water

MDPI

Article

Forecasting Groundwater Table in a Flood Prone Coastal City with Long Short-term Memory and Recurrent Neural Networks

Benjamin D. Bowes [1], Jeffrey M. Sadler [1], Mohamed M. Morsy [1,2], Madhur Behl [1,3] and Jonathan L. Goodall [1,*]

1 Dept. of Engineering Systems and Environment, Univ. of Virginia, 351 McCormick Rd., P.O. Box 400742, Charlottesville, VA 22904, USA; bdb3m@virginia.edu (B.D.B.); jms3fb@virginia.edu (J.M.S.); mmm4dh@virginia.edu (M.M.M.); mb2kg@virginia.edu (M.B.)
2 Irrigation and Hydraulics Dept., Cairo University, P.O. Box 12211, Giza 12614, Egypt
3 Dept. of Computer Science, Univ. of Virginia, 351 McCormick Rd., P.O. Box 400259, Charlottesville, VA 22904, USA
* Correspondence: goodall@virginia.edu

Received: 9 April 2019; Accepted: 22 May 2019; Published: 25 May 2019

Abstract: Many coastal cities are facing frequent flooding from storm events that are made worse by sea level rise and climate change. The groundwater table level in these low relief coastal cities is an important, but often overlooked, factor in the recurrent flooding these locations face. Infiltration of stormwater and water intrusion due to tidal forcing can cause already shallow groundwater tables to quickly rise toward the land surface. This decreases available storage which increases runoff, stormwater system loads, and flooding. Groundwater table forecasts, which could help inform the modeling and management of coastal flooding, are generally unavailable. This study explores two machine learning models, Long Short-term Memory (LSTM) networks and Recurrent Neural Networks (RNN), to model and forecast groundwater table response to storm events in the flood prone coastal city of Norfolk, Virginia. To determine the effect of training data type on model accuracy, two types of datasets (i) the continuous time series and (ii) a dataset of only storm events, created from observed groundwater table, rainfall, and sea level data from 2010–2018 are used to train and test the models. Additionally, a real-time groundwater table forecasting scenario was carried out to compare the models' abilities to predict groundwater table levels given forecast rainfall and sea level as input data. When modeling the groundwater table with observed data, LSTM networks were found to have more predictive skill than RNNs (root mean squared error (RMSE) of 0.09 m versus 0.14 m, respectively). The real-time forecast scenario showed that models trained only on storm event data outperformed models trained on the continuous time series data (RMSE of 0.07 m versus 0.66 m, respectively) and that LSTM outperformed RNN models. Because models trained with the continuous time series data had much higher RMSE values, they were not suitable for predicting the groundwater table in the real-time scenario when using forecast input data. These results demonstrate the first use of LSTM networks to create hourly forecasts of groundwater table in a coastal city and show they are well suited for creating operational forecasts in real-time. As groundwater table levels increase due to sea level rise, forecasts of groundwater table will become an increasingly valuable part of coastal flood modeling and management.

Keywords: groundwater table; forecast; recurrent neural network; long short-term memory; coastal flooding

1. Introduction

Storm events in low relief coastal areas can quickly raise the groundwater table, which is often relatively shallow [1,2]. During these events, infiltration and groundwater table response decrease the volume available for stormwater storage, therefore increasing runoff and, by extension, loads on stormwater systems [3]. Many coastal urban areas are also experiencing increased flooding due to land subsidence and climate change effects, such as sea level rise [4], increased precipitation, and more frequent extreme climactic events [5]. While there are several causes of flooding in coastal cities [6], the groundwater table level is a largely unrepresented factor and forecasting its variations can provide valuable information to aid in planning and response to storm events. Furthermore, because the groundwater table will rise as sea level rises [3,7–9], stormwater storage capacity will continue to decrease and inundation from groundwater may occur. Damage from groundwater inundation, which occurs through different mechanisms than overland flooding, can have significant impacts on subsurface structures [10,11]. Even if groundwater inundation does not regularly reach the land surface, increased duration of high groundwater table levels could have significant impacts on infrastructure [8,12–14] making groundwater table forecasting an increasingly important part of effectively modeling and predicting coastal urban flooding.

In the field of groundwater hydrology, models based on the physical principles of groundwater flow have traditionally been some of the main tools for understanding the mechanics of these systems [9,15–20]. Developing these models, however, requires extensive details about aquifer properties. In urban areas, this level of detail is hard to achieve at high resolutions because the subsurface is a complex mix of natural and anthropogenic structures such as varied geologic deposits, buried creeks or wetlands, roadbeds, building foundations, and sanitary and stormwater pipes. These factors should be considered when developing a physics-based model; if the necessary data are not available then assumptions and estimations must be substituted based on domain knowledge. Even if the data necessary to build a physics-based model are available, there is still the challenge of calibrating the model to adequately reflect reality.

Machine learning approaches are being increasingly used by hydrologists in order to mitigate the difficulties associated with physics-based models [6,21–27]. The advantage of such data-driven modeling is that physical relationships and the physical parameters needed to describe the physical environment do not need to be explicitly defined; the machine learning algorithm approximates the relationship between model inputs and outputs through an iterative learning process [28]. Neural networks (NN) have been used to model and predict nonlinear time series data, such as the groundwater table, and have been found to perform as well as, and in some cases, better than physics-based models [29,30]. Several studies have applied NN models on a daily or monthly time step to aquifers used for water supply in order to make forecasts appropriate for groundwater management. [31–36]. Few studies, however, have used NNs for predicting the groundwater table in unconfined surficial coastal aquifers where flooding is a major concern and a finer time scale is needed to capture the impacts of storm events [2].

Recurrent neural networks (RNNs) have been a popular choice for modeling groundwater time series data because they attempt to retain a memory of past network conditions. While RNNs have been successfully applied to groundwater modeling [31–34], it's been found that standard RNN architectures have difficulty capturing long term dependencies between variables [37]. This is due to two problems, (i) vanishing and (ii) exploding gradient, where weights in the network go to zero or become extremely large during model training. These two problems occur because the error signal can only be effectively backpropagated for a limited number of steps [38].

One of the most successful approaches to avoiding the vanishing and exploding gradient problems has been the long short-term memory (LSTM) variant of standard RNNs [38]. LSTM is able to avoid these training problems by eliminating unnecessary information being passed to future model states, while retaining a memory of important past events. In the field of natural language processing, LSTM has become a popular choice of neural network because of its ability to retain context over long

spans [39]. LSTM has also been effective for financial time series prediction [40] and for short-term traffic and travel time predictions [41,42] Despite the wide variety of applications, however, LSTM has only recently been used for hydrologic time series prediction [43,44]. For example, LSTM was found to outperform two simpler RNN architectures for predicting streamflow [45]. LSTM networks have also recently been used to model the groundwater table on a monthly time step in an inland agricultural area of China [46]. This agriculture focused study provides valuable information on the advantages of LSTM for groundwater level prediction over a basic feed-forward neural network (FFNN), but only presents predictions for one time step ahead. In a real-time flood forecasting application, however, longer forecasts of the groundwater table at short time intervals would be needed [2] and should include the use of forecast input data. LSTM models have yet to be evaluated for this type of application.

With the growing availability of large datasets and high performance computing, data-driven modeling techniques can now be evaluated for groundwater table forecasting. The objective of this study, therefore, is to compare RNN and LSTM neural networks for their ability to model and predict groundwater table changes in an unconfined coastal aquifer system with an emphasis on capturing groundwater table response to storm events. Based on prior research on this topic, it is expected that LSTM will outperform RNN for forecasting groundwater table levels. In this study, LSTM and RNN models were built for seven sites in Norfolk, Virginia USA, a flood prone coastal city. The models were trained and tested using observed groundwater table, sea level, and rainfall data as input. In addition to comparing RNN and LSTM neural networks, the effect of different training methods on model accuracy was evaluated by creating two unique datasets, one of the complete time series and one containing only periods identified as storms. The two types of datasets were bootstrapped and a statistical comparison of the two model types was made with *t*-tests to determine if differences in the results were significant. To ensure fair comparison, the hyperparameters of the RNN and LSTM networks were individually optimized with an advanced tuning technique instead of traditional ad-hoc methods. Once trained and evaluated, the RNN and LSTM models were tested with forecast sea level and rainfall input data to quantify the accuracy that could be expected in a real-time forecasting scenario.

This paper is structured as follows: First, a description of the study area, data and methodology used is given in Section 2. The methodology includes a description of the RNN and LSTM networks, input data preprocessing, and how models are trained and evaluated. The results of data preprocessing and modeling are then presented in Section 3 and discussed in Section 4. Conclusions are drawn in Section 5.

2. Materials and Methods

2.1. Study Area

The City of Norfolk, Virginia is located on the southern portion of the Chesapeake Bay along the eastern coast of the United States (Figure 1, inset). The city covers 171 km^2 of land with an average elevation of 3.2 m (above the North American Vertical Datum of 1988) and has 232 km of shoreline. Home to almost a quarter million people [47], Norfolk serves important economic and national security roles with one of the U.S.'s largest commercial ports, the world's largest naval base, and the North American Headquarters for the North Atlantic Treaty Organization (NATO). The larger Hampton Roads Region, of which Norfolk is a major part, has the second greatest risk from sea level rise in the U.S. and is surpassed only by New Orleans [48]. This risk is partly due to coupled sea level rise and regional land subsidence from groundwater withdrawals from the deep Potomac Aquifer for water supply and glacial isostatic adjustment [49]. Because of these and other factors, including low relief terrain and a regular hurricane season, the city and larger Hampton Roads region face increasingly frequent and severe recurrent flooding [4] which threatens its economic, military, and historic importance.

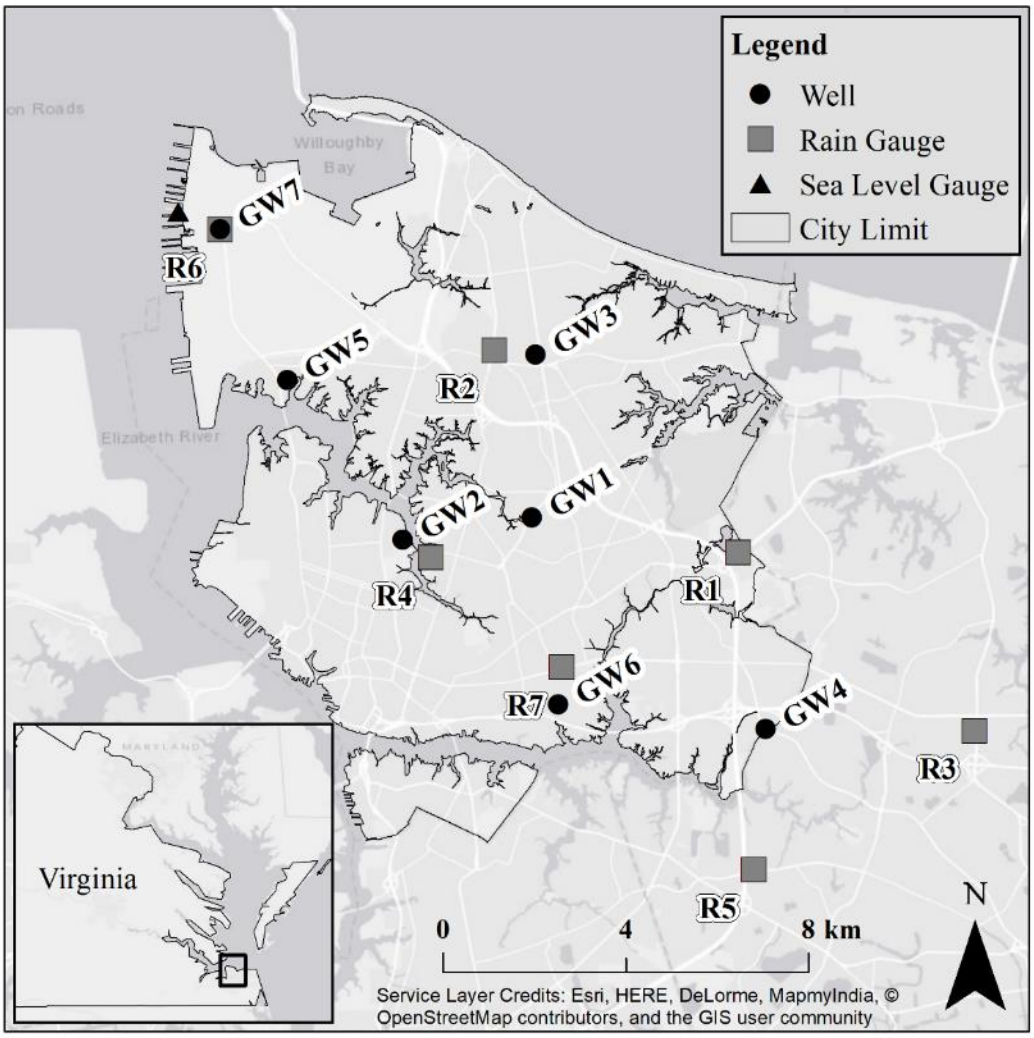

Figure 1. Location of gauges in Norfolk, Virginia.

2.2. Data

In order to predict groundwater table levels, the neural networks created in this study were trained and tested with the available groundwater table, rainfall, and sea level data as input. Input data was collected in two forms: observed and forecasted.

2.2.1. Observed Data

A unique dataset of groundwater table level observations for seven shallow monitoring wells in Norfolk was provided by the Hampton Roads Sanitation District (HRSD) (Figure 1, Table 1). Groundwater observations, in meters, are measured at a two minute time step and referenced to the North American Vertical Datum of 1988 (NAVD88). Observed rainfall data, in millimeters, also came from HRSD and was measured at a fifteen minute time step. Observed sea level data, in meters, was measured at a six minute time step, and is referenced to NAVD88. Sea level data came from the National Oceanic and Atmospheric Administration (NOAA) Sewells Point gauge [50]. The mean, minimum, and maximum sea level at this station is 0.11 m, −0.98 m, and 1.88 m, respectively. All of the observed data are for the period between 1 January 2010 and 31 May 2018.

Table 1. Groundwater table monitoring well details.

Well ID	Land Surface Elevation (m) [a]	Well Depth (m) [b]	Distance to Tidal Water (m)	Impervious Area (%) [c]	Groundwater Table Level (m) [a,d]		
					Minimum	Maximum	Mean
GW1	2.21	4.27	36	27	−0.678	0.883	−0.102
GW2	1.24	4.08	32	23	−0.670	1.476	0.635
GW3	4.35	5.18	668	42	1.197	3.844	2.026
GW4	3.24	4.57	777	53	0.659	2.021	1.075
GW5	1.72	2.53	32	20	−0.167	1.5562	0.492
GW6	2.35	3.23	41	30	0.259	2.012	0.742
GW7	2.57	4.60	650	73	0.200	1.750	0.707

[a] Referenced to North American Vertical Datum of 1988 (NAVD88); [b] Below land surface; [c] Percent of area classified as impervious within a 610 m buffer around well; [d] Statistics calculated from January 2010 to May 2018.

An examination of the observed data shows that each well has a different response to storm events (Figure 2). For instance, GW2 shows a large, rapid increase in the groundwater table from the first pulse of rainfall and GW4 shows more of a step response in the groundwater table to the three distinct pulses of rainfall. The groundwater level at GW6, however, shows a small, gradual increase in response to the storm event. While rainfall appears to be the main driver of groundwater table levels in all of these wells, sea level is also an important forcing factor which has a diminishing impact with increasing distance from the coast [3,51].

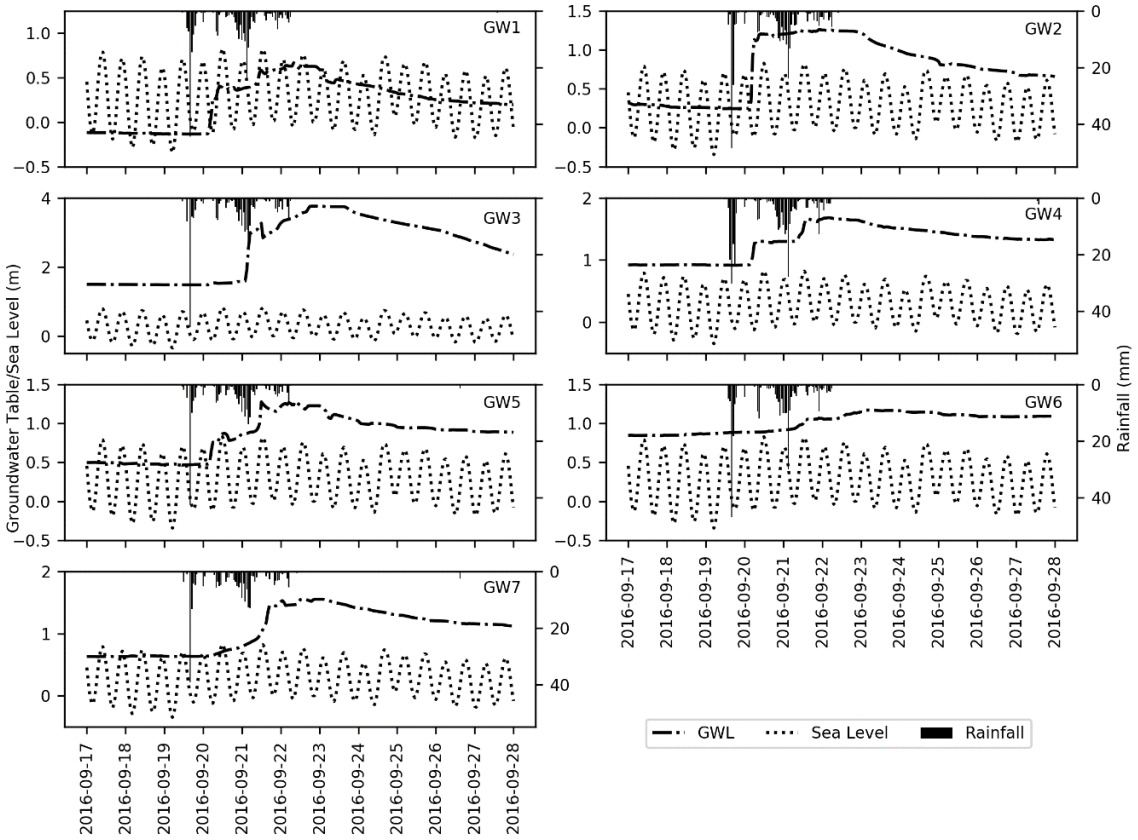

Figure 2. Hourly groundwater table level, sea level, and rainfall at individual wells for Tropical Storm Julia.

