN structure offers dynamic internal feedback in the system and leads to a reduction of trained parameters without needing prior knowledge of the lag time (Hsu et al., 1997). Other studies proved that RNN can produce better streamflow predictions for both single and multiple steps ahead (Anmala et al., 2000; Kumar et al., 2004).

Although RNN provides dynamical feedback in the system, it suffers from exploding or vanishing problems in the temporal evolution of backpropagation, especially for problems where the minimal time lag between input variables and corresponding outputs are long (Hochreiter & Schmidhuber., 1997). This gradient problem may result in oscillating weights, take a long time for model training, or not work at all (Hochreiter., 1991; Hochreiter & Schmidhuber., 1997). A novel recurrent network structure, Long Short-Term Memory (LSTM), introduced by Hochreiter & Schmidhuber in 1997, overcome these error backpropagation problems through enforcing constant error flow through internal states of special units of the model (Hochreiter & Schmidhuber., 1997). Compared to traditional RNN, LSTM has additional cell state or cell memory to store and control information flow within the LSTM cell (Hochreiter & Schmidhuber., 1997; Kratzert et al., 2018). More

recently, studies have shown that LSTM, which has internal memory and is capable of learning the long-term dependency between inputs and outputs, can achieve better results in streamflow prediction compared to ANN and RNN (Kratzert et al., 2018; Liu et al., 2020; Gao et al., 2020).

In addition to providing dynamical information in the system to make more precise predictions, other researchers tried to apply classification or clustering methods to better understand the insights or underlying processes of the data before making predictions. Studies show that the use of a self-organizing feature map (SOFM) [Kohonen, 1989], which uses neighborhood function to preserve the topological properties of the input space, has the ability to provide rapid and precise streamflow simulation [Hsu et al., 2002; Moradkhani et al., 2004]. Other research also indicates that with the use of K-means clustering [Anderberg, 1973] based on geomorphological factors, even a simple ANN model can achieve decent simulated results for the overland flow depths in space [Huang et al., 2021].

## 2.2 Overview of Deep Learning

Deep learning, enabled by deep neural networks, belongs to the family of machine learning, and the adjective ‘deep’ refers to the multiple layers of the model. Traditional machine learning models have limitations in processing the raw forms of natural data and require careful engineering and considerable domain knowledge to extract useful internal representations or patterns from the data (LeCun et al., 2015). However, deep learning models can be fed with raw data and automatically discover the representations through

composing non-linear modules where each can transform the representations from raw input into higher and more abstract levels, which models can identify (LeCun et al., 2015). The deep learning model, which consists of an advanced artificial neuron network (ANN), has received drastically increasing attention and momentum since 2012, when it won the machine learning contest (Shen., 2018). It turned out to be more capable of finding intricate structures in high-dimensional data than other machine learning techniques (LeCun et al., 2015), and has beaten records in image recognition (Krizhevsky et al., 2012) and speech recognition (Hinton et al., 2012).

Most deep learning algorithms involve optimization, which either maximizes or minimizes the objective function or cost function that measures the error between outputs and target patterns (Bengio et al., 2017). The model then modifies its internal adjustable parameters to reduce error, and those adjustable parameters are often called weights, which are real values that define the input-output of the model (LeCun et al., 2015). Successful use of a deep learning model largely depends on how well to optimize the weights and calibrate the model with given training data (Tao., 2017). To properly adjust the weights, the derivatives of them are calculated, because it tells us the change of error according to the increase or decrease of the weights and then the weights are adjusted in the opposite direction of the derivations, which is known as the backpropagation method (Rumelhart et al., 1986a).

The ideal scenario of deep learning is that there are abundant training data and corresponding accurate labels, and the test data has the same distribution of training data, while collecting sufficient data for training is expensive and unrealistic in most scenarios (Zhuang et al., 2020). In the application of hydrology modelling, we usually have limited observational data to train a deep learning model for accurate predictions, which often

results in model overfitting due to the shortage of training data. Recent study has shown that a recurrent deep learning model trained for streamflow forecasting in one watershed can achieve advantageous performance in a different watershed, even compared to calibrated process-based hydrological models (Kratzert et al., 2019). This evidence raises the promise that we can train a learning model at a basin with relatively sufficient data, and apply it to ungauged basins to generate reliable prediction.

Historically, physically-based models and deep learning models are often treated as two different fields (Reichstein et al., 2019), because they have very different scientific diagrams: one is theory-driven and the other is data driven. In fact, these two methods can be complementary. Physically-based models can provide interpretable and sufficient data through mathematical theory and equations, which deep learning models can learn the mechanisms of the systems from. In this case, the physically-based model can serve as a ‘teacher model’ and simulate potentially unlimited synthetic data that can train the deep learning model. Deep learning models, once trained, can achieve orders of magnitude faster than the original physically-based models without sacrificing too much accuracy (Reichstein et al., 2019).

## **Chapter 3**

### **3 Exploration of Deep Learning Models on Streamflow Simulations**

#### 3.1 Data and Materials

##### 3.1.1 Study Area

The Little Washita watershed in southwestern Oklahoma is selected as the experimental place. Hydrological and meteorological measurements of the watershed have been conducted for decades, which makes it an ideal region for researchers to study water conservation and basin hydrology due to its long-term data source (Starks et al., 2014, ARS., 2021). The watershed is 611 km<sup>2</sup>, covering the land usage including: range, pasture, forest, cropland, oil waste land, quarries, urban/highways, and water. The elevation ranges from 300m to 500 m, with soil textures ranging from fine sand to silty loam. It has mean annual precipitation of 760 mm and mean annual temperature of 16 degree Celsius. Most precipitation is received in the spring and fall, and summers are long, hot, and dry, while winters are short, temperate, and dry (Rosero et al., 2011).

##### 3.1.2 Gauged Precipitation and Soil Moisture

The Little Washita watershed is managed by The U.S.D.A. Agricultural Research Service's (ARS) Grazinglands Research Laboratory (GRL) in cooperation with Oklahoma State University and the Oklahoma Climatological Survey. Currently, there are 20 stations



Washita River near Cyril, OK; USGS 07327447 Little Washita River near Cement, OK; USGS 07327550 Little Washita River East of Ninnekah, OK, as shown in Figure 3-2. These three sites provide records of streamflow from the year of 1992 to present, and streamflow is tracked every 15 minutes. The streamflow of three sites were collected during 2017 - 2019 and were transformed into hourly scale to fit the precipitation and soil moisture data. The missing values were replaced with the values of previous time since the streamflow is continuous and does not change a lot within a very short time.



Figure 3-2: The locations of USGS streamflow sites in the Little Washita watershed.

#### 3.1.4 ParFlow-CLM Model Products

The ParFlow-CLM model provides potentially unlimited synthetic data that can be learned by a deep learning model. The simulated streamflow by ParFlow-CLM is selected as the target of the U-net model, and the input and simulated soil moisture of ParFlow-CLM are selected as the inputs for the U-net model. The inputs include:

1. Press: Atmospheric Pressure (pa)
2. DSWR: Downward Visible or Short-Wave radiation ( $W/m^2$ )

3. DLWR: Downward Infa-Red or Long-Wave radiation ( $\text{W/m}^2$ )
4. APCP: Precipitation rate( $\text{mm/s}$ )
5. Temp: Air temperature (K)
6. UGRD: West-to-East or U-component of wind (m/s)
7. VGRD: South-to-North or V-component of wind (m/s)
8. SPFH: Water-vapor specific humidity (kg/kg)
9. Saturation at the 20th model layer from the bottom (1-2m below ground surface)
10. Saturation at the 21th model layer from the bottom (0.4m-1m below ground surface)
11. Saturation at the 22nd model layer from the bottom (0.1m-0.4m below ground surface)
12. Saturation at the 23rd model layer from the bottom (surface layer; 0m-0.1m below ground surface)

### 3.1.5 Data Pre-processing

Considering some ungauged basins that there are limited hydrological records and gauged streamflow records were missing, previous streamflows were excluded in the input variables compared with other studies (Gao et al., 2020; Hu et al., 2018). To compensate for the absence of the streamflow stage information, the cumulative precipitation in the previous 1 day, 3 days, 7 days, 15 days and 30 days are calculated. Therefore, the input variables are summarized as following:

$$\text{Input} = f(p_{t-1}, p_{t-2}, \dots p_{t-6}, cp_{1d}, cp_{3d}, cp_{7d}, cp_{15d}, cp_{30d}, \text{soil}_{5\text{cm}}, \text{soil}_{25\text{cm}}, \text{soil}_{45\text{cm}})$$

where  $p_{t-1}$  refers to the precipitation at time of  $t-1$ ;  $cp_{1d}$  sums the cumulative precipitation in previous 1d at time of  $t-1$ ;  $soil_{5cm}$  defines the soil moisture underground 5 cm.

