Deep Temporal Neural Networks for Water Level Predictions of Watershed Systems

Jordan Huff[1], Jeremy Watts[1], Anahita Khojandi[1] *Member, IEEE*, and Jon Hathaway[2]

***Abstract*—Rainfall-runoff systems are complex hydrological environments that play a critical role in flood prevention. Currently, physics-based, process-driven computational models are often used to forecast future flooding events. However, these physics-based models are computationally expensive and require intensive physical measurements of hydrological environments beyond remote data collection. There is a growing body of literature that applies deep neural networks to time-series data for computationally efficient, real-time flooding predictions without the need for the complete virtual modeling of the hydrological system. However, these deep-learning networks' robustness at forecasting far into the future remains an open question. In this study, we examine the capabilities of Long Short-Term Memory (LSTM) networks and Temporal Convolutional Networks (TCN), state-of-the-art temporal deep neural networks, to forecast rainfall-runoff system depths. Specifically, this study leverages primary, multi-modal, time-series data collected by remote sensors in the watershed system of Conner Creek, a tributary of the Clinch River in eastern Tennessee. These data were collected in 5-minute intervals over a course of 5 months. Notably, the Conner Creek watershed system consists of four interconnected reservoir basins. We forecast the water level of each reservoir basin independently for times ranging from five minutes to two hours into the future. Our results show that both the LSTM and TCN can effectively model and forecast future reservoir basin water levels. Specifically, when averaged across the four reservoir basins, the LSTM has an mean absolute error (MAE), with a 95% confidence interval, of 0.158 $\pm$ 0.049 ft and 0.490 $\pm$ 0.260 ft at 5 minutes and 120 minutes into the future, respectively. In comparison, the TCN has an MAE of 0.258 $\pm$ 0.160 ft and 0.375 $\pm$ 0.245 ft at 5 minutes and 120 minutes into the future, respectively. Our results show that the LSTM model outperforms the TCN for near lead time forecasting; however, the TCN retains a greater relative accuracy at larger lead time forecasting periods (two hours). Nevertheless, both models can be considered effective at capturing future trends of watershed systems, demonstrating them to be powerful tools for use in flood risk management systems.**

***Index Terms*—Temporal Convolutional Neural Network, Long-short Term Memory, Watershed System, Water Level Prediction**

I. Introduction

Watershed systems, particularly in urban environments, are heavily affected by stormwater runoff. Although a portion of precipitation is always absorbed into the soil, intercepted by vegetation, and readmitted into the atmosphere through evapotransmition, heavy precipitation events can result in the creation of stormwater runoff, defined as precipitation which is not intercepted by one of the aforementioned methods [1], [2]. This runoff collects itself in various streams and bodies of water, including artificial reservoirs, whose role is to collect this excess stormwater, managing the effects of rainfall-runoff on the surrounding watershed [3]. Often, the creation of these reservoirs is necessary in heavily developed regions, where the natural methods of stormwater reduction are curtailed by vegetation loss. Alterations to land use and management are the primary urban methods for this phenomenon, including both active and passive management of the watershed systems [1], [4]. Over the past decades, this has taken form in the production of rainfall-runoff modeling.

Various critical features of the watershed region, including topography, soil type, land use, and land management, heavily affect the development of stormwater events [1]. To capture the complex nature of watershed systems, process-driven, physics-based models, such as PCSWMM, have historically been considered effective modeling solutions [5]–[7]. Developed for "stormwater, wastewater, watershed, and water distribution systems" [6], [8], PCSWMM and other process-driven models remained standard, but recent literature has demonstrated the potential for deep-learning models to match and surpass their predictive capabilities in certain environments [9]–[11]. Two such deep-learning models important to modern hydrological research are the Long Short-Term Memory (LSTM) and the Temporal Convolutional Network (TCN) [12]–[16]. LSTM models were developed to capture long-term temporal trends more effectively than other deep-learning models, specifically solving the vanishing gradient problem found in its predecessor, the Recurrent Neural Network (RNN) [17]. Recent literature, however, has suggested that, in the context of rainfall-runoff modeling, TCN models, developed as a sequential-modeling-focused successor to the Convolutional Neural Network (CNN), can improve prediction over similar machine learning methods [15], [18].

In this study, we compare two temporal deep-learning models (LSTM and TCN) in a rainfall-runoff context using primary data recorded at a watershed system in eastern Tennessee. This work builds on the existing literature to compare each models' effectiveness at lead time forecasting.

