Transfer Learning Approaches for Knowledge Discovery in Grid-based Geo-Spatiotemporal Data

Anonymous Author(s)
Affiliation
Address
email

Abstract

Extracting and meticulously analyzing geo-spatiotemporal features is crucial to recognize intricate underlying causes of natural events, such as floods. Limited evidence about hidden factors leading to climate change makes it challenging to predict regional water discharge accurately. In addition, the explosive growth in complex geo-spatiotemporal environment data that requires repeated learning by the state-of-the-art neural networks for every new region emphasizes the need for new computationally efficient methods, advanced computational resources, and extensive training on a massive amount of available monitored data. We, therefore, propose HydroDeep, an effectively reusable pretrained model to address this problem of transferring knowledge from one region to another by effectively capturing their intrinsic geo-spatiotemporal variance. Further, we present four transfer learning approaches on HydroDeep for spatiotemporal interpretability that improve Nash–Sutcliffe efficiency by 9% to 108% in new regions with a 95% reduction in time.

1 Introduction

Spatiotemporal variance influencing local water discharge is inevitable due to varying soil property, climate, and land usage among different regions. Training individual hydrological models to capture regional spatiotemporal features requires extensive training and computational resources. These regions, although varying in local characteristics, still follow fundamental hydrological dependencies. Transfer learning is a well-known solution to reuse a trained model to reduce training duration for a new dataset. Although most commonly applied in computer vision [1, 2, 3] and time-series prediction [4, 5, 6], recent works [7, 8] in other application domains show that these techniques have also been effective in knowledge guided neural networks that are powered by process-based (PB) mechanisms. In hydrology, PB models that rely on domain-specific principles and mathematical formulations, although often criticized as overly complex are elegant in describing large-scale patterns [9, 10] where knowledge of distributed state variables and physical constraints is essential. However, understanding a system's general organization does not provide insights into how the principal variables interact over space and time. [11] discuss all the prevalent challenges of distributed hydrological PB models. On the other hand, several works show an increase in deep learning (DL) applications in hydrology. A recent survey of 129 publications finds the use of various state-of-the-art DL network architectures, namely, convolutional neural networks (CNNs), long short-term memory networks (LSTMs), and gated recurrent units (GRUs) [12]. However, limited knowledge of underlying PB mechanisms in DL algorithms makes PB models irreplaceable in studying environmental systems. As a result, knowledge-guided deep neural networks are an active research area in hydrology [13, 14, 15] as well as other domains [16, 17, 18].

In this paper, we present HydroDeep that couples a PB hydro-ecological model with a combination of one-dimensional CNN and LSTM, to capture regional grid-based geo-spatiotemporal features of

Submitted to NeurIPS 2021 AI for Science Workshop.

a watershed contributing to water discharge that influence an event of flood. The combination of CNN and LSTM that our PB-DL based network uses showed promising results in computer vision [19], speech recognition and natural language processing [20], and other time-series analysis [21]. In our experiments, HydroDeep outperformed the Nash–Sutcliffe efficiency [22] of standalone CNNs and LSTMs by 1.6% and 10.5% respectively. We further propose a new application area of transfer learning to analyze similarities and dissimilarities in geo-spatiotemporal characteristics of watersheds while reducing the extensive training time required in training local DL models. Recent research in hydrology shows transfer learning has been applied in predicting water temperature in unmonitored lakes [23] and water quality prediction system [24]. In [25] transfer learning was applied for flood prediction, similar to us. However, they converted their time-series data to images to use transfer learning in a purely CNN architecture for time-series prediction with a reduction in computational cost by 18%. In this paper, we explore how transferrable geo-spatiotemporal features are when kept in their original spatiotemporal form.

2 Approach

2.1 HydroDeep

If $g_i \in \{g_1, g_2, \ldots, g_L\}$ is the spatial grid vector where L is the total number of grids covering a region having spatial coordinates $c_{g_i} \in \{(x_{g_1}, y_{g_1}), (x_{g_2}, y_{g_2}), \ldots, (x_{g_L}, y_{g_L})\}$, the distance to these grids from the nearest river or water source is $d_{g_i} \in \{d_{g_1}, d_{g_2}, \ldots, d_{g_L}\}$. On a certain day t, the grids have precipitation measurements p_{g_i} where $p_{g_i} \in \{p_{g_1}, p_{g_2}, \ldots, p_{g_L}\}$ which are mapped to their corresponding grid-based PB simulated runoff, r_{g_i} where $r_{g_i} \in \{r_{g_1}, r_{g_1}, \ldots, r_{g_L}\}$. The extent to which each grid's precipitation contributes to the river discharge depends extensively on the grid's distance to the nearest water source. The distance vector is thus transformed to a distance weight vector $\widetilde{d}_{g_i}$ such that higher distance weights are applied to p_{g_i} if the said grid is closer to a local river thus contributing more to the regional river discharge (Appendix Section 1). We denote weighted precipitation vector as $\widetilde{p}_{g_i} = \widetilde{d}_i \odot p_{g_i}$. On day t, the input vectors $\widetilde{p}_{g_i,t}$ and $r_{g_i,t}$ are mapped with respective daily river discharge observations $D_t \in \{D_1, D_2, \ldots, D_T\}$ where T is the total number of daily river discharge observations. Input vector x_t can be shown as $x_t = f[(\widetilde{p}_{g_i,t}), (r_{g_i,t})]$ where $i \in \{1, 2, \ldots, L\}$ and $t \in \{1, 2, \ldots, T\}$. From a multivariate time-series point of view, we denote our inputs as $X = (x_1, x_2, \ldots, x_T) \in \mathbb{R}^{n \times T}$ where $x_t \in \mathbb{R}^n$ and n denotes the total number of input variables per day which in our case is $2L$. We want to predict the corresponding target outputs D_t. The aim is to obtain a non-linear mapping between X and D. The motive behind the integration of CNN and LSTM in HydroDeep lies in capturing both the spatial and temporal dependencies of a watershed. The CNN layers help extract local geospatial features between the input variables and pass them to LSTMs to support temporal sequence prediction. HydroDeep has an initial input layer customizable to different input shapes (number of grids) to make our model easily transferable to other watersheds. If t_p denotes the prediction day, HydroDeep trains on inputs from $(t_p - 7)$ to $(t_p - 1)$, a weekly time window validated by empirical studies, to predict the output of day t_p. A continuous drought for days can be followed by a hurricane, leading to a flood overnight, making it crucial to include the target day t_p's inputs as the second input to our network, which is concatenated with the last LSTM layer's output, and are processed collectively by a fully connected layer. We use data from Jan 1, 2000, to Dec 31, 2011, for training and Jan 1, 2012, to Dec 31, 2016, for evaluation. We optimized the hyperparameters of HydroDeep by random search. (Appendix Section 2)