In this study, the aim is to predict the streamflow of 15 h in the future, and the lead time is set to 1h, 3h, 5h, 7h, 9h, 11h, 13h and 15h corresponding to the streamflow at time of  $t$ ,  $t+2$ ,  $t+4$ ,  $t+6$ ,  $t+8$ ,  $t+10$ ,  $t+12$  and  $t+14$ . The precipitation and soil moisture is normalized using the min-max method, as shown in equation 3.1.

$$x_{norm} = \frac{x - x_{min}}{x_{max} - x_{min}} \quad (3.1)$$

where  $x_{norm}$  denotes the variable after normalization;  $x_{min}$  and  $x_{max}$  refer to the minimum and maximum value of  $x$ .

In addition, considering the imbalance property of the streamflow data, the logarithm function with base 10 was used to transform the original streamflow values.

The ParFlow-CLM product data are normalized using the mean-std normalization method, shown in equation 3.2. This is because some data, like UGRD and VGRD, the negative values means the direction of the wind is opposite to the direction of positive values, the use of min-max normalization method is not proper here.

$$x_{norm} = \frac{x - x_{mean}}{x_{std}} \quad (3.2)$$

where  $x_{mean}$  refers to the mean value of  $x$  and  $x_{std}$  denote the standard deviation value of  $x$ .

## 3.2 Methodology

### 3.2.1 ANN

A typical ANN model has three parts: input layer, hidden layer and output layer, as shown in Figure 3-3. The process units in each layer are called neurons or nodes, and the number of neurons in each layer and the number of layers are not fixed. The number of neurons on a layer is called width, while the number of layers is called the depth of the network (Shen., 2018). The parameters that connect to each layer are called weights. In general, a neuron receives and sums the multiple inputs from the previous layer, and a bias is optional to add in this process. Then the value is sent to an activation function, like Sigmoid, Tanh or ReLu to achieve non-linear transformation. The update function were applied:

$$h = \sigma(Ux + b_h) \quad (3.3)$$

$$o = b_o + Vh \quad (3.4)$$

where  $\sigma$  is the sigmoid activation function, and  $x$  is the input vector and  $h$  is the hidden state.  $b_h$  and  $b_o$  are the bias vectors, and  $U$ ,  $V$  are the parameters vectors that correspond to the input-to-hidden and hidden-to-output, respectively.

When the input is transferred from the input layer to the output layer, the error, also called loss, between the target results and the output is calculated, and the training is the process that minimizes the error. The most common training method is the backpropagation method (Rumelhart et al., 1986), which is a gradient-based method that calculates the partial derivation of each parameter in each layer according to the chain rule and then updates weights to minimize loss. The training process stops when it reaches the maximum epochs

or the loss does not decrease within man-set epochs. After the ANN model is well trained, the inputs are forward to the model to make predictions.

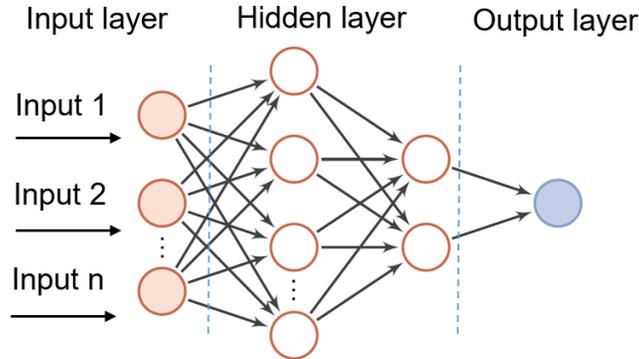


Figure 3-3: The typical structure of the ANN model. The circles represent the neurons in the model. The number of neurons in each layer can be adjusted by the user, and a bias is optional to add.

### 3.2.2 RNN

The RNN model is another type of neural network that has dynamic internal feedback within the network, and consists of a series of similar modules as shown in Figure 3-4. The  $x_t$ ,  $h_t$  and  $o_t$  refer to the input, state and out at time of  $t$ , respectively. Compared with the ANN model, the output of the RNN model at time  $t$  not only just depends on the current input  $x_t$ , but also depends on the current state of  $h_t$ . Therefore, this type of structure provides a better workflow for dynamic systems than the ANN model. Similarly, the value calculated at each module is sent to an activation function to achieve non-linear transformation. For each step from  $t = 1$  to  $t = n$ , the equations were summarized as :

$$h_t = \tanh(Ux_t + Wh_{t-1} + b_h) \quad (3.5)$$

$$o_t = b_o + Vh_t \quad (3.6)$$

where  $\tanh$  is the hyperbolic tangent activation function, and  $x_t$  is the input vector at time of  $t$  and  $h_{t-1}$  is the hidden state at time of  $t-1$ .  $b_h$  and  $b_o$  are the bias vectors, and  $U$ ,  $V$  and  $W$  are the parameters vectors that correspond to the input-to-hidden, hidden-to-hidden and hidden-to-output, respectively. (Zhang et al., 2018).

In theory, the RNN model can process all information regardless of the length of the sequences (Gao et al., 2020). Nevertheless, it suffers from exploding or vanishing gradient problems, especially for problems that the long time lag between input variables and corresponding outputs (Hochreiter and Schmidhuber., 1997).

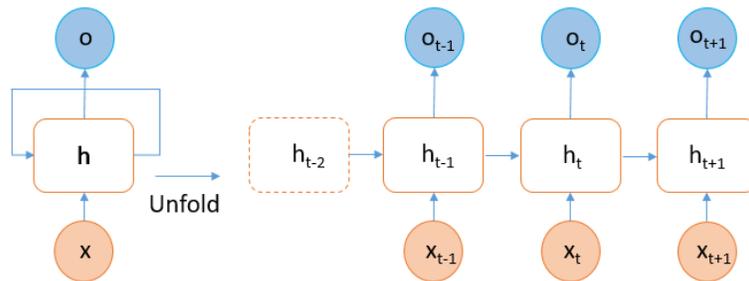


Figure 3-4: The structure of the RNN model. The right part is the unfolded picture of the left part. In the training process, the hidden state from the last time step is concated with new input to the model.

### 3.2.3 LSTM

The LSTM model is a variation of the RNN model, designed to overcome the weakness of the traditional RNN to learn long-term dependencies (Kratzert et al., 2018), and has

additional cells to store long-term and short-term information. Similar to the RNN model, the LSTM model has a series of modules, and the structure of each module is shown in Figure 3-5. The  $x_t$  is the input vector at time of  $t$ , and the  $h_{t-1}$  is the hidden state also the short-term information at time of  $t-1$ , and the  $c_{t-1}$  is the cell state also the long-term information at time of  $t-1$ . The generated  $c_t$  and  $h_t$  are then transferred to the next time step.

