Long Short-Term Memory Neural Network (LSTM-NN) for Aquifer Level Time Series Forecasting Using in-Situ Piezometric Observations

Ryan Solgi[1,1], Hugo A. Loaiciga[1,1], and Mark Kram[2,2]

[1]University of California Santa Barbara
[2]Groundswell Technologies, Inc.

January 20, 2023

Abstract

The application of neural networks (NN) in groundwater (GW) level prediction has been shown promising by previous works. Yet, previous works have relied on a variety of inputs, such as air temperature, pumping rates, precipitation, service population, and others. This work presents a long short-term memory neural network (LSTM-NN) for GW level forecasting using only previously observed GW level data as the input without resorting to any other type of data and information about a groundwater basin. This work applies the LSTM-NN for short-term and long-term GW level forecasting in the Edwards aquifer in Texas. The Adam optimizer is employed for training the LSTM-NN. The performance of the LSTM-NN was compared with that of a simple NN under 36 different scenarios with prediction horizons ranging from one day to three months, and covering several conditions of data availability. This paper's results demonstrate the superiority of the LSTM-NN over the simple-NN in all scenarios and the success of the LSTM-NN in accurate GW level prediction. The LSTM-NN predicts one lag, up to four lags, and up to 26 lags ahead GW level with an accuracy (R^2) of at least 99.89%, 99.00%, and 90.00%, respectively, over a testing period longer than 17 years of the most recent records. The quality of this work's results demonstrates the capacity of machine learning (ML) in groundwater prediction, and affirms the importance of gathering high-quality, long-term, GW level data for predicting key groundwater characteristics useful in sustainable groundwater management.

Long Short-Term Memory Neural Network (LSTM-NN) for Aquifer Level Time Series Forecasting Using in-Situ Piezometric Observations

Ryan (Mohammad) Solgi[1], Hugo A. Loáiciga[2], and Mark Kram[3]

[1] PhD student, University of California Santa Barbara (UCSB)

[2] Professor, Department of Geography, University of California Santa Barbara (UCSB)

[3] PhD, CTO at Groundswell Technologies, Inc.

Corresponding author: Ryan (Mohammad) Solgi (ryan.solgi@geog.ucsb.edu)

Key points:

- A long short-term memory neural network is proposed for groundwater level prediction using in-situ piezometric observations.
- The results demonstrated the capability of the proposed model for daily and monthly prediction of the Edward aquifer's water table.
- The study's results highlight the importance of gathering high-quality, long-term, groundwater level data.

Keywords: Long Short-Term Memory (LSTM), Neural Network, Adam Optimizer, Groundwater, Prediction, Edward Aquifer, Time Series.

Abstract

The application of neural networks (NN) in groundwater (GW) level prediction has been shown promising by previous works. Yet, previous works have relied on a variety of inputs, such as air temperature, pumping rates, precipitation, service population, and others. This work presents a long short-term memory neural network (LSTM-NN) for GW level forecasting using only previously observed GW level data as the input without resorting to any other type of data and information about a groundwater basin. This work applies the LSTM-NN for short-term and long-term GW level forecasting in the Edwards aquifer in Texas. The Adam optimizer is employed for training the LSTM-NN. The performance of the LSTM-NN was compared with that of a simple NN under 36 different scenarios with prediction horizons ranging from one day to three months, and covering several conditions of data availability. This paper's results demonstrate the superiority of the LSTM-NN over the simple-NN in all scenarios and the success of the LSTM-NN in accurate GW level prediction. The LSTM-NN predicts one lag, up to four lags, and up to 26 lags ahead GW level with an accuracy (R^2) of at least 99.89%, 99.00%, and 90.00%, respectively, over a testing period longer than 17 years of the most recent records. The quality of this work's results demonstrates the capacity of machine learning (ML) in groundwater prediction, and affirms the importance of gathering high-quality, long-term, GW level data for predicting key groundwater characteristics useful in sustainable groundwater management.

1. Introduction

Previous works have been studied water resources management methods to address sustainability of different water resources systems (Solgi, et al., 2015, 2016a, 2016b, 2020; Bozorg-Haddad et al., 2017). Sustainable operation of water resources, groundwater systems among them, is contingent upon accurate groundwater level tracking and prediction. Hydraulic head in groundwater systems affects surface water sustainability, sea water intrusion in coastal zones, soil stability, stream flow, and other key hydrologic functions. For these reasons the prediction of the groundwater level is central to the management of aquifer resources. Nevertheless, the nonlinearity of the governing groundwater flow equations, the heterogeneity and anisotropy of aquifers, and the complex interconnection of surface and groundwater systems, associated uncertainties, and anthropogenic effects (i.e., withdrawal and managed aquifer recharge) render the task of forecasting groundwater levels challenging. A variety of methods have been developed for the purpose of groundwater level prediction (Orsborn, 1966; Yakowitz, 1976; Hipel and McLeod, 1994; Sahoo and Jha, 2013; Suryanarayana et al., 2014; Wunsch et al., 2018; Takafuji et al., 2019). Several machine learning techniques have been successfully applied to groundwater head prediction. Rajaee et al. (2019) reviewed machine learning (ML) methods for groundwater modeling including neural networks (NN), adaptive neuro-fuzzy inference system (ANFIS), genetic programming (GP), and support vector machine (SVM) among others.

The majority of the ML models applied to groundwater level prediction require a variety of inputs such as precipitation, air temperature, evaporation, service population, surface-water systems data (i.e., reservoir storage, river discharge), pumping rates, and so forth (Coulibaly et al., 2001; Sun, 2013; Sahoo et al., 2017; Adiat et al., 2020; Khedri et al., 2020). Only a limited number of models have been presented which successfully forecast the groundwater level relying on the historical groundwater hydraulic head as the sole model input. Yang et al. (2009) presented a NN to predict groundwater level in Western Jillian, China. The network consists of six input nodes (receiving six successive previous lags of monthly average groundwater level), one hidden layer with 10 sigmoid nodes, and it predicts the monthly average ground level one month ahead. The latter authors demonstrated the superiority of the NN over the autoregressive (AR) model. Chen et al. (2010) implemented the self-organizing map (SOM) technique to determine the hyperparameters (i.e., the number of hidden/unit layers) of a radial basis function network (RBFN) for groundwater level prediction for a case study in Taiwan. The inputs to the model were the past 13, 12, and one lag monthly average groundwater levels. The latter authors employed the model for 1-month (one step) ahead prediction. Chen et al. (2011) applied autoregressive integrated moving average (ARIMA) and the semivariogram to determine the best set of lags of historical groundwater level as the inputs to a NN to predict one-month ahead groundwater level. Kisi and Shiri (2012) applied a modified wavelet neuro fuzzy model to predict groundwater level up to three days ahead where the inputs to the network were at most five lags of observed daily GW depths. Maheswaran and Khosa (2013) presented a wavelet neural network (WA-ANN) for GW prediction using monthly data. Yang et al. (2015) studied the performance of WA-ANN in comparison to integrated time series model employing monthly average GW data in an island in China.

A long short-term memory (LSTM) network is a variety of recurrent neural networks (RNNs) introduced by Hochreiter and Schmidhuber (1997), and has been successfully applied to executing elaborate machine learning tasks like speech recognition and machine translation (Houdt et al., 2020). The LSTM architecture is superior to other RNNs as it provides a deep network but does not suffer from vanishing gradient shortcomings prevalent in other RNNs. Therefore, the LSTM networks seem well suited for modeling dependencies imbedded in timeseries. LSTM has been successfully applied for some other machine learning tasks; yet, its application to GW level forecasting has been limited. Zhang et al. (2018) implemented an LSTM network to predict GW level where the inputs to the network were monthly water diversions, evaporation, precipitation, air temperature, and time. Bowes et al (2019) applied an LSTM network to predict GW level for flood control purposes using observed GW table, precipitation, and sea level data. The latter two studies showed the capability of LSTM in predicting groundwater levels.

ML techniques of the NN variety have demonstrated good performance in GW level prediction. Most previous related works, however, have predicted the GW table using NN models based on a variety of inputs (e.g., temperature, pumping, precipitation, service population, and others). Besides the fact that the choice of the input commonly depends on data availability, the task of finding the best set of inputs is a challenging task and varies among groundwater basins. To address these difficulties this paper introduces and applies a long short-term memory neural network (LSTM-NN) for GW level forecasting relying only on previously observed GW level data. LSTM has been successfully applied to predict various types of timeseries; nevertheless, to the best of our knowledge, it has not been applied to forecast GW timeseries relying only on previous in-situ piezometric observations. This work tests the application of the LSTM-NN trained by the Adam optimizer to forecast GW level where the only input to the network is observed GW level. Most pertinent studies carried out GW level prediction based on other sources of data, such as precipitation, due to the lack of enough piezometric data. GW data have become more widely available worldwide, opening new avenues for developing and testing novel algorithms predicting groundwater phenomena. The GW data themselves can be viewed as a hydrologic footprint in a basin when no other data are available. This work presents and tests the LSTM-NN for forecasting short-term and long-term groundwater level, and evaluates its performance in the Edwards Balcones Fault Zone aquifer of south-central Texas. The LSTM-NN's performance is compared with that of a simple NN. This work's results demonstrate the capacity of ML in groundwater prediction, and affirm the importance of gathering high-quality, long-term, GW level data for the purpose of predicting key groundwater characteristics.

2. Methodology

This study's objective is to predict the future groundwater level from in-situ groundwater level data. This is accomplished with an LSTM-NN and calculated results are compared with those of a simple NN. The studied networks were optimized using the Adam optimizer. The performance of the LSTM-NN was evaluated under several scenarios of data availability and prediction horizons. This section presents the applied simple NN, followed by a description of the proposed LSTM-NN's architecture and the Adam optimizer, and a definition of the prediction scenarios.