II. Study Area and Data Acquisition

In this study, we analyze the watershed area of Conner Creek, a tributary of the Clinch River in eastern Tennessee. This watershed system includes four dry extended detention

We kindly thank Aaron Akin at the University of Tennessee for their work in data collection [1].

[1]Jordan Huff, Jeremy Watts, and Anahita Khojandi are with the Department of Industrial and Systems Engineering, University of Tennessee, Knoxville, TN, USA (email: khojandi@utk.edu).

[2]Jon Hathaway is with the Department of Civil and Environmental Engineering, University of Tennessee, Knoxville, TN, USA.

reservoirs - artificial reservoirs designed to prevent flooding by retaining runoff water for at least 24 hours [1]. The largest basin is capable of retaining approximately 14,770 m^3 of water, with a maximum depth of 10 ft. A diagram of the watershed system is provided in Fig. 1.

The Conner Creek watershed system has been outfitted with on-basin equipment, including a controllable valve, a water depth sensor, and a rain gauge, to enable the active monitoring and management of the system by an environmental engineering research team at the University of Tennessee [1]. Specifically, this system utilizes an HC-SR04 ultrasonic depth sensor, and a "tipping bucket" style rain gauge for data collection. Data are also recorded at two stream gauging stations, "upstream" and "downstream," referring to their relative positions to the basins. At these locations, depth and various water quality metrics are recorded at the same rate as the basin data.

The primary data used in this study consists of 6,579 data points of 5-minute intervals recorded between April 12 and May 4, 2020. Included in this data are the depths of the four basins (referred to as Ponds 1-4), as well as upstream depth, downstream depth, and rainfall. Additionally, 18,014 data points collected between January 1 and March 27, 2021 were used as warm-start training data. These data were not included for testing the models' predictive performances, as they were recorded with 10-minute intervals, requiring interpolation to match the initial data set. As this process may introduce bias, these secondary data were used only in training. For preprocessing the data, a Kalman Filter [19] was applied to remove erroneous measurements and smooth trends. This type of smoothing process estimates real values from weighted means of noisy data [20]. For in-model scaling, Sklearn's min-max scaling function was used to normalize the data before predictions were made [21].

III. Methods

In this study, we compare the performance of two temporal deep-learning methods. Traditional deep-learning methods, such as neural networks, solve complex problems through "networks," represented as a series of computational nodes [22]. There are no less than three layers in a standard neural network: 1) an input layer, 2) one or more hidden layers, and 3) an output layer. The input layer contains as many nodes as there are features of the data set. The synapses that connect these nodes to those in the hidden layers hold "weights" that determine the importance of individual features within the hidden nodes. Through these weights, decisions are made as to what inputs/signals and their associated magnitude will be used in the final model. The manipulation of these weights allows for neural networks to "learn" what information is most important in creating accurate predictions. However, traditional neural networks do not account for the measurements' history and only make "snapshot" predictions. As such, traditional neural networks have limited use in time-series contexts, where the change is highly characteristic of the prediction [17].

A. LSTM

Long short term memory (LSTM) networks are a form of neural network that are designed to incorporate historical information in predictions. In order to do so, LSTM networks are formed by a line of cells, each incorporating a series of gates that allow for the systematic management of the information stored in memory. Each gate utilizes a neural-network-based structure, and work together in what essentially can be thought of as a process of three neural-network-powered filters. These filters - the forget gate, the memory gate, and the output gate - give the network capabilities for removing, filtering, and passing along data [17], [23], [24]. To accomplish this, LSTMs utilize "lookback windows," that collate a set of time-series data to which the LSTM makes a single prediction. In this study, we utilized Tensorflow-Keras for its Python LSTM library [25] and sequential modeling capabilities. The bidirectional version of the LSTM was chosen due to it's ability to make fewer assumptions about the distribution of the data [26], [27]. A lookback window of 24 data points, corresponding to two hours of data, was utilized and the model was fitted with a batch size of 12 and a validation split of 0.3. The structure of the model consisted of a bidirectional LSTM with 100 epochs, a dropout layer of 0.2, and a dense layer with linear activation.