2.2.2. Forecast Data

In order to simulate a real-time forecast scenario, archived forecast data were collected for three months: September, 2016, January, 2017, and May 2018. These months were selected because archived forecast data was available and had both dry periods and storm events. The storm events in the

archived forecast data ranged from unnamed storms to Hurricane Hermine and Tropical Storm Julia, which has an estimated return period of 100–200 years, based on the 24 and 48 h rainfall [52]. Forecast rainfall was generated by the High-Resolution Rapid Refresh (HRRR) model, a product of the National Center for Environmental Prediction (NCEP), which generates hourly forecasts of meteorological conditions, including total surface precipitation, for the coming 18 h with a resolution of 3 km^2. These data are archived by the Center for High Performance Computing at the University of Utah [53] and was accessed from that database.

Forecast sea level data for the Sewells Point station was gathered from NOAA [54] for the same three months as the rainfall forecasts. These sea level data were downloaded at an hourly time step, and is referenced to NAVD88. The model used to generate sea level predictions at this station is based on the harmonic constituents of the observed tide cycle [55,56]. While harmonic predictions can closely match the observed sea level under normal weather conditions, they do not include any storm surge effects.

2.3. Methodology

This study was carried out through the workflow detailed in Figure 3. As such, this section is divided into three main subsections: Data preprocessing, neural network modeling, and results post-processing. Links to model code and data are given in the Supplemental Data section at the end of this article.

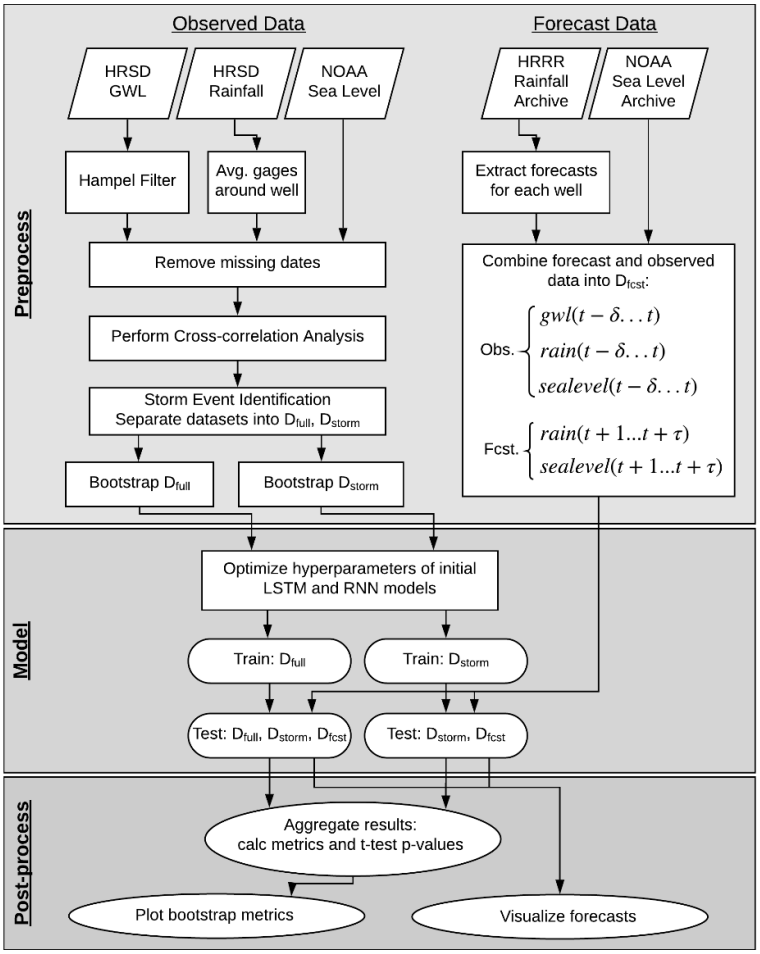

Figure 3. Study workflow detailing major steps in the data preprocessing, neural network modeling, and results post-processing.

2.3.1. Input Data Preprocessing

Data preprocessing involves a number of steps for observed and forecast data (Figure 3). Raw groundwater table observations were filtered with a Hampel filter [57] to remove large erroneous values. This filter used the standard deviation of the observations within a single day (720 two minute observations) rolling window as a threshold; any observations greater than the threshold were replaced by the rolling median. All of the raw observed data were aggregated to an hourly time step to match the format of the forecast data. Groundwater table and sea levels were aggregated using the hourly mean value and rainfall is the cumulative hourly total. Because some wells did have several months of missing data, any time steps with any missing values were removed. For wells without an immediately adjacent rain gauge, the rainfall at the well was assumed to be the mean of the surrounding rain gauges (Table 2).

Table 2. Rain gauges associated with each well based on geographic proximity.

Well ID	Rain Gauge (s)
GW1	R1 R2 R4 R7
GW2	R4
GW3	R2
GW4	R1 R3 R5 R7
GW5	R2 R6
GW6	R7
GW7	R6

To prepare the filtered and continuous data as model input, the time series of each variable (groundwater table, sea level, and rainfall) was shifted to include relevant past observations, based on an appropriate lag δ, and observations up to the forecast horizon τ (18 h in this study to correspond to the HRRR model forecast horizon). Lags for each well represent the delay between a rainfall or sea level observation and the corresponding response of the groundwater table and were identified by cross-correlation analysis (see Section 3.1.1). After shifting the time series of each variable, all data were normalized to values between 0–1 and combined into an input matrix or tensor and a label tensor. Each row in the input tensor contains three vectors: Groundwater table $\mathbf{gwl_I}$, rainfall **rain**, and sea level **sea**. Each row in the label tensor is a vector of groundwater table values $\mathbf{gwl_L}$ to be predicted (Table 3).

Table 3. Input and label tensors for neural network modeling.

Inputs	Labels
$\mathbf{gwl_I} = \{t - \delta \ldots t\}$ $\mathbf{rain} = \{t - \delta \ldots t + \tau\}$ $\mathbf{sea} = \{t - \delta \ldots t + \tau\}$	$\mathbf{gwl_L} = \{t + 1 \ldots t + \tau\}$

Preprocessing of forecast data, which is retrieved at an hourly time step, consists of two steps (Figure 3). First, the time series of HRRR rainfall data, which is a gridded product over the continental United States, has to be extracted for the coordinates of each well. Second, the forecast data have to

be inserted into the correct locations in the input tensor. Specifically, the observed rainfall and sea level data in columns (t + 1) to (t + τ) has to be replaced with the corresponding forecast data. This creates a dataset D_{fcst} that includes both observed and forecast data as specified in Figure 3. The same normalization from 0–1 used for the observed data was applied to the forecast data.

2.3.2. Input Variable Cross-Correlation Analysis

Parsing the relationships between a rainfall or sea level observation and the corresponding groundwater table response is a crucial component of input data preprocessing. This response time is called the lag δ and can be separated into two components: δ_R between rainfall and groundwater table response and δ_S between sea level and groundwater table response. The appropriate δ_R and δ_S, in hours, for each well was approximated by a cross correlation analysis [25]. This process involves shifting one signal in relation to the other until a rainfall or sea level observation lines up with its corresponding groundwater table response. The highest cross correlation value (CCF) corresponds to the most influential δ_R or δ_S.

2.3.3. Storm Event Response Identification

In order to evaluate the performance of RNN and LSTM models for groundwater table forecasting during storm events, two training datasets were used (Figure 3). The first training set D_{full} represents the continuous time series data and includes both dry and wet days. The second training set D_{storm} consists only of time periods that were identified as storm events. D_{storm} was created through a filtering process using the gradient and peaks of the observed groundwater table values. For any storm event, the starting time of the event was based on locating the local maxima of the gradient of the groundwater table and looking backward in time to the first occurrence of zero gradient. A peak finding algorithm [58] was then used to locate the peak of the groundwater table that occurred after the corresponding starting time; peak values were used as the end point of the storm.

2.3.4. Bootstrapping Datasets

Bootstrapping was used to generate many datasets with characteristics similar to the observed datasets. While bootstrapping is generally done by selecting values at random and combining them into a new dataset, special techniques are needed to preserve the dependence in time series data. In order to bootstrap the D_{full} datasets in a manner appropriate for time series data, circular block bootstrapping with replacement was used [59]. The block size was based on the average storm length found when creating the storm datasets for each well. Because the D_{storm} datasets were already separated into blocks of different time periods, they were bootstrapped by randomly sampling the blocks with replacement. By creating one thousand bootstrap replicates of each dataset, a normal distribution of error can be approximated when the models are trained and tested. The first 70% of each bootstrapped dataset was taken as the training data and the remaining 30% was used as the test set.

2.3.5. Recurrent Neural Networks

RNNs [60] have been specifically designed to capture the structure that is often inherent in time series data. They do this by passing the output, or state, of the hidden layer neurons, which represent what has been learned at the previous time steps, as an additional input to the next time step (Figure 4A). RNN training was done with back-propagation through time (BPTT) [61], or some variant, to adjust network weights based on the error gradient with respect to both the network weights and the previous hidden states. Because gradients can change exponentially during this process, they tend to either vanish or explode. In this study, a fully connected RNN [62] was used and the output was calculated by stacking a fully connected layer on top of the RNN cell. The product of the output layer is the groundwater table level for the forecast horizon τ. The RNN calculations can be written as:

$$h_t = \tan h(Wx_t + Uh_{t-1} + b) \quad (1)$$

$$y_t = Vh_t + b \tag{2}$$

where h_t is the hidden state, y_t is the output, and x_t is the input vector. The input, hidden, and output weights are represented by W, U, and V, respectively, and b is the bias. The hyperbolic tangent activation function is noted as tanh.

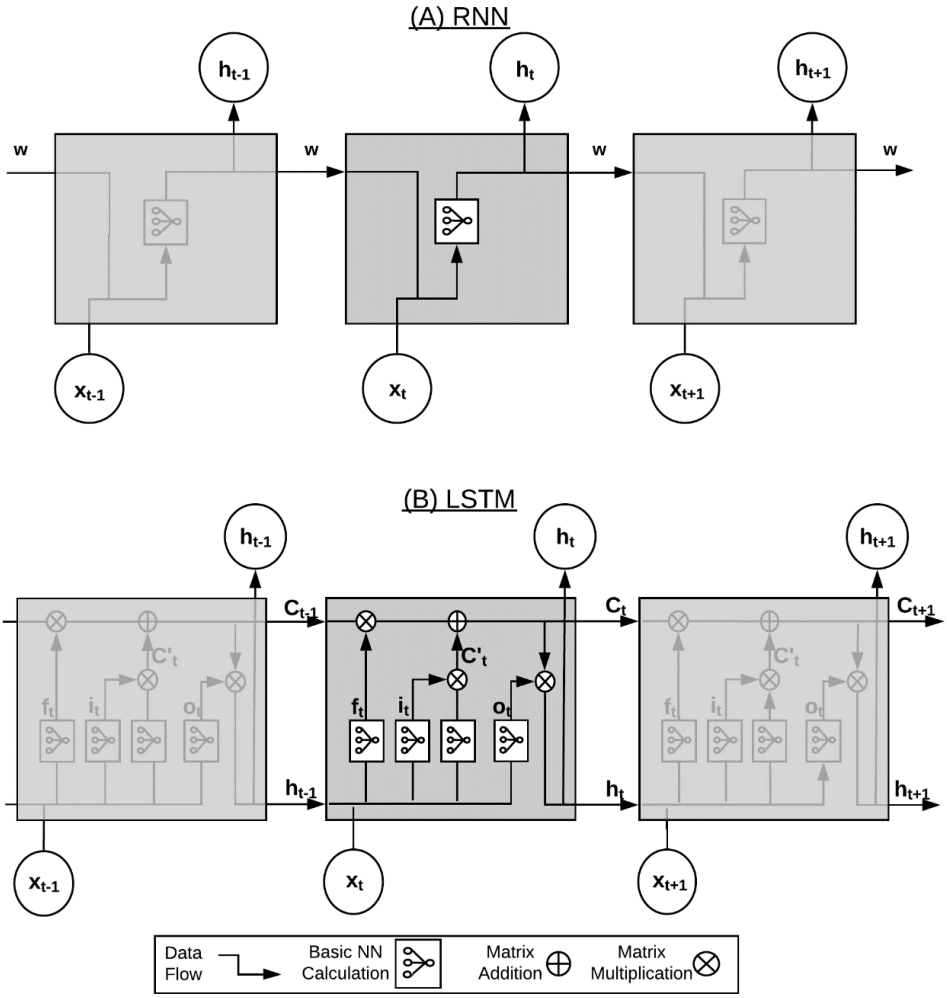

Figure 4. Recurrent neural network (RNN) (**A**) and long short-term memory (LSTM) (**B**) model architectures. Merging lines show concatenation and splitting lines represent copies of matrices being sent to different locations.

2.3.6. Long Short-term Memory Neural Networks

LSTM neural networks are a type of RNN that were developed to overcome the vanishing and exploding gradient obstacles of traditional RNNs [38]. The LSTM architecture (Figure 4B) minimizes gradient problems by enforcing constant error flow between hidden cell states, without passing through an activation function. In addition to this constant error path, an LSTM cell contains three multiplicative units known as gates: The forget gate, the input gate, and the output gate. Because each gate acts as a neuron, it can learn what inputs and cell states are important for predicting the output through the process of passing inputs forward, back propagating the error, and adjusting the weights. The processes within the LSTM cell can be represented with the following equations:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f) \tag{3}$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i) \tag{4}$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o) \quad (5)$$

$$C'_t = \tan h(W_c x_t + U_c h_{t-1} + b_c) \quad (6)$$

$$C_t = f_t{}^{\circ} C_{t-1} + i_t{}^{\circ} C'_t \quad (7)$$

$$h_t = \tan h(C_t)^{\circ} o_t \quad (8)$$

$$yt = Vh_t + b \quad (9)$$

where f_t, i_t, and o_t represent the forget, input, and output gates, respectively. The new cell state candidate values and updated cell state are represented by C'_t and C_t, respectively. Element-wise multiplication of vectors is represented by ° and the sigmoid activation function is noted as σ.

While studies have experimented with different gate configurations, significant improvements over the standard configuration were not found [63]. This study uses LSTM cells with three gates [62]. The network output was calculated by stacking a fully connected layer on top of the LSTM cell. The product of the output layer is the groundwater table level for forecast horizon τ.

2.3.7. Hyperparameter Tuning

Hyperparameter tuning has traditionally been done in an ad-hoc manner through manual trial and error or random search [22,24,25]. This type of tuning can be efficient, but is hard to reproduce or compare fairly [64]; with the increasing complexity of network architectures, more formal methods of hyperparameter optimization are also emerging. In this study, tuning was accomplished for each model type and for each well using a sequential model-based optimization (SMBO) search with the tree-structured Parzen estimator (TPE) algorithm, a Bayesian optimization approach [65]. Given the search history of parameter values and model loss, TPE suggests hyperparameter values for the next trial which are expected to improve the model loss (reduce root mean squared error (RMSE), in this case). As the number of trials increases, the search history grows and the hyperparameter values chosen become better.

The Hyperas library [66] implements the SMBO/TPE technique and was used in this study to advance what has been done in previous research. For example, when comparing four types of neural networks, Zhang et al. [67] simply stated that a trial and error procedure was used to select the best network architecture. When predicting groundwater levels, Zhang et al. [46] presented results for a trial and error optimization of LSTM hyperparameters, but then state that the same hyperparameters were used for the much simpler architecture of FFNN models. By not optimizing the hyperparameters of the FFNN it is more difficult to draw comparisons with the LSTM. Optimizing the hyperparameters of both the LSTM and RNN models in this study allowed each model the best chance to perform well.

The hyperparameters tuned for each model in this study were the number of neurons, the activation function, the optimization function, the learning rate, and the dropout rate (Table 4). The number of neurons influences the model's ability to fit a complex function. The dropout rate helps prevent overfitting by randomly dropping some connections between neurons during training [68]. A minimum value of 10% ensures some dropout is used, as the natural tendency would be for models to not have any connections dropped during training. The combination of hyperparameters for each model type that resulted in the lowest RMSE, based on 100 trials, was used in the final models.

Table 4. Hyperparameter choices explored.

Hyperparameter	Type	Options Explored
Number of Neurons	Choice	10, 15, 20, 40, 50, 75
Activation Function	Choice	Rectified Linear Unit (relu), Hyperbolic tangent (tanh), Sigmoid
Optimization Function	Choice	Adam, Stochastic Gradient Descent (SGD), Root Mean Square Propagation (RMSProp)
Learning Rate	Choice	1×10^{-3}, 1×10^{-2}, 1×10^{-1}
Dropout Rate	Continuous	0.1–0.5

2.3.8. Model Training and Evaluation

All the models for this study were built with the Keras deep learning library for Python [62] using the Tensorflow backend [69]. Model training was carried out on the Rivanna HPC at the University of Virginia using either one NVIDIA Tesla K80 or P100 graphical processing unit (GPU), depending on which was available at the time of execution (Figure 3). RNN and LSTM models were trained for each well using each of the one thousand bootstrap datasets for both the D_{full} and the D_{storm} datasets (Figure 5). At each time step, models were fed input data and output a vector of forecast groundwater table levels, as shown in Table 3. During training, the models sought to minimize the cost function, which is the RMSE between predicted and observed values, by iteratively adjusting the network weights. After training, the D_{full} models were tested on the D_{full}, D_{storm}, and D_{fcst} test sets. Likewise, the D_{storm} models were tested on the D_{storm} and D_{fcst} test sets.