2.1. Simple neural network (NN)

A typical simple neural network consists of an input layer, one or several hidden layers, and one output layer (see, e.g., Kelleher and Tierney, 2018). Each layer consists of one or several cells (units). Each cell's output $y_{i,j}$ is a function of the weighted sum of all the inputs to the cell. The simple NNs employ an activation function as expressed by the following equations:

$$y_{i,j} = f(\sum_l [W_{i,j}(l, j-1) \times y_{l,j-1}] + b_{i,j}) \quad (1)$$

in which, the sigmoid function $f(.)$ is defined as follows:

$$f(x) = \frac{1}{1+exp(-x)} \quad (2)$$

where $y_{i,j}$ = the output of cell *i* of layer *j*, $W_{i,j}(l, j-1)$ = the weight of the connection between cell *i* of layer *j* and cell *l* of layer *j*-1, and $b_{i,j}$ = the bias of cell *i* of layer *j*.

The simple NN herein comprises of one hidden layer with 10 sigmoid cells, one input layer with P input cells where P = the number of previous lags of observed GW level, and one output layer with one cell whose activation is the identity function as depicted in Figure 1. The simple NN is fully connected, meaning that the cells of each layer are connected to all the cells of the previous and next layers. The number of parameters of the simple NN is equal to $(P \times 10) + 21$ (number of cells in the hidden layer × [number of input cells + number of cells in the output layer + 1] + number of cells in the output layer). For example, when inputs of the network are three lags of previously observed GW level, the simple NN has 51 parameters including connection weights and biases. These are the parameters which must be optimized to minimize the predictive error of the network.

2.2. Long short-term memory neural network (LSTM-NN)

In a recurrent neural network (RNN) in addition to adjacent layers, the cells of each layer are connected to other cells of the same layer and may have self-feedback connections where one of the inputs to a cell at time step t is the output of the cell at time step $t-1$. An LSTM network is a partially connected RNN made of LSTM cells. Each LSTM cell consists of a memory (gm), input (gin), output (gout), and forget (gf) gates. The output $y_{i,j}^{(t)}$ of every LSTM-NN cell in feed-forward networks is calculated as follows (Staudemeyer and Morris, 2019):

$$y_{i,j}^{(t)} = gout_{i,j}^{(t)} \times h(state_{i,j}^{(t)}) \tag{3}$$

where $gout_{i,j}^{(t)}$ = the output of the output gate of cell i in layer j at time t, $state_{i,j}^{(t)}$ = the state of cell i in layer j at time t, $h(.)$ represents the hyperbolic tangent function (calculated by Equation (9)), and $state_{i,j}^{(t)}$ denotes the current state of cell i in layer j at time t, which is given by:

$$state_{i,j}^{(t)} = gf_{i,j}^{(t)} \times state_{i,j}^{(t-1)} + gm_{i,j}^{(t)} \times gin_{i,j}^{(t)} \tag{4}$$

The input, forget, output, and memory gates of cell i in layer j at time t are respectively denoted by $gin_{i,j}^{(t)}$, $gf_{i,j}^{(t)}$, $gout_{i,j}^{(t)}$, and $gm_{i,j}^{(t)}$, and are calculated as follows:

$$gin_{i,j}^{(t)} = f(\sum_l \left[W_{i,j}^{in}(l, j-1) \times y_{l,j-1}^{(t)}\right] + \sum_l \left[W_{i,j}^{in}(l,j) \times y_{l,j}^{(t-1)}\right] + b_{i,j}^{in}) \tag{5}$$

$$gf_{i,j}^{(t)} = f(\sum_l \left[W_{i,j}^{f}(l, j-1) \times y_{l,j-1}^{(t)}\right] + \sum_l \left[W_{i,j}^{f}(l,j) \times y_{l,j}^{(t-1)}\right] + b_{i,j}^{f}) \tag{6}$$

$$gout_{i,j}^{(t)} = f(\sum_l \left[W_{i,j}^{out}(l, j-1) \times y_{l,j-1}^{(t)}\right] + \sum_l \left[W_{i,j}^{out}(l,j) \times y_{l,j}^{(t-1)}\right] + b_{i,j}^{out}) \tag{7}$$

$$gm_{i,j}^{(t)} = h(\sum_l \left[W_{i,j}^{m}(l, j-1) \times y_{l,j-1}^{(t)}\right] + \sum_l \left[W_{i,j}^{m}(l,j) \times y_{l,j}^{(t-1)}\right] + b_{i,j}^{m}) \tag{8}$$

in which, $y_{i,j}^{(t)} =$ the output of cell *i* of layer *j* at time *t*, $W_{i,j}^{in}(l,j)$, $W_{i,j}^{f}(l,j)$, $W_{i,j}^{out}(l,j)$, and $W_{i,j}^{m}(l,j)$ = the weights of the connection from cell *l* of layer *j* to the input, forget, output, and memory gates of cell *i* of layer *j*, respectively; $b_{i,j}^{in}$, $b_{i,j}^{f}$, $b_{i,j}^{out}$, and $b_{i,j}^{m}$ = the bias of the input, forget, output, and memory gates of cell *i* of layer *j*, respectively and $f(.)$ denotes the sigmoid activation function introduced in equation (2). Notice that unlike other gates the memory gate's activation function is the hyperbolic tangent function instead of the sigmoid function.

The hyperbolic tangent function $h(.)$ is defined as follows:

$$h(x) = \frac{exp(x) - exp(-x)}{exp(x) + exp(-x)} \tag{9}$$

Figure 2 depicts a schematic of the applied LSTM-NN in the current study. The applied LSTM-NN has one input cell, a hidden layer which consists of 10 LSTM cells, and one output cell with the identity activation function. Unlike the simple NN which needs *P* input cells to receive *P* lags of previously observed GW level, the LSTM-NN has only one input cell regardless of the number of lags, and the LSTM-NN receives the previous lags as a sequence. In fact, unlike the simple-NN, the feedforward of the LSTM-NN is a temporal process as shown in Figure 2. The outputs of the LSTM cells are not passed to the output cell of the network after the network receives the first input (i.e., the oldest lag, which here is Lag *P*). Instead, the outputs of the LSTM layer are received by itself in addition to the next input (i.e., Lag (*P*-1)). This procedure continues until the network receives the last input (which is lag 1). At this stage the outputs of the LSTM cells are passed to the output cell of the network. Notice that in general an LSTM network may have more than one input and output cell. This study applies one input cell because there is one kind of data (the GW level timeseries) used as the input to the model. Also, one output cell is used because one lag GW level is forecasted. Figure 3 depicts the structure of a single LSTM cell of the applied LSTM-NN and its connections. It is seen in Figure 3 that an LSTM cell has several gates each of which are directly connected to the input cell and other LSTM cells. In fact, each of these gates is a cell with its own activation function, connection weights, and biases. The state of an LSTM cell is stored inside the cell and contributes to the next output of the cell as shown in Figure 3 and Equation (4). The specific structure of the LSTM cells allows temporal dependencies to be captured. Such a characteristic makes an LSTM network an ideal candidate for the task of timeseries prediction (for further reading about the LSTM architecture, see Staudemeyer and Morris, 2019).

Regardless of the number of input lags, the LSTM-NN has only one input cell; therefore the number of parameters of the LSTM-NN does not depend on *P* (the number of lags of GW level observation) unlike the simple NN. The applied LSTM-NN in this study has 10 LSTM cells, one input cell, and one output cell, therefore the total number of parameters including weights and biases is a fixed value and is equal to 491 (number of LSTM cells × 4 × [number of input cells + number of LSTM cells + 1] + number of cells in the output layer × [number of LSTM cells + 1]). Among two identical networks (one simple and one LSTM) with the same number of cells the LSTM network has a greater number of parameters because every LSTM cell has four gates each of which is connected to all of the other cells in the same layer and the previous layer while a simple cell only is connected to the cells of the previous layer. This increases the number of parameters which must be optimized for the LSTM network. With the recent advanced techniques for training neural networks optimizing networks of such size with several hundreds of parameters is straighforward. Thus, such an increase in the number of parameters does not pose a serious computational burden. On the other hand, however, the LSTM-NN has an advantage over the simple NN when there are long-term dependencies in the timeseries, and many lags are required as the input to the network for accurate prediction. In such a case, the independency of the number of parameters of the LSTM to the length of the input sequence provides better scalability.

2.3. Adam optimizer for the training phase

Training of a neural network requires that its parameters (weights, biases) be optimized to minimize the network's prediction error. The training objective function is to minimize the mean squared error (MSE) defined below:

$$MSE = F(\theta) = \frac{\sum_{t=1}^{N}\left(z_t - \hat{z}_t(\theta)\right)^2}{N} \quad (10)$$

in which, $N =$ the number of observations of a phenomenon under study (say, the groundwater level), z_t = the observed value at time *t*, $\hat{z}_t(\theta)$ = the predicted value at time *t*, it is the output of the neural network and a function of the parameter vector θ, where $\theta = (\theta_1, \theta_2, \ldots, \theta_l, \ldots, \theta_M)$ and $\theta_l = l^{th}$ parameter of the network to be optimized, and $M =$ the total number of trainable parameters of the network. During training of the parameters (coefficients) of the network $F(\theta)$ is treated as a stochastic function evaluated at batches of the train data set. This means that the training data set is divided into several batches (subsets) randomly and the parameters are updated based on the partial evaluation of the objective function in each batch.

This study applies the Adam optimizer to obtain the neural network's parameters. The Adam optimizer is a stochastic gradient-based optimization algorithm introduced by Kingma and Ba (2015). It differs from traditional stochastic gradient descent (SGD) algorithms in that it assigns adaptive individual learning rates to each neural network parameter separately and updates them based on the estimates of the first and second moments of the gradients. Traditional SGD algorithms, on the other hand, use a single learning rate for all the neural network's parameters. The Adam optimizer has proven successful in solving deep machine learning problems and domains with sparse gradients. It is therefore a suitable choice when working with LSTM networks that commonly have several times more parameters than simple neural networks. Optimization of such networks with traditional SGDs is computationally burdensome, while the Adam optimizer is more effective for the task.