B. TCN

Temporal Convolutional Networks (TCN) are a similar temporal deep-learning method used in sequential modeling. The uniqueness of the TCN stems from its "convolutional" structure, which comes from the TCN's predecessor, the Convolutional Neural Network (CNN). The TCN's convolutional layers are combined with pooling, activation, and dropout layers that operate in parallel with a 1x1 convolutional layer, which maintains the length of the layers [28]. These parallel operations exist in what are known as "residual blocks." These blocks make transformations of the data in the same manner as the convolutional blocks in the feature extraction portion of CNNs. Multiple residual blocks are often included before the final activation layer for non-linearity and 1x1 convolution for dimensionality reduction. Because of their unique structures and analogous results, LSTM and TCN models are commonly compared in literature [29], [30]. Specifically in this study, we utilized the Tensorflow supplemental "Keras-TCN" library [31] for TCN functionality alongside the use of standard Tensorflow-Keras [25] for sequential modeling. A lookback window of 24 data points - corresponding to two hours of data - was implemented, and the model was fitted with a batch size of 12 and a validation split of 0.3. The structure of the model consisted of one TCN layer and one dense layer.

IV. Results

We compare the performance of a LSTM and TCN model in a lead time forecasting scenario, testing their effectiveness at predicting water level at t_{i+k} time in the future. In

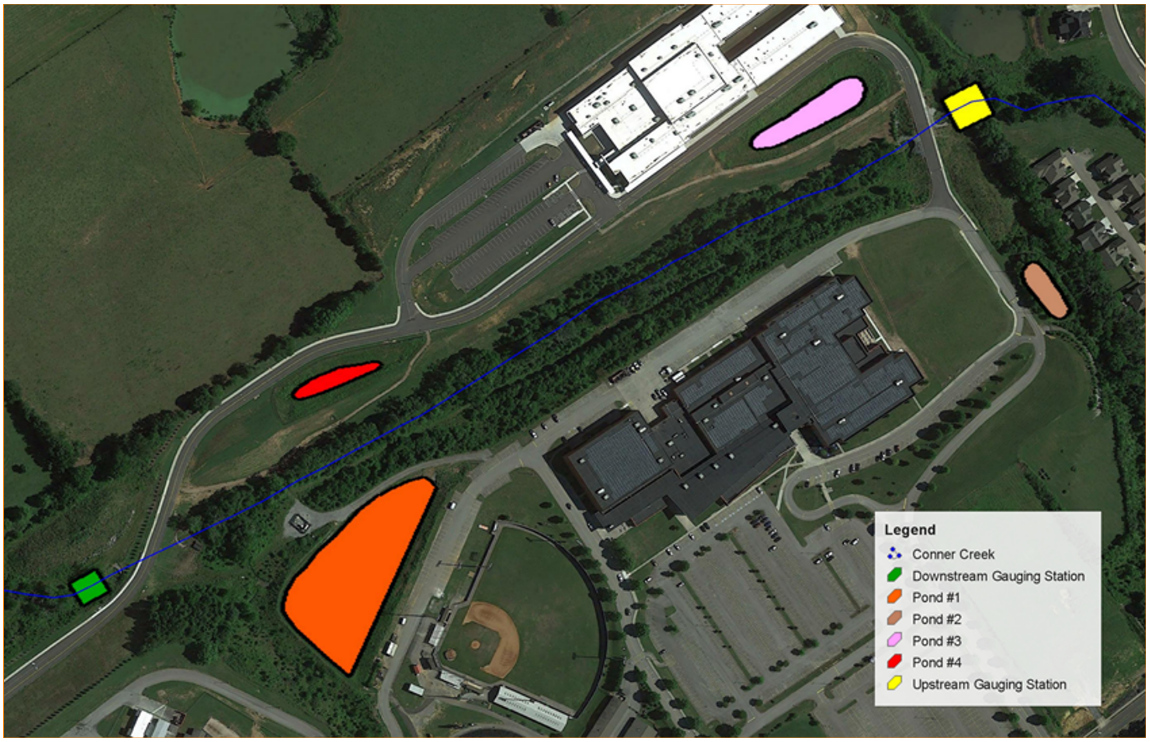

Fig. 1. Diagram of the Conner Creek watershed system. The system consists of four artificial reservoirs (Ponds 1-4). These reservoirs are interconnected. Water depth is independently measured in each pond as well as at the upstream and downstream gauging station. Measurements were made at 5-minute intervals [1].

traditional temporal deep-learning models, the forecast lead time is set to one time-step beyond the training window, i.e., t_1. However, in this study, we adjust t such that t_i predicts a point further into the future beyond the training window. Specifically, we manipulated t_i by moving it's associated forecast lead time k time into the future beyond t_i, resulting in a t_{i+k} forecast lead time, which predicts points further into the future. This process is shown in Fig. 2.