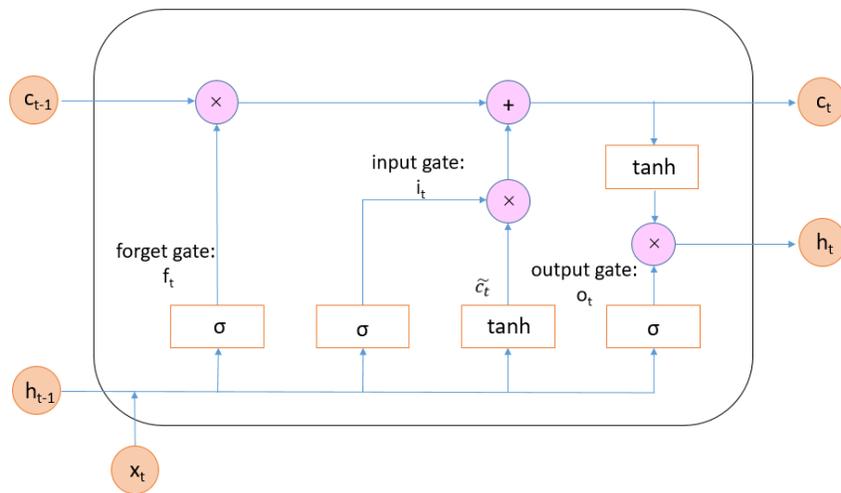


Figure 3-5: The structure of a LSTM module. It has three additional gates: forget gate, input gate and output gate, and each gate serves different functions and purposes.

A LSTM module has three gates: forget gate, input gate and output gate, and each gate serves different functions and purposes. The forget gate determines which information of the cell state  $c_{t-1}$  will be forgotten. In the next step, a potential update vector  $\tilde{c}_t$  for the cell state is calculated ranging from -1 to 1. In addition, the input gate determines which information of  $\tilde{c}_t$  is used to update the cell state in the current step. The third and the last gate is the output gate, which specifies what information of the cell state  $c_t$  is used to

generate new hidden state  $h_t$  (Kratzert et al., 2018).

The parameters in the LSTM module are given by the following equations:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \quad (3.7)$$

$$\tilde{c}_t = \tanh(W_{\tilde{c}} x_t + U_{\tilde{c}} h_{t-1} + b_{\tilde{c}}) \quad (3.8)$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \quad (3.9)$$

$$c_t = f_t \otimes c_{t-1} + i_t \otimes \tilde{c}_t \quad (3.10)$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (3.11)$$

$$h_t = \tanh(c_t) \otimes o_t \quad (3.12)$$

where  $W_f$ ,  $U_f$ ,  $W_{\tilde{c}}$ ,  $U_{\tilde{c}}$ ,  $W_i$  and  $U_i$  are learnable weight matrices, and the  $\otimes$  is element-wise multiplication of two vectors. Similarly,  $b_f$ ,  $b_{\tilde{c}}$ ,  $b_i$  and  $b_o$  are the bias vectors.  $f_t$ ,  $i_t$  and  $o_t$  are value vectors of the forget gate, the input gate and the output gate after activation function transformation.

### 3.2.4 U-net

Before the introduction of the U-net model, we firstly describe the structure of Convolution Neural Network (CNN), since U-net is a ‘fully convolutional’ network (Long et al., 2015).

The CNN model is designed to process input data in the form of multiple arrays, such as 1D signals, 2D images and 3D video images, and the four highlights that CNN takes advantage of data are local connections, shared weights, pooling and use of many layers

(LeCun et al., 2015). For the data like images and videos, there are stronger correlations within the local patches than remote components, so the CNN model extracts and takes advantage of the local features to build large-scale patterns (Pan et al., 2019). This is achieved by the convolution and pooling layer. The convolution is to use a weighted function, also known as filter bank, to multiply the input data to obtain a smoothed estimate of the position of the spaceship (Bengio., 2017). Different layers use different filter banks, but the filters share weights in each layer, so it greatly reduces the number of parameters compared to a fully connected layer, like the ANN model (Shen., 2018). Then the result of this local weighted sum is passed through a nonlinear activation function, like rectified linear activation function ReLU (LeCun et al., 2015).

After that, a pooling function is used to replace the output of the layer at each location with a summary static of the neighboring outputs (Bengio., 2017), thereby reducing the dimension of the representation and make small shifts and distortions invariant (LeCun et al., 2015). For instance, the max pooling operation retrieves the maximum results from a rectangular patch. Multiple stages of convolution, non-linearity and pooling layers can be stacked, and the learned features in the first layer are typically the edges or orientations in the image. The second layer can detect particular arrangements of the edges regardless of small variations in the edge positions, and the third layer can assemble these features into larger combinations and the following layers can detect objects as the combinations of these parts (LeCun et al., 2015). Fully-connected layers are following the last pooling layer to achieve classification and regression task, and backpropagation method (Rumelhart et al., 1986a) is used to train and update parameters in each filter.

The U-net model is a fully convolutional neural network, which provides end-to-end,

pixels-to-pixels mapping (Long et al., 2015; Ronneberger et al., 2015). The U-net model applies a convolution based contracting path to capture the resolved dynamical field information, and a symmetric transposed convolution based expanding path that gradually refines precipitation field estimation. Skip connections between symmetrical convolution and transposed convolution blocks ease the training by forcing the deeper layers to learn meaningful representations that are not well captured by shallower layers (He et al., 2016). The structure of the U-net model is shown in Figure 3-6, and the number of layers is changed considering the size of our input shape.

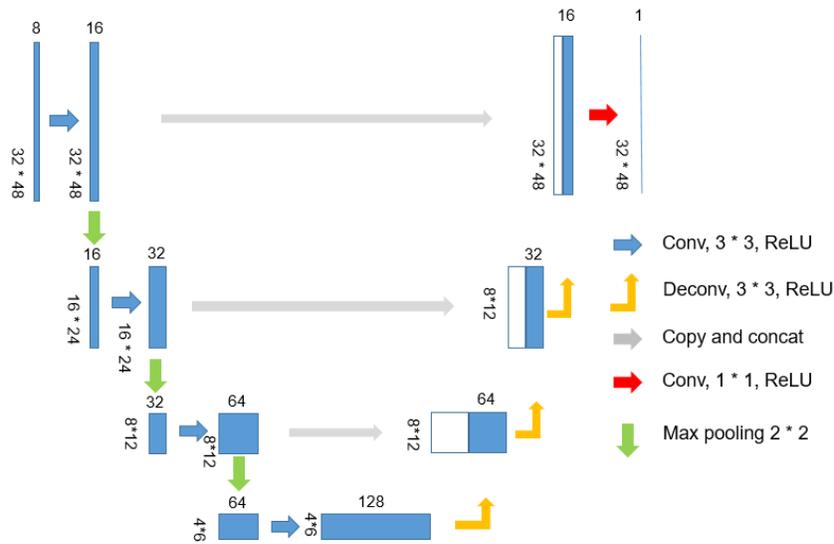


Figure 3-6: The structure of the U-net model.

### 3.3 Model Development

#### 3.3.1 Training and Evaluation Methods

The root mean square error (RMSE) between the simulated streamflow and the observed

streamflow is selected as the loss function, and the equation is shown as:

$$RMSE = \sqrt{\frac{1}{n} (S_{obs} - S_{simu})^2} \quad (3.13)$$

Where  $S_{obs}$  is the streamflow observation and  $S_{simu}$  is the streamflow simulation.

The backpropagation method (Rumelhart et al., 1986) is to compute partial derivative of the parameters in the both convolution and fully-connected layers based on chain rule, and the Adam method (Kingma et al., 2014) is chosen as the optimization method to update the parameters.

The Pearson correlation coefficient ( $r$ ) between the streamflow simulations and observations serves as the metric to evaluate model performance.

$$r = \frac{cov(S_{obs} - S_{simu})}{\sigma_{S_{obs}} * \sigma_{S_{simu}}} \quad (3.14)$$

where  $cov$  is the covariance and  $\sigma$  is the standard deviation.

### 3.3.2 Implementation

The lead time is set to 1, 3, 5, 7, 11, 13 and 15h, and each simulations were repeated 5 times to determine the lower and upper boundaries, so 35 models were built for each of the ANN, RNN and LSTM model at Ninnekah site to study the performance of three models on different lead time. To test the performance of three models on the size of the watershed, the lead time is set to 1h at Cyril, Cement and Ninnekah site, so 9 models were developed in this session. For the U-net model, the lead time is set to 1h to evaluate the effectiveness

of deep learning model learning from the physically-based model.