The Adam optimizer has several hyperparameters. These are the step size (α), exponential decay rates for the estimates of moments (β_1 and β_2), and a small value (ε), which must be specified. The recommended initial hyperparameters are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and $\varepsilon = 10^{-8}$ (Kingma and Ba, 2015). The algorithm starts by initializing the vector θ (neural network parameters), and 1st moment vector ($m^{(0)}$), 2nd moment vector ($v^{(0)}$), and iteration (s) by setting them equal to zero. The moments and vector θ (neural network parameters) are updated iteratively until the algorithm converges. At the beginning of each iteration the gradient vector is calculated as follows:

$$g^{(s)} = \nabla_\theta F(\theta^{(s-1)}) \tag{11}$$

in which, $g^{(s)}$ = gradient vector at iteration s with respect to the objective function F, $\theta^{(s-1)}$ = parameter vector at iteration $s-1$, and ∇_θ denotes the gradient vector obtained by differentiating the objective function F with respect to the components of the parameter vector θ.

Next, the first and second moments are updated as follows:

$$m^{(s)} = \beta_1 \cdot m^{(s-1)} + (1-\beta_1) \cdot g^{(s)} \tag{12}$$

$$v^{(s)} = \beta_2 \cdot v^{(s-1)} + (1-\beta_2) \cdot {g^{(s)}}^2 \tag{13}$$

in which, $m^{(s)}$and $v^{(s)}$ denote respectively the first and second moments at iteration s, and ${g^{(s)}}^2 = g^{(s)} \odot g^{(s)}$ ($\odot$ denotes the Hadamard product, which involves elementwise multiplication). Note that all operations on vectors are elementwise.

An iteration updates the parameters as follows:

$$\theta^{(s)} = \theta^{(s-1)} - \alpha \cdot \frac{\widehat{m}^{(s)}}{\sqrt{\hat{v}^{(s)}} + \varepsilon} \tag{14}$$

$$\widehat{m}^{(s)} = \frac{m^{(s)}}{1-\beta_1^s} \tag{15}$$

$$\hat{v}^{(s)} = \frac{v^{(s)}}{1-\beta_2^s} \tag{16}$$

in which, $\theta^{(s)}$ = updated vector of neural network parameters at iteration s, β_1^s and β_2^s denote β_1 and β_2 to the power s, respectively. The Adam optimizer terminates when a specific number of epochs are completed; otherwise, the updated parameters are used as initial values to start the next iteration. In every epoch the parameters are updated with respect to the whole train data set.

Every epoch consists of several data batches each of which is a subset of the train dataset. Therefore, every iteration of the Adam optimizer uses a batch (sample) of the train dataset to update the parameters. The Adam optimizer performs step size annealing. When the ratio $\frac{\hat{m}^{(s)}}{\sqrt{\hat{v}^{(s)}}}$ is small, the step size is small. For instance, close to optimal parameter values the magnitude of the aforementioned ratio tends to zero resulting in small step sizes for the updating of parameters. For further reading about the Adam optimizer and its convergence properties see Kingma and Ba (2015).

2.4. Scenarios

This work evaluates the performance of the applied neural networks (the LSTM-NN and the simple NN) under three scenarios named D, MA, and MM, and several prediction horizons corresponding to each scenario. Scenario D refers to the situation when daily data are available. In this instance the networks are trained using daily data. Scenarios MA and MM refer to the situation in which monthly average GW level data and monthly minimum data are available, respectively. In all of the cases the inputs and outputs of the networks are consistent; for example, if the LSTM-NN forecast monthly groundwater level this means the input to the model is a set of monthly groundwater observations, and, furthermore, the LSTM-NN is trained with monthly data.

Scenario D is employed to evaluate the performance of the networks in predicting the GW level from one day (step) ahead to 30 days (steps) ahead. In the case of monthly scenarios the performance of the models is evaluated with one, two- and three steps (months) ahead predictions. Also, the number associated with each scenario refers to the prediction horizon. Thus, scenarios D1 and D20 denote one-day and 20-day ahead predictions, respectively; or scenario MA2 refers to two-month ahead prediction of monthly average GW level.

3. Evaluation criteria

The following criteria are applied to test the LSTM-NN's and simple NN's performances in forecasting the GW level. The Nash-Sutcliffe model efficiency coefficient (NSE) was also calculated and it was always equal to the coefficient of determination (R^2). Therefore, only the R^2 is reported.

3.1. Coefficient of determination (R^2)

The coefficient of determination (R^2) measures the level of statistical association between the observed and predicted time series. The value of this criteria varies from zero to one. The higher the value of R^2, the better the prediction, with a value equal to one indicating a perfect fit between observed and predicted time series. An R^2 equal to zero means a lack of association between predictions and observation. R^2 is calculated as follows:

$$R^2 = 1 - \frac{\sum_{t=1}^{N}(z_t - \hat{z}_t)^2}{\sum_{t=1}^{N} z_t{}^2 - \frac{\sum_{t=1}^{N} \hat{z}_t{}^2}{N}} \tag{17}$$

3.2. Mean squared error (MSE)

The MSE measures the discrepancy between the observed and predicted time series. In this study the LSTM-NN and the simple NN are trained to minimize the MSE (Equation (10)). A lower MSE implies a better prediction. Although during training phase in order to optimize the parameters (coefficients) of the network, $F(\theta)$ was treated as a stochastic function evaluated at batches of the train data set, in the results section the reported MSE is calculated with respect to the whole data set (encompassing the testing and training data sets).

3.3. Mean absolute error (MAE)

The mean absolute error is used to measure the accuracy of the time series predictions. Unlike the MSE the MAE avoids the contribution of large errors to the value of the index. The MAE is calculated as follows:

$$MAE = \frac{\sum_{t=1}^{N}|z_t - \hat{z}_t|}{N} \tag{18}$$

4. Case study: the Edwards aquifer

This work evaluated the performances of the LSTM-NN and simple NN in forecasting GW level in the Edwards (Balcones Fault Zone region) aquifer. The reason for this choice of aquifer was the availability of a long-term daily GW level time series dating back to 1932, the regional water-supply, and ecologic importance of the Edwards aquifer. The Edwards aquifer is located in south-central Texas and its catchment area encompass all or part of 13 counties in south-central Texas, USA. The hydrogeologic and groundwater management of the Edwards aquifer have been described in detail in several studies (see, e.g., Loáiciga et al. 2000; Loáiciga 2017, 2019; Sharp et al., 2019). The Edwards aquifer is a highly productive, confined, karst aquifer which has an upstream drainage area, a recharge (unconfined aquifer) region, a transition zone (between unconfined and confined conditions), and a confined zone. The Edwards aquifer encompasses an area approximately 290 km long and its width varies from 8 to 65 km. The Edwards aquifer is the primary water source for the City of San Antonio and various neighboring areas. The average annual discharge to springs plus groundwater withdrawal by pumping wells is approximately equal to 57,419 $\times 10^6$ m^3, of which 49.94, 28.02, 13.64, 4.23, and 4.16 percent are allocated to spring discharge, and to meet municipal and military, irrigation, domestic and livestock, and industrial water demands, respectively. Also, groundwater rights and the effect of groundwater withdrawal on several native species of animals and plants in the Edwards aquifer have been contentious over decades. The degradation of aquatic ecosystems in the Edwards aquifer caused by groundwater withdrawal led to the listing of some endemic species (i.e., the Fountain darter, the Comal Springs riffle beetle, the Texas blind salamander, etc.) as endangered by the US Fish and Wildlife Service.

5. Data

The GW level data used in this study are maximum daily water level at index well J-17 in San Antonio, Texas, recorded from 11/12/1932 until 7/31/2020 (in total 31,239 days) reported by The Edward aquifer authority (EAA, https://www.edwardsaquifer.org/eaa/). The EAA archives the highest water level observed every day. The EAA reports GW level in feet above mean sea level (msl). Prior to using the data for training the networks and prediction, the data were standardized (Standardized data = [the original data – the mean of the data] / the standard deviation of the data). The entire available GW level data were used for daily scenarios (scenario D); in the case of monthly scenarios, the minimum and average of daily GW levels were calculated for every month to create two distinct monthly GW level time series. The partial data (i.e., less than one month duration) for the last and first month were removed and the rest of the data were used for prediction with the monthly scenarios (Scenarios MA and MM). Therefore, the data period for monthly scenarios started in December 1932 and ended in July 2020 (1,043 months in total). Figures 4 and 5 depict the available data for the daily and monthly scenarios, respectively. The available data were divided into training and testing data sets such that the older 80 percent of the data were applied for training and the most recent 20 percent of the time series were used for testing.

6. Results

The proposed LSTM-NN's predictive skill was evaluated and compared with that of the simple NN. Three goodness-of-fit criteria (R^2, MSE, and MAE) were employed for the evaluative comparison. First, the effect of the number of lags of previously observed GW level as the input to the simple NN on its performance was evaluated. This was done to ensure that the simple NN would achieve the best possible results. The simple NN with three past lags of GW level observations as input provided the best result among a set of candidates including 1, 2, 3, 4, 5, 6, 12, and 24 previous lags for monthly scenarios and 1, 2, 3, 4, 5, 6, 7, 14, 21, and 30 previous lags for daily scenarios. The same test applied to the LSTM-NN demonstrated that the past three lags performed well for the LSTM-NN, also. There are a few cases when other numbers of previous lags (e.g., six lags) performed better as the input to the LSTM-NN; yet, the improvement was negligible. Therefore, three lags of previously observed GW level were initially selected as the input to the neural networks to provide a baseline for their comparison. Also, the performance of the LSTM-NN is compared with that of the simple NN when the past 47 lags are used as the input. In this case the simple NN has 47 input cells, and, consequently its total number of parameters is 491 which is equal to the number of parameters of the LSTM-NN. This comparison is made to study the performance of the networks when they have the same number of parameters, number of hidden cells, and they receive the same inputs.