In this study, measurements are recorded in 5-minute intervals. Therefore, a forecast lead time at t_1 is predicting 5 minutes into the future and can be referred to as $t_{k=5min}$. To determine the robustness of the models on forecasting the ponds' depths, the models predicted future pond depths from 15 minutes to 120 minutes in 15-minute intervals. These eight comparisons, along with the $t_{k=5min}$ base case, were performed for both the bidirectional LSTM and the TCN for each pond individually. The models predicted pond depth using upstream depth, downstream depth, rainfall, and the time-series data for the same pond's depth. 10-fold cross-validation was performed for each comparison. Each model's performance was determined via the mean absolute error (MAE) averaged across all cross-validation folds, as well as the 95% confidence interval [32].

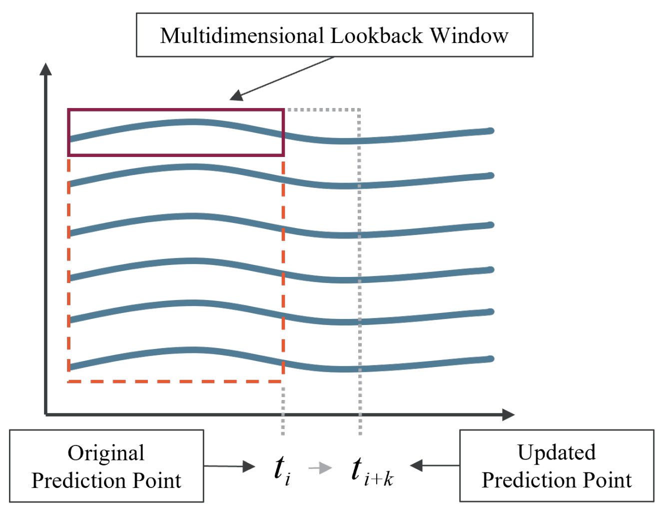

Fig. 2. An overview of the forecast lead time adjustment. Here multiple time-series data are provided to the deep-learning models. Testing data is represented in maroon and training data in orange. Lookback windows are constructed from the data, providing the models with a temporal understanding of the data. Traditionally, the model will predict point t_i; however, we adjust this point such that the lookback period is used to predict the future point t_{i+k}, providing a forecast for pond depth.

A. Pond 1 Predictive Performance

Pond 1 is considered the primary focus of this study, as it is the largest reservoir, and therefore, has the most consistent and gradual change in pond depth. The LSTM's and TCN's predictive capabilities when predicting the depth of Pond 1 are displayed in Figs. 3 and 4, which show each predicted point along with their true in-stream measurements. In Figs. 3 and 4, predictions are made at 5 minutes and 120 minutes, respectively. The predictions decrease in accuracy as the forecast lead time increases. The average MAE and 95% confidence interval across the 10-folds are provided in Table I. Specifically, for $t_{k=5min}$, the LSTM's MAE, with a 95% confidence interval, is 0.212 ± 0.057 ft, whereas for $t_{k=120min}$, the results are 0.918 ± 0.544 ft. Similarly, the TCN's MAE at $t_{k=5min}$ is 0.320 ± 0.153 ft, whereas for $t_{k=120min}$, is 0.681 ± 0.476 ft.

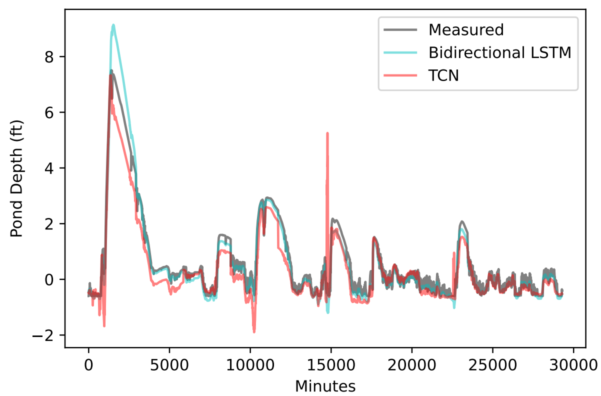

Fig. 3. Comparison of the bidirectional LSTM's and TCN's predictions for Pond 1 depth with a 5-minute forecast lead time. Each prediction is plotted along with the measured Pond 1 depth. This image is a composite of all predictions via 10-fold cross-validation.