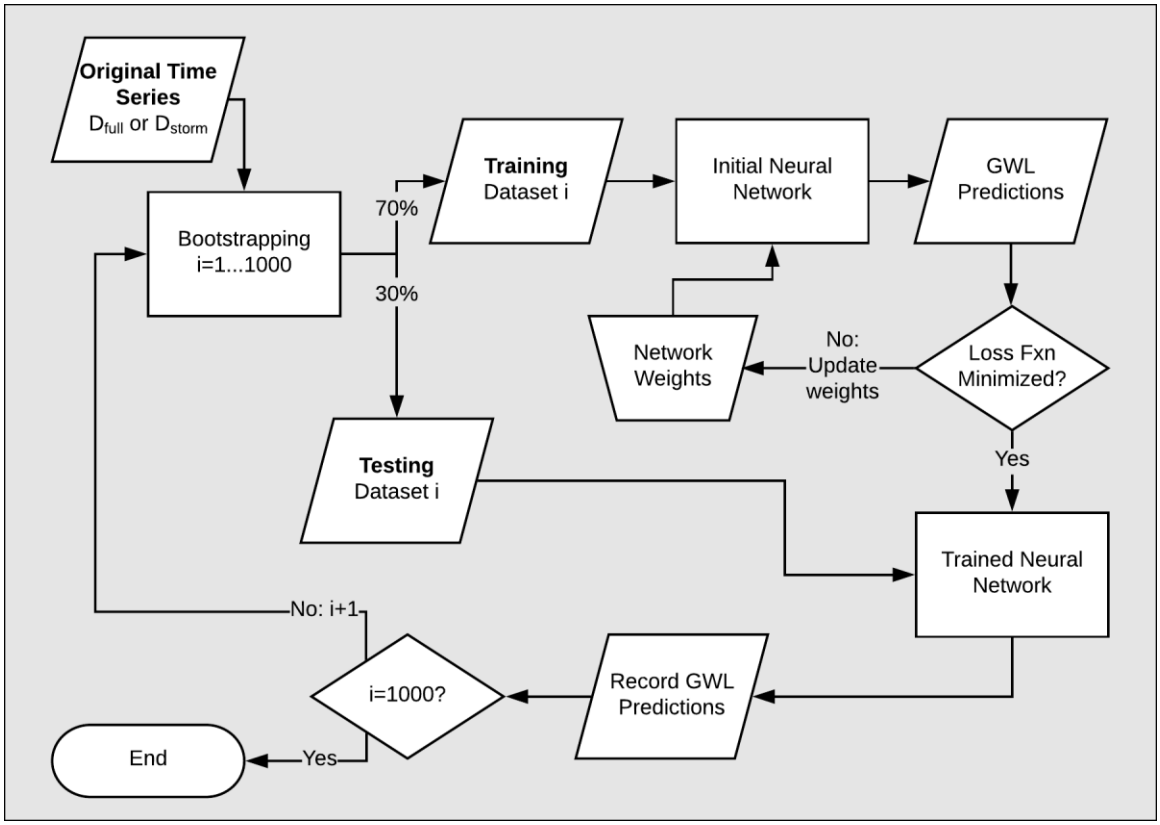

Figure 5. Model training and evaluation with bootstrapped datasets.

Besides being the training cost function, RMSE was also the main metric used for model evaluation. Additionally, the mean absolute error (MAE) was also calculated. Values approaching zero are preferred for both metrics. Both RMSE and MAE were calculated by comparing the predicted water table level (18 predictions at each time step) to the observed values for the corresponding time periods.

To help prevent overfitting and increase the ability of models to generalize, early stopping was used in addition to dropout. Early stopping ends the training process once the cost function has failed to decrease by a threshold value after 5 epochs.

2.3.9. Results Post-Processing

Results post-processing consisted mainly of aggregating model predictions and RMSE values, performing *t*-tests for model comparison, and visualization (Figure 3). Before these actions, however, all predicted values were post-processed to cap predicted groundwater table levels at the land surface elevation for each well.

A number of hypotheses were formulated to test the effects of model type and training dataset on forecast accuracy (Table 5). For example, it was hypothesized that LSTM models would have a lower mean RMSE than RNN models when trained and tested with the D_{full} dataset (Table 5, Comparison ID A). The hypotheses were evaluated using *t*-tests to evaluate whether or not there was a statistically significant difference between the mean of the 1000 RMSEs between two models [70]. In order to reject a null hypothesis that the two models have identical average values, the p-value from the *t*-test would need to be significant (less than 0.01).

Table 5. *t*-test null hypotheses for model type and training data comparison.

Comparison ID	Null Hypothesis	Testing Data
A	RMSE(LSTM, D_{full}) = RMSE(RNN, D_{full})	D_{full}
B	RMSE(LSTM, D_{storm}) = RMSE(RNN, D_{storm})	D_{storm}
C	RMSE(RNN, D_{storm}) = RMSE(RNN, D_{full})	
D	RMSE(LSTM, D_{storm}) = RMSE(LSTM, D_{full})	
E	RMSE(LSTM, D_{full}) = RMSE(RNN, D_{full})	D_{fcst}
F	RMSE(LSTM, D_{storm}) = RMSE(RNN, D_{storm})	
G	RMSE(RNN, D_{storm}) = RMSE(RNN, D_{full})	
H	RMSE(LSTM, D_{storm}) = RMSE(LSTM, D_{full})	

3. Results

The results of this study are divided into two subsections. The first subsection, data preprocessing results, describes the findings of the cross correlation analysis, the storm event identification, and the hyperparameter tuning for each well and model type. The second subsection, model results, describes the model performance and the statistical evaluation of differences between models and training data types. This subsection concludes with a visualization of model predictions.

3.1. Data Preprocessing Results

3.1.1. Input Variable Cross-Correlation Analysis

Using cross correlation analysis, appropriate median lags δ for the entire period of record were found for each well (Table 6). Rainfall lags δ_R were generally expected to increase with a greater distance between the land surface and the groundwater table. It was found δ_R did increase with greater depth to the groundwater table when GW2 and GW3 were compared. At GW2, δ_R was 26 h and the mean groundwater table depth was 0.61 m (Table 1) while at GW3 δ_R was 59 h and the mean groundwater table depth was 2.32 m. At the other wells, however, this relationship did not hold. For example, GW1 had the same δ_R as GW2, but the mean groundwater table depth was very similar to that of GW3 (2.31 m). Other characteristics that influence infiltration rate, such as vertical hydraulic conductivity, porosity, impermeable surfaces, or the configuration of the stormwater system appear to have had a large effect on δ_R at these wells. In addition, sea level may also be influencing groundwater table levels at some or all of these wells.

Table 6. Rainfall δ_R and sea level δ_S lags found for each well.

Well ID	δ_R (h)	δ_S (h)
GW1	26	19
GW2	26	18
GW3	59	–
GW4	25	17
GW5	28	–
GW6	48	–
GW7	58	51

The impact of sea level lags δ_S on the groundwater table was more difficult to determine than rainfall lags δ_R, indicating that sea level does not have as much impact on certain wells; there did not seem to be clear correlations for GW3, GW5, or GW6. It was expected that the impact of sea level would decrease with greater distance between a given well and the closest tidal waterbody influencing it. However, this did not seem to have a strong relationship. GW4, for example, was the farthest well from a tidal water body but had the shortest δ_S, suggesting that tidal water may have a more direct route to this location. While a strong correlation between sea level and groundwater table was not found for three wells, it was deemed that sea level could still be an important input variable for models at those wells because of their proximity to tidal water bodies [71,72]. In order to keep the data preprocessing consistent, and because δ_S values could not be found for all wells and the δ_S values found were always shorter than δ_R values, δ_R was taken as the lag value for all input variables.

3.1.2. Storm Event Response Identification

The storm identification process produced a unique dataset and a different average storm duration and total number of events for each well (Table 7). Average storm duration, the average length in hours of the identified periods, was used as the block size for bootstrapping the D_{full} datasets. The storm events identified for each well also accounted for the majority of total rainfall, indicating that the method is capturing large rainfall events. Storm surge is also being captured at most wells as shown by the positive increase in mean sea level for the storm events compared to the D_{full} datasets (Table 7). Figure 6 shows an example of storms found with this process; large responses of the groundwater table are captured, but smaller responses are excluded.

Table 7. Storm characteristics for each well.

Well ID	Average Storm Duration (h)	Number of Events	% of Total Rain	% Increase in Mean Sea Level over D_{full}
GW1	83	239	75	27
GW2	82	307	85	36
GW3	137	155	57	18
GW4	89	254	67	18
GW5	91	149	60	64
GW6	120	295	60	0
GW7	132	166	63	0

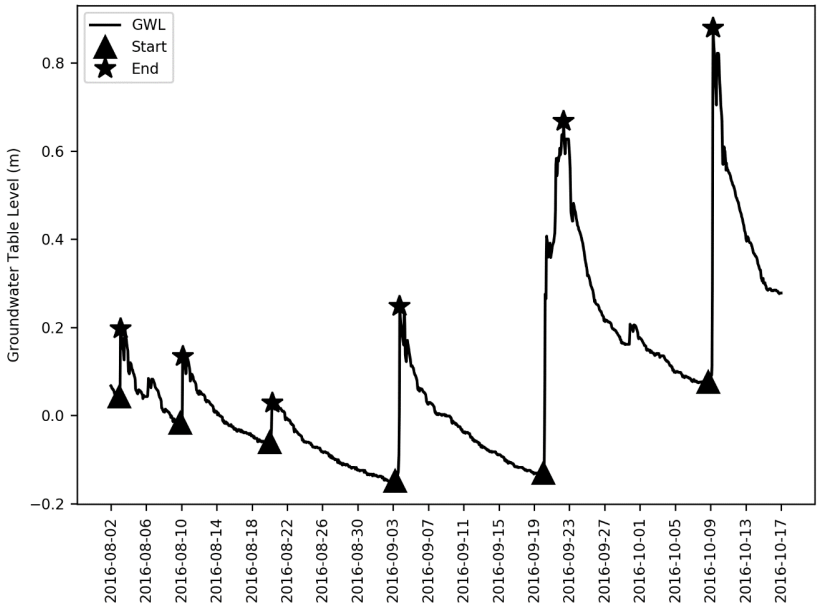

Figure 6. Detail of identified storm periods found for well GW1.

3.1.3. Hyperparameter Tuning

Tuned hyperparameters were generally consistent across wells and model types (Tables 8 and 9). Dropout rates ranged from just above the minimum of 0.1 to a high of 0.355. The preferred activation function was the hyperbolic tangent, except for the GW5 RNN. In all cases the Adam optimization function performed the best with its recommended learning rate of 10^{-3} [73]. The largest number of neurons possible (75) was used in five of the seven RNN (Table 8) and LSTM (Table 9) models. The other models of each type used a midrange number of neurons (40 or 50).

Table 8. Tuned hyperparameters for RNN models.

Well	Dropout Rate	Activation Function	Optimization Function	Learning Rate	Neurons
GW1	0.126	tanh	adam	10^{-3}	40
GW2	0.340	tanh	adam	10^{-3}	75
GW3	0.320	tanh	adam	10^{-3}	75
GW4	0.111	tanh	adam	10^{-3}	75
GW5	0.127	relu	adam	10^{-3}	75
GW6	0.145	tanh	adam	10^{-3}	75
GW7	0.104	tanh	adam	10^{-3}	40

Table 9. Tuned hyperparameters for LSTM models.

Well	Dropout Rate	Activation Function	Optimization Function	Learning Rate	Neurons
GW1	0.355	tanh	adam	10^{-3}	75
GW2	0.106	tanh	adam	10^{-3}	40
GW3	0.166	tanh	adam	10^{-3}	75
GW4	0.102	tanh	adam	10^{-3}	75
GW5	0.103	tanh	adam	10^{-3}	50
GW6	0.251	tanh	adam	10^{-3}	75
GW7	0.177	tanh	adam	10^{-3}	75

3.2. Model Results

3.2.1. Network and Training Data Type Comparison

The results in this subsection address hypotheses A–D (Table 5), which compare performance of the two model types trained using the two different datasets. All of these comparisons had significant p-values (<0.001). This shows that the null hypotheses that two models have identical average values was rejected and there are significant differences in RMSE for different model types and training datasets. The distributions of RMSE values for all bootstrap models in this subsection is available in Appendix A; corresponding MAE values are available in Appendix C.

When trained with either D_{full} or D_{storm}, LSTM models have lower mean RMSE values than RNN models (Figure 7A,B), as hypothesized (Table 5, A and B). LSTM models trained and tested with D_{full} had average RMSE values that were lower than RNN models by 49%, 38%, and 18% for the t + 1, t + 9, and t + 18 predictions, respectively. LSTM's advantage over RNN decreased as the prediction horizon increased. Similarly, LSTM models trained and tested with D_{storm} had lower average RMSE values than RNN models by 50%, 55%, and 36% for the t + 1, t + 9, and t + 18 predictions when tested on D_{storm}, respectively.

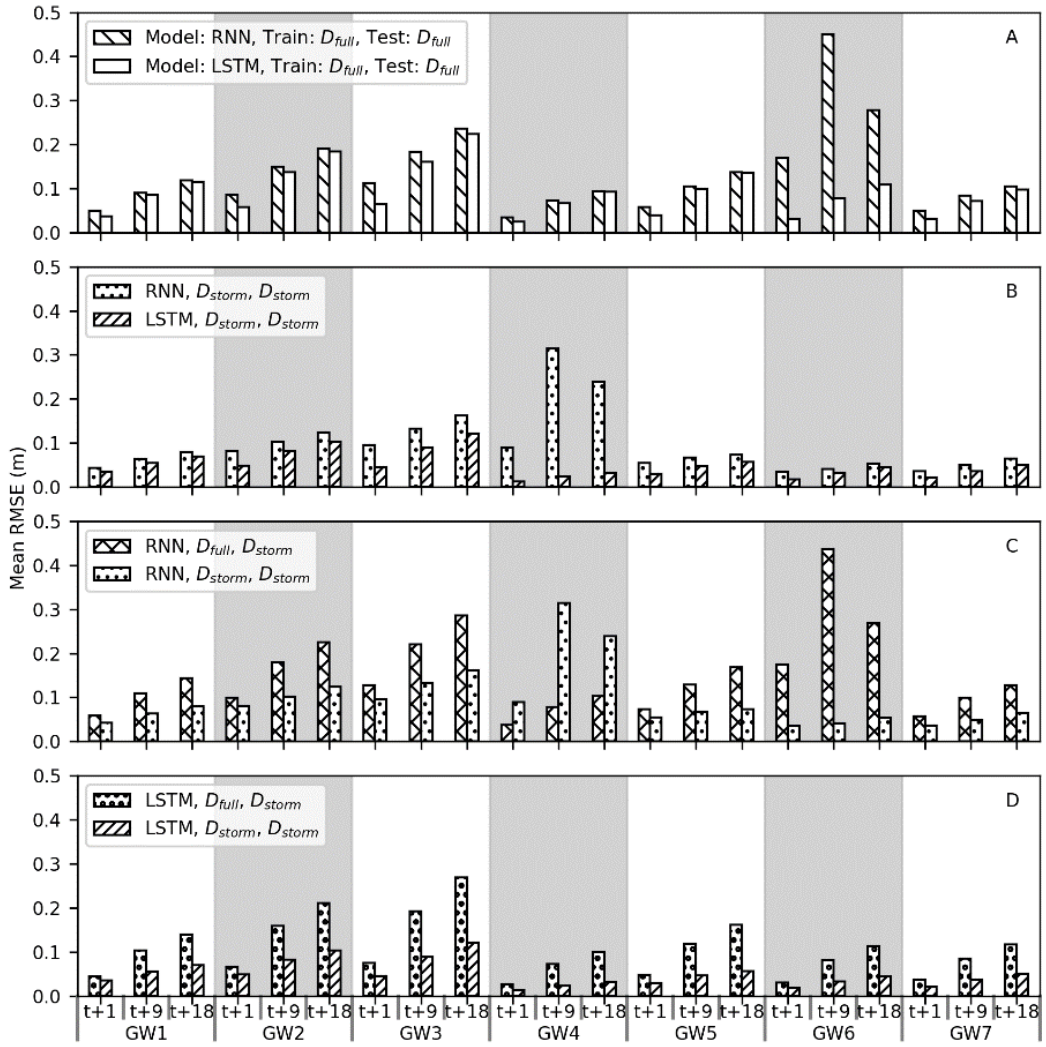

Figure 7. Mean root mean squared error (RMSE) values for each model type and training dataset treatment at each well and forecast period. Subplot letters correspond to the hypothesis being tested (Table 5) and are comparisons of (**A**) RNN and LSTM models trained and tested with D_{full} (**B**) RNN and LSTM models trained and tested with D_{storm} (**C**) RNN models trained with either D_{full} or D_{storm} and tested on D_{storm} (**D**) LSTM models trained with either D_{full} or D_{storm} and tested on D_{storm}.

When tested with D_{storm}, the models trained with D_{storm} outperformed the models trained with D_{full} (Figure 7C,D), with the exception of the RNN for GW4. In this scenario, the models trained with D_{storm} had RMSE values that were lower than models trained with D_{full} by an average of 33%, 39%, and 42% for the RNN models and by an average of 40%, 58%, and 56% for the LSTM models for the t + 1, t + 9, and t + 18 predictions, respectively. The improvement in performance when using D_{storm} as opposed to D_{full}, increased with the prediction horizon. While this was true for both model types, the performance improvement for LSTM was greater than for the RNN.

In most cases the model error increased as the prediction horizon increased. This held for all of the LSTM models, but not with the RNN at GW4 and GW6 for certain datasets. For example, the RNN trained and tested on D_{storm} (Figure 7B,C) had a larger RMSE for the t + 9 prediction than the t + 18 prediction. This pattern is the same for the GW6 RNN (Figure 7A,C) and may have been caused by some combination of hyperparameters and/or some unknown error in the dataset. Causes of individual errors in these types of models, however, are very difficult to pinpoint [25].

3.2.2. Real-Time Forecast Scenario

By training and testing models with observed data, comparisons can be made between model types and training datasets in terms of performance (as shown in Figure 8). The performance of these models, however, also needs to be evaluated in a real-time scenario that includes forecast conditions of rainfall and sea level level. The mean RMSE values from testing the models and data treatments with the D_{fcst} test set are shown in Figure 8 and correspond to hypotheses E–H (Table 5). The distributions of RMSE values for all bootstrap models in this subsection is available in Appendix B; corresponding MAE values are available in Appendix C.