Since the rainfall-runoff event is rare compared to no rainfall event and streamflow is imbalanced data, the batch size and learning rate are set to 2 and 0.0001, respectively, to gradually learn the hydrological behaviors. Early stop strategy is used to avoid model overtraining: training stops as soon as the loss of validation does not decrease within a specific epoch, and the patience is set to 50.

These experiments were implemented in Google Colab, which is a computing platform that provides free computing resources to the public. The python packages include Numpy (Harris et al.,2020), Pandas (McKinney et al., 2011), Pytorch (Paszke et al., 2019) and Matplotlib (Hunter et al., 2007) are selected to process data, build models and visualize results. The GPU Tesla P100 is used to accelerate model training.

### 3.3 Results and Discussion

#### 3.3.1 Performance of ANN, RNN and LSTM models on Different Lead Time

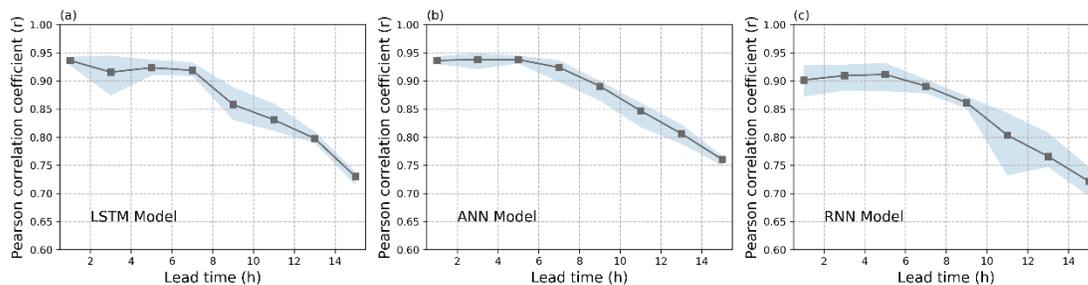


Figure 3-7: Pearson correlation coefficients (r) of (a) LSTM models, (b) ANN models and (c) RNN models with different lead time on validation dataset at the Ninnekah site. The black line with square is the average value from five repeated simulations and the blue

contour shows the minimum and maximum of five simulations.

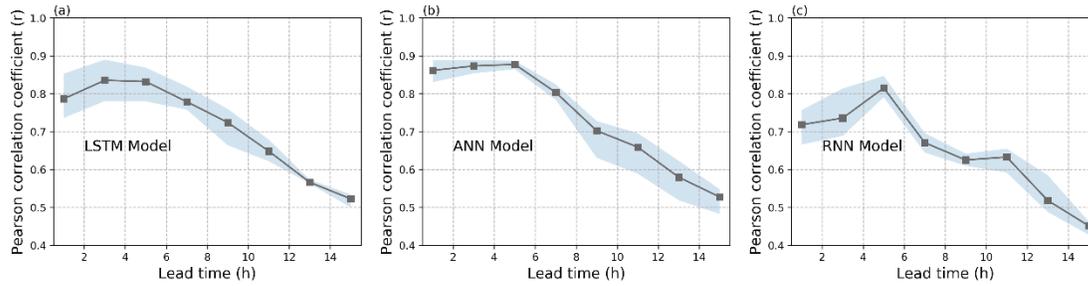


Figure 3-8: Pearson correlation coefficients ( $r$ ) of (a) LSTM models, (b) ANN models and (c) RNN models with different lead time on training dataset at the Ninnekah site. The black line with square is the average value from five repeated simulations and the blue contour shows the minimum and maximum of five simulations.

Figure 3-7 and Figure 3-8 show the Pearson correlation coefficients of LSTM, ANN and RNN models on different lead time on validation dataset and calibration dataset at the Ninnekah site, respectively. Because the initial weights are randomly generated in the training process, the results may fluctuate and be trapped in different local minimums, so the experiments are run five times to obtain relatively stable results. The black line with square is the average value of five simulations and the blue contour represents the minimum and maximum value of the five simulations. From the results, we can tell that all three models can achieve relatively stable and promising results on the lead time of 1 – 5 h on the validation process. However, when the lead time is larger than 5 h, the  $r$  decays as the lead time increases. In particular, RNN models have wild fluctuation simulations when the lead time is over 9 h. This is expected since the prediction of streamflow in further future is more challenging (Gao et al., 2020). On the calibration, all three models show best calibration results on the lead time of 5 h. Since the Ninnekah site is close to the outlet of

the watershed and it takes time for the water from the different location of the watershed to the outlet point, the models can better represent the mechanism of rainfall-runoff at larger lead time.

In terms of the  $r$ , ANN model shows best and most stable results, while LSTM model shows slightly worse simulated results and the result of RNN model is much worse. Compared with conventional thinking that more complex structures should bring more prominent results, the ANN model has the simplest structure and least parameters even shows best performance. This is because the soil moisture at different depths is able to represent the current state and the ANN model takes advantage of the current state and new input formation to achieve nonlinear mapping between inputs and outputs. However, the LSTM and RNN models that have dynamical feedback within the structure contact new information and hidden state, then these structures are processing high dimensional and redundant information, which limits their performance.

### 3.3.2 Performance of ANN, RNN and LSTM Models on Different Sites.

In this section, the performance of ANN, RNN and LSTM models at different sites are compared, and it can be found at Table 3.1. Since the Cyril site is at the upper watershed and the rainfall-runoff response is quite quick, the lead time is set to 1h for three sites for comparison. The  $r$  and RMSE for the three methods at the three sites is shown in Table 3.1. Figure 3-9 shows the observations and the ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Ninnekah site, (a) original plot and (b) semi-log plot. Since there is a large gap between the minimum value and maximum value of the

streamflow, the semi-log is provided to show the performance on the simulation of the low flow. In general, three models all can achieve good simulations. Although they kind of underestimate the peak, they generate quick responses and can predict the time of the peak accurately. The ANN and LSTM can better simulate the low flow at the spring season, but RNN have better simulations in low flow at the winter season. In addition, the streamflow simulated by all the three models dry quicker than the observations, because there are very limited precipitation during that time, and the input variables for precipitation are zeros, which leads to smaller simulated results.

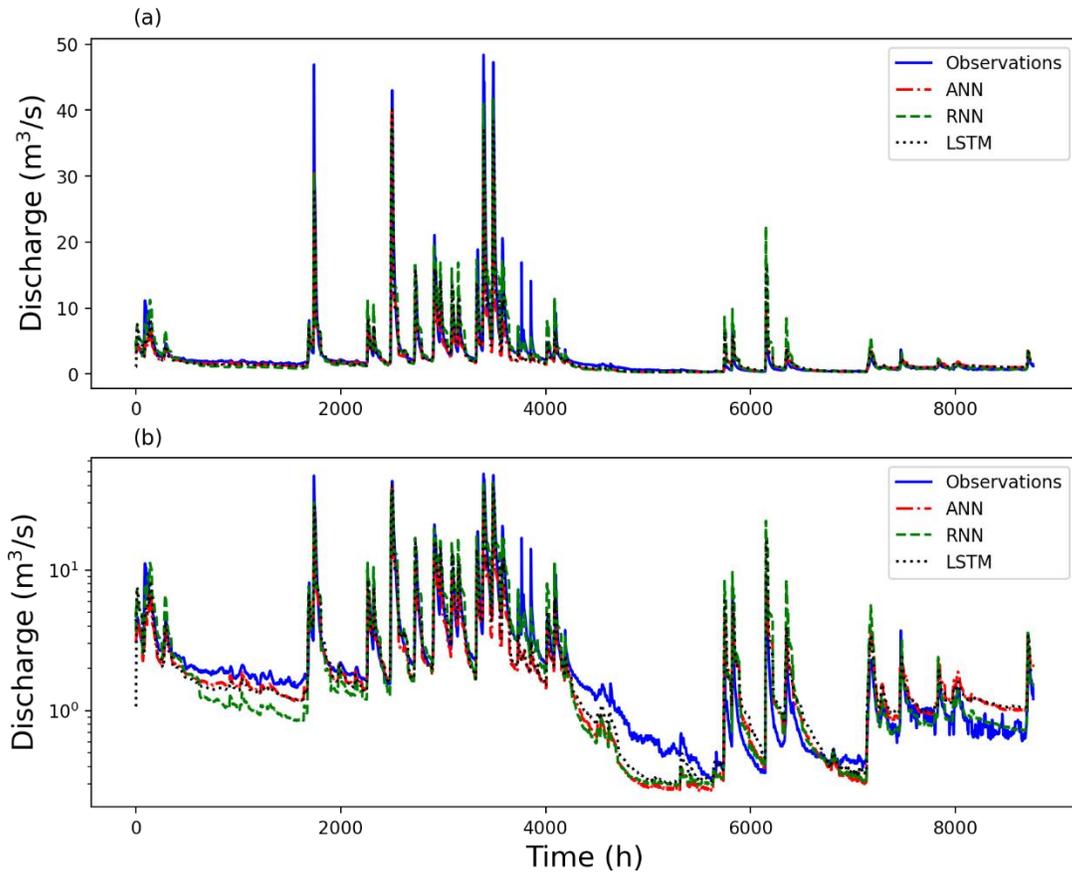


Figure 3-9: Observations and ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Ninnekah site, (a) original plot and (b) semi-log plot.