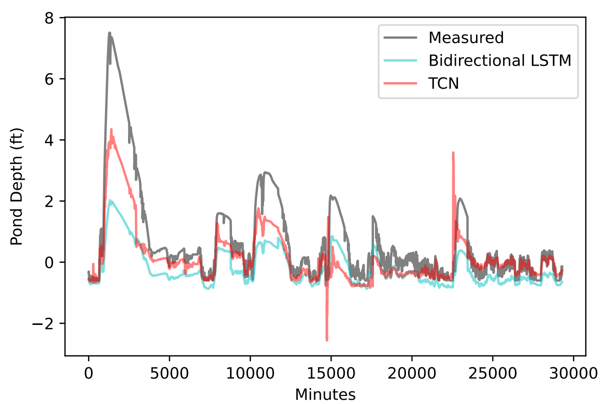

Fig. 4. Comparison of the bidirectional LSTM's and TCN's predictions for Pond 1 depth with a 2-hour forecast lead time. Each prediction is plotted along with the measured Pond 1 depth. This image is a composite of all predictions via 10-fold cross-validation.

B. Generalizability of Models

To better understand the generalizability of the temporal deep-learning models, we examine their predictive performance over the remaining three reservoirs (Ponds 2-4). In comparison to Pond 1, the remaining three ponds have less volume and have greater fluctuations based on the intake of rainfall-runoff. The average MAE and 95% confidence interval averaged across all cross-validation folds of the LSTM and TCN are presented in Fig. 5. The models have similar performance for Pond 2 and Pond 4, and overall reduced performance for Pond 3, especially with forecast lead times further into the future.

Specifically, for Pond 2, the LSTM had MAE results of 0.123 ± 0.037 ft at prediction time $t_{k=5min}$ and 0.254 ± 0.118 ft at time $t_{k=120min}$. For Pond 3, the results were 0.174 ± 0.053 ft at time $t_{k=5min}$ and 0.507 ± 0.149 ft at time $t_{k=120min}$. Finally, for Pond 4, the results were 0.117 ± 0.047 ft at time $t_{k=5min}$ and 0.282 ± 0.229 ft at time $t_{k=120min}$. Comparatively, for Pond 2, the TCN had MAE results of 0.257 ± 0.235 ft at time $t_{k=5min}$ and 0.185 ± 0.163 ft at time $t_{k=120min}$. For Pond 3, the results were 0.243 ± 0.098 ft at time $t_{k=5min}$ and 0.430 ± 0.202 ft at time $t_{k=120min}$. For Pond 4, the results were 0.214 ± 0.153 ft at time $t_{k=5min}$ and 0.204 ± 0.137 ft at time $t_{k=120min}$.

V. Discussion and Future Work

Forecasting watershed reservoir depths is a critical function for the prediction of flooding events and watershed management. Our results show that both the bidirectional LSTM and the TCN are capable of accurately predicting the depths of basins in rainfall-runoff systems. These results are consistent with the literature, which has found that temporal deep-learning models work well in this context [9]–[11]. However, the models do not perform consistently when predicting points further into the future. While, in general, both models perform well, each has trade-offs which may impact which model performs best in a given context.

A. Influence of Forecasting Time

Both the bidirectional LSTM and TCN models have reduced predictive performance at larger forecast lead times. Namely, for the LSTM, the average MAE of Pond 1 predictions increases from 0.212 ± 0.057 ft to 0.918 ± 0.544 ft from 5 minutes to 2 hours. This represents an average depreciation in predictive capabilities of 0.706 ft, or a relative percentage change of 433%, whereas the TCN obtains an average MAE of Pond 1 predictions of 0.320 ± 0.153 ft to 0.681 ± 0.476 ft from 5 minutes to 2 hours. This represents an average depreciation in predictive capabilities of 0.361 ft, or a relative percentage change of 213%. Both models' reduction in predictive performance is largely due to the increasing difficulty of the problem at hand, as the trends further into the future are less dependent on the trends of the near past. However, while the models' performances depreciate when predicting further into the future, they both are capable of following trends. It is interesting to note that as the prediction time moves further into the future, both models appear to be less accurate at predicting the magnitude of the pond depth while still predicting these trends relatively accurately. This can be observed in Figs. 3 and 4.