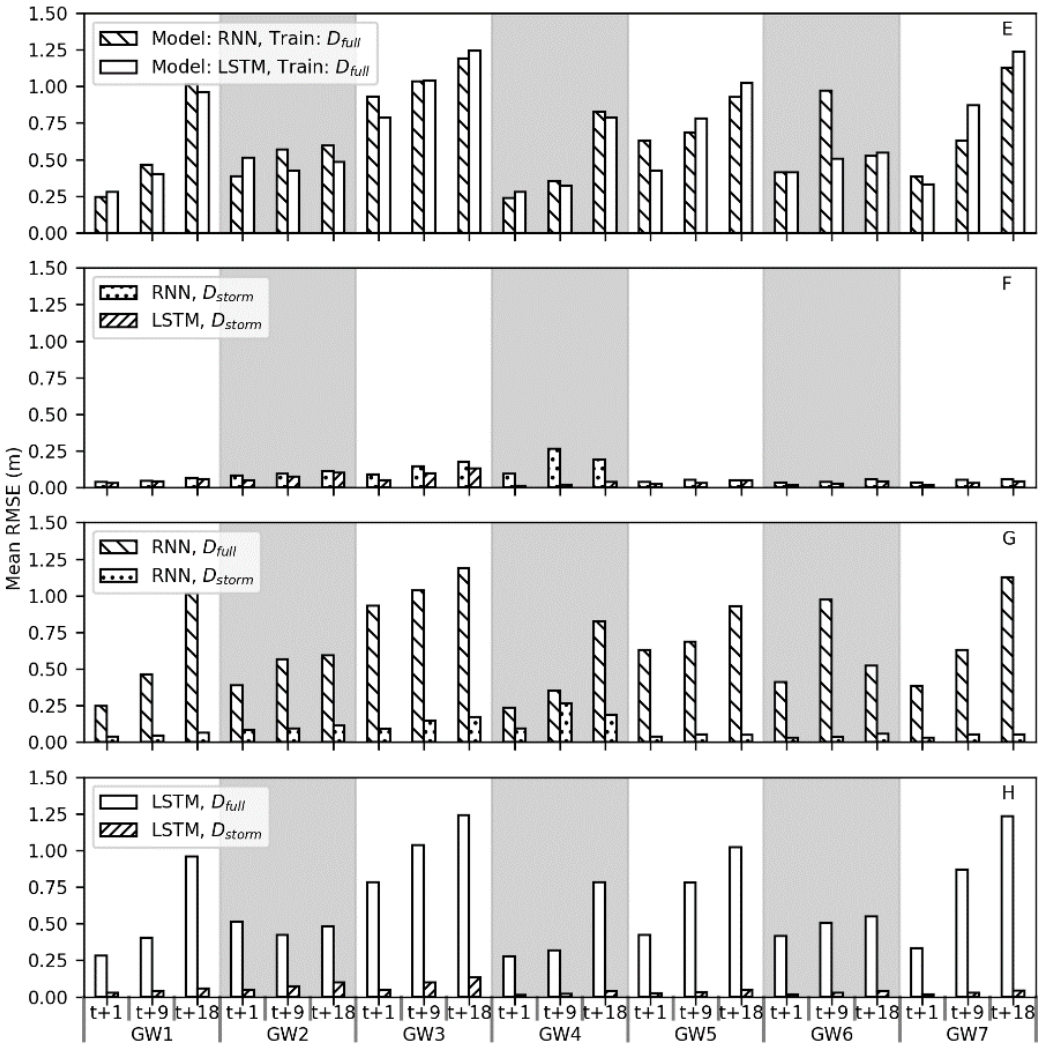

Figure 8. Mean RMSE values from the forecast test set D_{fcst} for each model type and training dataset treatment at each well and forecast period. Subplot letters correspond to the hypothesis being tested (Table 5) and are comparisons of (**E**) RNN and LSTM models trained with D_{full} (**F**) RNN and LSTM models trained with D_{storm} (**G**) RNN models trained with either D_{full} or D_{storm} (**H**) LSTM models trained with either D_{full} or D_{storm}.

In the real-time use simulation, models trained on D_{storm} (Figure 8F–H) performed much better than those trained with D_{full} (Figure 8E), which had RMSE values of up to nearly 1.25 m. In contrast to the difference training data type made, model architecture only made a small difference in performance (Figure 8E,F). All differences seen in Figure 8E were statistically significant at the 0.001 level, except GW3 at t + 9 and GW6 at t + 1 where the results were almost identical. The comparisons in Figure 8F–H all had significant p-values.

Visualizations from the real-time forecasting scenario (Figure 9) complement the aggregate metrics from bootstrap testing of models and training data treatments and demonstrate the response of predicted groundwater table levels to a storm when using D_{fcst} as input data. The forecasts at GW1 are shown in Figure 9 for Tropical Storm Julia, which impacted Norfolk in late September of 2016. The initial rainfall from this storm on the 19th caused the groundwater table to spike early on the 20th. Subsequent rainfall on the 20th, 21st, and 22nd maintained the elevated groundwater table level. The LSTM model trained with D_{full} has greatly increasing error as the forecast horizon grows (Figure 9 t + 1, t + 9, t + 18) and tends to be overly impacted by sea level fluctuations. In contrast, the predicted groundwater table level from the LSTM model trained with D_{storm} has much better agreement with the observed groundwater table levels, even as the forecast horizon increases.

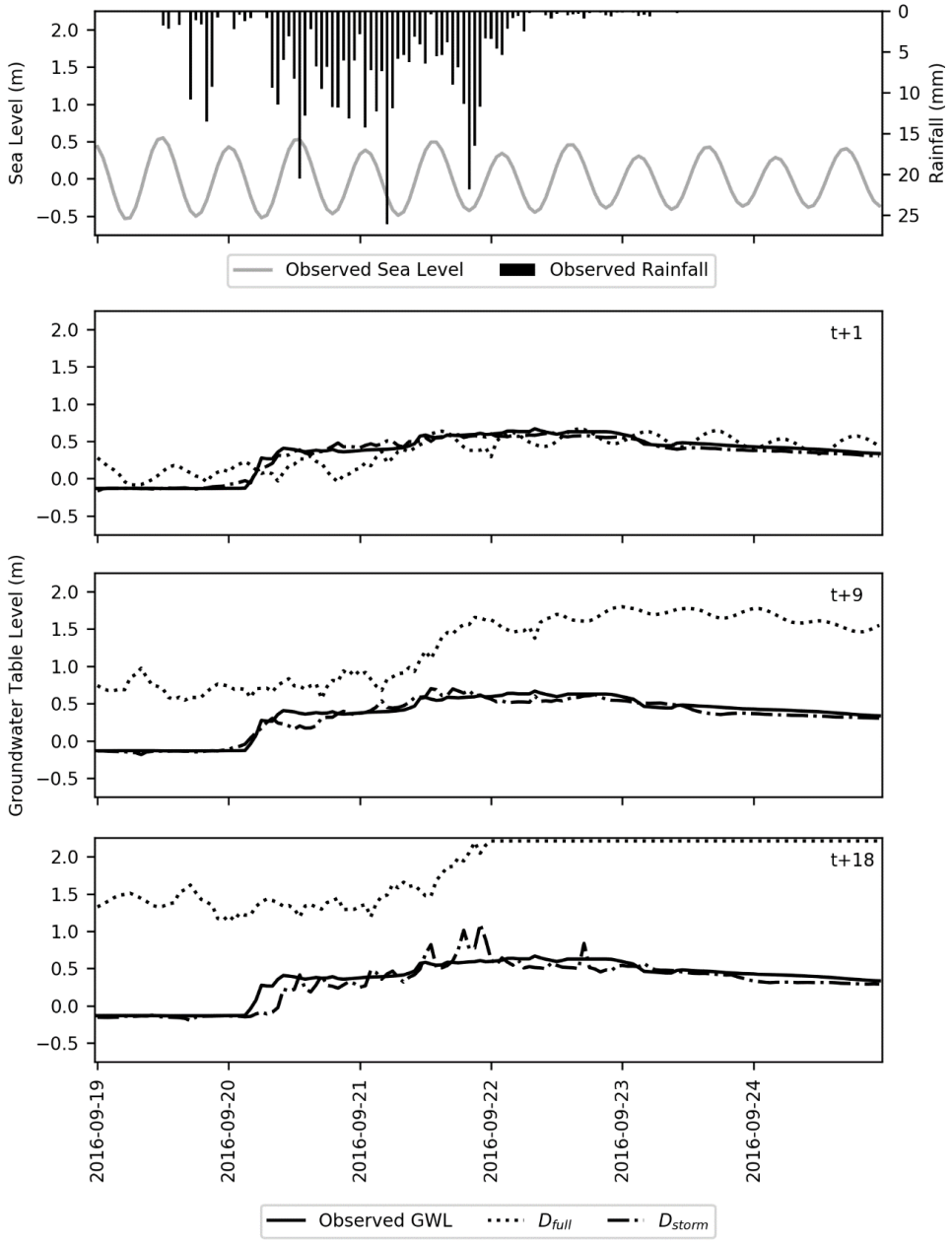

Figure 9. Comparison of groundwater table observations and forecasts at GW1 from LSTM models trained with the D_{full} and D_{storm} training sets.

4. Discussion

The results of hypothesis testing (Table 5) indicate that both model type and the training data influenced the accuracy of groundwater table forecasts. The LSTM architecture was better able to learn the relationships between groundwater table, rainfall, and sea level than the simpler RNN. Additionally, models trained with storm data D_{storm} outperformed models trained with the full dataset D_{full} when tested on either observed for forecast data. In the real-time scenario one reason for this difference in performance could be the structure of the test set D_{fcst}. These results indicate that the structure of the time series data in D_{storm} and D_{fcst} are more closely aligned, as opposed to the time series structure of D_{full} and D_{fcst}. The models trained on D_{full} also have to learn groundwater table response with many observations where no rainfall occurred. In contrast, models trained on D_{storm}, which have a higher proportion of observations with rainfall, may have a clearer pattern to learn.

In the real-time forecasting scenario, both RNN and LSTM models trained with D_{storm} demonstrated predictive skill, forecasting groundwater table levels with low RMSE values (Figure 8F). Models trained with D_{full} however performed much worse because of the noisier signal they had to learn (Figure 9) and are not suitable for use in a real-time forecasting scenario. Across all wells, averaged RMSE values for the RNN models were 0.06 m, 0.1 m, and 0.1 m for the t + 1, t + 9, and t + 18 predictions, respectively. Averaged RMSE values for the LSTMs were slightly lower at 0.03 m, 0.05 m, and 0.07 m for the t+1, t+9, and t+18 predictions, respectively. While there is limited research on the use of LSTMs for forecasting groundwater table, these results are comparable with the work of J. Zhang et al. [46], who reported RMSE values for one-step ahead prediction of monthly groundwater table at six sites ranging from 0.07 m to 0.18 m. The current work makes advances by showing that both LSTM and RNN can accurately forecast groundwater table response to storm events at an hourly time step, with forecast input data, and at longer prediction horizons all of which are necessary in a coastal urban environment.

Because the effect of sea level on the groundwater table is heavily dependent on well location and soil characteristics not included in this study, a sensitivity analysis was performed by removing sea level from the D_{full} and D_{storm} data sets and retraining and retesting the models. Of the wells that were not correlated with sea level, GW3 and GW6 performed better without sea level data. Using RNN models trained with D_{full}, there was an average decrease in RMSE of 12% for GW3 and 41% for GW6. The only exception to this is the GW6 RNN trained with D_{storm} which performed much worse without sea level. For LSTM models trained with D_{full} however, there was only a 3% decrease in RMSE for GW3 and a 2% decrease for GW6. The third well that was not correlated with sea level, GW5, was worse without sea level for the RNN trained with D_{full}; the average increase in RMSE was 17%. Removing sea level at this well had no change in RMSE for the LSTM models trained with D_{full}. This particular well is only 32 m from the coast so the influence of sea level seems reasonable. When models were trained with D_{storm} excluding sea level, across all well there was an average increase in RMSE of 8% for RNN models and no change for LSTM models. This demonstrates that sea level data is important for groundwater table prediction during storms for wells close to the coast and this is captured effectively by the D_{storm} datasets (Table 7). This analysis indicated that RNN models were much more sensitive to the inputs used than LSTM models. As designed, the structure of LSTM models allowed them to filter out noisy data and have little to no change in RMSE if sea level data was removed, especially when using the best performing combination of LSTM and D_{storm} training data.

The results of this study illustrate the trade-off between model complexity and performance that has implications beyond creating forecasts. The increased complexity of LSTM models, in terms of gates that learn and the constant error pathway, allowed them to have more predictive skill than the RNN models for forecasting groundwater table response to storm events. Additionally, the structure of LSTM models allowed them to filter out noise from the sea level signal which RNN struggled to do. Most of the comparisons presented in the Results had significant p-values; because of the large sample size (1000) however, even a very small difference in RMSE values between two models was considered significant. For example, the differences between LSTM and RNN models trained with D_{storm} in the

real-time forecasting scenario were statistically significant (Figure 8F). The average difference in the RNN and LSTM RMSE values, however, was only 0.03 m, 0.05 m, and 0.03 m for the t + 1, t + 9, and t + 18 predictions, respectively. If these groundwater table forecasts were to be used as additional input to a rainfall-runoff model to predict flooding, it seems unlikely that the small differences between RNN and LSTM models would have a large impact, especially when compared to other factors like rainfall variability and storm surge timing.

The increased complexity of the LSTM models, while they had better performance than the RNN models, also increased their computational cost. The main difference in computational cost of the LSTM and RNN in this study was the length of training time. When trained on an HPC with either an NVIDIA Tesla K80 or P100 GPU or a smaller NVIDIA Quadro P2000 GPU on a desktop computer, wall-clock training time for LSTM models was approximately three times that of RNN models. Factors in training time include hyperparameters, such as the number of neurons in the hidden layer, which were relatively similar between model types. Once models are trained, groundwater table forecasts are obtained by a forward pass of input data through the network; this time was short and comparable for both models. For this groundwater table forecasting application training time was not a major concern, but if the application was time sensitive and the models were frequently retrained, RNNs could be an appropriate choice that does not sacrifice much in terms of accuracy.

Because forecast data were used as model input in the real-time scenario, it's important to note some of the uncertainties that dataset might introduce. HRRR rainfall data are a product of a numerical forecast model and as such is subject to the uncertainty of that model, which includes the transformation of radar reflectivity data into precipitation amounts [74]. Additionally, the uncertainty of HRRR forecasts will increase the farther into the future they are. NOAA sea level forecasts, as previously mentioned, are based only on the harmonic constituents of the astronomical tide cycle. For rainfall-dominated storm events this type of forecast may be accurate enough as a model input, but any storm surge from hurricanes or nor'easters would not be included. This could result in under prediction of groundwater table levels. While archived storm surge predictions were not available for this study, in a real scenario predictions of storm surge could be incorporated into the model input.

The neural networks and data processing techniques presented in this paper are applicable to other coastal cities facing sea level rise and recurrent flooding. Because there is a lack of groundwater table data in most locations however, the direct transferability of the models created for Norfolk should be explored in other locations were observational data are not available. Even in Norfolk, questions still remain about how much data, both temporally and spatially, is needed to accurately forecast groundwater table levels using the methods presented in this study. In this study, at least eight years of data were available for each well, but other researchers have found acceptable results when training neural networks with more [32,33] and less [2,71] time series data. Based on our sensitivity analysis, rainfall is the most important input for the models. However, sea level data was from a single station; if there were more sea level gauges throughout the city it could provide a more accurate input for these models to learn from. The groundwater table monitoring network in Norfolk consists of only seven wells; while this network is a valuable source of data, it may not be dense enough to accurately represent the groundwater table across the complex urban landscape. The city is divided by many tidal rivers and stormwater conveyances and the effects these features have on the groundwater table maybe highly localized. Areas where groundwater table level is important to flooding are likely not well represented by a distant monitoring well. Research has been done with kriging to determine potential densities of groundwater monitoring [75] and rain gauge networks [76]. A similar approach may be valuable in Norfolk or comparable cities to determine the optimal density of monitoring networks when planning for and adapting to climate change and sea level rise.

5. Conclusions

The objective of this study was to compare two types of neural networks, RNN and LSTM, for their ability to predict groundwater table response to storm events in a coastal environment. The study

area was the city of Norfolk, Virginia where time series data from 2010–2018 were collected from seven shallow groundwater table wells distributed throughout the city. Two sets of observed data, the full continuous time series D_{full} and a dataset of only time periods with storm events D_{storm}, were bootstrapped and used to train and test the models. An additional dataset D_{fcst} including forecasts of rainfall and sea level was used to evaluate model performance in a simulation of real-time model application. Statistical significance in model performance was evaluated with *t*-tests.

Major conclusions from this study, in light of the hypotheses described in Table 4 are:

- Both model type and training data are important factors in creating skilled predictions of hourly groundwater table using observed data:
    - Using D_{full}, LSTM had a lower average RMSE than RNN (0.09 m versus 0.14 m, respectively)
    - Using D_{storm}, LSTM had a lower average RMSE than RNN (0.05 m versus 0.10 m, respectively)
- The best predictive skill was achieved using LSTM models trained with D_{storm} (average RMSE = 0.05 m) versus RNN models trained with D_{storm} (average RMSE = 0.10 m)
- LSTM has better performance than RNN but requires approximately 3 times more time to train
- In a real-time scenario using observed and forecasted input data, accurate forecasts of groundwater table were created with an 18 h horizon:
    - LSTM: Average RMSE values of 0.03, 0.05, and 0.07 m, for the t + 1, t + 9, and t + 18h forecasts, respectively
    - RNN: Average RMSE values of 0.06, 0.10, and 0.10 m, for the t + 1, t + 9, and t + 18h forecasts, respectively

Forecasts of groundwater table levels are not common; in many locations even direct measurements of the groundwater table are not widely available. As sea levels rise and storms become more extreme, however, forecasts of groundwater table will become an increasingly important part of flood modeling. In low-lying coastal areas, sea level rise, stormwater infiltration, and storm surge could cause groundwater inundation. Even if groundwater inundation does not occur, increased duration of high groundwater table levels could have significant impacts on infrastructure. Forecasts of groundwater table, an often overlooked part of coastal urban flooding, can provide valuable information on subsurface storage available for stormwater and help inform infrastructure management and planning.

Supplementary Materials: Model code is available on Github at: https://github.com/UVAdMIST/Norfolk_Groundwater_Model. Data is available on Hydroshare at: http://www.hydroshare.org/resource/813dedd3568b4ef3897202988c14a522.

Author Contributions: Conceptualization, B.D.B., J.M.S., M.M.M. and J.L.G.; Methodology, B.D.B. and M.B.; Software, B.D.B., J.M.S., and M.M.M.; Validation, B.D.B.; Formal Analysis, B.D.B.; Investigation, B.D.B.; Resources, B.D.B.; Data Curation, B.D.B., J.M.S., and M.M.M.; Writing—Original Draft Preparation, B.D.B.; All authors contributed to the final version of the manuscript.

Funding: This work was funded as part of a National Science Foundation grant: Award #1735587 (CRISP—Critical, Resilient Interdependent Infrastructure Systems and Processes).