Table 3-1: The Pearson correlation coefficient (r) and RMSE at three sites.

Site	Metric	Calibration			Validation		
		ANN	RNN	LSTM	ANN	RNN	LSTM
Ninnekah	r	0.878	0.766	0.751	0.931	0.900	0.948
	RMSE	2.570	3.330	3.100	1.790	1.880	1.350
Cement	r	0.810	0.772	0.807	0.936	0.903	0.905
	RMSE	0.995	1.110	1.210	0.562	0.722	0.726
Cyril	r	0.623	0.547	0.582	0.818	0.640	0.745
	RMSE	0.549	0.770	0.520	0.358	0.660	0.396

Figure 3-10 shows the observed and ANN, RNN and LSTM model simulated results at lead time of 1h on the validation dataset at the Cement site, (a) original plot and (b) semi-log plot. In general, the LSTM and ANN model show better simulated results than the RNN model at the Cement site. The ANN model has better simulated results in low flow in summer, while it overestimates the low flow in autumn, winter and spring. The LSTM model underestimates the low flow in the summer, but it has better simulations in low flow at other seasons. However, the simulations of the RNN model fluctuate largely within the short time. Since the RNN model has dependence on the very short internal hidden state, the results oscillate if the input change rapidly. The Cement site has smaller recharge area than the Ninnekah site, so the average value of the rainfall records within the area may have larger variations and leads to oscillation of the simulations of the RNN model.

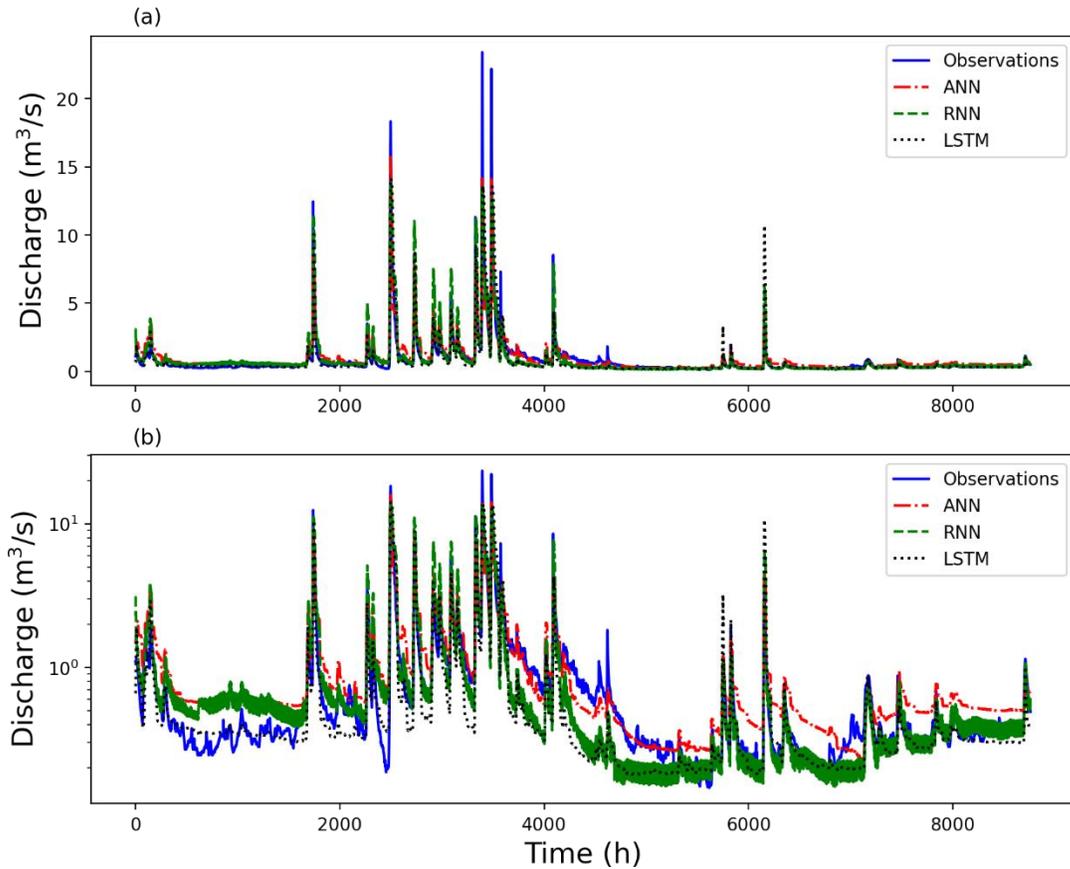


Figure 3-10: Observations and ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Cement site, (a) original plot and (b) semi-log plot.

Figure 3-11 shows the observations and the ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Cyril site, (a) original plot and (b) semi-log plot. The Cyril site is at the upper watershed, and the majority of the streamflow is very small value. The minimum, mean, maximum streamflow are 0.00014, 0.16 and 20 m<sup>3</sup>/s, respectively. The highly imbalanced data makes the learning model hard to identify its pattern, and has limitations in high flow simulation. As expected, three models have limited abilities to simulate the peak. As for the low value, the LSTM model can provide relatively better simulations than the ANN and RNN models.