2.2 Transfer Learning

An event of flood or drought is heavily dependent on environmental drivers. Likewise, each of these drivers' inherent local spatiotemporal patterns is bound to be unique based on their geographical location. Consequently, one model that has learned the local spatiotemporal patterns of a region will fail to perform accurately for a geographically distant region with different characteristics. As a result, the model should be retrained perpetually for every new unique region. Alternatively, a global model can be trained on a larger area covering many watersheds to address this problem, but it will fail to capture the local patterns. Besides, both these methods are expensive as they require more training time and computational power to train such a model of global extent. Therefore, we use transfer learning to reuse HydroDeep's knowledge from one region to another. More formally, transfer learning consists of a domain $\widetilde{D}$ and a task $\widetilde{T}$ where the domain $\widetilde{D}$ is the marginal probability distribution $P(X)$ over an input feature space $X = \{x_1, x_2, \ldots, x_N\}$ where N is the total number of input features. Given a domain $\widetilde{D} = \{X, P(X)\}$, a task $\widetilde{T}$ consists of a conditional probability distribution $P(Y|X)$ over a label space Y. The conditional probability distribution is usually learned

from the pairs $\{x_i, y_i\}$ in the training samples where $x_i \in X$ and $y_i \in Y$. Suppose there is a source domain $\widetilde{D}_{source}$ with a source task $\widetilde{T}_{source}$ and a target domain $\widetilde{D}_{target}$ with a target task $\widetilde{T}_{target}$, through transfer learning we try to learn the target conditional probability distribution $P(Y_{target}|X_{target})$ in $\widetilde{D}_{target}$, from the knowledge learned from $\widetilde{D}_{source}$ and $\widetilde{T}_{target}$ [26].

3 Experimental Design

3.1 Dataset and the Process-based Model - DLEM

For experiments, we use a PB hydro-ecological model named the Dynamic Land Ecosystem Model (DLEM) that mimics the plant physiological, biogeochemical, and hydrological processes in the plant-soil-water-river continuum [27, 28]. In DLEM, the design of grid-to-grid connection tracks significant features of a region, including within-grid heterogeneity, grid-to-grid flow, and land-aquatic linkage. DLEM models the water movement from land to aquatic systems at a daily time step with each grid cell comprising multiple land cover types, rivers, and lakes with their area percentage prescribed by land-use history data [29]. Research shows that DLEM has been extensively validated against measurements from LTER, NEON, Ameri Flux, USDA crop yield survey, and USGS gauge monitoring and are widely used to quantify the spatiotemporal variations in the pool and fluxes of water, carbon, and nitrogen coupling (water-C-N) at the site, and regional scales [27, 28, 30, 31, 32]. The preliminary results from DLEM at the outlet of the Mississippi and Atchafalaya river basin (MARB) show that the variations of daily river discharge are very close to the USGS observed river discharge over the years [29].

We pulled dataset from the U.S. Geological Survey's (USGS) daily discharge measurements for Iowa Streams having daily discharge measurements of 23 watersheds covering the state of Iowa [33]. Alongside, we use daily precipitation at a 5-arc-min resolution generated from high-resolution gridded meteorological data products from station observations by the Climatic Research Unit (CRU) of the University of East Anglia [34], and North America Regional Reanalysis (NARR) dataset from a combination of modeled and observed data [35]. We also use DLEM-simulated surface and subsurface runoff to guide our DL network [29]. Climate, land management, and environmental drivers steer DLEM simulation and are used to represent our "best estimate" of land-to-aquatic surface and subsurface runoff across the watershed. For our experiment, we chose the Thompson Fork Grand River basin at David City (watershed 13) as our source domain due to its smaller size and transferred its knowledge to 5 target domains - West Nodaway River near Shambaugh (watershed 14), East Nishnabotna River near Shenandoah (watershed 15), Turkey River near Garber (watershed 4), South Skunk River near Oskaloosa (watershed 10), and Rock River near Hawarden (watershed 23) [33]. Watershed 13 (w13) has 29 grids with each grid having two input time-series signals corresponding to precipitation and DLEM runoff. Daily river discharge observations are given as labels to all 29 grids.

3.2 Regional Knowledge Transfer for Spatiotemporal Analysis

We have experimented with four transfer learning approaches to reuse HydroDeep's knowledge in predicting river discharge in distant watersheds (Figure 1). The goal was to find the best approach to transfer HydroDeep's grid-based spatiotemporal knowledge from one watershed to another to reduce the required training iterations on a new region and interpret the geo-spatiotemporal similarities and dissimilarities between the source and the target. In the first approach, we preserve the original HydroDeep's spatiotemporal knowledge from the source w13 and use it solely in T-HydroDeep-1 (T-HD-1) to test the new target watersheds. Secondly, we transfer the original HydroDeep's spatiotemporal knowledge in T-HydroDeep-2 (T-HD-2) and allow it to finetune on the target. In the third and fourth approaches, we take turns in finetuning just the temporal features in T-HydroDeep-3 (T-HD-3) and the spatial features in T-HydroDeep-4 (T-HD-4). The CNN layers and the LSTM layers are responsible for learning spatial and temporal features, respectively. When we finetune one kind of layer, we freeze the other to keep the originally learned features intact. Table 1 shows the observations from our ablation study.

4 Results

The watersheds w14 and w15, although being adjacent to the source w13 [33], are observed to have distinct spatial features as T-HD-4 proved to be the best approach for both the watersheds. We also observed that the second-best approach for w14 is T-HD-1, which means w13 and w14 have similar temporal features, but since they vary in spatial features, only the spatial layers needed to

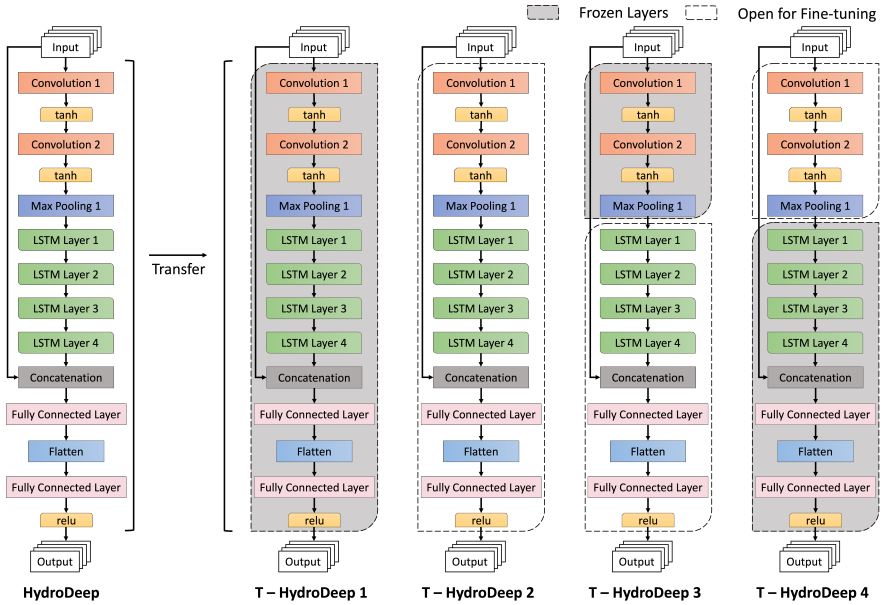


Figure 1: Four transfer learning approaches in reusing HydroDeep's layers for a new target.