	Pond 1								
	5 mins	15 mins	30 mins	45 mins	60 mins	75 mins	90 mins	105 mins	120 mins
LSTM	0.212 ± 0.057	0.199 ± 0.079	0.203 ± 0.049	0.302 ± 0.135	0.465 ± 0.327	0.488 ± 0.178	0.635 ± 0.326	0.795 ± 0.478	0.918 ± 0.544
TCN	0.320 ± 0.153	0.392 ± 0.291	0.411 ± 0.368	0.527 ± 0.413	0.593 ± 0.318	0.582 ± 0.269	0.387 ± 0.189	0.574 ± 0.209	0.681 ± 0.476

TABLE I

THE MAE AND 95% CONFIDENCE INTERVAL, MEASURED IN FEET, AVERAGED ACROSS 10-FOLD CROSS-VALIDATION FOR BOTH THE BIDIRECTIONAL LSTM AND THE TCN. THESE RESULTS SHOW THE MODELS' PERFORMANCE WHEN PREDICTING THE DEPTH OF POND 1 WITH DIFFERENT FORECAST LEAD TIMES, RANGING FROM 15 MINUTES TO 2 HOURS IN 15-MINUTE INTERVALS, AS WELL AS A BASE CASE OF 5 MINUTES.

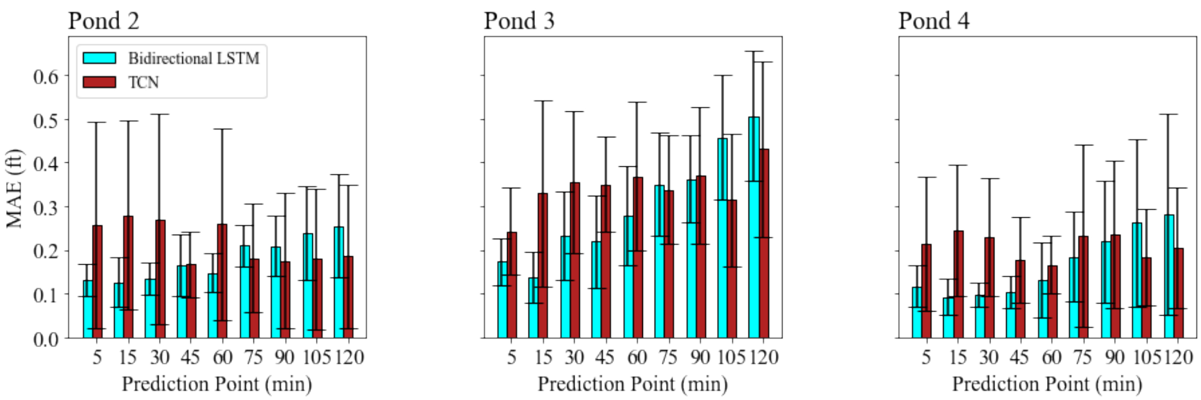

Fig. 5. Comparison of the average of the bidirectional LSTM's and TCN's predictions of depth for Ponds 2-4 with forecast lead times from 5 minutes to 2 hours. The MAE is presented along with the 95% confidence interval. The MAE and 95% confidence interval for each prediction time are averaged over all cross-validation folds.

B. Model Performance Comparison

The differences between the two models' decreases in accuracy demonstrate a contrasting narrative to the trends in MAE between $t_{k=5min}$ and $t_{k=75min}$. In these first six interactions, the bidirectional LSTM outperforms the TCN with lower average MAE. However, both models' predictive performances become more comparable for the tests after $t_{k=75min}$. This is demonstrated in the results for every ponds' prediction. Namely, for forecast lead times between $t_{k=90min}$ and $t_{k=120min}$, the TCN has a lower average MAE than the LSTM. However, the consistency of the models' performances, represented by the 95% confidence interval, is also an important consideration. The TCN has a much wider confidence interval during short-term predictions ($t_{k\leq 75min}$) than the LSTM. Specifically, the LSTM average confidence interval across all ponds (Ponds 1-4) at $t_{k=5min}$ is 0.049 ft and at $t_{k=120min}$ is 0.260 ft, where as the TCN's average confidence interval at $t_{k=5min}$ is 0.160 ft and at $t_{k=120min}$ is 0.245 ft. The differences in the confidence intervals suggest that the LSTM is more consistent with its predictions and may be applicable to a wider variety of modeling environments, especially those necessitating a shorter forecasting lead time.