Table 1 lists the results corresponding to the training and testing data sets for each NN with respect to the monthly scenarios, and selected daily scenarios D1, D10, D20, and D30 where three previous lags are the inputs of the NNs. The accuracy of the LSTM-NN is consistently better than that of the simple NN by several percentages with respect to the goodness-of-fit criteria. For example, for one-day ahead prediction under scenario D1 the R^2 of the LSTM-NN prediction for the test data set is equal to 99.89% whereas the R^2 of the simple network is 95.32%, demonstrating about 4.5% improvement in the accuracy of prediction. The LSTM-NN's R^2 for the testing data set under scenarios D10, D20, and D30 is at least 4% better than those of the simple NN. It is seen in Table 1 that the same pattern exists for monthly scenarios. For instance, the LSTM-NN's R^2 for scenarios MA1 (one-month ahead prediction of the average GW level) and MM1 (one-month ahead prediction of minimum GW level) with the testing data set are at least 8% better than those of the simple NN. The results listed in Table 1 indicate that on average the R^2 of the LSTM-NN for testing data set features about a 5% improvement over the simple-NN. The MSE and MAE produced goodness-of-fit results consistent with those of the R^2, such that the MSE and MAE of the LSTM-NN have lower values (i.e., they indicate better predictive skill) than those of the simple-NN. For instance, the MSE of the LSTM-NN for the testing data set under scenario D1 is equal to two percent (2%) of the MSE of the simple-NN. The results for the training data set demonstrates the same pattern.

The number of parameters of the simple NN is smaller than that of the LSTM-NN when the inputs to the NNs are three previous lags. Therefore, the improved results might be assumed to be due to the greater number of parameters and not to the LSTM architecture. To resolve this dilemma Table 2 lists the results corresponding to the training and testing data sets for each NN with respect to the monthly scenarios, and selected daily scenarios D1, D10, D20, and D30 featuring the past 47 lags as the inputs to the NNs. The simple NN requires 47 input cells to receive 47 previous lags as the input. The simple NN with 47 input cells has 491 parameters. However, as discussed earlier, the number of parameters of the LSTM does not depend on the number of input lags. Therefore, in this case, both NNs have the same number of parameters. Nevertheless, the results listed in Table 2 show the accuracy of the LSTM-NN is always better than that of the simple NN. Also, comparing Table 1 and Table 2, it is seen that increasing the number of input lags may improve or worsen the accuracy of the NNs. For example, for scenario D1 the results of Table 2 (47 lags) are slightly better than those of Table 1 (three lags) for both NNs. However, the accuracy for scenarios MA3 and MM3 is reduced for both NNs in Table 2 in comparison to Table 1. It is noteworthy that for scenario MA3 and MM3 when 47 input lags were used (Table 2) the difference between the training and testing accuracies increased. This shows that neither the LSTM-NN's nor for the simple NN's prediction accuracy increases with increasing number of lags. We found that three previous lags perform best for the GW level time series of the Edwards Aquifer. This finding may or may not apply to other basins.

The results of the LSTM-NN for the rest of the daily scenarios (i.e., the ones that are not reported in Table 1, such as scenarios D2, D3) are presented in Table 3 for the training and testing data sets using three past lags. For the sake of brevity, the results corresponding to the simple NN are not listed considering that the LSTM-NN featured a consistently better performance than the simple-NN. It is evident from Table 3 that the LSTM-NN's predictive skill of the daily GW level is good up to 30-day ahead predictions. Specifically, the LSTM-NN predicts up to four-day ahead GW level with an R^2 above 99%. One-week predictions features an R^2 as high as 98.26%. Two- week ahead predictions achieved an R^2 above 95%; and predictions of the GW level up to 26-day ahead achieved an R^2 of at least 90%. Therefore, it can be stated that the LSTM-NN successfully predicts the GW level of the Edwards aquifer several-day ahead with high accuracy.

Figures 6, 7, and 8 display the observed and predicted time series calculated with the LSTM-NN and the simple NN for the testing data set corresponding respectively to daily scenarios (D1, D10, D20, and D30), and average monthly scenarios (MM1 through MA3), and minimum monthly scenarios (MM1 through MM3) using three previous input lags. Figure 6 demonstrates a good fit achieved by the LSTM-NN, whereas the simple NN frequently failed to render accurate predictions of the time series. It is seen in Figures 7 and 8 that, although the results of the monthly predictions are not as good as that of daily scenarios, the results of the LSTM-NN network are consistently better than those of the simple-NN; furthermore, the predictive skill of the simple-NN decreases with increasing length of the prediction horizon. It is noteworthy that the length of the available data for the monthly scenarios was shorter than that of the daily scenarios. Even for a long prediction horizon (e.g., MA3) the LSTM-NN predicts the monthly GW level well but with a slight shift to the right.

The improvement of the LSTM-NN over the simple NN is critical. It is seen in Figure 6 that the simple NN produced poor prediction of extreme events (peaks and troughs) of the GW levels. However, the LSTM-NN predicted all the extreme events of the GW levels observed during a testing period of 17 years with a very good accuracy even up to 30 days ahead (for scenario D30). The performance of the LSTM-NN demonstrated in this work is noteworthy considering that this work applied in-situ observed GW level records without using any other kind of predictive data, such as precipitation and pumping rates, or other hydrogeologic information about the groundwater basin. The Edwards aquifer is subjected to various stresses, such as groundwater withdrawal and recharge-zone land-use changes, and its geohydrology is complex. Nevertheless, the LSTM-NN successfully predicted the GW level in the training stage and with high accuracy in the testing phase without relying on any information about anthropogenic factors (e.g., population, water usage, etc.) or hydrologic factors (e.g., precipitation, runoff, spring discharge, groundwater withdrawal, etc.), and only by applying a long data set of observed GW levels. This work's results demonstrate the importance of long-term, multi-decadal, groundwater monitoring for the purpose of constructing accurate predictive machine learning methods.

7. Conclusion

This work introduced the LSTM-NN for GW level forecasting, and compared the results of the LSTM-NN with those achieved by a simple NN in predicting long-term and short-term GW level in the Edwards aquifer, Texas. The predictive skill of the NNs was evaluated under multiple daily, monthly average, and monthly minimum scenarios considering several prediction horizons and data availability. The goodness-of-fit criteria R^2, MSE, and MAE established that the performance of the LSTM-NN was superior to that of the simple-NN. For example, it was shown that the R^2 of the LSTM-NN was on average about 5% superior to that of the simple-NN. Also, The LSTM-NN was able to predict one day, up to four days, and up to 26 days ahead GW level with an accuracy (R^2) of at least 99.89%, 99.00%, and 90.00%, respectively. This level of predictive skill was achieved using the GW level as the only input to the NNs without resorting to any other kind of data and information about the groundwater basin. It is noteworthy that the Edward aquifer is subjected to various stresses, such as groundwater withdrawal and recharge-zone land-use changes, and its geohydrology is complex. This successful application of machine learning to GW level prediction for such a complex basin emphasizes the importance of gathering high quality and long-term GW level data. In addition, the rising awareness worldwide for the need of accurate and long-term groundwater monitoring creates an ideal juncture for resorting to machine learning algorithms to support decision making in groundwater management. Lastly, this work has revealed that long-term GW level data serves as a footprint of hydrologic and anthropogenic influence in groundwater basins.

8. Acknowledgement and Data Availability Statement

The authors thank the Department of Geography, University of California Santa Barbara (UCSB) for its financial support of this research.

The data used in this study is open access and is provided by the Edward aquifer authority (EAA, https://www.edwardsaquifer.org/eaa/).

9. References

Adiat K. A. N., Ajayi, O. F., Akinlalu, A. A., and Tijani, I. B. (2020). "Prediction of groundwater level in basement complex terrain using artificial neural network: a case of ljebu-Jesa, southwestern, Nigeria." Applied Water Science, 10(8).

Bows, B. D., Sadler, J. M., Morsy, M. M., Behl, M., and Goodall, J. L. (2019). "Forecasting groundwater table in a flood prone coastal city with long short-term memory and recurrent neural networks." Water, 11, 1098, doi: 10.3390/w11051098.

Bozorg-Haddad, Solgi, M., and Loáiciga, H. A. (2017). "Investigation of climatic variability with hybrid statistical analysis." Water Resources Management, 31(1), 341-353.

Chen L. H., Chen, C. T., and Lin, D. W. (2011). "Application of integrated back-propagation network and self-organizing map for groundwater level forecasting." Journal of Water Resources Planning and Management, 137(4), 352-365.

Chen, L. H., Chen, C. T., and Pan, Y. G. (2010). "Groundwater level prediction using SOM-RBFN multisite model." Journal of Hydrologic Engineering, 15(8), 624-631.

Coulibaly, P. Anctil, F., Aravena, R., and Bobee B. (2001). “Artificial neural network modeling of water table depth fluctuations.” Water Resources Research, 37(4), 885-896.

Hipel, K. W., and McLeod, A. I. (1994). “Time series modeling of water resources and environmental systems.” Elsevier, Amsterdam.

Hochreiter, S. and Schmidhuber, J. (1997). “Long short-term memory.” Neural Computation, 9, 1735-1780.

Houdt, G. V., Mosquero, C., and Napoles, G. (2020). “A review on the long short-term memory model.” Artificial Intelligence Review, doi: 10.1007/s10462-020-09838-1.

Kelleher, J.D, Tierney, B. (2018). Data Science. The MIT Press, Cambridge, MA.