be finetuned (T-HD-4) to improve prediction accuracy. Similarly, the result shows that w15 has more distinct spatial features than temporal features from w13, as T-HD-2 shows the second-best performance, just behind T-HD-4. However, w4 and w10, both being far from w13, show distinct spatiotemporal features as T-HD-2 shows the best performance. Note that w13 has only 29 spatial grids, whereas both w4 and w10 have 61 and 65 grids, respectively. HydroDeep's knowledge is transferred to targets almost double the area of the source and still achieved the best performances among all the watersheds included in our experiment. This supports our arguments that more data availability will increase HydroDeep's performance; transfer learning approaches work in targets larger in area than the source and can be used to analyze watersheds' spatiotemporal characteristics. The original HydroDeep was pretrained for 300 iterations on w13 in ~ 3138 seconds to achieve 0.63 NSE outperforming the baseline neural network architectures - CNN by 1.6%, LSTM by 10.5%, GRU by 26%, and Bi-directional LSTM by 53.6% that are commonly used for hydrological modeling (Appendix Section 3). In contrast, the transferred models on the new targets (w14, w15, w4, w10, and w23) achieved significant performance in just 20 training iterations in ~ 155 seconds, a $\sim 95\%$ reduction in time. In Table 1, training HydroDeep (HD) from scratch in each target watershed remains low when run for 20 iterations. Thereby, using transfer learning, the prediction performance on the individual watersheds increased by 44% in w14, 108% in w15, 15% in w4, 9% in w10, and 39% in w23 in terms of NSE.

Table 1: Performance comparison of untrained HydroDeep and four transfer learning approaches where HydroDeep was pretrained on w13 with 29 grids before being transferred to other watersheds of varying distance and size. (Appendix Section 4)

Target Watersheds	**No. of grids**	**HD** (NSE)	**T-HD-1** (NSE)	**T-HD-2** (NSE)	**T-HD-3** (NSE)	**T-HD-4** (NSE)	**Time** (seconds)
Watershed 14 (w14)	34	0.27	0.33	0.30	0.32	**0.39**	154.35 ± 20.81
Watershed 15 (w15)	39	0.24	0.46	0.47	0.42	**0.50**	154.87 ± 20.38
Watershed 4 (w4)	61	0.71	0.76	**0.82**	0.81	0.82	156.52 ± 20.44
Watershed 10 (w10)	65	0.80	0.82	**0.87**	0.86	0.86	157.15 ± 21.40
Watershed 23 (w23)	32	0.36	0.45	0.38	**0.50**	0.46	154.21 ± 20.84

5 Conclusion and Future Work

This paper illustrates a new application of transfer learning techniques in interpreting geo-spatiotemporal characteristics of watersheds with limited computational resources and a 95% reduction in time. We believe that a smaller grid-scale resolution will help HydroDeep to capture local features on a finer scale. In the future, we will run our experiments on more watersheds to better quantify the performance of HydroDeep and its variants. Further, we plan to perform extensive research on how to select the source watershed(s) more effectively.

References

[1] Chuanqi Tan, Fuchun Sun, Tao Kong, Wenchang Zhang, Chao Yang, and Chunfang Liu. A survey on deep transfer learning. In *International conference on artificial neural networks*, pages 270–279. Springer, 2018.

[2] Yusuf Celik, Muhammed Talo, Ozal Yildirim, Murat Karabatak, and U Rajendra Acharya. Automated invasive ductal carcinoma detection based using deep transfer learning with whole-slide images. *Pattern Recognition Letters*, 133:232–239, 2020.

[3] Rafael Pires de Lima and Kurt Marfurt. Convolutional neural network for remote-sensing scene classification: Transfer learning analysis. *Remote Sensing*, 12(1):86, 2020.

[4] Jun Ma, Jack CP Cheng, Changqing Lin, Yi Tan, and Jingcheng Zhang. Improving air quality prediction accuracy at larger temporal resolutions using deep learning and transfer learning techniques. *Atmospheric Environment*, 214:116885, 2019.

[5] Priyanka Gupta, Pankaj Malhotra, Jyoti Narwariya, Lovekesh Vig, and Gautam Shroff. Transfer learning for clinical time series analysis using deep neural networks. *Journal of Healthcare Informatics Research*, 4(2):112–137, 2020.

[6] Vasileios Lampos, Maimuna S Majumder, Elad Yom-Tov, Michael Edelstein, Simon Moura, Yohhei Hamada, Molebogeng X Rangaka, Rachel A McKendry, and Ingemar J Cox. Tracking covid-19 using online search. *NPJ digital medicine*, 4(1):1–11, 2021.

[7] Somdatta Goswami, Cosmin Anitescu, Souvik Chakraborty, and Timon Rabczuk. Transfer learning enhanced physics informed neural network for phase-field modeling of fracture. *Theoretical and Applied Fracture Mechanics*, 106:102447, 2020.

[8] Souvik Chakraborty. Transfer learning based multi-fidelity physics informed deep neural network. *Journal of Computational Physics*, 426:109942, 2021.

[9] A Kleidon, Stan Schymanski, and M Stieglitz. Thermodynamics, irreversibility, and optimality in land surface hydrology. In *Bioclimatology and natural hazards*, pages 107–118. Springer, 2009.

[10] J Wang and Rafael L Bras. A model of surface heat fluxes based on the theory of maximum entropy production. *Water resources research*, 45(11), 2009.

[11] Simone Fatichi, Enrique R Vivoni, Fred L Ogden, Valeriy Y Ivanov, Benjamin Mirus, David Gochis, Charles W Downer, Matteo Camporese, Jason H Davison, Brian Ebel, et al. An overview of current applications, challenges, and future trends in distributed process-based models in hydrology. *Journal of Hydrology*, 537:45–60, 2016.

[12] Muhammed Sit, Bekir Z Demiray, Zhongrun Xiang, Gregory J Ewing, Yusuf Sermet, and Ibrahim Demir. A comprehensive review of deep learning applications in hydrology and water resources. *Water Science and Technology*, 2020.