Acknowledgments: The authors thank the Hampton Roads Sanitation District for access to their data and the Advanced Research Computing Services at the University of Virginia for HPC assistance.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

Appendix A

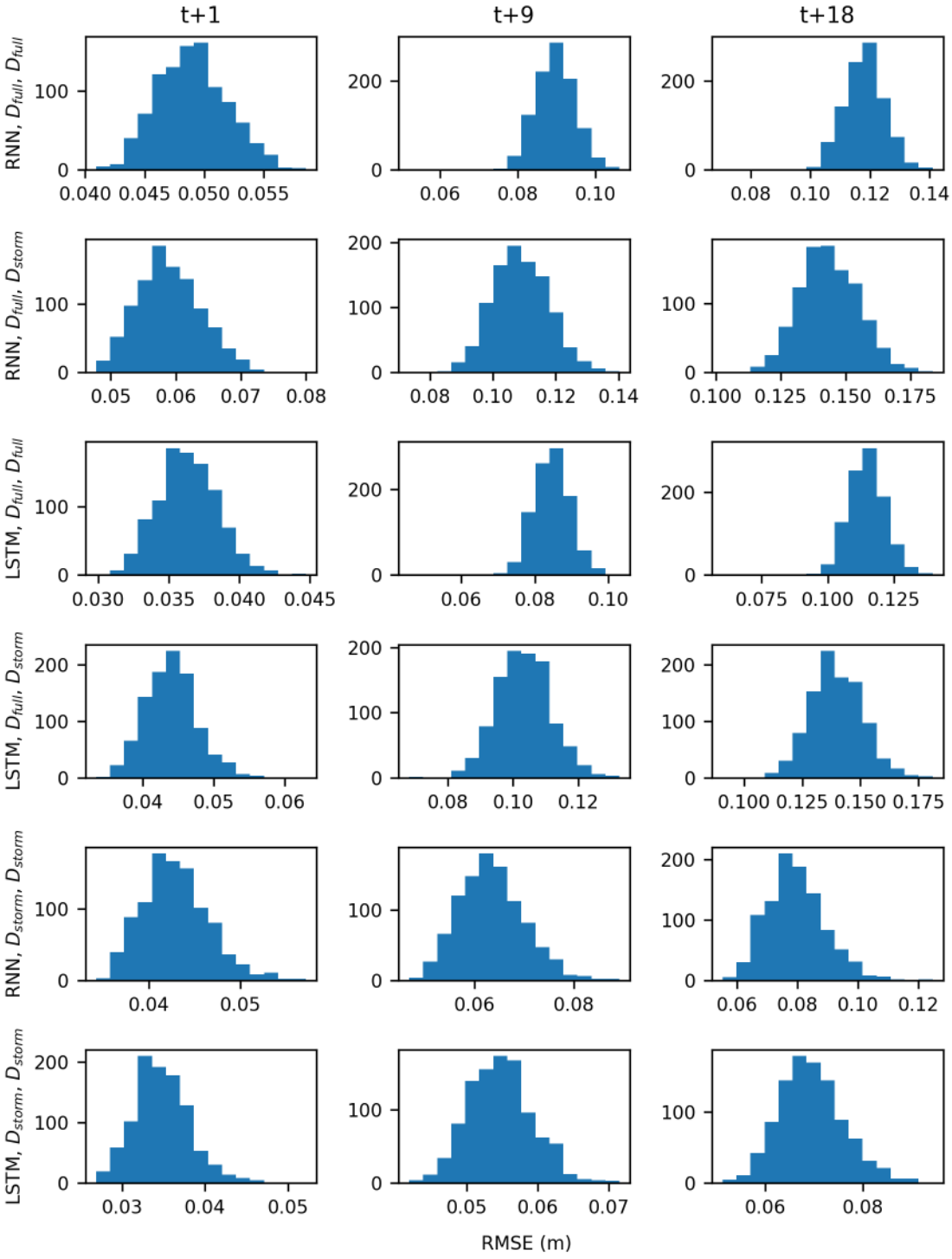

Figure A1. RMSE distributions for GW1 using observed data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

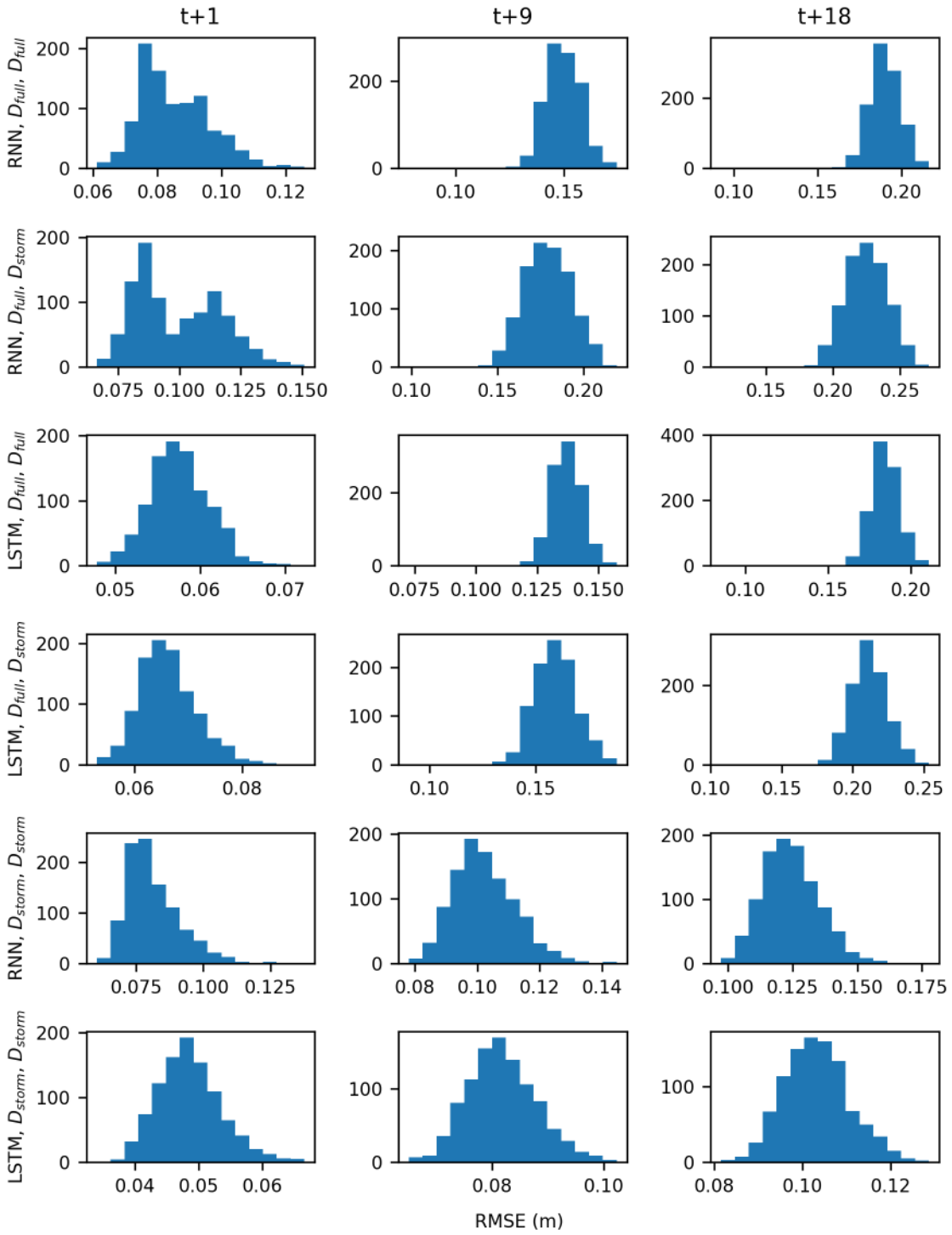

Figure A2. RMSE distributions for GW2 using observed data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

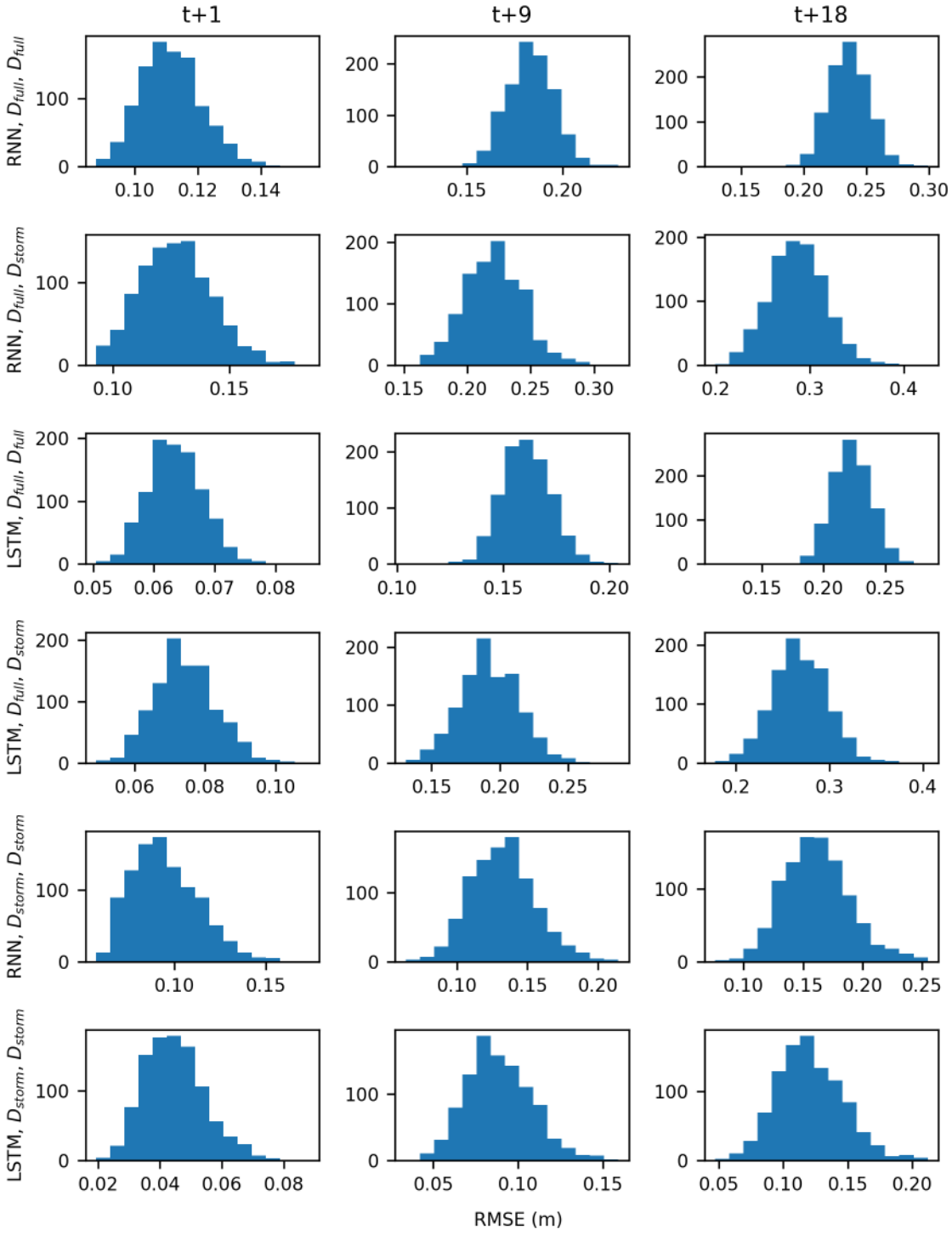

Figure A3. RMSE distributions for GW3 using observed data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

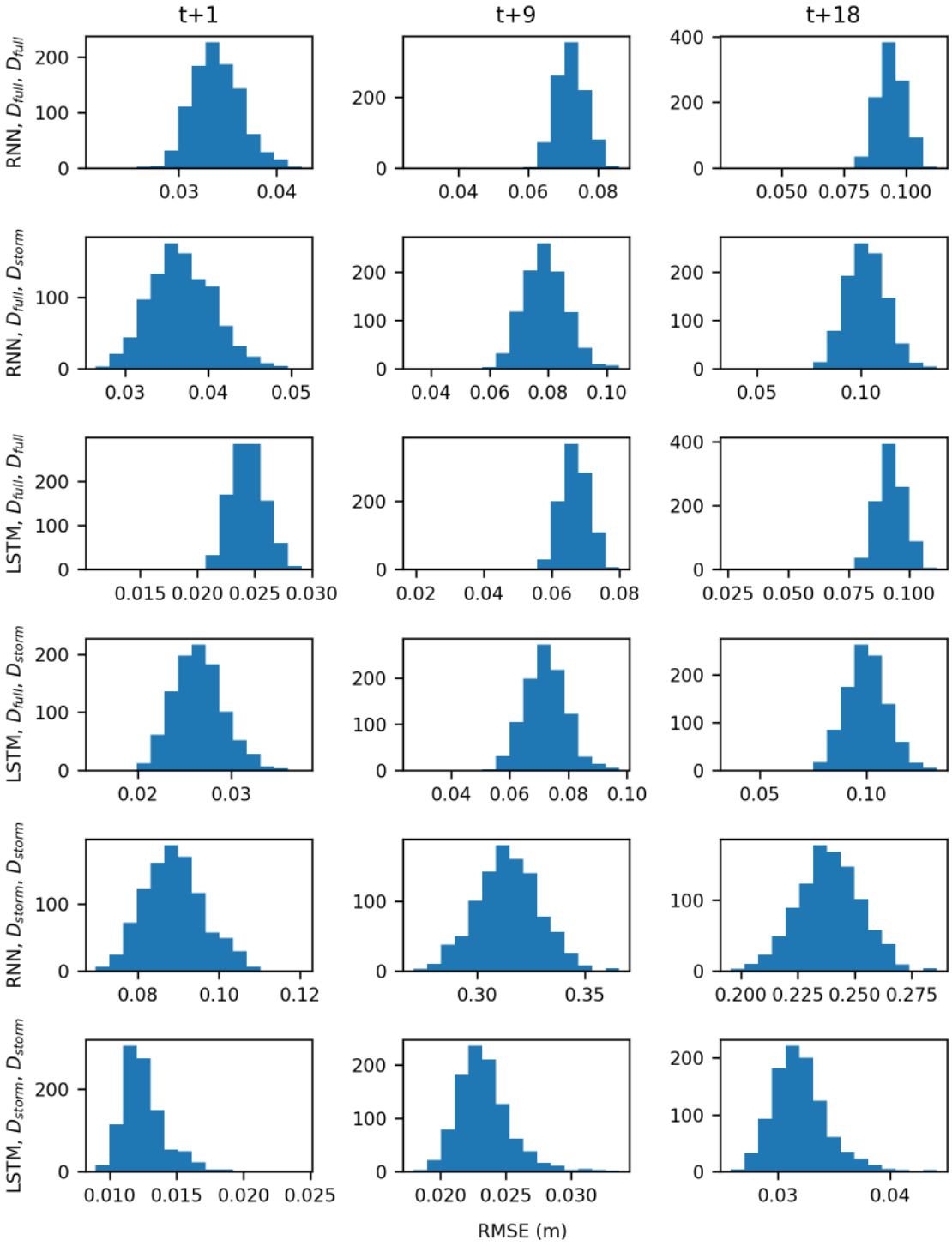

Figure A4. RMSE distributions for GW4 using observed data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

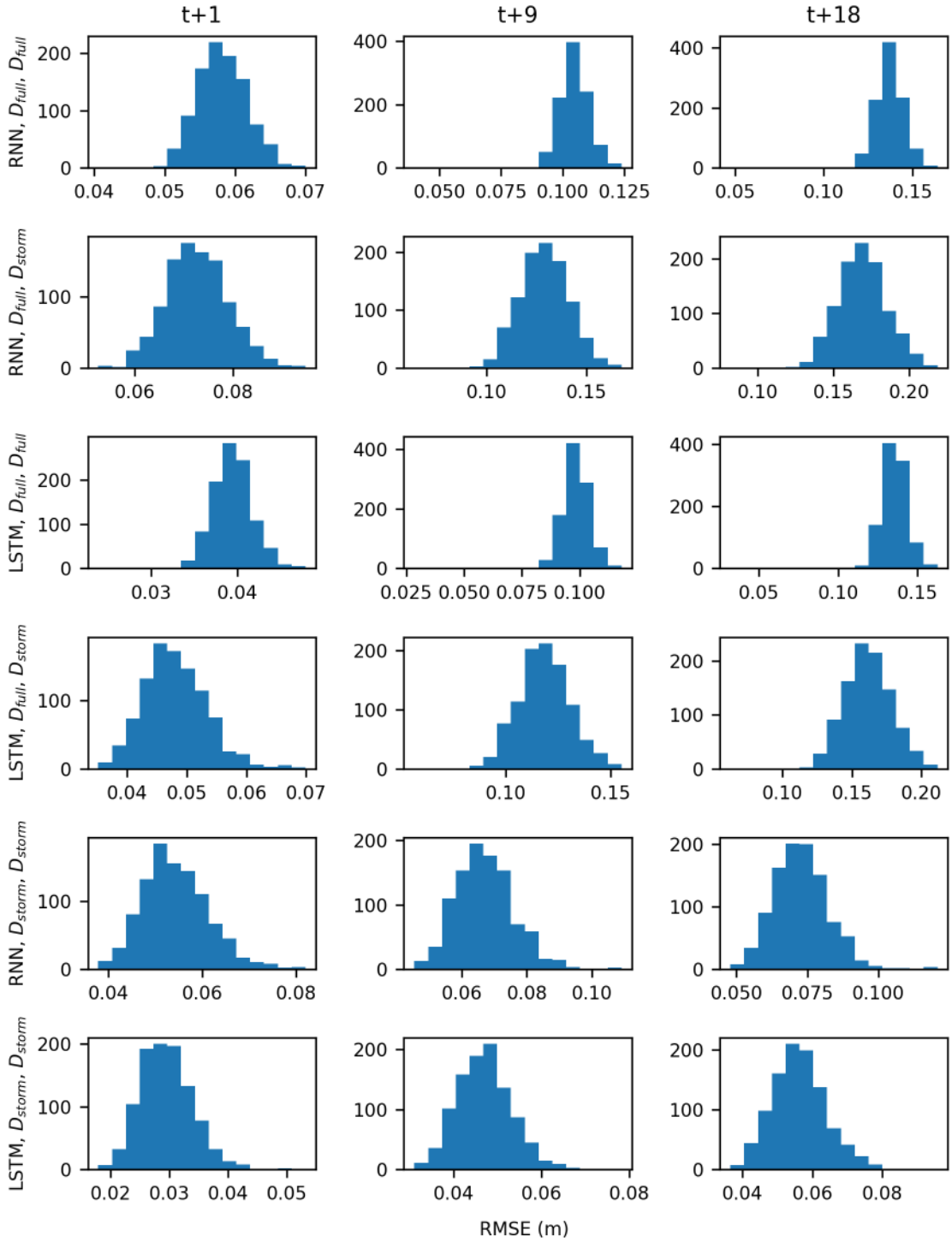

Figure A5. RMSE distributions for GW5 using observed data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