Khedri, A., Kalantari, N., and Vadiati, M. (2020). “Comparison study of artificial intelligence method for short term groundwater level prediction in the northeast Gachsaran unconfined aquifer.” Water Supply, 20(3), 909-921.

Kingma, D. P. and Ba, J. L. (2015). “Adam: a method for stochastic optimization.” International Conference on Learning Representations, San Diego, CA, US, May 7-May 9.

Kisi, O. and Shiri, J. (2012). “Wavelet and neuro-fuzzy conjunction model for predicting water table depth fluctuations.” Hydrology Research, 43(3), 286-300.

Loáiciga, H. A. (2017). “The safe yield and climatic variability: Implications for groundwater management.” Groundwater Journal, 55(3), 334–345.

Loáiciga, H. A., and Schofield, M. (2019). “Climate variability, climate change, and Edwards Aquifer water fluxes.” *in* Sharp, J. M., Jr., Green, R. T., and Schindel, G. M., eds. “The Edwards aquifer: the past, present, and future of a vital water resource.” Geological Society of America Memoir 215, 223–237.

Loáiciga, H. A., Maidment, D. R., and Valdes, J. B. (2000). “Climate-change impacts in a regional karst aquifer, Texas, USA.” Journal of Hydrology, 227, 173–194.

Maheswaran, R. and Khosa, R. (2013). “Long term forecasting of groundwater levels with evidence of non-stationary and nonlinearity characteristics.” Computers and Geosciences, 52, 422-436.

Orsborn J. F. (1966). “The prediction of piezometric levels in observation wells based on prior occurrences.” Water Resources Research, 2(1).

Rajaee, T., Ebrahimi, H., and Nourani, V. (2019). “A review of the artificial intelligence methods in groundwater level modeling.” Journal of Hydrology, doi: https://doi.org/10.1016/j.jhydrol.2018.12.037.

Sahoo, S. and Jha, M. K. (2013). “Groundwater level prediction using multiple linear regression and artificial neural network techniques: a comparative study.” Hydrogeology Journal, 21, 1865-1887.

Sahoo, S., Russo, T. A., Elliott, J., and Foster, I. (2017). “Machine learning algorithms for modeling groundwater level changes in agricultural regions of the US.” Water Resources Research, 53, 3878-3895.

Sharp, J.M., Green, R.T., Schindel, G.M. (2019). The Edwards Aquifer: the past, present, and future of vital resource. The Geological Society of America, Memoir 215, Boulder, Colorado, USA.

Solgi, M., Bozorg-Haddad, O., and Loáiciga, H. A. (2016a). “The enhanced honey-bee mating optimization algorithm for water resources optimization.” Water Resources Management, 31, 885-901.

Solgi, M., Bozorg-Haddad, O., and Loáiciga, H. A. (2020). “A multi-objective optimization model for operation of water distribution networks.” Water Supply, 20(7), 2630-2647.

Solgi, M., Bozorg-Haddad, O., Seifollahi-Aghmiuni, S., and Loáiciga, H. A. (2015). “Intermittent operation of water distribution networks considering equanimity and justice principles.” Journal of Pipeline Systems Engineering and practice, 6(4), 04015004.

Solgi, M., Bozorg-Haddad, O., Seifollahi-Aghmiuni, S., Ghasemi-Abiazani, P., and Loáiciga, H. A. (2016b). “Optimal operation of water distribution networks under water shortage considering water quality.” Journal of Pipeline Systems Engineering and Practice, 7(3), 04016005.

Staudemeyer, R. C. and Morris, E. R. (2019). “Understanding LSTM-a tutorial into long short-term memory recurrent neural networks.” ArXiv, abs/1909.09586.

Sun A. Y. (2013). “Predicting groundwater level changes using GRACE data.” Water Resources Research, 49, 5900-5912.

Suryanarayana, C., Sudheer, C. Mahammood, V., and Panigrahi, B. K. (2014). “An integrated wavelet-support vector machine for groundwater level prediction in Visakhapatnam, India.” Neurocomputing, 145, 324-335.

Takafuji, E. H. D. M., Rocha, M. M. D., and Manzione, R. L. (2019). “Groundwater level prediction/forecasting and assessment of uncertainty using SGS and ARIMA models: a case study in the Bauru Aquifer System (Brazil).” Water Resources Research, 28(2), 487-503.

Wunsch, A., Liesch, T., and Broda, S. (2018). “Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX).” Journal of Hydrology, 567, 743-758.

Yakowitz, S. (1976). “Model-free statistical methods for water table prediction.” Water Resources Research, 12(5).

Yang, Q., Hou, Z., Wang, Y., Zhao, J., and Delgado, J. (2015). “A comparative study of shallow groundwater level simulation with WA-ANN and ITS model in a coastal island of south China.” Arabian Journal of Geosciences, 8, 6583-6593.

Yang, Z. P., Lu, W. X., Long, Y. Q., and Li, P. (2009). “Application and comparison of two prediction models for groundwater levels: A case study in Western Jilin Province, China.” Journal of Arid Environments, 73, 487-492.

Zhang, J., Zhu, Y., Zhang, X., Ye, M., and Yang, J. (2018). "Developing a long short-term memory (LSTM) based model for predicting water table depth in agricultural areas." Journal of Hydrology, 561, 918-929.

Table 1. The results of LSTM-NN and Simple-NN for the training and testing data sets for all monthly and selected daily scenarios using the past three lags.

	Testing					
	LSTM-NN			Simple-NN		
Scenario	**R^2**	**MSE**	**MAE**	**R^2**	**MSE**	**MAE**
D1	99.89	0.001	0.023	95.32	0.050	0.137
D10	96.51	0.038	0.138	91.97	0.087	0.213
D20	92.01	0.086	0.212	87.83	0.131	0.270
D30	87.53	0.135	0.266	83.65	0.177	0.313
MA1	88.96	0.120	0.252	80.81	0.208	0.357
MA2	75.62	0.265	0.391	69.59	0.331	0.452
MA3	62.61	0.407	0.489	58.93	0.447	0.523
MM1	88.38	0.125	0.245	80.21	0.213	0.361
MM2	74.36	0.277	0.394	68.02	0.346	0.467
MM3	61.15	0.421	0.506	56.53	0.471	0.551
	Training					
D1	99.90	0.001	0.019	95.73	0.043	0.123
D10	97.05	0.030	0.111	92.81	0.072	0.181
D20	92.71	0.073	0.178	88.84	0.112	0.231
D30	88.34	0.117	0.231	84.86	0.152	0.274
MA1	89.47	0.106	0.228	80.68	0.194	0.320
MA2	77.39	0.227	0.352	70.29	0.299	0.405
MA3	66.23	0.340	0.439	60.80	0.395	0.471
MM1	89.61	0.104	0.226	80.14	0.199	0.328
MM2	76.71	0.233	0.358	68.33	0.318	0.423
MM3	62.80	0.374	0.461	56.84	0.434	0.501

Table 2. The results of LSTM-NN and Simple-NN for the training and testing data sets for all monthly and selected daily scenarios using past 47 lags.

	Testing					
	LSTM-NN			Simple-NN		
Scenario	**R^2**	**MSE**	**MAE**	**R^2**	**MSE**	**MAE**
D1	99.91	0.001	0.021	96.08	0.042	0.121
D10	96.26	0.040	0.143	92.58	0.080	0.201
D20	91.51	0.092	0.222	88.40	0.126	0.260
D30	85.25	0.160	0.288	84.33	0.170	0.303
MA1	87.36	0.128	0.260	71.89	0.286	0.415
MA2	70.65	0.300	0.414	57.56	0.433	0.512
MA3	47.49	0.540	0.562	44.75	0.567	0.585
MM1	86.18	0.138	0.263	71.27	0.288	0.408
MM2	67.83	0.324	0.423	56.45	0.438	0.504
MM3	38.67	0.621	0.610	43.35	0.574	0.583
	Training					
D1	99.91	0.001	0.018	96.63	0.034	0.102
D10	97.20	0.028	0.110	93.55	0.065	0.170
D20	93.28	0.067	0.175	89.53	0.105	0.221
D30	90.03	0.100	0.221	85.43	0.146	0.267
MA1	89.95	0.103	0.225	82.72	0.178	0.310
MA2	79.55	0.211	0.341	76.26	0.245	0.371
MA3	69.71	0.313	0.430	70.82	0.301	0.415
MM1	90.15	0.101	0.244	83.30	0.171	0.308
MM2	78.16	0.224	0.352	76.88	0.237	0.367
MM3	66.07	0.348	0.457	71.21	0.295	0.414

Table 3. The LSTM-NN results for the training and testing data sets under the daily scenarios using past three lags (except those listed in Table 1).