[13] Shuyu Yang, Dawen Yang, Jinsong Chen, and Baoxu Zhao. Real-time reservoir operation using recurrent neural networks and inflow forecast from a distributed hydrological model. *Journal of Hydrology*, 579:124229, 2019.

[14] Nanzhe Wang, Dongxiao Zhang, Haibin Chang, and Heng Li. Deep learning of subsurface flow via theory-guided neural network. *Journal of Hydrology*, 584:124700, 2020.

[15] Xiaowei Jia, Jared Willard, Anuj Karpatne, Jordan S Read, Jacob A Zwart, Michael Steinbach, and Vipin Kumar. Physics-guided machine learning for scientific discovery: An application in simulating lake temperature profiles. *ACM/IMS Transactions on Data Science*, 2(3):1–26, 2021.

[16] Anuj Karpatne, William Watkins, Jordan Read, and Vipin Kumar. Physics-guided neural networks (pgnn): An application in lake temperature modeling. *arXiv preprint arXiv:1710.11431*, 2017.

[17] Anurag Ajay, Jiajun Wu, Nima Fazeli, Maria Bauza, Leslie P Kaelbling, Joshua B Tenenbaum, and Alberto Rodriguez. Augmenting physical simulators with stochastic neural networks: Case study of planar pushing and bouncing. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3066–3073. IEEE, 2018.

[18] Filippo Masi, Ioannis Stefanou, Paolo Vannucci, and Victor Maffi-Berthier. Thermodynamics-based artificial neural networks for constitutive modeling. *Journal of the Mechanics and Physics of Solids*, 147:104277, 2021.

[19] Jeffrey Donahue, Lisa Anne Hendricks, Sergio Guadarrama, Marcus Rohrbach, Subhashini Venugopalan, Kate Saenko, and Trevor Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2625–2634, 2015.

[20] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3156–3164, 2015.

[21] Chiou-Jye Huang and Ping-Huan Kuo. A deep cnn-lstm model for particulate matter (pm2.5) forecasting in smart cities. *Sensors*, 18(7):2220, Jul 2018.

[22] Daniel N Moriasi, Jeffrey G Arnold, Michael W Van Liew, Ronald L Bingner, R Daren Harmel, and Tamie L Veith. Model evaluation guidelines for systematic quantification of accuracy in watershed simulations. *Transactions of the ASABE*, 50(3):885–900, 2007.

[23] Jared Willard, Jordan Stuart Read, Alison Appling, Samantha Oliver, Xiaowei Jia, Paul C Hanson, Hilary A Dugan, Robert Ladwig, and Vipin Kumar. Predicting water temperature dynamics of unmonitored lake systems with meta transfer learning. In *AGU Fall Meeting Abstracts*, volume 2020, pages H166–0030, 2020.

[24] Zeng Chen, Huan Xu, Peng Jiang, Shanen Yu, Guang Lin, Igor Bychkov, Alexey Hmelnov, Gennady Ruzhnikov, Ning Zhu, and Zhen Liu. A transfer learning-based lstm strategy for imputing large-scale consecutive missing data and its application in a water quality prediction system. *Journal of Hydrology*, 602:126573, 2021.

[25] Nobuaki Kimura, Ikuo Yoshinaga, Kenji Sekijima, Issaku Azechi, and Daichi Baba. Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. *Water*, 12(1):96, 2020.

[26] Nikolay Laptev, Jiafan Yu, and Ram Rajagopal. Applied timeseries transfer learning. 2018.

[27] Mingliang Liu, Hanqin Tian, Qichun Yang, Jia Yang, Xia Song, Steven E Lohrenz, and Wei-Jun Cai. Long-term trends in evapotranspiration and runoff over the drainage basins of the gulf of mexico during 1901–2008. *Water Resources Research*, 49(4):1988–2012, 2013.

[28] Chaoqun Lu and Hanqin Tian. Net greenhouse gas balance in response to nitrogen enrichment: perspectives from a coupled biogeochemical model. *Global change biology*, 19(2):571–588, 2013.

[29] C. Lu, J. Zhang, H. Tian, W. Crumpton, M. Helmers, W. Cai, C. Hopkinson, and S. Lohrenz. Increased extreme precipitation challenges nitrogen load reduction to the Gulf of Mexico. *Communications Earth & Environment*, page Accepted, 2020.

[30] Qichun Yang, Hanqin Tian, Marjorie AM Friedrichs, Charles S Hopkinson, Chaoqun Lu, and Raymond G Najjar. Increased nitrogen export from eastern north america to the atlantic ocean due to climatic and anthropogenic changes during 1901–2008. *Journal of Geophysical Research: Biogeosciences*, 120(6):1046–1068, 2015.

[31] Zhen Yu, Chaoqun Lu, Hanqin Tian, and Josep G Canadell. Largely underestimated carbon emission from land use and land cover change in the conterminous united states. *Global change biology*, 25(11):3741–3752, 2019.

[32] Chaoqun Lu, Jien Zhang, Peiyu Cao, and Jerry L Hatfield. Are we getting better in using nitrogen?: Variations in nitrogen use efficiency of two cereal crops across the united states. *Earth's Future*, 7(8):939–952, 2019.

[33] Christopher S Jones, Jacob K Nielsen, Keith E Schilling, and Larry J Weber. Iowa stream nitrate and the gulf of mexico. *PloS one*, 13(4):e0195930, 2018.

[34] Timothy D. Mitchell and Philip D. Jones. An improved method of constructing a database of monthly climate observations and associated high-resolution grids. *International Journal of Climatology*, 25(6):693–712, 2005.

[35] Fedor Mesinger, Geoff DiMego, Eugenia Kalnay, Kenneth Mitchell, Perry C. Shafran, Wesley Ebisuzaki, Dušan Jović, Jack Woollen, Eric Rogers, Ernesto H. Berbery, Michael B. Ek, Yun Fan, Robert Grumbine, Wayne Higgins, Hong Li, Ying Lin, Geoff Manikin, David Parrish, and Wei Shi. North American Regional Reanalysis. *Bulletin of the American Meteorological Society*, 87(3):343–360, 03 2006.