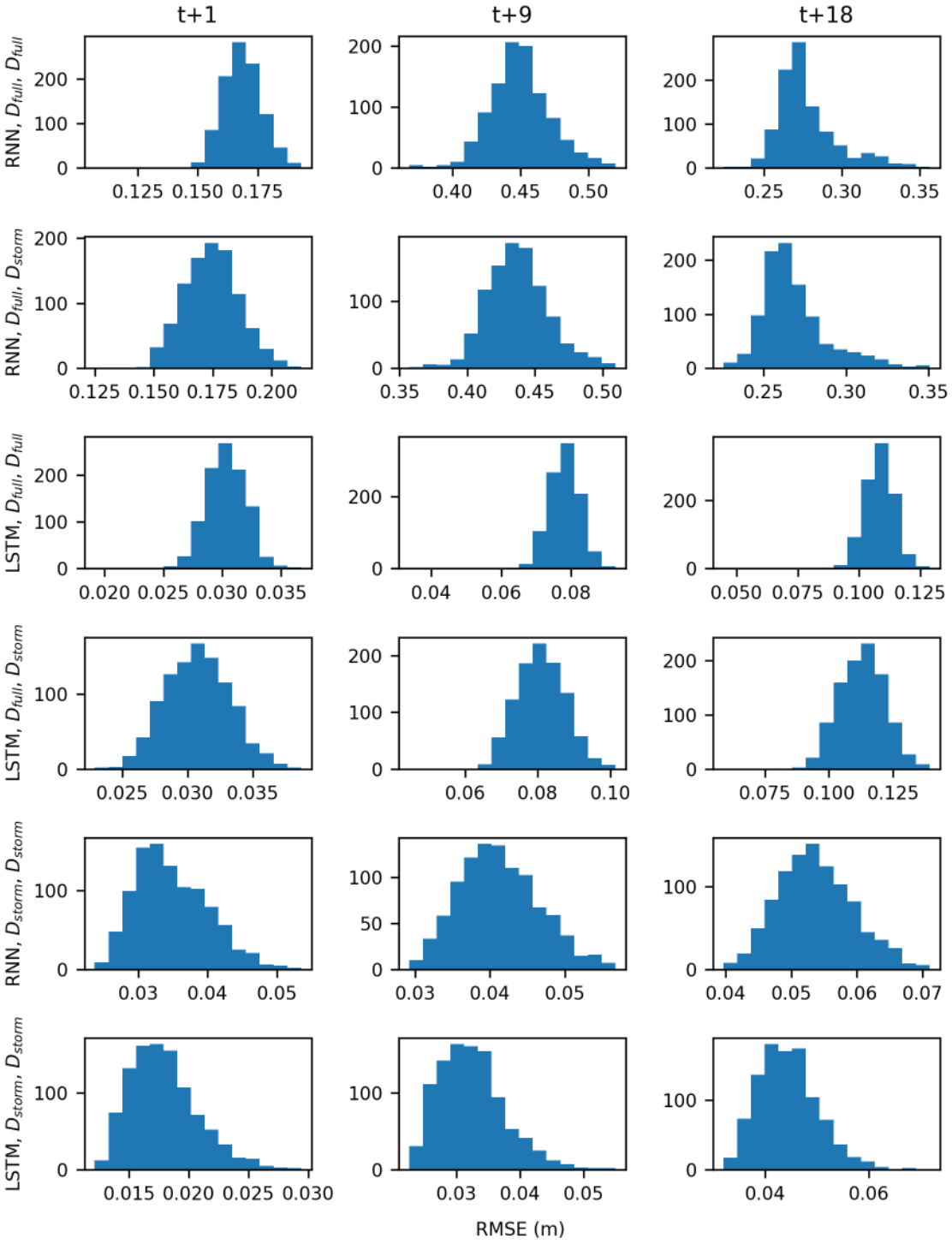

Figure A6. RMSE distributions for GW6 using observed data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

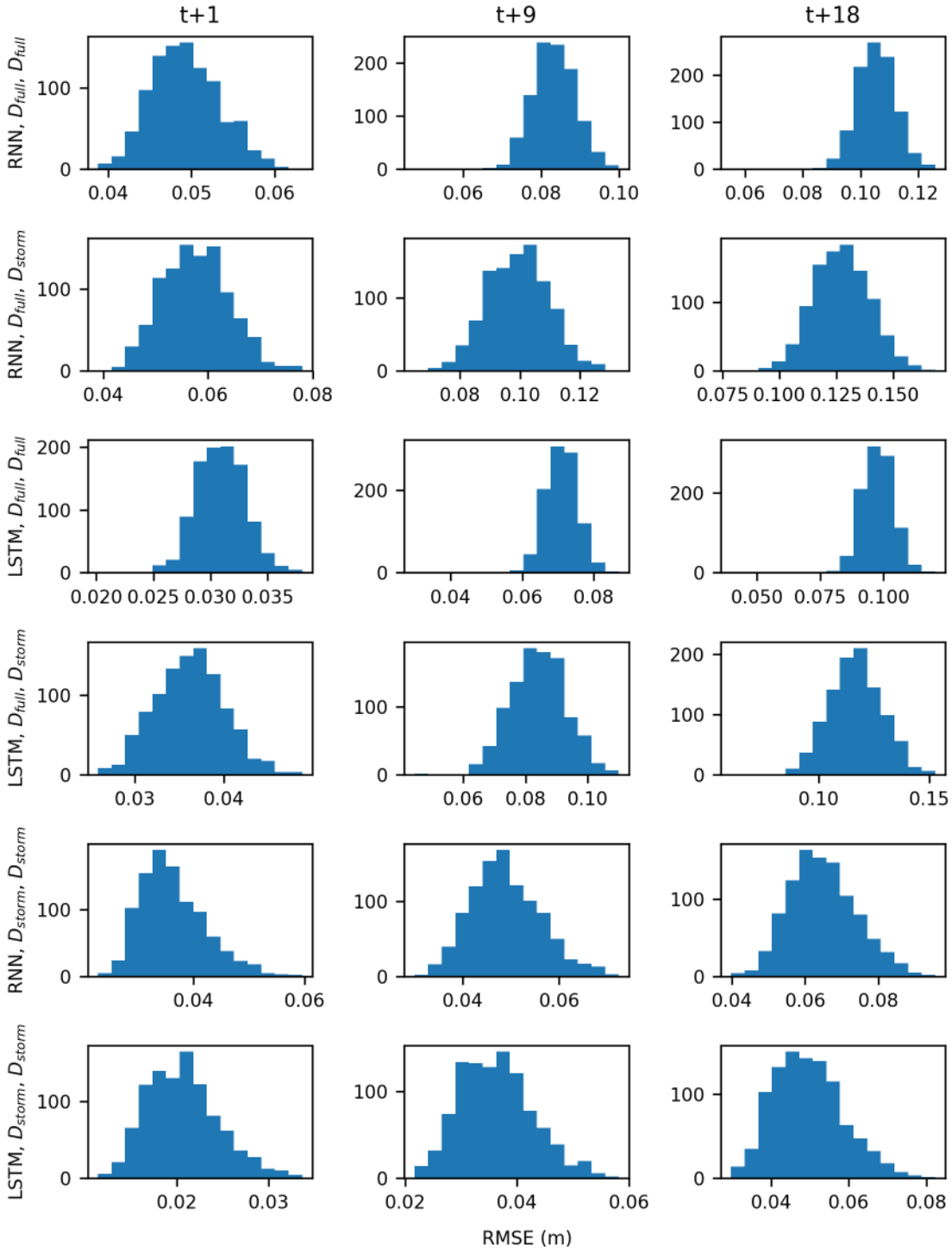

Figure A7. RMSE distributions for GW7 using observed data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

Appendix B

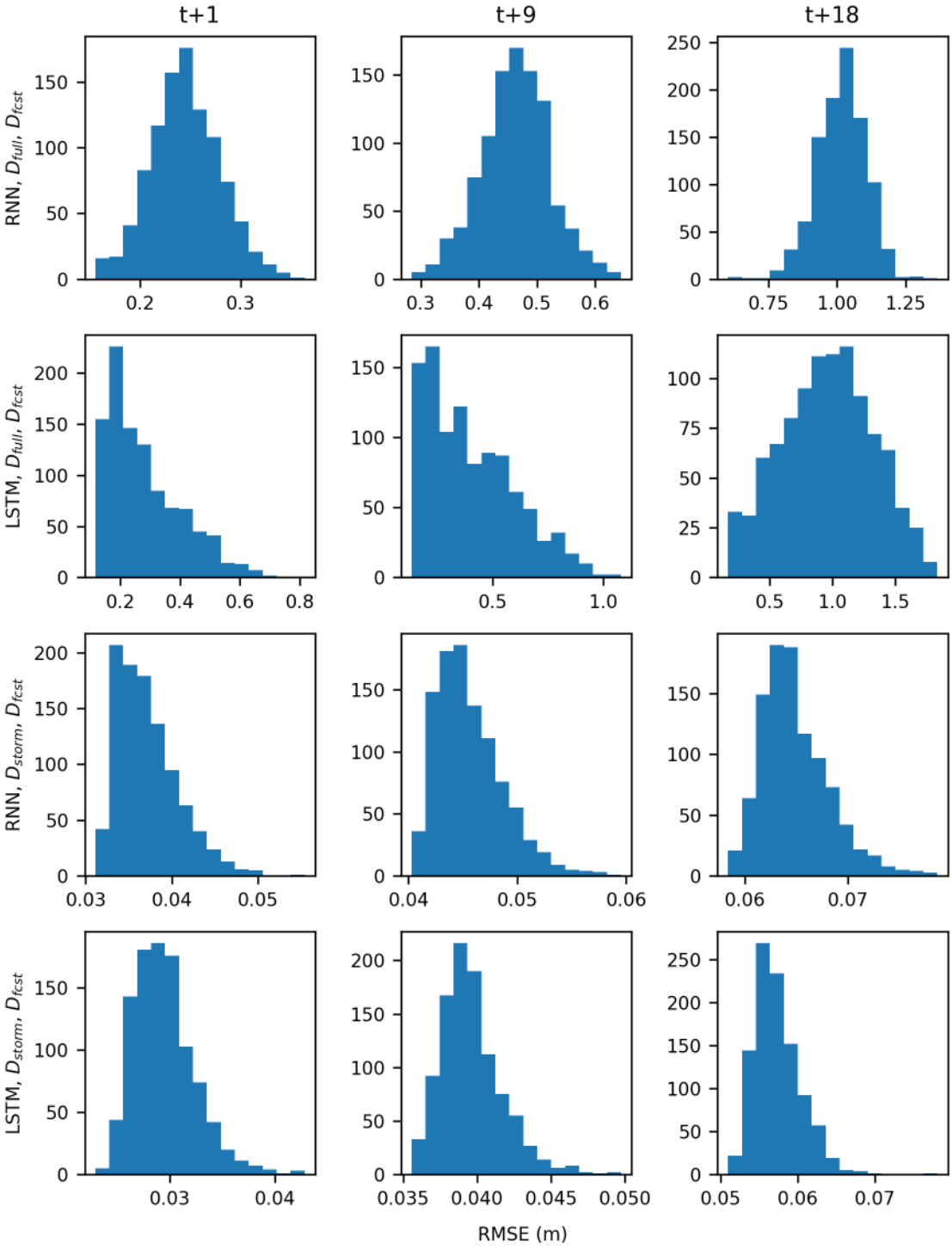

Figure A8. RMSE distributions for GW1 using forecast input data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

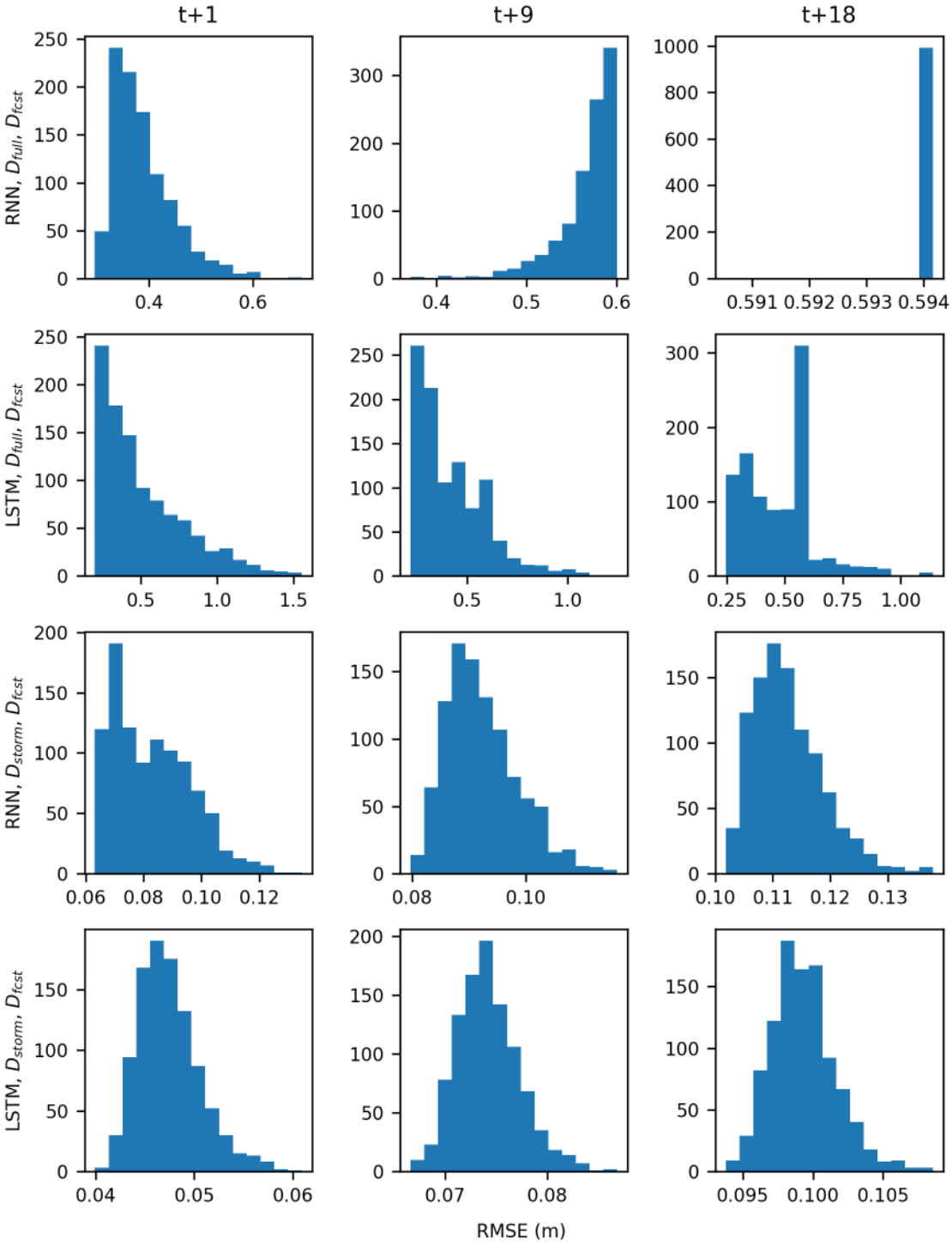

Figure A9. RMSE distributions for GW2 using forecast input data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

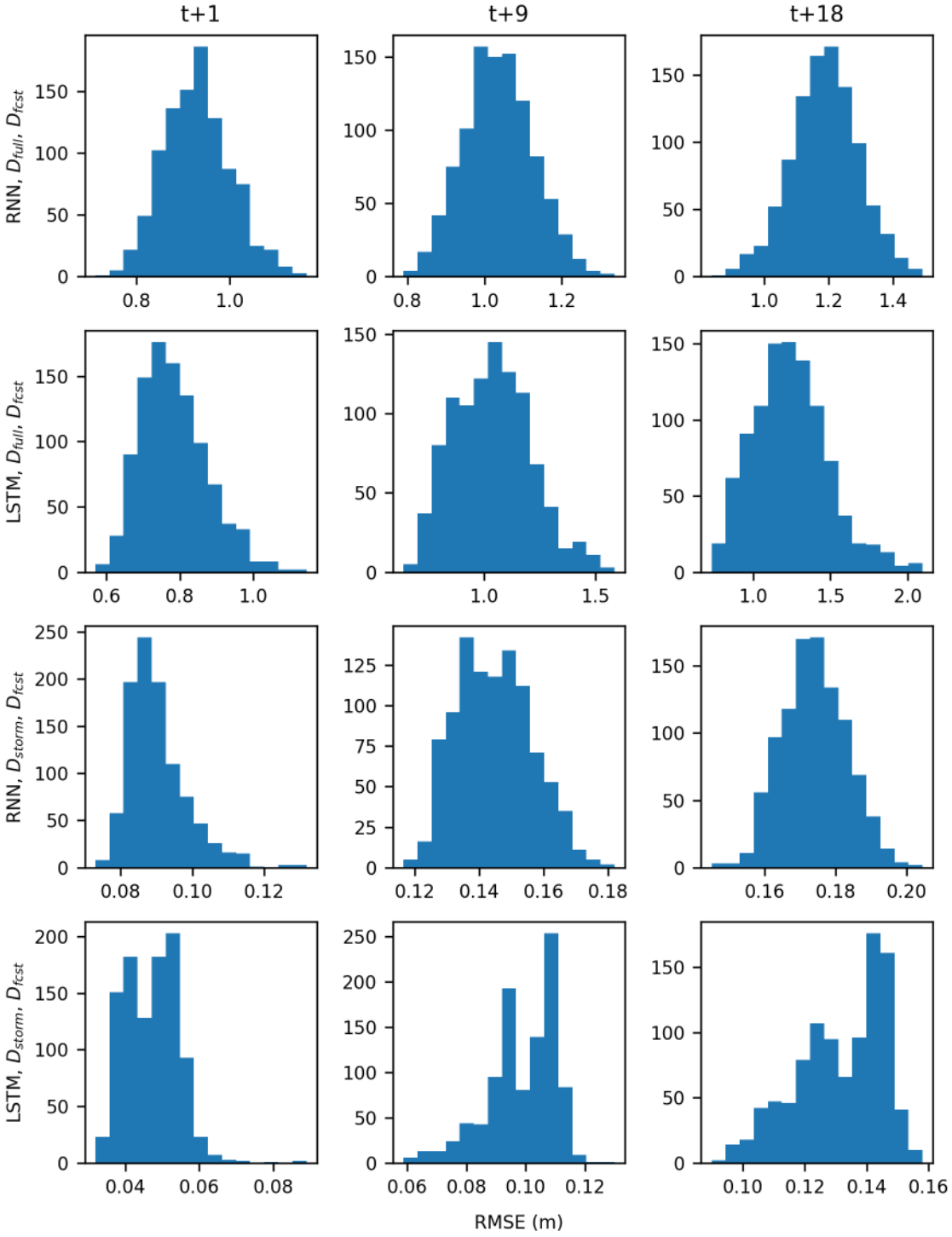

Figure A10. RMSE distributions for GW3 using forecast input data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