	LSTM-NN					
	Training			**Testing**		
Scenario	**R^2**	**MSE**	**MAE**	**R^2**	**MSE**	**MAE**
D2	99.73	0.003	0.033	99.67	0.004	0.041
D3	99.48	0.005	0.045	99.37	0.007	0.056
D4	99.20	0.008	0.056	99.04	0.010	0.070
D5	98.90	0.011	0.066	98.68	0.014	0.083
D6	98.59	0.014	0.075	98.33	0.018	0.093
D7	98.26	0.017	0.083	97.96	0.022	0.103
D8	97.88	0.021	0.093	97.51	0.027	0.115
D9	97.46	0.025	0.102	97.00	0.032	0.127
D11	96.62	0.034	0.119	96.02	0.043	0.147
D12	96.21	0.038	0.126	95.56	0.048	0.156
D13	95.80	0.042	0.133	95.16	0.052	0.163
D14	95.40	0.046	0.139	94.76	0.057	0.169
D15	94.97	0.050	0.146	94.29	0.062	0.178
D16	94.51	0.055	0.153	93.78	0.067	0.187
D17	94.05	0.060	0.160	93.29	0.072	0.194
D18	93.59	0.064	0.166	92.83	0.078	0.201
D19	93.14	0.069	0.172	92.39	0.082	0.207
D21	92.29	0.077	0.183	91.63	0.090	0.217
D22	91.84	0.082	0.189	91.18	0.095	0.223
D23	91.37	0.086	0.195	90.69	0.101	0.230
D24	90.90	0.091	0.201	90.22	0.106	0.236
D25	90.45	0.096	0.206	89.75	0.111	0.242
D26	90.02	0.100	0.211	89.32	0.115	0.247
D27	89.61	0.104	0.216	88.92	0.120	0.251
D28	89.21	0.108	0.221	88.51	0.124	0.255
D29	88.78	0.112	0.226	88.04	0.129	0.260

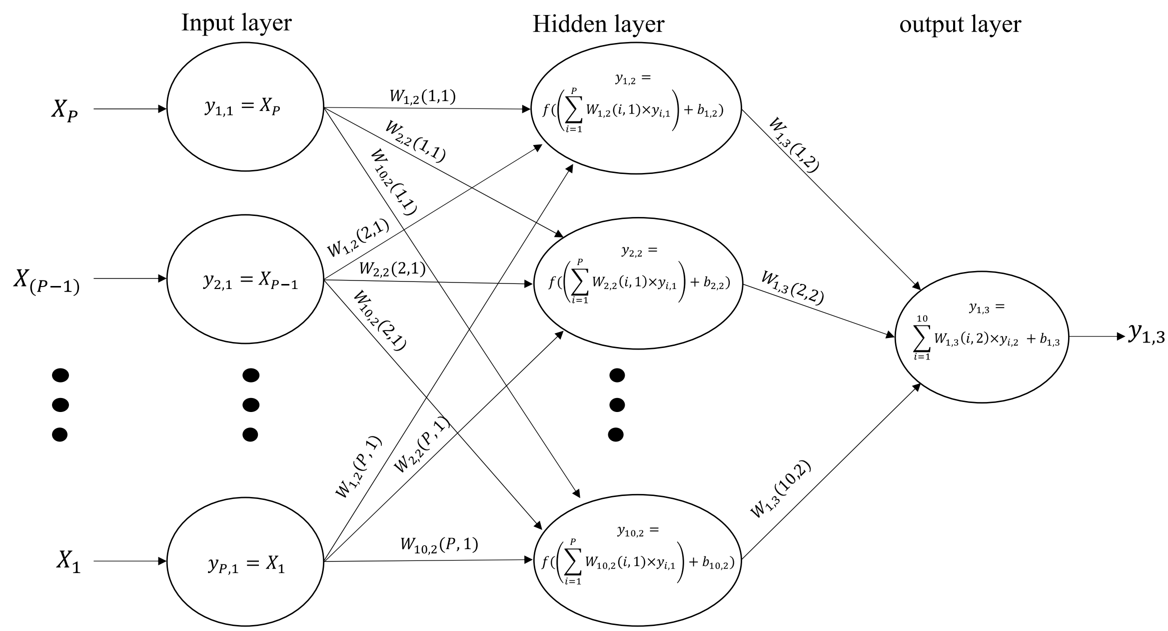


Figure 1. A schematic of a simple NN which has P input cells, one hidden layer with 10 sigmoid cells, and one sigmoid output cell (X_P is the lag P of the previous GW level observation).

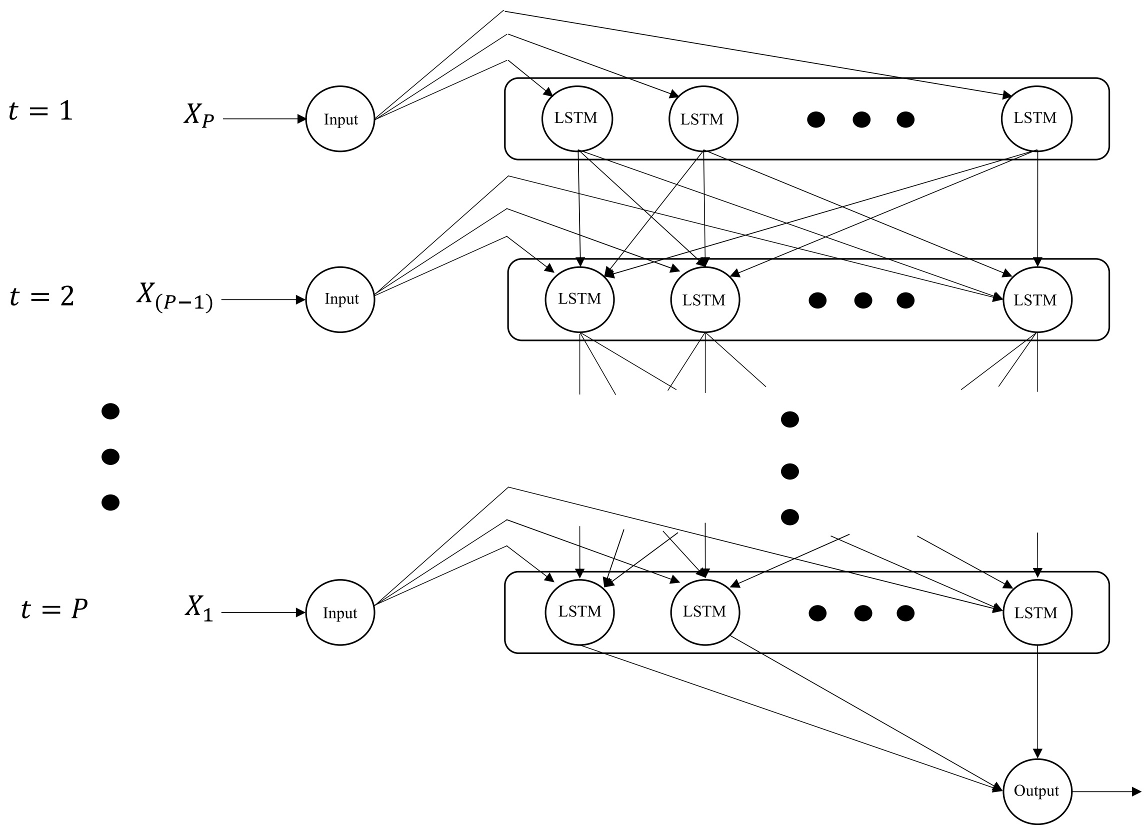


Figure 2. A Schematic of the LSTM-NN with one input cell, one hidden layer with LSTM cells, and one sigmoid output cell (X_P is the lag P of previous GW level observation) and its temporal process for receiving a sequence of past P lags and generating output.

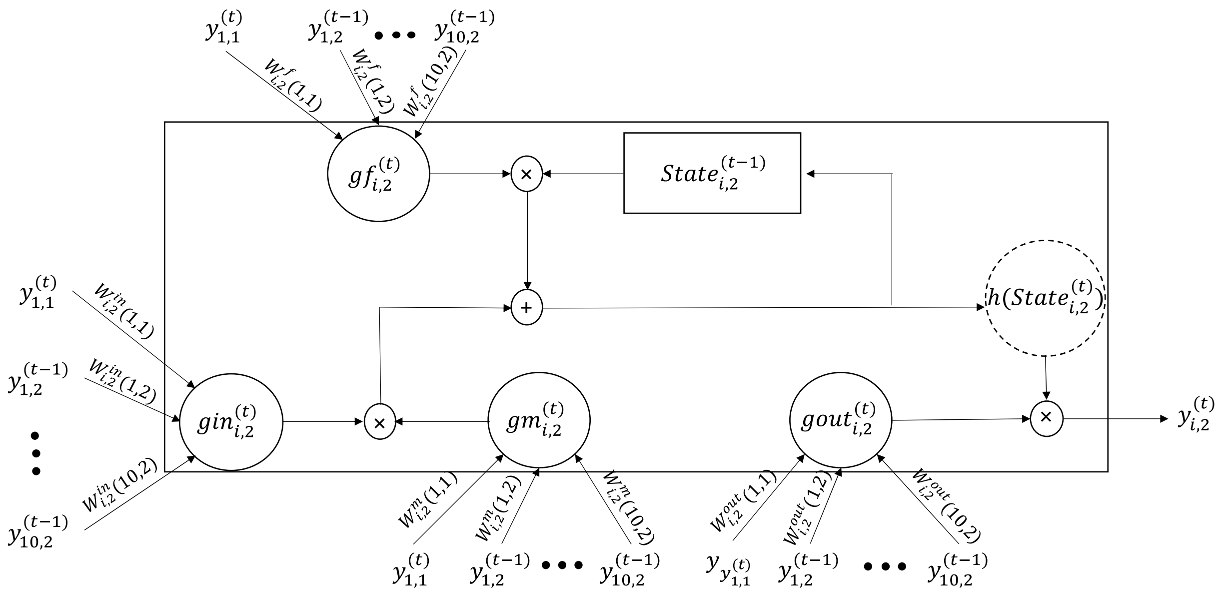


Figure 3. The architecture of *i*-th LSTM cell in the LSTM-NN whose first layer is the input layer with only one input cell (whose output at time *t* is $y_{1,1}^{(t)}$), and whose second layer is the LSTM layer with 10 LSTM cells (whose outputs are $y_{1,2}^{(t-1)}$ to $y_{10,2}^{(t-1)}$).