Transfer Learning Approaches for Knowledge Discovery in Grid-based Geo-Spatiotemporal Data

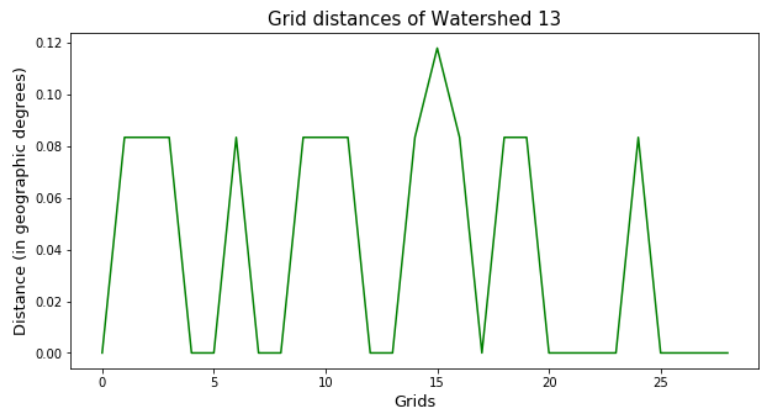
Technical Appendix

Table of Contents

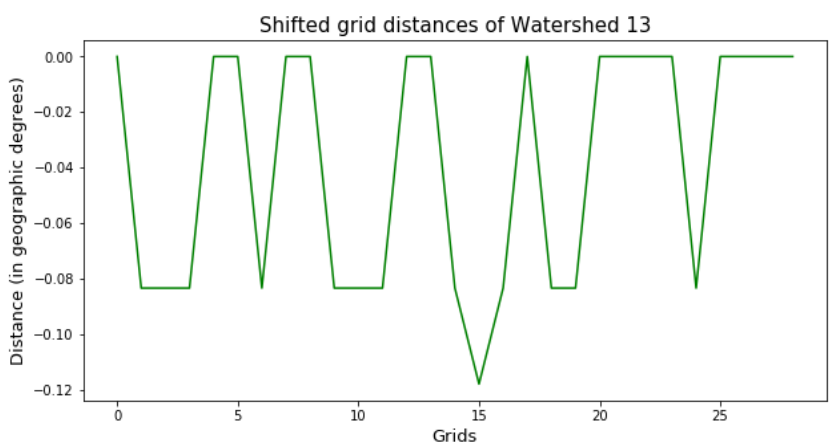
Section	Item	Page No.
1	Transformation of distance into spatial weights for precipitation **(Related to Section 2.1)**	**2-3**
2	Hyperparameter Tuning for HydroDeep - Learning Rate, Time-Window **(Related to Section 2.1)**	**4-5**
3	Evaluation of HydroDeep's Performance **(Related to Section 2.1, 4)**	**6-7**
4	Evaluation of Transfer Learning Approaches - Supplementary Figure 2. **(Related to Section 4)**	**8-10**
5	Hardware Specification	**11**

1. Transformation of distance into spatial weights for precipitation

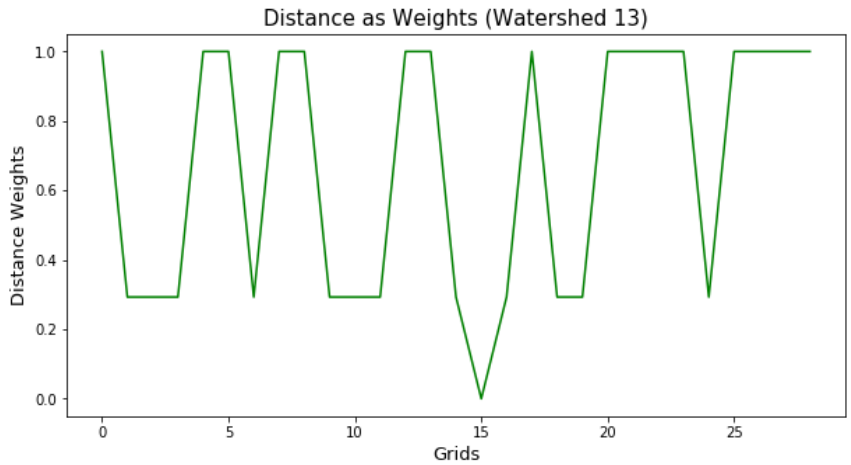
The distance between each grid and its nearest local river is calculated by the Euclidean distance tool of ArcMap 10.5. These distances add weights to the grid based precipitation measurements such as the farthest grid contributes the lowest weight to the local river and finally into the river outlet where river drainage discharge has been recorded. The raw distances (Supplementary Fig 1a.) were first transformed, so that the closest grids to the local rivers having the shortest distances is at 0, and the farthest distances lie at negative (Supplementary Fig 1b.). The transformed distances are then scaled between 0 and 1, which means the shortest distance is brought to 1 having the highest contributing weight, whereas the highest distance at 0 will have the lowest contributing weight to the concerned local river grid (Supplementary Fig 1c.).



Supplementary Figure 1a. The raw distances of each grid of watershed 13 from its nearest local river. Watershed 13 has 29 grids. The distance is measured in geographic degree. (Related to Section 2.1)



Supplementary Figure 1b. The raw distance after undergoing transformation. (Related to Section 2.1)



Supplementary Figure 1c. The weight of each grid is based on their distance from the nearest local river. The closest grids with the smallest distances lie at 1.0, whereas grid 15 that is the farthest, has zero weight. (Related to Section 2.1)

2. Hyperparameter Tuning for HydroDeep

2.1. Learning Rate

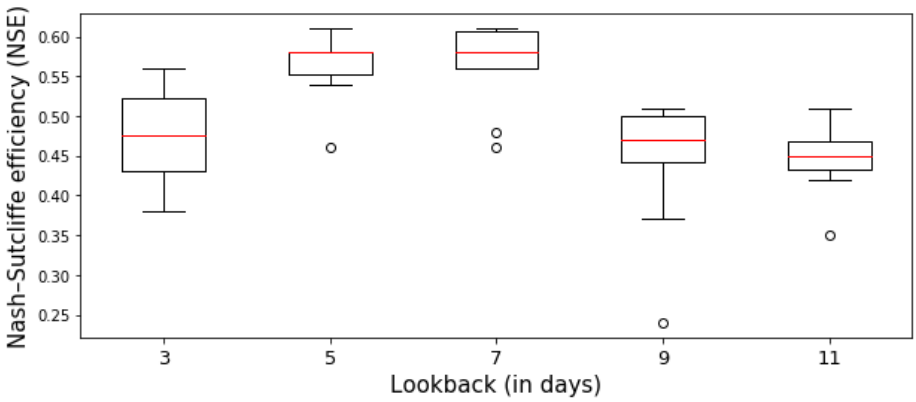
Choosing an optimal learning rate is challenging since a higher learning rate can make the network stuck in a sub-optimal set of weights, whereas a lower learning rate may result in a longer training process. We did an extensive random search in finding the optimal learning rate for HydroDeep.

Supplementary Table 1. Learning Rate tuning results of HydroDeep while training with 500 epochs set with early stop and a patience of 50 epochs. The network was run with each of the learning rate values multiple times, and the best results for each are shown here.