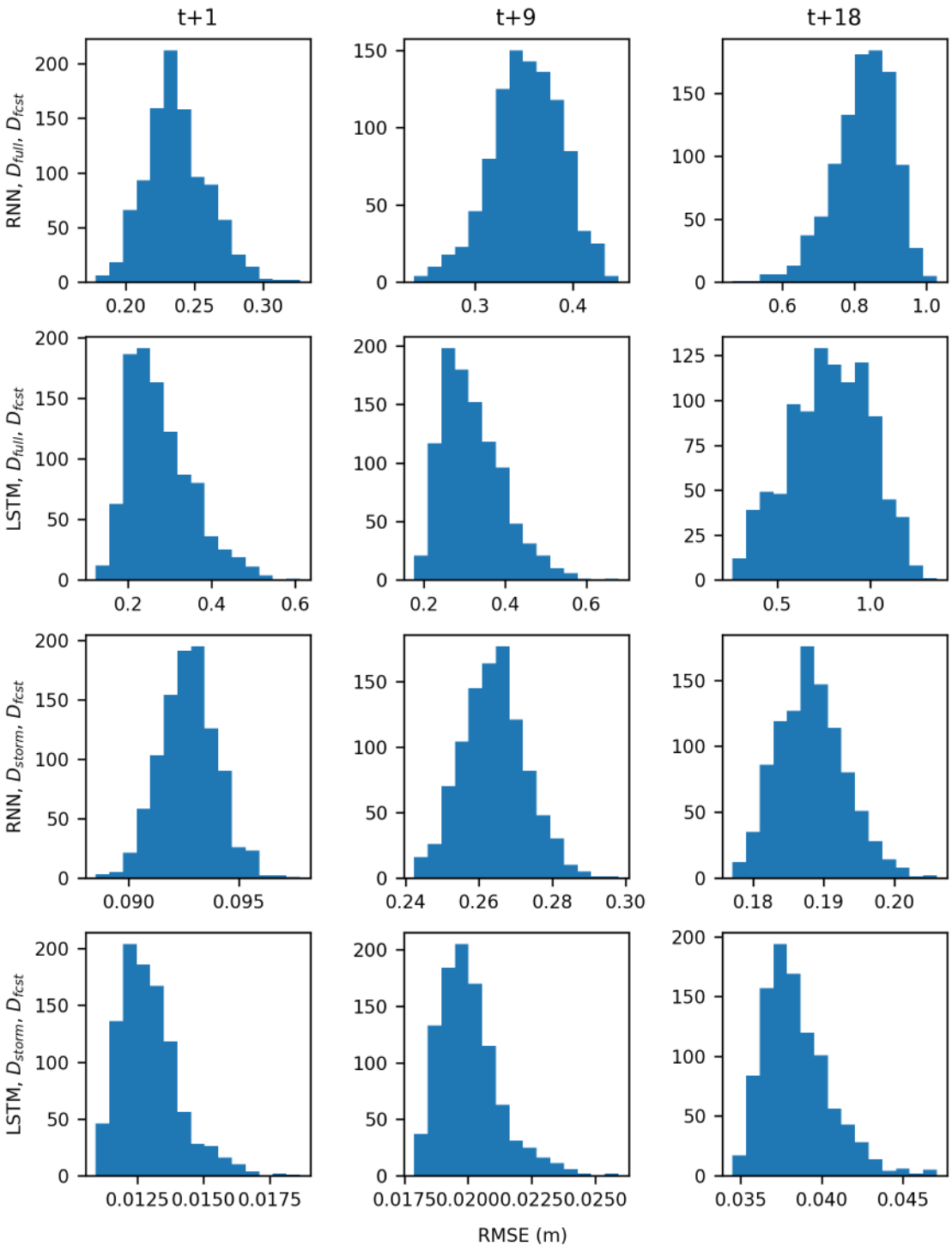

Figure A11. RMSE distributions for GW4 using forecast input data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

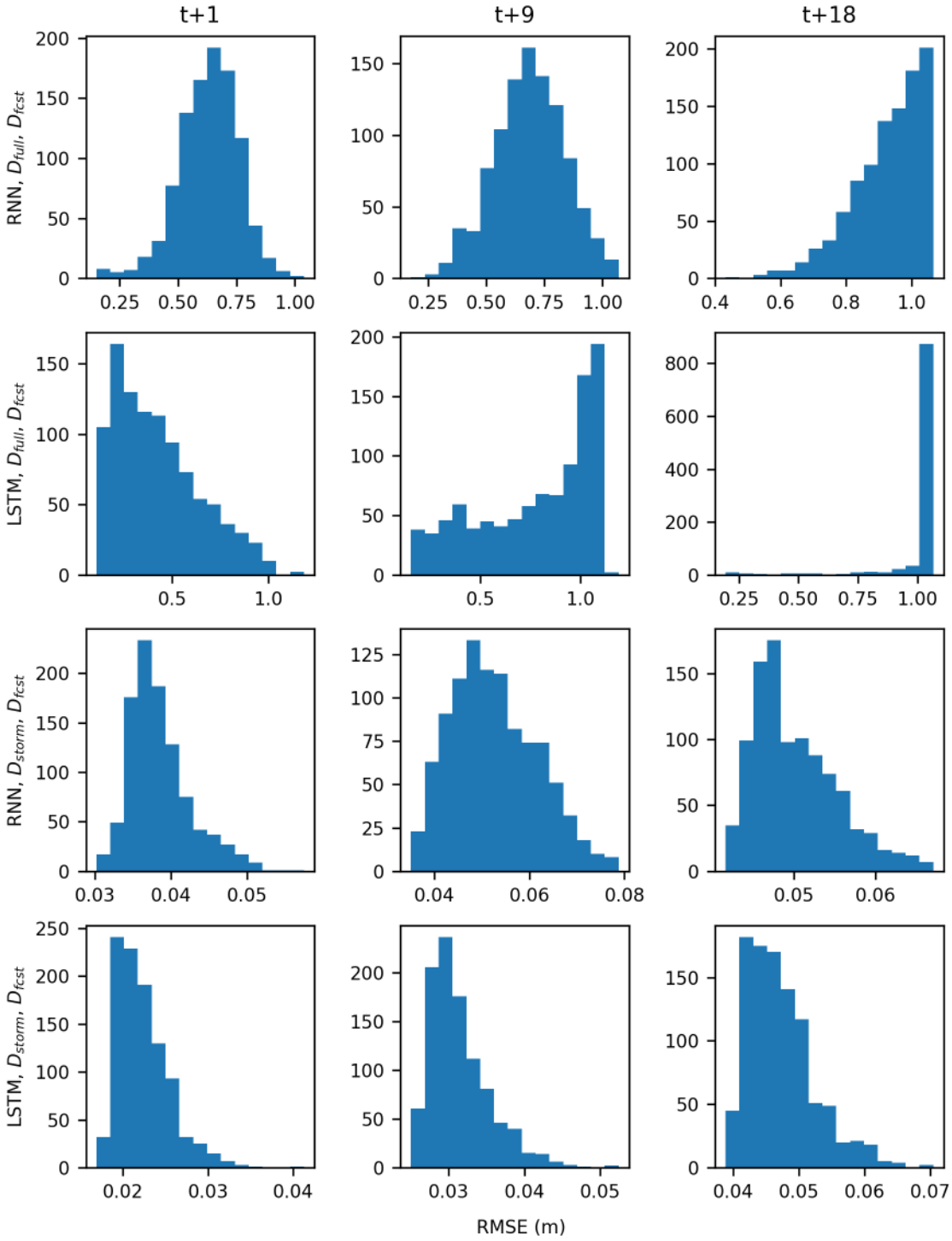

Figure A12. RMSE distributions for GW5 using forecast input data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

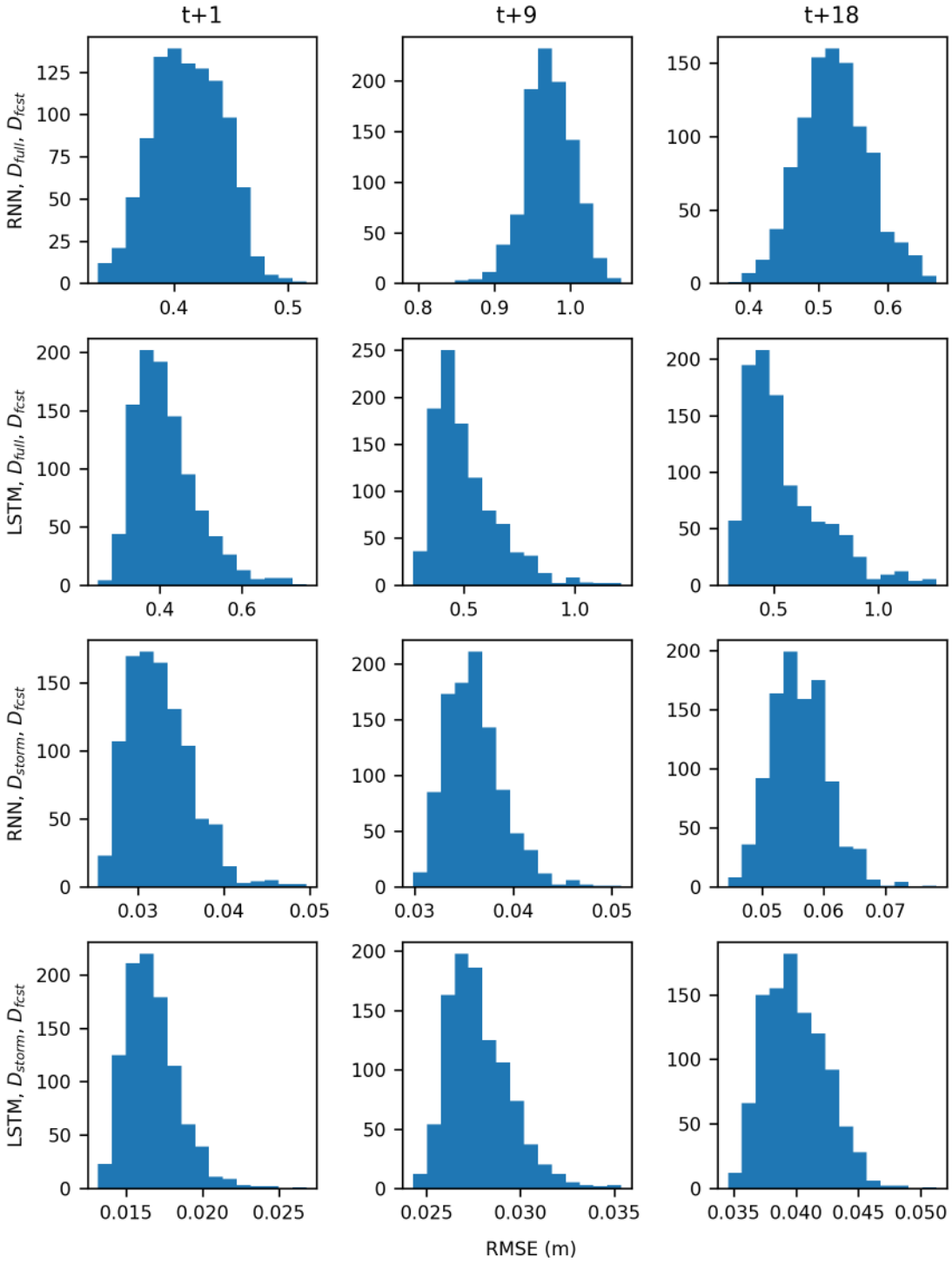

Figure A13. RMSE distributions for GW6 using forecast input data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

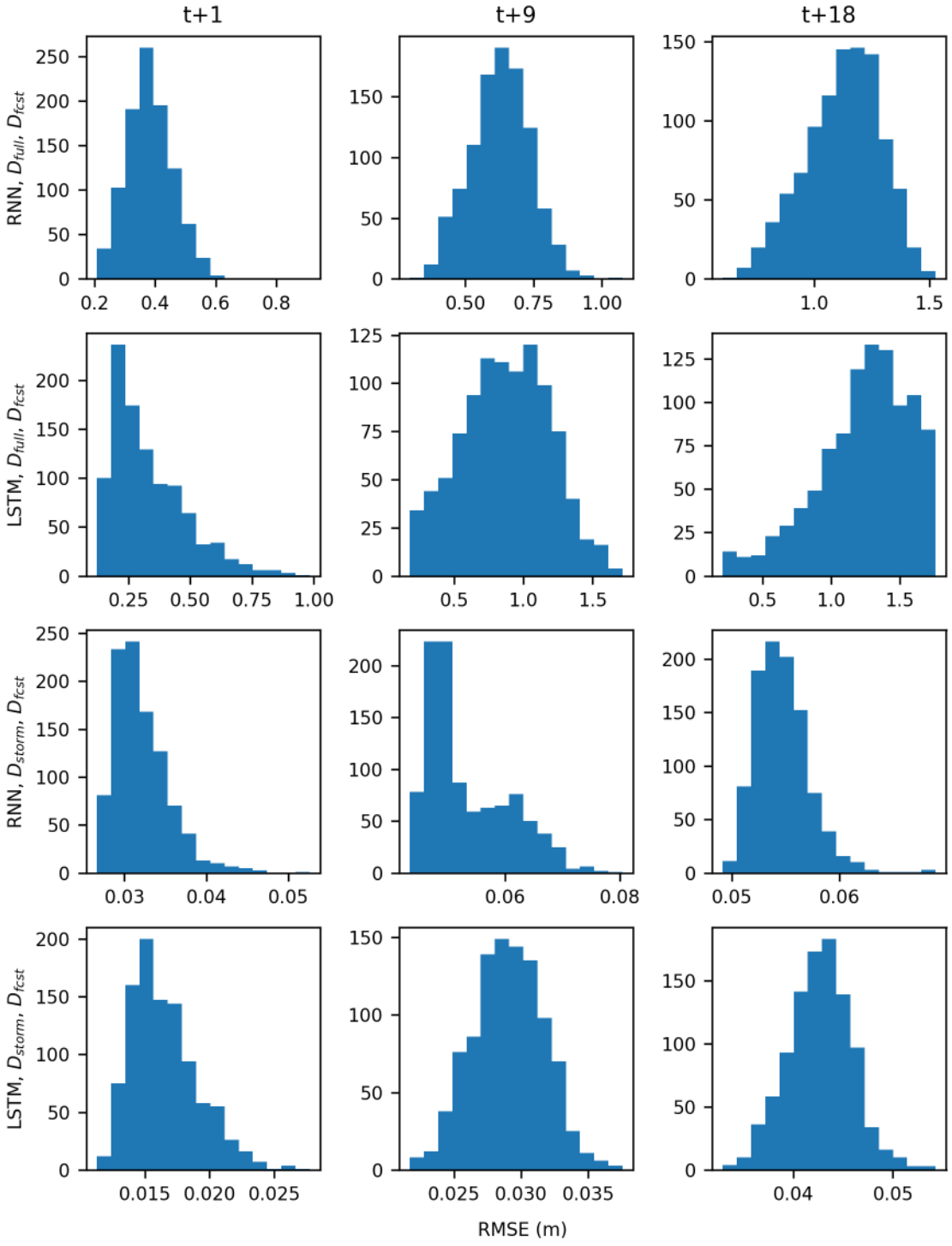

Figure A14. RMSE distributions for GW7 using forecast input data. Columns represent the forecast horizons t + 1, t + 9, and t + 18. Rows are specified as model type, training data, and testing data.

Appendix C

Table A1. Mean mean absolute error (MAE) values for each model type and training dataset treatment at each well and forecast period when tested on observed data.

Model Type	Training Data	Testing Data	Forecast Period	GW1	GW2	GW3	GW4	GW5	GW6	GW7
RNN	D_{full}	D_{full}	t + 1	0.031	0.060	0.072	0.019	0.035	0.116	0.029
			t + 9	0.049	0.089	0.099	0.036	0.054	0.410	0.044
			t + 18	0.069	0.118	0.127	0.052	0.075	0.236	0.060
RNN	D_{full}	D_{storm}	t + 1	0.038	0.072	0.080	0.022	0.047	0.121	0.034
			t + 9	0.064	0.114	0.119	0.042	0.074	0.397	0.054
			t + 18	0.092	0.151	0.157	0.060	0.102	0.228	0.076
LSTM	D_{full}	D_{full}	t + 1	0.020	0.029	0.021	0.008	0.016	0.013	0.014
			t + 9	0.040	0.067	0.053	0.027	0.039	0.032	0.028
			t + 18	0.061	0.102	0.087	0.046	0.063	0.052	0.045
LSTM	D_{full}	D_{storm}	t + 1	0.025	0.033	0.026	0.010	0.020	0.013	0.016
			t + 9	0.056	0.083	0.070	0.032	0.053	0.034	0.036
			t + 18	0.084	0.128	0.116	0.054	0.084	0.057	0.058
RNN	D_{storm}	D_{storm}	t + 1	0.030	0.060	0.069	0.069	0.039	0.026	0.026
			t + 9	0.041	0.068	0.080	0.288	0.045	0.028	0.033
			t + 18	0.051	0.085	0.095	0.208	0.048	0.036	0.043
LSTM	D_{storm}	D_{storm}	t + 1	0.024	0.031	0.023	0.008	0.017	0.012	0.013
			t + 9	0.036	0.049	0.037	0.015	0.027	0.019	0.021
			t + 18	0.045	0.066	0.052	0.023	0.033	0.027	0.029

Table A2. Mean MAE values for each model type and training dataset treatment at each well and forecast period when tested on forecast data D_{fcst}.