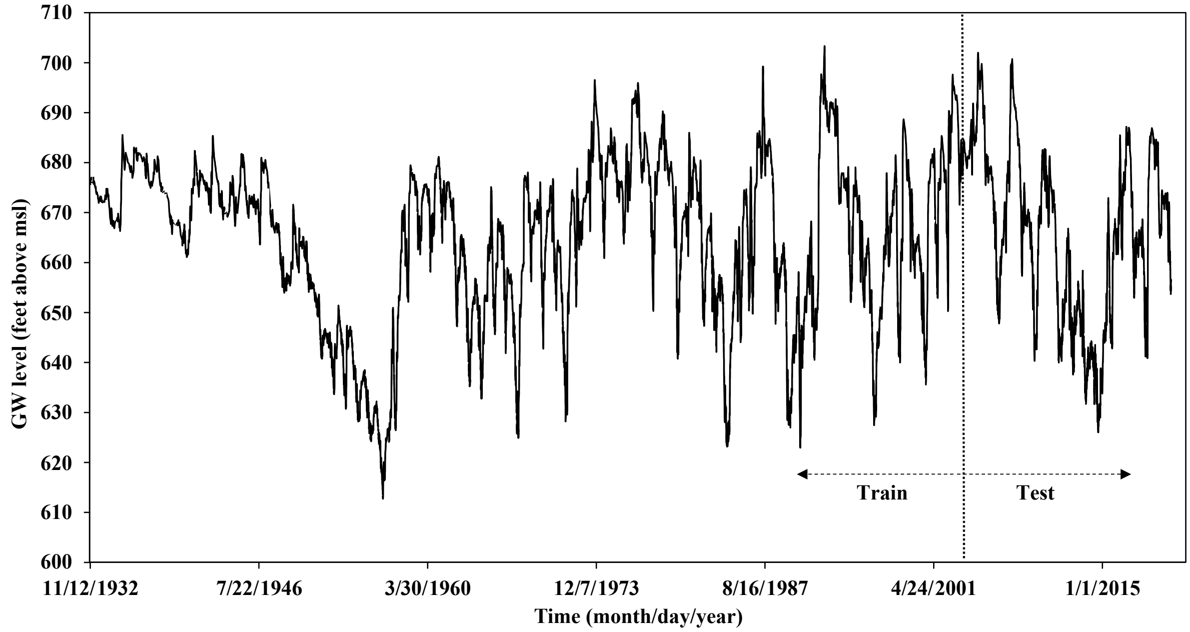
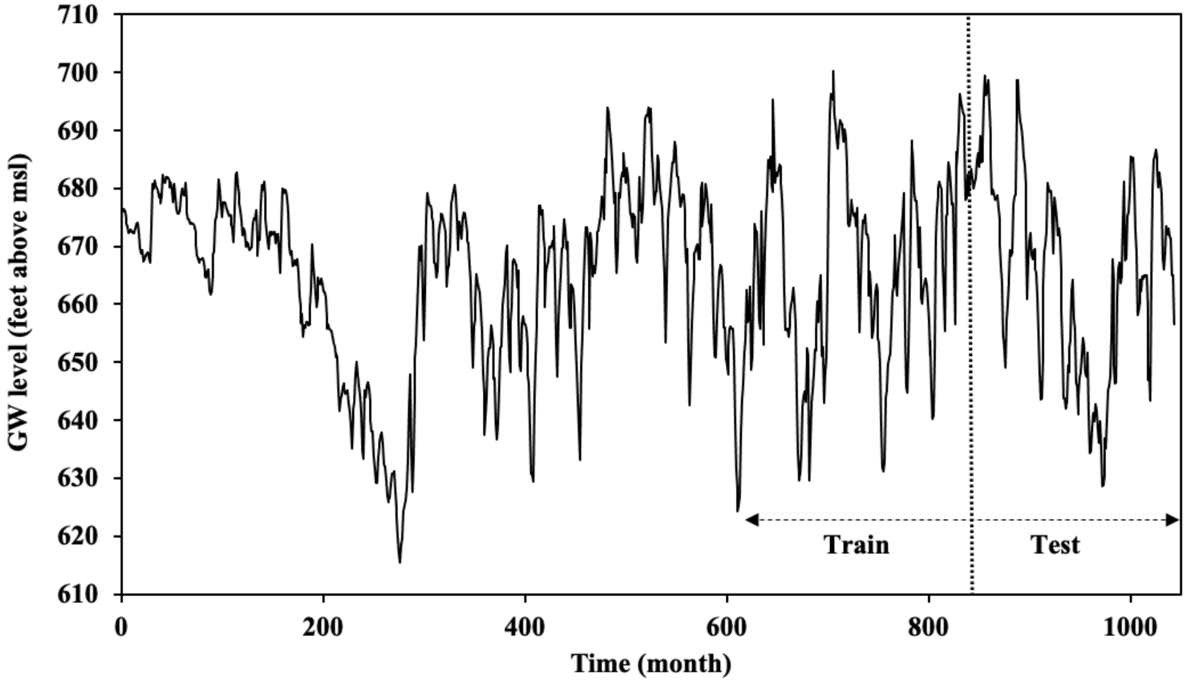
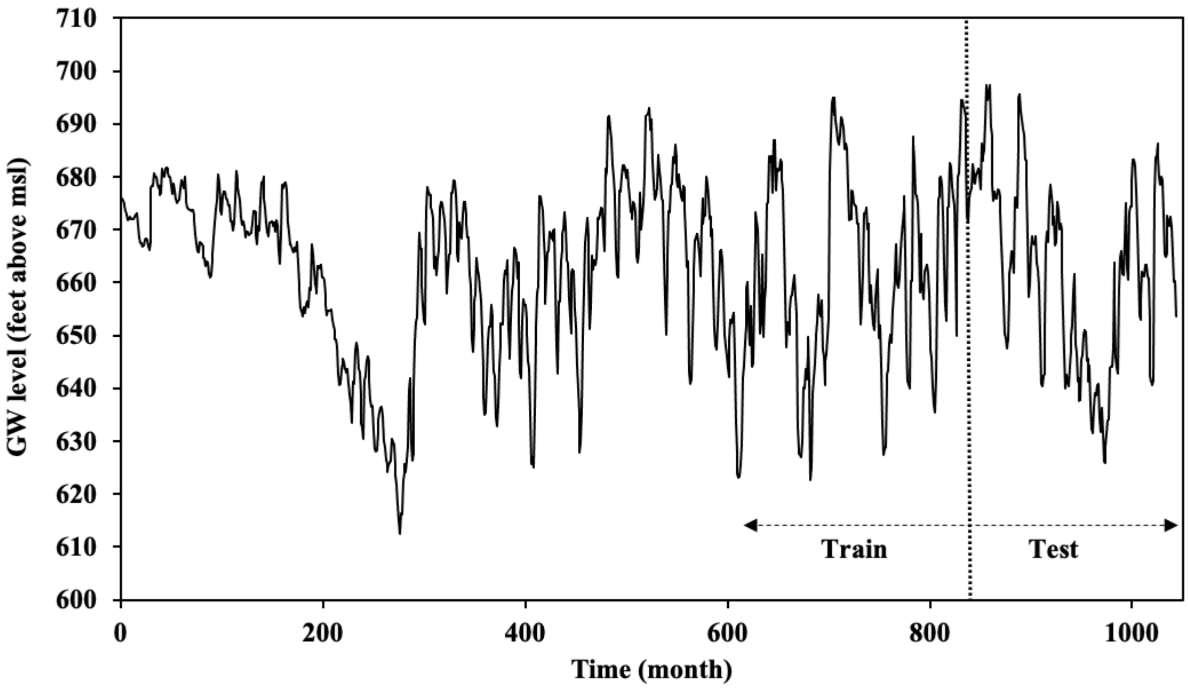


Figure 4. Daily GW level time series of the Edwards aquifer (1 foot = 0.3048 m).



(a)



(b)

Figure 5. Monthly GW level time series of Edward aquifer used for (a) monthly average (b) monthly minimum scenarios (December 1932 through July 2020), (1 foot = 0.3048 m).