Model Run No.	Learning Rate	NSE on Training Dataset	NSE on Test Dataset
1	0.0001	0.78	0.57
2	0.00004	0.83	0.50
3	0.00003	0.81	0.57
4	0.000025	0.80	0.60
5	0.000021	0.80	0.51
6	0.000019	0.80	0.62
7	0.00002	0.87	0.63
8	0.0000201	0.80	0.60

2.2. Time-window

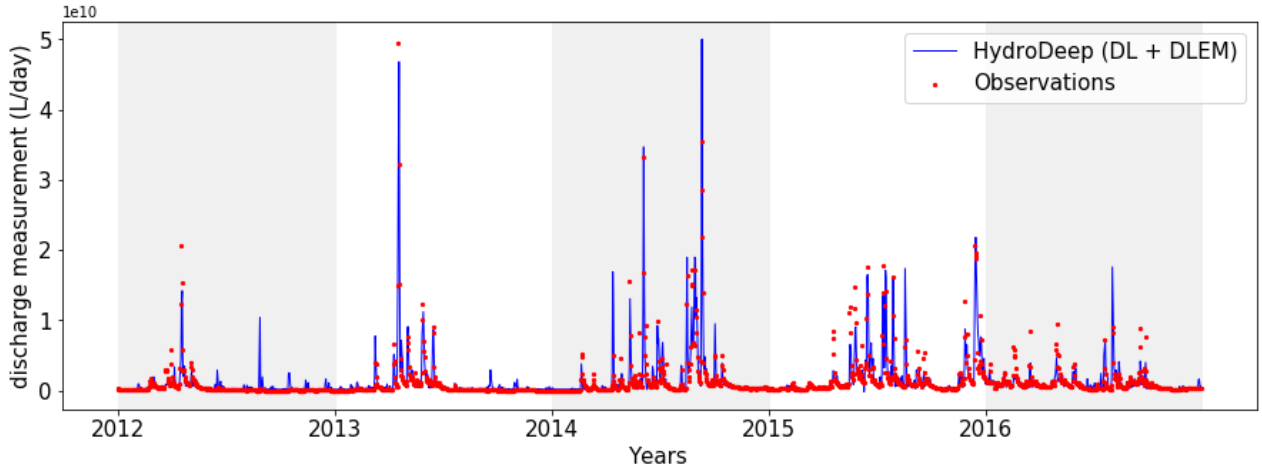
Out of all the hyperparameters, HydroDeep was most susceptible to the lookback (time-window) parameter. In the first iteration, the lookback window sets itself from the start day of the training sample-set and then slides over with a step value of 1 day per iteration for the following iterations. We experimented with lookback values varying from 3 to 11 days and observed that increasing the number of days did not consistently improve the performance. The value was set manually before each run to observe the NSE performance (Supplementary Figure 2a). From our analysis, a lookback period of 7 days in HydroDeep has achieved the best performance in NSE throughout.



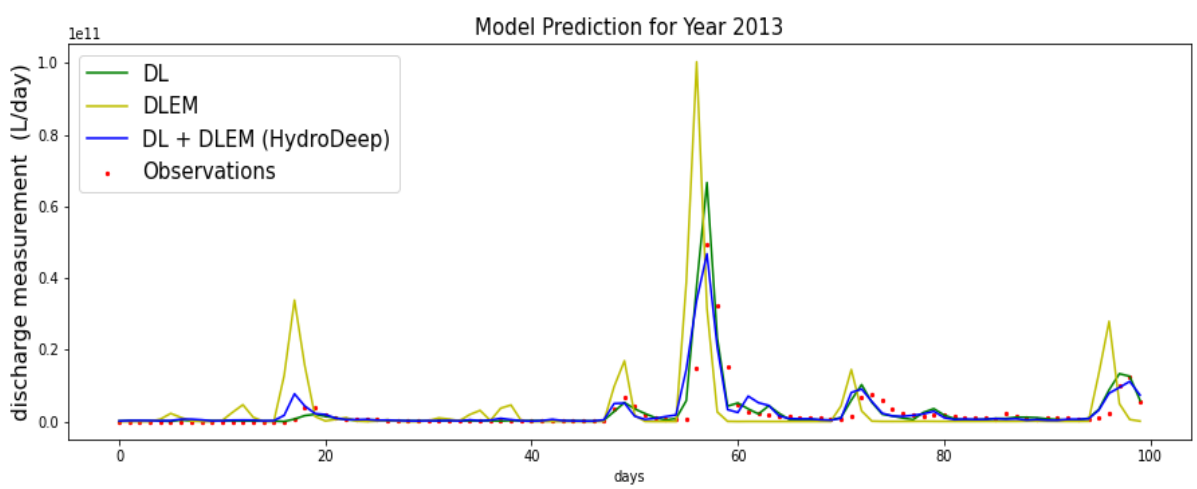
Supplementary Figure 2a. NSE of different model runs on watershed 13 with various lookback (lag) values in HydroDeep

3. Evaluation of HydroDeep's performance

After the training is complete, Hydrodeep's performance is evaluated on the test dataset (Supplementary Figure 3a). Supplementary Figure 3b shows an arbitrarily chosen window of 3 months from 2013 in w13. We compare the prediction power of the three approaches, DL which represents a similar neural network architecture as HydroDeep but without DLEM simulated runoff as input, standalone DLEM, and our proposed HydroDeep (with DLEM), against the observations to evaluate where they stand in comparison. According to our results, Hydrodeep performs better than its counterparts.



Supplementary Figure 3a. The HydroDeep predicted daily river discharge of w13 is plotted against river gauge measurements from Jan 1, 2012 to Dec 31, 2016



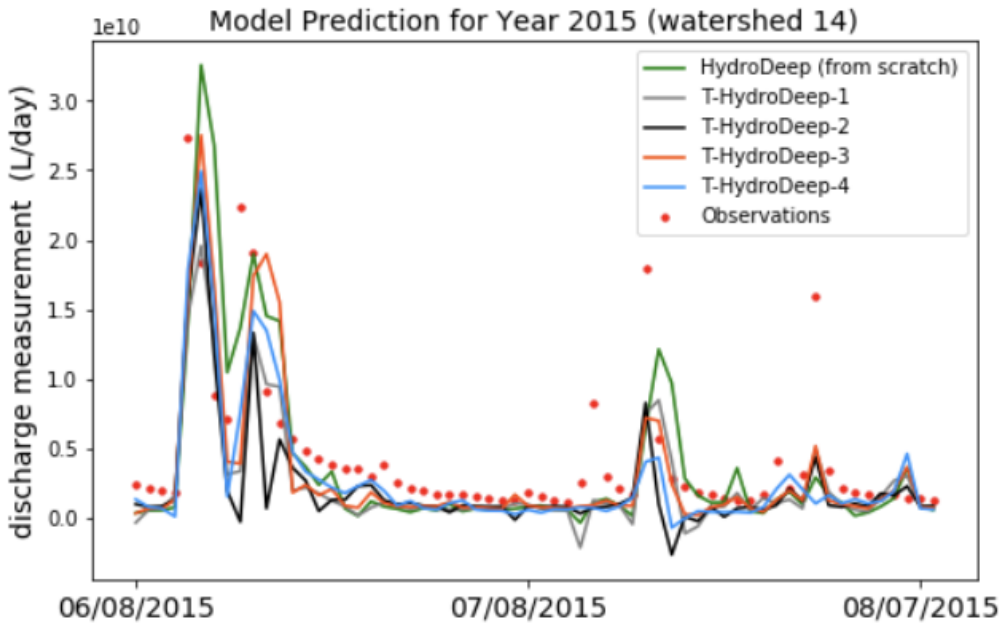
Supplementary Figure 3b. We present a closer look at a period from February 20, 2013 to May 20, 2013 in watershed 13 where HydroDeep, with the use of DLEM, outperforms the other models. DL (without DLEM) achieves an NSE of 0.49 compared to HydroDeep's 0.63.