Model Type	Training Data	Testing Data	Forecast Period	GW1	GW2	GW3	GW4	GW5	GW6	GW7
RNN	D_{full}	D_{fcst}	t + 1	0.211	0.308	0.881	0.206	0.613	0.369	0.356
			t + 9	0.439	0.513	1.001	0.333	0.668	0.960	0.608
			t + 18	0.998	0.537	1.131	0.800	0.913	0.493	1.113
LSTM	D_{full}	D_{fcst}	t + 1	0.235	0.454	0.716	0.199	0.394	0.346	0.295
			t + 9	0.374	0.362	0.976	0.285	0.759	0.440	0.853
			t + 18	0.939	0.421	1.178	0.764	1.011	0.488	1.222
RNN	D_{storm}	D_{fcst}	t + 1	0.027	0.064	0.064	0.068	0.027	0.026	0.023
			t + 9	0.032	0.060	0.096	0.241	0.037	0.026	0.036
			t + 18	0.038	0.073	0.106	0.160	0.034	0.037	0.034
LSTM	D_{storm}	D_{fcst}	t + 1	0.022	0.029	0.027	0.007	0.014	0.012	0.012
			t + 9	0.028	0.044	0.038	0.012	0.019	0.019	0.017
			t + 18	0.037	0.059	0.055	0.022	0.025	0.027	0.025

References

1. Giambastiani, B.M.S.; Colombani, N.; Greggio, N.; Mastrocicco, M.A.M. Coastal aquifer response to extreme storm events in Emilia-Romagna, Italy. *Hydrol. Process.* **2017**, *31*, 1613–1621. [CrossRef]
2. Taormina, R.; Chau, K.-W.; Sethi, R. Artificial neural network simulation of hourly groundwater levels in a coastal aquifer system of the Venice lagoon. *Eng. Appl. Artif. Intell.* **2012**, *25*, 1670–1676. [CrossRef]
3. Rotzoll, K.; Fletcher, C.H. Assessment of groundwater inundation as a consequence of sea-level rise. *Nat. Clim. Chang.* **2012**, *3*, 477–481. [CrossRef]
4. Sweet, W.V.; Park, J. From the extreme to the mean: Acceleration and tipping points of coastal inundation from sea level rise. *Earth's Future* **2014**, *2*, 579–600. [CrossRef]
5. Wuebbles, D.J.; Fahey, D.W.; Hibbard, K.A.; Dokken, D.J.; Stewart, B.C.; Maycock, T.K. *Climate Science Special Report: Fourth National Climate Assessment, Volume I*; Global Change Research Program: Washington, DC, USA, 2017.
6. Sadler, J.M.; Goodall, J.L.; Morsy, M.M.; Spencer, K. Modeling urban coastal flood severity from crowd-sourced flood reports using Poisson regression and Random Forest. *J. Hydrol.* **2018**, *559*, 43–55. [CrossRef]

7. Bjerklie, D.M.; Mullaney, J.R.; Stone, J.R.; Skinner, B.J.; Ramlow, M.A. *Preliminary Investigation of the Effects of Sea-Level Rise on Groundwater Levels in New Haven, Connecticut*; U.S. Geological Survey: Reston, VA, USA, 2012.
8. Hoover, D.J.; Odigie, K.O.; Barnard, P. Sea-level rise and coastal groundwater inundation and shoaling at select sites in California, USA. *J. Hydrol. Reg. Stud.* **2017**, *11*, 234–249. [CrossRef]
9. Masterson, J.P.; Pope, J.P.; Fienen, M.N.; Monti, J., Jr.; Nardi, M.R.; Finkelstein, J.S. *Assessment of Groundwater Availability in the Northern Atlantic Coastal Plain Aquifer System From Long Island, New York, to North Carolina*; U.S. Geological Survey: Reston, VA, USA, 2016.
10. Kreibich, H.; Thieken, A.H. Assessment of damage caused by high groundwater inundation. *Water Resour. Res.* **2008**, *44*, 9409. [CrossRef]
11. Abboud, J.M.; Ryan, M.C.; Osborn, G.D. Groundwater flooding in a river-connected alluvial aquifer. *J. Flood Risk Manag.* **2018**, *11*, e12334. [CrossRef]
12. Bloetscher, F.; Romah, T.; Berry, L.; Hammer, N.H.; Cahill, M.A. Identification of physical transportation infrastructure vulnerable to sea level rise. *J. Sustain. Dev.* **2012**, *5*, 40–51.
13. Flood, J.F.; Cahoon, L.B. Risks to coastal wastewater collection systems from sea-level rise and climate change. *J. Coast. Res.* **2011**, *274*, 652–660. [CrossRef]
14. Sadler, J.M.; Haselden, N.; Mellon, K.; Hackel, A.; Son, V.; Mayfield, J.; Blase, A.; Goodall, J.L. Impact of sea-level rise on roadway flooding in the hampton roads region, virginia. *J. Infrastruct. Syst.* **2017**, *23*, 05017006. [CrossRef]
15. Chang, S.W.; Nemec, K.; Kalin, L.; Clement, T.P. Impacts of climate change and urbanization on groundwater resources in a Barrier Island. *J. Environ. Eng.* **2016**, *142*, D4016001. [CrossRef]
16. Doble, R.C.; Pickett, T.; Crosbie, R.S.; Morgan, L.K.; Turnadge, C.; Davies, P.J. Emulation of recharge and evapotranspiration processes in shallow groundwater systems. *J. Hydrol.* **2017**, *555*, 894–908. [CrossRef]
17. Heywood, C.E.; Pope, J.P. *Simulation of Groundwater Flow in the Coastal Plain Aquifer System of Virginia*; U.S. Geological Survey: Reston, VA, USA, 2009; p. 115.
18. Masterson, J.P.; Garabedian, S.P. Effects of sea-level rise on ground water flow in a coastal aquifer system. *Ground Water* **2007**, *45*, 209–217. [CrossRef] [PubMed]
19. Park, E.; Parker, J.C. A simple model for water table fluctuations in response to precipitation. *J. Hydrol.* **2008**, *356*, 344–349. [CrossRef]
20. Pauw, P.S.; Oude Essink, G.H.P.; Leijnse, A.; Vandenbohede, A.; Groen, J.; van der Zee, S.E.A.T.M. Regional scale impact of tidal forcing on groundwater flow in unconfined coastal aquifers. *J. Hydrol.* **2014**, *517*, 269–283. [CrossRef]
21. Fahimi, F.; Yaseen, Z.M.; El-Shafie, A. Application of soft computing based hybrid models in hydrological variables modeling: A comprehensive review. *Theor. Appl. Climatol.* **2017**, *128*, 875–903. [CrossRef]
22. Govindaraju, R.S. Artificial neural networks in hydrology. I: Preliminary concepts by the asce task committee on application of artificial neural networks in hydrology. *J. Hydrol. Eng.* **2000**, *5*, 115–123.
23. Govindaraju, R.S. Artificial neural networks in hydrology. II: Hydrologic applications. *J. Hydrol. Eng.* **2000**, *5*, 124.
24. Maier, H.R.; Jain, A.; Dandy, G.C.; Sudheer, K.P. Methods used for the development of neural networks for the prediction of water resource variables in river systems: Current status and future directions. *Environ. Model. Softw.* **2010**, *25*, 891–909. [CrossRef]
25. Maier, H.R.; Dandy, G.C. Neural networks for the prediction and forecasting of water resources variables: A review of modelling issues and applications. *Environ. Model. Softw.* **2000**, *15*, 101–124. [CrossRef]
26. Yang, T.; Asanjan, A.A.; Welles, E.; Gao, X.; Sorooshian, S.; Liu, X. Developing reservoir monthly inflow forecasts using artificial intelligence and climate phenomenon information. *Water Resour. Res.* **2017**, *53*, 2786–2812. [CrossRef]
27. Yaseen, Z.M.; El-shafie, A.; Jaafar, O.; Afan, H.A.; Sayl, K.N. Artificial intelligence based models for stream-flow forecasting: 2000–2015. *J. Hydrol.* **2015**, *530*, 829–844. [CrossRef]
28. Solomatine, D.P.; Ostfeld, A. Data-driven modelling: Some past experiences and new approaches. *J. Hydroinform.* **2008**, *10*, 3–22. [CrossRef]
29. Karandish, F.; Šimůnek, J. A comparison of numerical and machine-learning modeling of soil water content with limited input data. *J. Hydrol.* **2016**, *543*, 892–909. [CrossRef]

30. Mohanty, S.; Jha, M.K.; Kumar, A.; Panda, D.K. Comparative evaluation of numerical model and artificial neural network for simulating groundwater flow in Kathajodi–Surua Inter-basin of Odisha, India. *J. Hydrol.* **2013**, *495*, 38–51. [CrossRef]
31. Chang, F.-J.; Chang, L.-C.; Huang, C.-W.; Kao, I.-F. Prediction of monthly regional groundwater levels through hybrid soft-computing techniques. *J. Hydrol.* **2016**, *541*, 965–976. [CrossRef]
32. Coulibaly, P.; Anctil, F.; Aravena, R.; Bobée, B. Artificial neural network modeling of water table depth fluctuations. *Water Resour. Res.* **2001**, *37*, 885–896. [CrossRef]
33. Daliakopoulos, I.N.; Coulibaly, P.; Tsanis, I.K. Groundwater level forecasting using artificial neural networks. *J. Hydrol.* **2005**, *309*, 229–240. [CrossRef]
34. Guzman, S.M.; Paz, J.O.; Tagert, M.L.M. The use of NARX neural networks to forecast daily groundwater levels. *Water Resour. Manag.* **2017**, *31*, 1591–1603. [CrossRef]
35. Nayak, P.C.; Rao, Y.R.S.; Sudheer, K.P. Groundwater Level Forecasting in a Shallow Aquifer Using Artificial Neural Network Approach. *Water Resour. Manag.* **2006**, *20*, 77–90. [CrossRef]
36. Sahoo, S.; Jha, M.K. Groundwater-level prediction using multiple linear regression and artificial neural network techniques: A comparative assessment. *Hydrogeol. J.* **2013**, *21*, 1865–1887. [CrossRef]
37. Bengio, Y.; Simard, P.; Frasconi, P. Learning long-term dependencies with gradient descent is difficult. *IEEE Trans. Neural Netw.* **1994**, *5*, 157–166. [CrossRef] [PubMed]
38. Hochreiter, S.; Schmidhuber, U. Long short-term memory. *Neural Comput.* **1997**, *9*, 1735–1780. [CrossRef] [PubMed]
39. Graves, A.; Mohamed, A.; Hinton, G. Speech recognition with deep recurrent neural networks. In Proceedings of the 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, Vancouver, BC, Canada, 26–31 May 2013; IEEE: Piscataway, NJ, USA, 2013; pp. 6645–6649.
40. Fischer, T.; Krauss, C. Deep learning with long short-term memory networks for financial market predictions. *Eur. J. Oper. Res.* **2018**, *270*, 654–669. [CrossRef]
41. Liu, Y.; Wang, Y.; Yang, X.; Zhang, L. Short-term travel time prediction by deep learning: A comparison of different LSTM-DNN models. In Proceedings of the 2017 IEEE 20th International Conference on Intelligent Transportation Systems (ITSC), Yokohama, Japan, 16–19 October 2017; pp. 1–8.
42. Zhao, Z.; Chen, W.; Wu, X.; Chen, P.C.Y.; Liu, J. LSTM network: A deep learning approach for short-term traffic forecast. *IET Intell. Transp. Syst.* **2017**, *11*, 68–75. [CrossRef]
43. Hu, C.; Wu, Q.; Li, H.; Jian, S.; Li, N.; Lou, Z.; Hu, C.; Wu, Q.; Li, H.; Jian, S.; et al. Deep learning with a long short-term memory networks approach for rainfall-runoff simulation. *Water* **2018**, *10*, 1543. [CrossRef]
44. Liang, C.; Li, H.; Lei, M.; Du, Q. Dongting lake water level forecast and its relationship with the three gorges dam based on a long short-term memory network. *Water* **2018**, *10*, 1389. [CrossRef]
45. Tian, Y.; Xu, Y.-P.; Yang, Z.; Wang, G.; Zhu, Q.; Tian, Y.; Xu, Y.-P.; Yang, Z.; Wang, G.; Zhu, Q. Integration of a parsimonious hydrological model with recurrent neural networks for improved streamflow forecasting. *Water* **2018**, *10*, 1655. [CrossRef]
46. Zhang, J.; Zhu, Y.; Zhang, X.; Ye, M.; Yang, J. Developing a Long Short-Term Memory (LSTM) based model for predicting water table depth in agricultural areas. *J. Hydrol.* **2018**, *561*, 918–929. [CrossRef]
47. USCB. U.S. Census Bureau QuickFacts: Norfolk city, Virginia. Available online: https://www.census.gov/quickfacts/fact/table/norfolkcityvirginia/PST045217 (accessed on 5 February 2019).
48. Fears, D. Built on sinking ground, Norfolk tries to hold back tide amid sea-level rise. *Washington Post*. 2012. Available online: https://www.washingtonpost.com/national/health-science/built-on-sinking-ground-norfolk-tries-to-hold-back-tide-amid-sea-level-rise/2012/06/17/gJQADUsxjV_story.html?noredirect=on&utm_term=.fc9be59c217a (accessed on 4 January 2019).
49. Eggleston, J.; Pope, J. *Land Subsidence and Relative Sea-Level Rise in the Southern Chesapeake Bay Region*; US Geological Survey Circular: Reston, VA, USA, 2013; Volume Circular 1392.
50. NOAA Sewells Point—Station Home Page—NOAA Tides & Currents. Available online: https://tidesandcurrents.noaa.gov/stationhome.html?id=8638610 (accessed on 29 October 2018).
51. Freeze, R.A.; Cherry, J.A. *Groundwater*; Prentice Hall, Inc.: Englewood Cliffs, NJ, USA, 1979; ISBN 0133653129.
52. Smirnov, D.; Giovannettone, J.; Lawler, S.; Sreetharan, M.; Plummer, J.; Workman, B.; Batten, B.; Rosenberg, S.; Mcglone, D. *Analysis of Historical and Future Heavy Precipitation*; City of Virginia Beach Department of Public Works: Virginia Beach, VA, USA, 2018.

53. Blaylock, B.K.; Horel, J.D.; Liston, S.T. Cloud archiving and data mining of high-resolution rapid refresh forecast model output. *Comput. Geosci.* **2017**, *109*, 43–50. [CrossRef]
54. NOAA. Tide Predictions—NOAA Tides and Currents. Available online: https://tidesandcurrents.noaa.gov/noaatidepredictions.html?id=8638610 (accessed on 4 January 2019).
55. NOAA. Tide Predictions—Help—NOAA Tides and Currents. Available online: https://tidesandcurrents.noaa.gov/PageHelp.html (accessed on 4 January 2019).
56. NOAA. Harmonic Analysis. Available online: https://tidesandcurrents.noaa.gov/harmonic.html (accessed on 4 January 2019).
57. MathWorks Outlier Removal Using Hampel Identifier. Available online: https://www.mathworks.com/help/signal/ref/hampel.html (accessed on 5 February 2019).
58. SciPy. SciPy.Signal.Find-Peaks-Scipy v1.2.1 Reference Guide. Available online: https://docs.scipy.org/doc/scipy/reference/generated/scipy.signal.find_peaks.html (accessed on 7 March 2019).
59. Shalizi, C.R. Bootstrapping Time Series. In *Advanced Data Analysis from an Elementary Point of View*; Cambridge University Press: Cambridge, UK, 2018; pp. 587–590. Available online: http://www.stat.cmu.edu/~{}cshalizi/ADAfaEPoV/ (accessed on 31 October 2018).
60. Elman, J.L. Finding Structure in Time. *Cogn. Sci.* **1990**, *14*, 179–211. [CrossRef]
61. Werbos, P.J. Backpropagation through time: What it does and how to do it. *Proc. IEEE* **1990**, *78*, 1550–1560. [CrossRef]
62. Chollet, F. Keras. Available online: https://keras.io2015 (accessed on 8 June 2018).
63. Greff, K.; Srivastava, R.K.; Koutnik, J.; Steunebrink, B.R.; Schmidhuber, J. LSTM: A search space odyssey. *IEEE Trans. Neural Netw. Learn. Syst.* **2017**, *28*, 2222–2232. [CrossRef]
64. Bergstra, J.; Yamins, D.; Cox, D.D. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. In Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA, USA, 16–21 June 2013.
65. Bergstra, J.S.; Bardenet, R.; Bengio, Y.; Kégl, B. Algorithms for Hyper-Parameter Optimization. In *Advances in Neural Information Processing Systems 24*; Shawe-Taylor, J., Zemel, R.S., Bartlett, P.L., Pereira, F., Weinberger, K.Q., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2011; pp. 2546–2554.
66. Pumperla, M. Hyperas. Available online: http://maxpumperla.com/hyperas/ (accessed on 7 November 2018).
67. Zhang, D.; Lindholm, G.; Ratnaweera, H. Use long short-term memory to enhance internet of things for combined sewer overflow monitoring. *J. Hydrol.* **2018**, *556*, 409–418. [CrossRef]
68. Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R. Dropout: A simple way to prevent neural networks from overfitting. *J. Mach. Learn. Res.* **2014**, *15*, 1929–1958.
69. Abadi, M.; Barham, P.; Chen, J.; Chen, Z.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Irving, G.; Isard, M.; et al. TensorFlow: A system for large-scale machine learning. In Proceedings of the 12th USENIX Symposium on Operating Systems Design and Implementation, Savannah, GA, USA, 2–4 November 2016; USENIX: Savannah, GA, USA, 2016; pp. 265–283.
70. SciPy. SciPy.stats.ttest_ind—SciPy v1.2.1 Reference Guide. Available online: https://docs.scipy.org/doc/scipy/reference/generated/scipy.stats.ttest_ind.html (accessed on 7 March 2019).
71. Yoon, H.; Jun, S.-C.; Hyun, Y.; Bae, G.-O.; Lee, K.-K. A comparative study of artificial neural networks and support vector machines for predicting groundwater levels in a coastal aquifer. *J. Hydrol.* **2011**, *396*, 128–138. [CrossRef]
72. Moss, A.; Marani, M. Coastal Water Table Mapping: Incorporating Groundwater Data into Flood Inundation Forecasts. Master's Thesis, Duke University, Durham, NC, USA, 2016.
73. Kingma, D.P.; Ba, J. Adam: A Method for Stochastic Optimization. *ArXiv* **2014**, arXiv:1412.6980.
74. Krajewski, W.F.; Smith, J.A. Radar hydrology: Rainfall estimation. *Adv. Water Resour.* **2002**, *25*, 1387–1394. [CrossRef]
75. Ran, Y.; Li, X.; Ge, Y.; Lu, X.; Lian, Y. Optimal selection of groundwater-level monitoring sites in the Zhangye Basin, Northwest China. *J. Hydrol.* **2015**, *525*, 209–215. [CrossRef]
76. Sadler, J.M.; Goodall, J.L.; Asce, M.; Morsy, M.M. Effect of rain gauge proximity on rainfall estimation for problematic urban coastal watersheds in Virginia Beach, Virginia. *J. Hydrol. Eng.* **2017**, *22*, 04017036. [CrossRef]