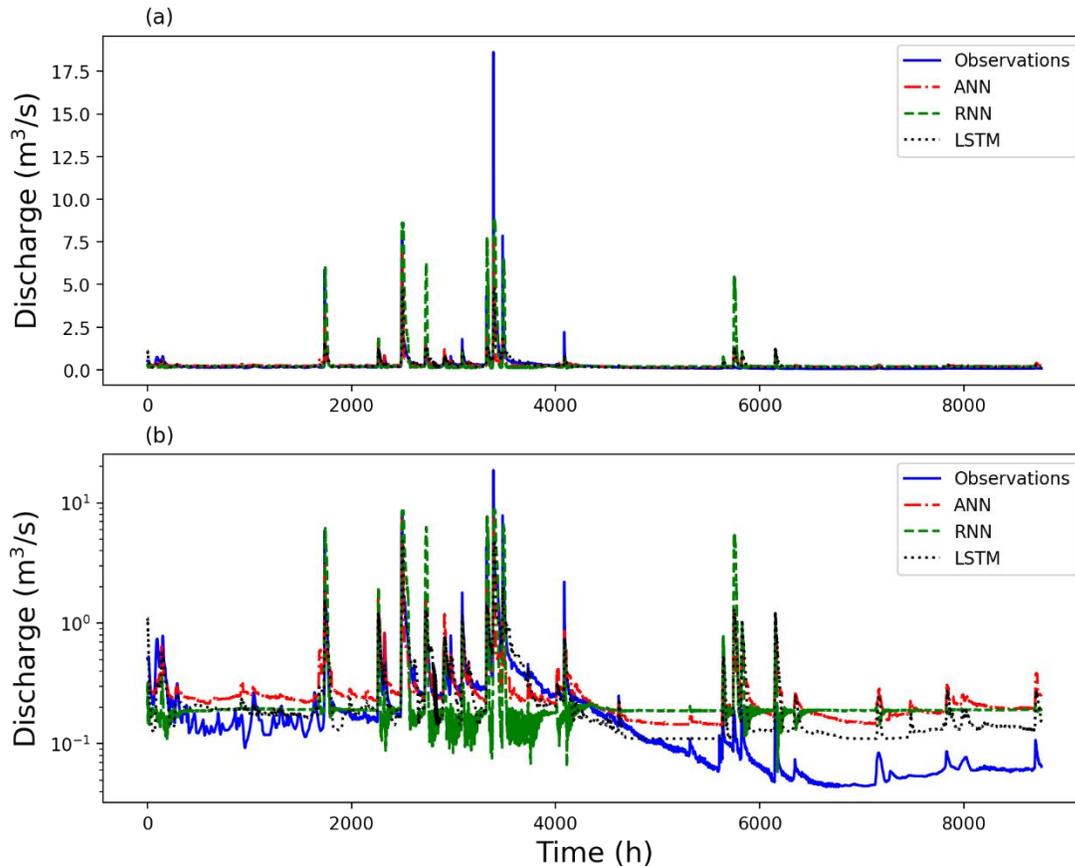


Figure 3-11: Observations and ANN, RNN and LSTM model simulations at lead time of 1h on the validation dataset at the Cyril site, (a) original plot and (b) semi-log plot.

### 3.3.3 Performance of the U-net Model on Learning From the ParFlow-CLM Model.

The performance of the U-net model learning from the ParFlow-CLM model is shown in Figure 3-12. In general, the U-net model represents good patterns on the results from the ParFlow-CLM model. The average  $r$  of validation and calibration results in the whole domain are 0.770 and 0.820, and 2.88 m<sup>3</sup>/s and 0.729 m<sup>3</sup>/s for RMSE, respectively. Some pixels whose color is white indicate the values of these pixels are constant zero, so the  $r$  cannot be calculated due to mathematical constraints that zeros cannot be divided. For

example, the values of grid (9, 37) continue to be zeros throughout the time, as shown in Figure 3-13. This also suggests that the U-net model can learn the complex situations in the domain that the streamflow remains to be zero.

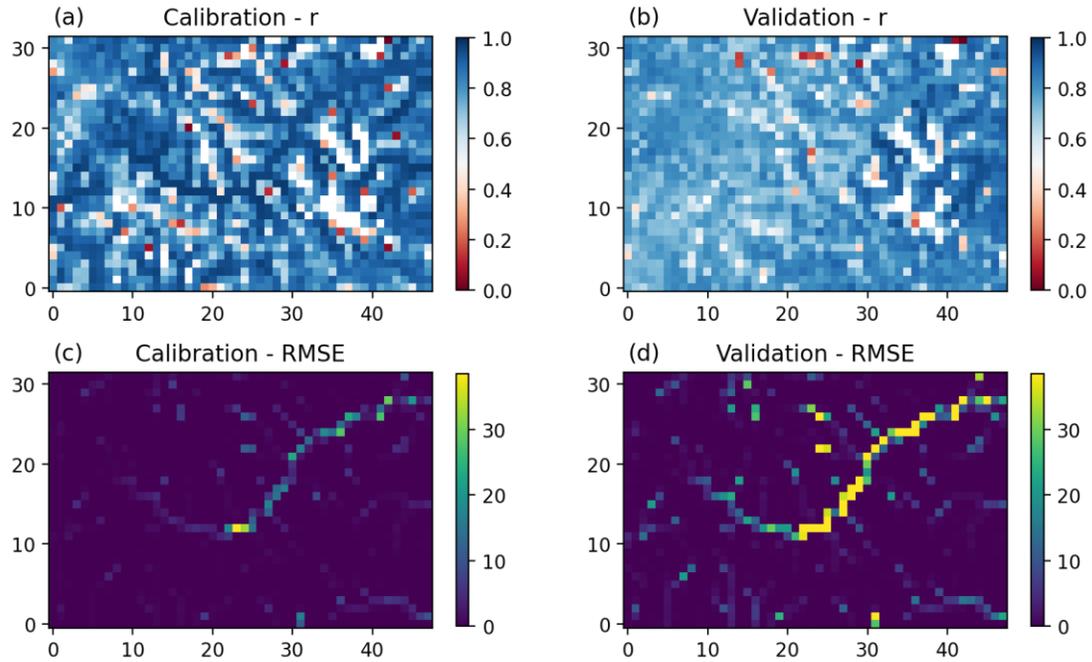


Figure 3-12: Performance of the U-net model on learning from the ParFlow-CLM model, (a) calibration – r, (b) validation – r, (c) calibration – RMSE and (d) validation – RMSE. Each pixel is calculated by the ParFlow-CLM and the U-net simulations.

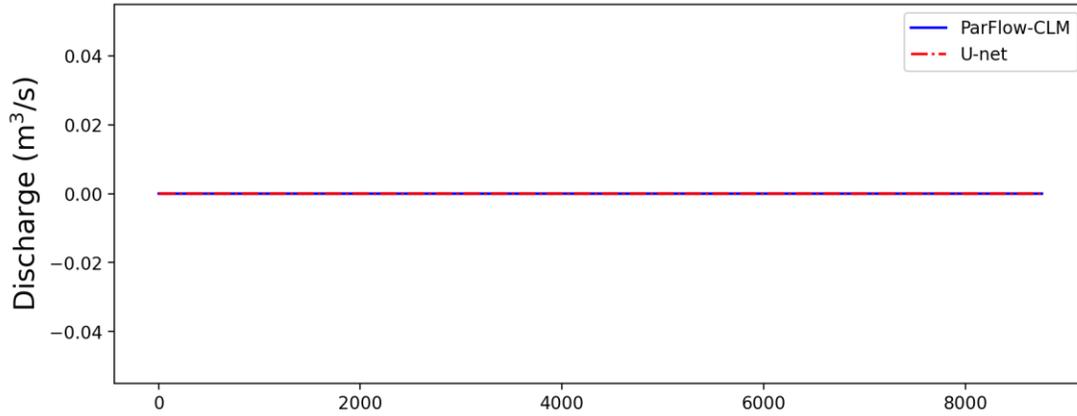


Figure 3-13: Simulations of the ParFlow-CLM and the U-net model for validation on the grid (9, 37).