Supplementary Table 2. Evaluation of HydroDeep's performance on watershed 13 compared to benchmark neural network architectures.

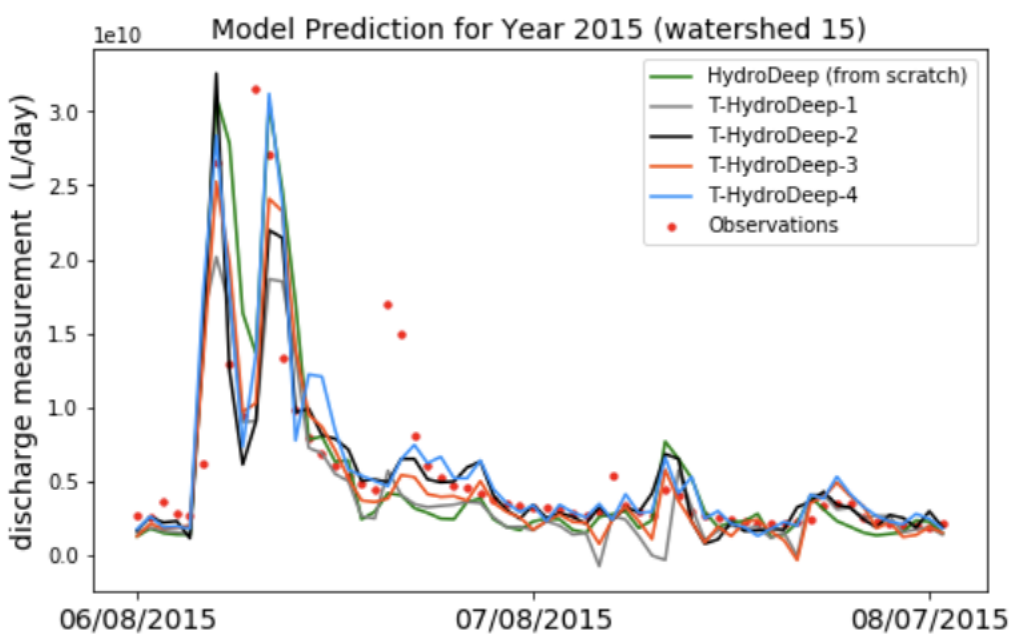
Models	Type	NSE	PBIAS	RSR
HydroDeep	DL + PB	0.63	10.54	0.61
Convolutional Neural Network (CNN)	DL + PB	0.62	-22.81	0.62
Long Short-term Memory Network (LSTM)	DL + PB	0.57	-27.21	0.65
Gated recurrent Unit (GRU)	DL + PB	0.50	-20.35	0.71
Bi-directional LSTM	DL + PB	0.41	-3.91	0.77

HydroDeep on w13 achieves an NSE of 0.63 outperforming the baseline neural network architectures - CNN by 1.6%, LSTM by 10.5%, GRU by 26%, and Bi-directional LSTM by 53.6% that are commonly used for hydrological modeling (Supplementary Table 2). We use DLEM (PB) simulated runoff and similar preprocessing of all the grid-based input vectors for a fair assessment of the benchmark architectures.

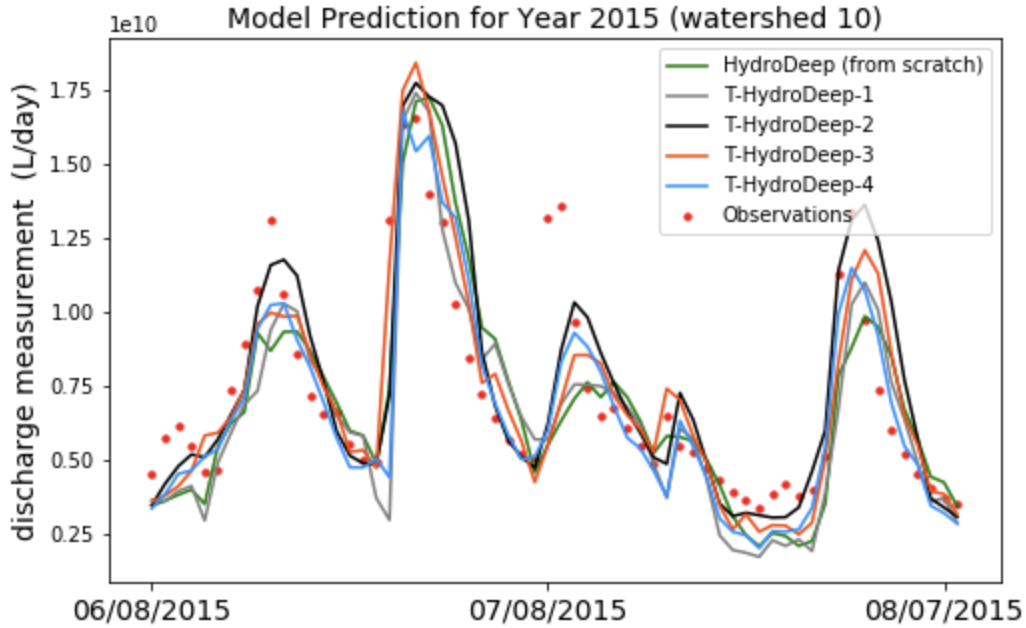
4. Evaluation of 4 Transfer Learning Approaches - T-HydroDeep-1, T-HydroDeep-2, T-HydroDeep-3, T-HydroDeep-4, with HydroDeep (with no pretraining on source watershed 13). (Related to Section 4)