The potential reasoning behind this finding may be found in the nature of the models' predictions. The bidirectional LSTM appears to be more conservative in its predictions and is often more consistent in capturing the shapes of trends, rather than the magnitude. This is demonstrated in its steady increase in MAE. The TCN, however, while overall having less variance in accuracy throughout the experiment, is also consistently reactive and can make unrealistic predictions well into the negative, even though the measured data never falls below -0.6 ft. Due to the 'black box' nature of deep-learning algorithms, one may only speculate as to why the TCN model was more prone to these inaccuracies than the LSTM model, but these results suggest the TCN has a more 'reactive' nature in this context.

VI. CONCLUSION

Watershed systems provide unique problems in regards to flood prevention and general forecasting. Since the development of deep-learning models has brought about the potential to surpass the process-driven models that are commonplace in hydrology, it has become an important field of study to determine the extent of these deep-learning models' capabilities. This study has developed a unique method for determining the accuracy of deep-learning models when forecasting the future of rainfall-runoff systems. Our results suggest that both LSTM and TCN models are effective at forecasting rainfall-runoff trends. Specifically, the LSTM model has a lower average MAE for near forecasting lead times (less than 75 minutes). However, the TCN is comparable to the LSTM for forecast lead times ranging from 75 to 120 minutes into the future. Further, the TCN has larger confidence intervals, especially notable for near-term predictions; however, the LSTM has reduced consistency as the prediction time increases further into the future, suggesting it may be less reliable on large time scales. The results presented in this study demonstrate the ability for these two models to be incorporated in flood risk management systems with expanded uses within the realm of watershed management.

This work is subject to certain limitations and constraints. Primarily, the real-world nature of the collected data leads to erroneous measurements. To reduce the influence of unrealistic spiking (likely caused by sensor error or environmental interference), a Kalman filter was applied to all time-series data to smooth trends. Further, as this data set is limited to several months of data collection, it is unable to be generalized to understand the effects of seasonality on the predictive capabilities of the models.

To more fully understand the comparison and generalizability of the temporal methods, additional work is required to model other watersheds systems with dynamic interactions in a variety of regions. Additionally, further analysis on the distribution of the LSTM and TCN predictions in regard to seasonality (data spanning longer time frames) is necessary to develop literature encompassing the potential of deep-learning flood forecasting.

Further, there is the potential application of LSTM and TCN models as the data-driven predictive forecasting method underpinning real-time control experiments and dynamic alerting systems. The incorporation of deep-learning models to such real-time control systems has the potential to provide significantly more cost-effective alternatives to the process-based models which are the standard approach in watershed management literature. Therefore, additional work to determine the relative cost-performance and efficacy of these methods for dynamic control is required.

References

[1] A. A. Akin, "Impact and application of real-time control on stormwater systems," *University of Tennessee*, 2021.

[2] R. L. Huffman, D. D. Fangmeier, W. J. Elliot, S. R. Workman, and G. Schwab, *Soil and water conservation engineering*. American Society of Agricultural and Biological Engineers St. Joseph, 2011.

[3] F. Interagency Stream Restoration Working Group (US), *Stream corridor restoration: Principles, processes, and practices*. National Technical Info Svc, 1998.

[4] T. Dunne and L. B. Leopold, *Water in environmental planning*. Macmillan, 1978.

[5] P. Rees and J. Schoen, "PCSWMM evaluation," *Water Resources Research Centre, University of Massachusetts: Boston, MA, USA*, 2009.

[6] V. Hamouz, P. Møller-Pedersen, and T. M. Muthanna, "Modelling runoff reduction through implementation of green and grey roofs in urban catchments using PCSWMM," *Urban Water Journal*, vol. 17, no. 9, pp. 813–826, 2020.

[7] L. M. Sidek, L. H. C. Chua, A. S. M. Azizi, H. Basri, A. S. Jaafar, and W. C. Moon, "Application of PCSWMM for the 1-D and 1-D–2-D modeling of urban flooding in Damansara Catchment, Malaysia," *Applied Sciences*, vol. 11, no. 19, p. 9300, 2021.

[8] W. James, L. A. Rossman, and W. R. C. James, "User's guide to SWMM 5:[based on original USEPA SWMM documentation]," CHI, 2010.

[9] G. Mao, M. Wang, J. Liu, Z. Wang, K. Wang, Y. Meng, R. Zhong, H. Wang, and Y. Li, "Comprehensive comparison of artificial neural networks and long short-term memory networks for rainfall-runoff simulation," *Physics and Chemistry of the Earth, Parts A/B/C*, vol. 123, p. 103026, 2021.