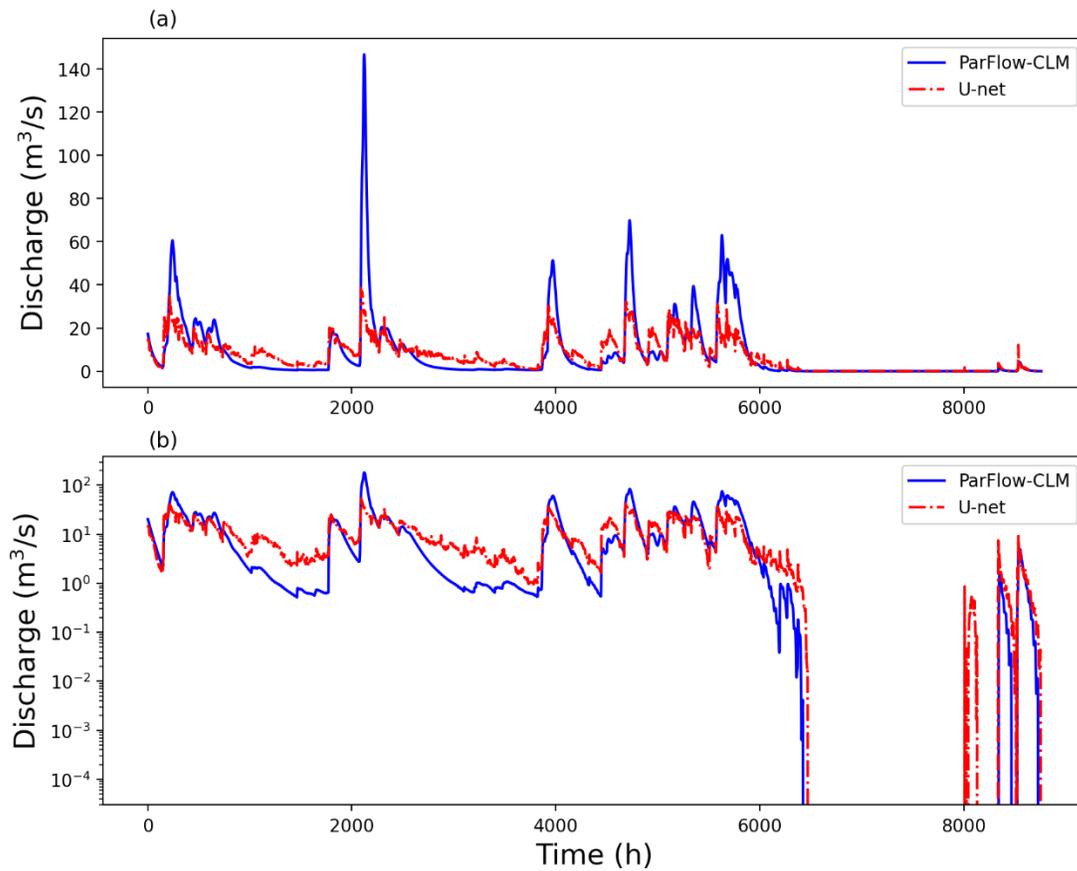


Figure 3-14: Simulations of the ParFlow-CLM and the U-net model for validation on the grid (26, 39), (a) original plot and (b) semi-log plot.

The simulated results of grid (26, 39) that is close to the Ninnekah is shown in Figure 3-14. In general, the U-net model can learn the pattern of ParFlow-CLM model, and achieves the  $r$  of 0.800 and RMSE of 10.2 m<sup>3</sup>/s. However, it has limitations in learning the peak and decays more slowly than the ParFlow-CLM model. This is because the rainfall event is relatively rare within the domain throughout the time and the streamflows are non zeros when there are no rainfall events, so the model underestimates the influence of the rainfall on streamflow and relies more on other input variables. What's more, the simulations from ParFlow-CLM in the time of 6300-8000 h are zeros and the results cannot be transferred into logarithmic scale, so the values vanish in the Figure 3-14 (b) in this period.

## Chapter 4

### 4 Summary and Conclusions

#### 4.1 Conclusions

Compared to the physically-based models, deep learning models can use less computation resources to generate the results without knowing the detailed understanding of the physical characteristics. However, Current rainfall-runoff models heavily rely on prior streamflow information, while we only have limited hydrologic observations in the real world application. Therefore, it is necessary to explore using limited hydrologic information to bring accurate, fast and reliable streamflow simulation and forecasting.

In this study, three deep learning methods ANN, RNN and LSTM models were evaluated for the streamflow simulation and forecasting at multiple lead time and different sites. One particular structure of CNN model U-net was used to learn the physical patterns of the physically-based ParFlow-CLM model to generate pixel-to-pixel streamflow mapping. The main findings and contributions can be followed as follows:

- (1) All three models can bring reliable and accurate streamflow simulation and forecasting at the lead time of 1-5 h, but the performance becomes worse when the lead time is larger than 5 h.
- (2) All three models show promising streamflow forecasting at the Ninnekah and Cement site, but have limited application at the Cyril site. This is because Cyril's site has more imbalanced data that makes deep learning models hard to identify its internal pattern.

(3) Compared to conventional thoughts that more complex models are supposed to generate much better results, the ANN model can achieve competent results as the LSTM model does and outperform the RNN model.

(4) Based on the prediction of the peak and low flow on different sites, the LSTM model shows more consistent and reliable streamflow simulation results and can work better in different basin scales.

(5) The U-net model can learn the patterns of ParFlow-CLM model well and achieves average  $r$  of 0.770 and 0.820 in the whole domain for validation and calibration, respectively. However, it has limitations in simulating the peak.

#### 4.2 Deficiency and future work

As mentioned above, the imbalanced data limits the performance of the deep learning models, and the rare events like peak less update the models compared with other parts of data in the training process, which makes the models hard to capture this situation. In addition, since the streamflow data is continuous, traditional data augmentation methods are also not proper here. Better normalization ways demand to be found to change the distribution of the data to ease the training process, and the model structures need to be adjusted to gradually learn the pattern of the hydrological processes for both lumped deep learning models and U-net model. For the U-net model, since it is trained by the synthetic data simulated by the ParFlow-CLM model, how to add ground truth to calibrate the model and apply it into real world application remains to be explored.

## REFERENCE

- Ahmed, A. A., Deo, R. C., Raj, N., Ghahramani, A., Feng, Q., Yin, Z., & Yang, L. (2021). Deep Learning Forecasts of Soil Moisture: Convolutional Neural Network and Gated Recurrent Unit Models Coupled with Satellite-Derived MODIS, Observations and Synoptic-Scale Climate Index Data. *Remote Sensing*, 13(4), 554.
- Anderberg, M. R. (1973). The broad view of cluster analysis. *Cluster analysis for applications*, 1-9.
- Anmala, J., Zhang, B., & Govindaraju, R. S. (2000). Comparison of ANNs and empirical approaches for predicting watershed runoff. *Journal of water resources planning and management*, 126(3), 156-166.
- ARS. (2021). Available online at : <https://ars.mesonet.org/>
- Ashby, S. F., & Falgout, R. D. (1996). A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear science and engineering*, 124(1), 145-159.
- Bengio, Y., Goodfellow, I., & Courville, A. (2017). *Deep learning* (Vol. 1). Massachusetts, USA:: MIT press.
- Clark, M. P., Bierkens, M. F., Samaniego, L., Woods, R. A., Uijlenhoet, R., Bennett, K. E., ... & Peters-Lidard, C. D. (2017). The evolution of process-based hydrologic models: historical challenges and the collective quest for physical realism. *Hydrology and Earth System Sciences*, 21(7), 3427-3440.
- Dai, Y., Zeng, X., Dickinson, R. E., Baker, I., Bonan, G. B., Bosilovich, M. G., ... & Yang, Z. L. (2003). The common land model. *Bulletin of the American Meteorological Society*, 84(8), 1013-1024.
- Ek, M. B., Mitchell, K. E., Lin, Y., Rogers, E., Grunmann, P., Koren, V., ... & Tarpley, J. D. (2003). Implementation of Noah land surface model advances in the National Centers for Environmental Prediction operational mesoscale Eta model. *Journal of Geophysical Research: Atmospheres*, 108(D22).
- Fang, K., Shen, C., Kifer, D., & Yang, X. (2017). Prolongation of SMAP to spatiotemporally seamless coverage of continental US using a deep learning neural network. *Geophysical Research Letters*, 44(21), 11-030.
- Gao, S., Huang, Y., Zhang, S., Han, J., Wang, G., Zhang, M., & Lin, Q. (2020). Short-term runoff prediction with GRU and LSTM networks without requiring time step optimization during sample generation. *Journal of Hydrology*, 589, 125188.
- Gassman, P. W., Reyes, M. R., Green, C. H., & Arnold, J. G. (2007). The soil and water assessment tool: historical development, applications, and future research directions. *Transactions of the ASABE*, 50(4), 1211-1250.
- Giles, C. L., Kuhn, G. M., & Williams, R. J. (1994). Dynamic recurrent neural networks:

Theory and applications. *IEEE Transactions on Neural Networks*, 5(2), 153-156.