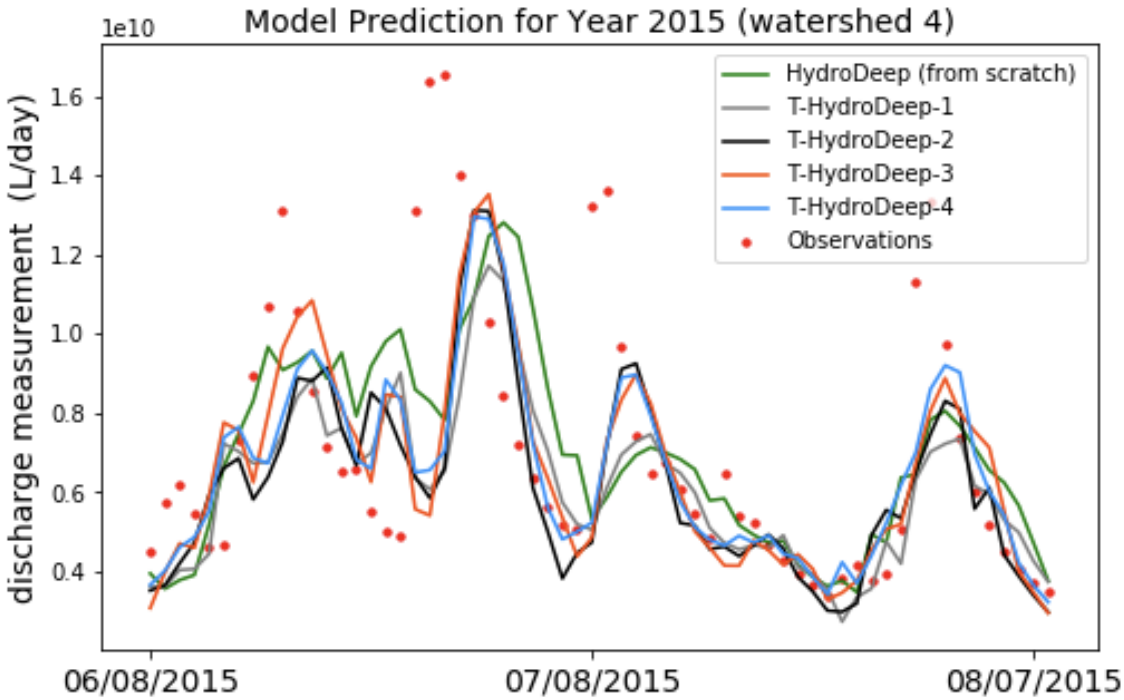
Supplementary Figure 4a. Watershed 14 (Related to Section 4)



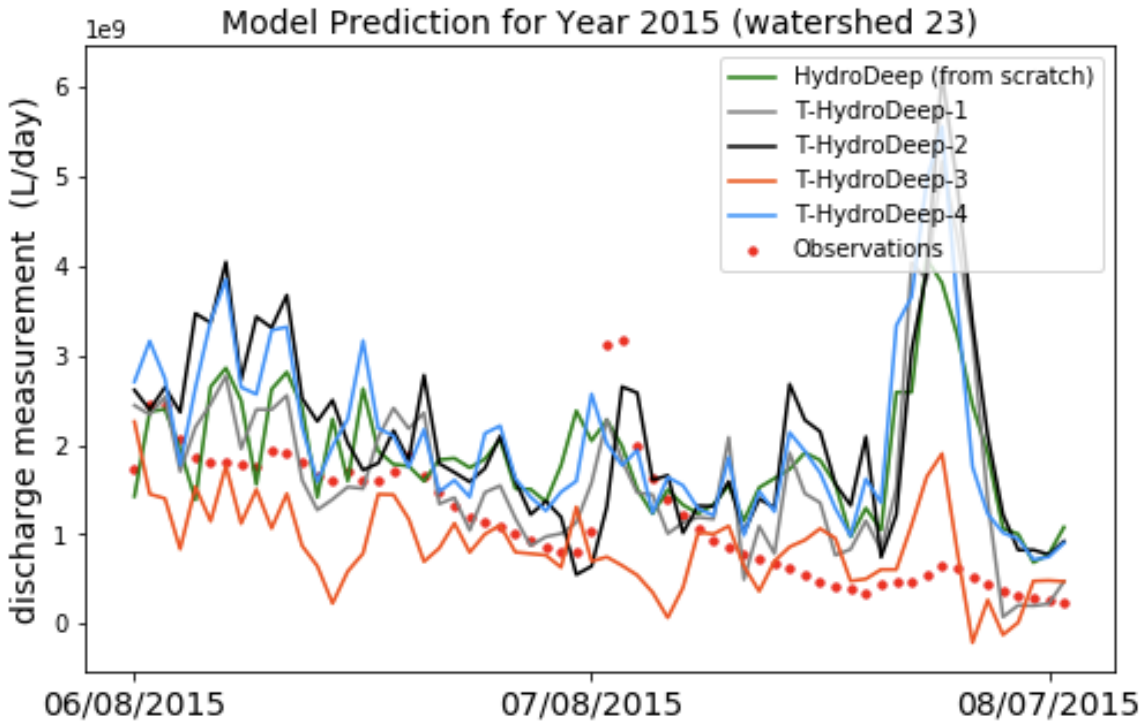
Supplementary Figure 4b. Watershed 15 (Related to Section 4)



Supplementary Figure 4c. Watershed 10 (Related to Section 4)



Supplementary Figure 4d. Watershed 4 (Related to Section 4)



Supplementary Figure 4e. Watershed 23 (Related to Table 4)

Supplementary Table 3. RRMSE of the Transfer Learning approaches. (Related to Section 4)

Approach/Watersheds	**W14** (RRMSE)	**W15** (RRMSE)	**W10** (RRMSE)	**W4** (RRMSE)	**W23** (RRMSE)
HydroDeep	2.2797	1.3379	0.6389	0.7747	2.3256
T-HydroDeep-1	2.1784	1.1262	0.6111	0.7057	2.1469
T-HydroDeep-2	2.2201	1.1209	**0.5139**	**0.6064**	2.2751
T-HydroDeep-3	2.2002	1.1709	0.5311	0.6289	**2.0508**
T-HydroDeep-4	**2.0722**	**1.0811**	0.5400	0.6066	2.1264

5. Hardware Specification

The extensive search to optimize hyperparameters for HydroDeep on the source watershed was done on the following hardware specification.

No. of Nodes	Processors/Node	Cores/Node	Memory/ Node	Interconnect	Local $TMPDIR Disk	Accelerator Card	CPU-Hour Cost Factor
1	Two 18-Core Intel Skylake 6140	36	192 GB	100G IB	1.5 TB	1 NVIDIA Tesla V100-32GB	2.7

The final training time for optimized HydroDeep and its variants T-HydroDeep-1, T-HydroDeep-2, T-HydroDeep-3, T-HydroDeep-4 were measured on Intel® Core™ i5-7200U CPU @ 2.50GHz × 4 with Intel® HD Graphics 620 (KBL GT2) and 12 GB RAM