[10] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger, "Rainfall–runoff modelling using long short-term memory (LSTM) networks," *Hydrology and Earth System Sciences*, vol. 22, no. 11, pp. 6005–6022, 2018.

[11] H. Fan, M. Jiang, L. Xu, H. Zhu, J. Cheng, and J. Jiang, "Comparison of long short term memory networks and the hydrological model in runoff simulation," *Water*, vol. 12, no. 1, p. 175, 2020.

[12] C. Hu, Q. Wu, H. Li, S. Jian, N. Li, and Z. Lou, "Deep learning with a long short-term memory networks approach for rainfall-runoff simulation," *Water*, vol. 10, no. 11, p. 1543, 2018.

[13] Z. Xiang, J. Yan, and I. Demir, "A rainfall-runoff model with LSTM-based sequence-to-sequence learning," *Water Resources Research*, vol. 56, no. 1, p. e2019WR025326, 2020.

[14] Y. Xu, C. Hu, Q. Wu, S. Jian, Z. Li, Y. Chen, G. Zhang, Z. Zhang, and S. Wang, "Research on particle swarm optimization in LSTM neural networks for rainfall-runoff simulation," *Journal of Hydrology*, vol. 608, p. 127553, 2022.

[15] K. Lin, S. Sheng, Y. Zhou, F. Liu, Z. Li, H. Chen, C.-Y. Xu, J. Chen, and S. Guo, "The exploration of a temporal convolutional network combined with encoder-decoder framework for runoff forecasting," *Hydrology Research*, vol. 51, no. 5, pp. 1136–1149, 2020.

[16] Z. Xiang and I. Demir, "Fully distributed rainfall-runoff modeling using spatial-temporal graph neural network," 2022.

[17] S. Siami-Namini, N. Tavakoli, and A. S. Namin, "A comparison of ARIMA and LSTM in forecasting time series," in *2018 17th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 1394–1401, IEEE, 2018.

[18] Y. Chen, Y. Kang, Y. Chen, and Z. Wang, "Probabilistic forecasting with temporal convolutional neural network," *Neurocomputing*, vol. 399, pp. 491–501, 2020.

[19] R. E. Kalman, "A new approach to linear filtering and prediction problems," *Transactions of the ASME–Journal of Basic Engineering*, vol. 82, no. Series D, pp. 35–45, 1960.

[20] G. Dean, "An introduction to Kalman filters," *Measurement and Control*, vol. 19, no. 2, pp. 69–73, 1986.

[21] J.-M. Jo, "Effectiveness of normalization pre-processing of big data to the machine learning performance," *The Journal of the Korea Institute of Electronic Communication Sciences*, vol. 14, no. 3, pp. 547–552, 2019.

[22] C. Gershenson, "Artificial neural networks for beginners," *arXiv preprint cs/0308031*, 2003.

[23] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, 2019.

[24] R. C. Staudemeyer and E. R. Morris, "Understanding LSTM–a tutorial into long short-term memory recurrent neural networks," *ArXiv preprint arXiv:1909.09586*, 2019.

[25] F. Chollet *et al.*, "Keras." https://keras.io, 2015.

[26] M. Schuster and K. K. Paliwal, "Bidirectional recurrent neural networks," *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, 1997.

[27] R. Zhao, R. Yan, J. Wang, and K. Mao, "Learning to monitor machine health with convolutional bi-directional LSTM networks," *Sensors*, vol. 17, no. 2, p. 273, 2017.

[28] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," *ArXiv preprint arXiv:1511.07122*, 2015.

[29] S. Gopali, F. Abri, S. Siami-Namini, and A. S. Namin, "A comparison of TCN and LSTM models in detecting anomalies in time series data," in *2021 IEEE International Conference on Big Data (Big Data)*, pp. 2415–2420, IEEE, 2021.

[30] Y. Xu, C. Hu, Q. Wu, Z. Li, S. Jian, and Y. Chen, "Application of temporal convolutional network for flood forecasting," *Hydrology Research*, vol. 52, no. 6, pp. 1455–1468, 2021.

[31] P. Remy, "Temporal convolutional networks for Keras." https://github.com/philipperemy/keras-tcn, 2020.

[32] J. Qi, J. Du, S. M. Siniscalchi, X. Ma, and C.-H. Lee, "On mean absolute error for deep neural network based vector-to-vector regression," *IEEE Signal Processing Letters*, vol. 27, pp. 1485–1489, 2020.