- Haiyan, C., Yaning, C., Weihong, L., Xinming, H., Yupeng, L., & Qifei, Z. (2018). Identifying evaporation fractionation and streamflow components based on stable isotopes in the Kaidu River Basin with mountain–oasis system in north-west China. *Hydrological Processes*, 32(15), 2423-2434.
- Halff, A. H., Halff, H. M., & Azmoodeh, M. (1993, July). Predicting runoff from rainfall using neural networks. In *Engineering hydrology* (pp. 760-765). ASCE.
- Harris, C. R., Millman, K. J., van der Walt, S. J., Gommers, R., Virtanen, P., Cournapeau, D., ... & Oliphant, T. E. (2020). Array programming with NumPy. *Nature*, 585(7825), 357-362.
- He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).
- Hinton, G., Deng, L., Yu, D., Dahl, G. E., Mohamed, A. R., Jaitly, N., ... & Kingsbury, B. (2012). Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal processing magazine*, 29(6), 82-97.
- Hochreiter, S. (1991). Untersuchungen zu dynamischen neuronalen Netzen. Diploma, Technische Universität München, 91(1).
- Hochreiter, S., & Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8), 1735-1780.
- Hsu, K. L., Gupta, H. V., & Sorooshian, S. (1995). Artificial neural network modeling of the rainfall-runoff process. *Water resources research*, 31(10), 2517-2530.
- Hu, C., Wu, Q., Li, H., Jian, S., Li, N., & Lou, Z. (2018). Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water*, 10(11), 1543.
- Huang, P. C., Hsu, K. L., & Lee, K. T. (2021). Improvement of Two-Dimensional Flow-Depth Prediction Based on Neural Network Models By Preprocessing Hydrological and Geomorphological Data. *Water Resources Management*, 35(3), 1079-1100.
- Hunter, J. D. (2007). Matplotlib: A 2D graphics environment. *IEEE Annals of the History of Computing*, 9(03), 90-95.
- Kingma, D. P., & Ba, J. (2014). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.
- Kohonen, T. (1989). Self-organizing feature maps. In *Self-organization and associative memory* (pp. 119-157). Springer, Berlin, Heidelberg.
- Kollet, S. J., & Maxwell, R. M. (2006). Integrated surface–groundwater flow modeling: A free-surface overland flow boundary condition in a parallel groundwater flow model. *Advances in Water Resources*, 29(7), 945-958.
- Kratzert, F., Klotz, D., Brenner, C., Schulz, K., & Herrnegger, M. (2018). Rainfall–runoff modelling using long short-term memory (LSTM) networks. *Hydrology and Earth*

System Sciences, 22(11), 6005-6022.

- Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25, 1097-1105.
- Kollet, S. J., & Maxwell, R. M. (2008). Capturing the influence of groundwater dynamics on land surface processes using an integrated, distributed watershed model. *Water Resources Research*, 44(2).
- Kratzert, F., Klotz, D., Herrnegger, M., Sampson, A. K., Hochreiter, S., & Nearing, G. S. (2019). Toward improved predictions in ungauged basins: Exploiting the power of machine learning. *Water Resources Research*, 55(12), 11344-11354.
- Kuffour, B. N., Engdahl, N. B., Woodward, C. S., Condon, L. E., Kollet, S., & Maxwell, R. M. (2020). Simulating coupled surface–subsurface flows with ParFlow v3. 5.0: capabilities, applications, and ongoing development of an open-source, massively parallel, integrated hydrologic model. *Geoscientific Model Development*, 13(3), 1373-1397.
- Kumar, D. N., Raju, K. S., & Sathish, T. (2004). River flow forecasting using recurrent neural networks. *Water resources management*, 18(2), 143-161.
- LeCun, Y., Bengio, Y., & Hinton, G. (2015). Deep learning. *nature*, 521(7553), 436-444.
- Li, J., Hsu, K. L., & Jiang, A. L. (2020, December). Applying Deep Learning Models for Catchment Scale Streamflow Prediction. In *AGU Fall Meeting 2020*. AGU.
- lin Hsu, K., Gupta, H. V., & Sorooshian, S. (1997). Application of a recurrent neural network to rainfall-runoff modeling. In *Proceedings of the 1997 24th Annual Water Resources Planning and Management Conference* (pp. 68-73). ASCE.
- Liu, M., Huang, Y., Li, Z., Tong, B., Liu, Z., Sun, M., ... & Zhang, H. (2020). The applicability of LSTM-KNN model for real-time flood forecasting in different climate zones in China. *Water*, 12(2), 440.
- Liu, W., Park, S., Bailey, R. T., Molina-Navarro, E., Andersen, H. E., Thodsen, H., ... & Trolle, D. (2020). Quantifying the streamflow response to groundwater abstractions for irrigation or drinking water at catchment scale using SWAT and SWAT–MODFLOW. *Environmental Sciences Europe*, 32(1), 1-25.
- Long, J., Shelhamer, E., & Darrell, T. (2015). Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 3431-3440).
- Maier, H. R., & Dandy, G. C. (2000). Neural networks for the prediction and forecasting of water resources variables: a review of modelling issues and applications. *Environmental modelling & software*, 15(1), 101-124.
- Maxwell, R. M., & Miller, N. L. (2005). Development of a coupled land surface and groundwater model. *Journal of Hydrometeorology*, 6(3), 233-247.

- McCulloch, W. S., & Pitts, W. (1943). A logical calculus of the ideas immanent in nervous activity. *The bulletin of mathematical biophysics*, 5(4), 115-133.
- McKinney, W. (2011). pandas: a foundational Python library for data analysis and statistics. *Python for High Performance and Scientific Computing*, 14(9), 1-9.
- Moradkhani, H., Hsu, K. L., Gupta, H. V., & Sorooshian, S. (2004). Improved streamflow forecasting using self-organizing radial basis function artificial neural networks. *Journal of Hydrology*, 295(1-4), 246-262.
- Noori, N., & Kalin, L. (2016). Coupling SWAT and ANN models for enhanced daily streamflow prediction. *Journal of Hydrology*, 533, 141-151.
- Pan, B., Hsu, K., AghaKouchak, A., & Sorooshian, S. (2019). Improving precipitation estimation using convolutional neural network. *Water Resources Research*, 55(3), 2301-2321.
- Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., ... & Chintala, S. (2019). Pytorch: An imperative style, high-performance deep learning library. arXiv preprint arXiv:1912.01703.
- Reichstein, M., Camps-Valls, G., Stevens, B., Jung, M., Denzler, J., & Carvalhais, N. (2019). Deep learning and process understanding for data-driven Earth system science. *Nature*, 566(7743), 195-204.
- Ronneberger, O., Fischer, P., & Brox, T. (2015, October). U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention* (pp. 234-241). Springer, Cham.
- Rosero, E., Gulden, L. E., Yang, Z. L., De Goncalves, L. G., Niu, G. Y., & Kaheil, Y. H. (2011). Ensemble evaluation of hydrologically enhanced Noah-LSM: Partitioning of the water balance in high-resolution simulations over the Little Washita River experimental watershed. *Journal of Hydrometeorology*, 12(1), 45-64.
- Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088), 533-536.
- Shamseldin, A. Y. (1997). Application of a neural network technique to rainfall-runoff modelling. *Journal of hydrology*, 199(3-4), 272-294.
- Shamseldin, A. Y., Nasr, A. E., & O'connor, K. M. (2002). Comparison of different forms of the Multi-layer Feed-Forward Neural Network method used for river flow forecasting. *hydrology and earth system sciences*, 6(4), 671-684.
- Shen, C. (2018). A transdisciplinary review of deep learning research and its relevance for water resources scientists. *Water Resources Research*, 54(11), 8558-8593.
- Starks, P. J., Fiebrich, C. A., Grimsley, D. L., Garbrecht, J. D., Steiner, J. L., Guzman, J. A., & Moriasi, D. N. (2014). Upper Washita River experimental watersheds: Meteorologic and soil climate measurement networks. *Journal of environmental quality*, 43(4), 1239-1249.

- Tao, Y. (2017). A Deep Learning Framework for Precipitation Estimation from Bispectral Satellite Information (Doctoral dissertation, UC Irvine).
- Tongal, H., & Booij, M. J. (2018). Simulation and forecasting of streamflows using machine learning models coupled with base flow separation. *Journal of hydrology*, 564, 266-282.
- Zhang, J., Zhu, Y., Zhang, X., Ye, M., & Yang, J. (2018). Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *Journal of hydrology*, 561, 918-929.
- Zhu, M. L., Fujita, M., & Hashimoto, N. (1994). Application of neural networks to runoff prediction. In *Stochastic and statistical methods in hydrology and environmental engineering* (pp. 205-216). Springer, Dordrecht.
- Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., ... & He, Q. (2020). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1), 43-76.