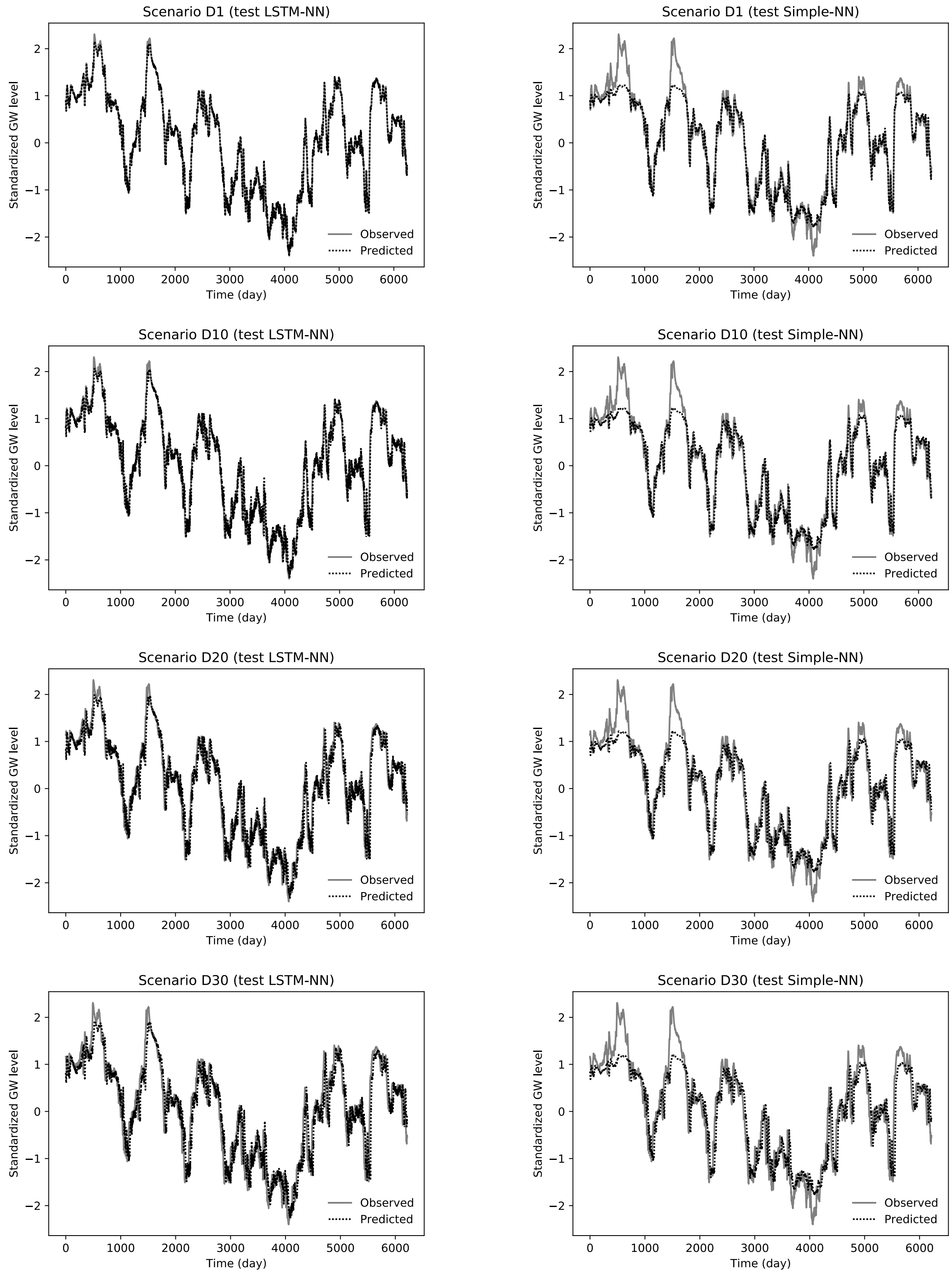


Figure 6. The results of daily GW level prediction corresponding to testing data sets achieved by the LSTM-NN (left) and the simple-NN (right) for several daily scenarios using three past lags.

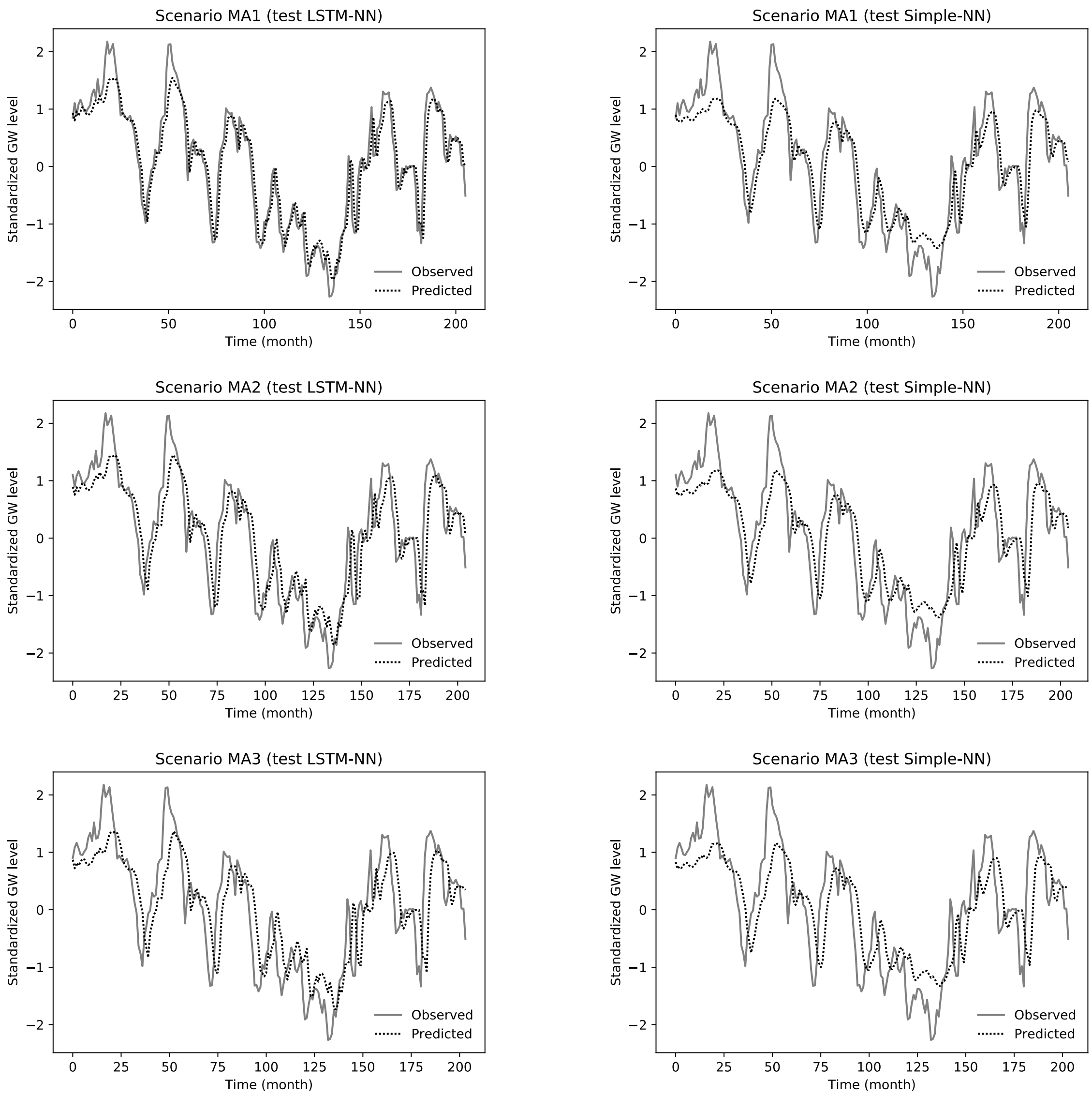


Figure 7. The results of monthly average GW level prediction corresponding to the testing data sets achieved by the LSTM-NN (left) and the simple-NN (right) under scenarios MA1, MA2, and MA3 using three past lags.

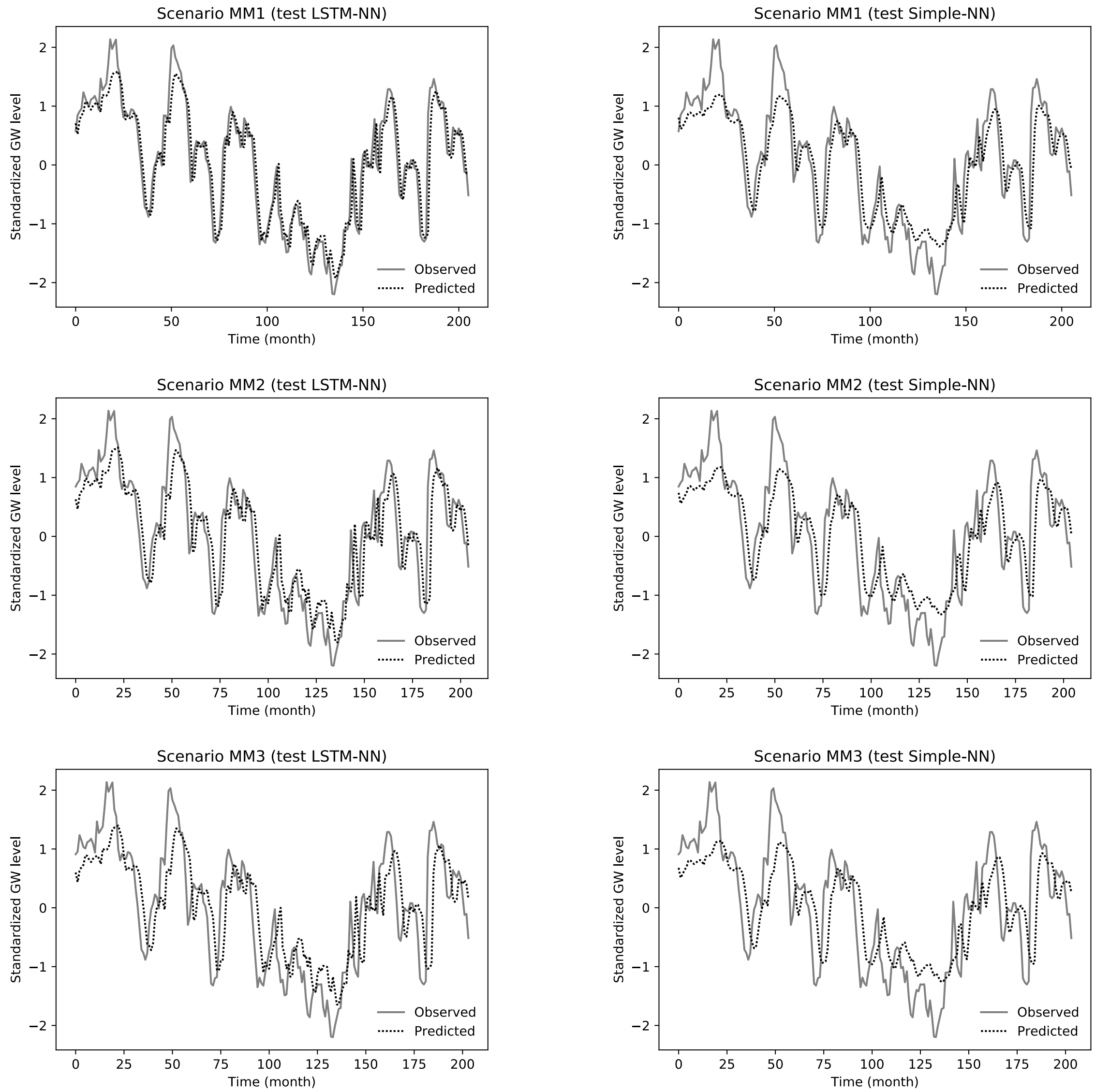


Figure 8. The result of monthly minimum GW level prediction corresponding to the testing data sets achieved by the LSTM-NN (left) and the simple-NN (right) under scenarios MM1, MM2, and MM3 using three past lags.