

## Article

# Enhancing Accuracy of Groundwater Level Forecasting with Minimal Computational Complexity Using Temporal Convolutional Network

Adnan Haider <sup>1</sup>, Gwanghee Lee <sup>2</sup>, Turab H. Jafri <sup>3,4</sup>, Pilsun Yoon <sup>5</sup>, Jize Piao <sup>6</sup>  and Kyoungson Jhang <sup>1,2,\*</sup> 

- <sup>1</sup> Department of Environmental and IT Convergence, Chungnam National University, Daejeon 34134, Republic of Korea
- <sup>2</sup> Department of Computer Science, Chungnam National University, Daejeon 34134, Republic of Korea
- <sup>3</sup> NUST Institute of Civil Engineering, National University of Sciences and Technology (NUST), Islamabad 44000, Pakistan
- <sup>4</sup> Department of Civil and Environmental Engineering, Pusan National University, Busan 46241, Republic of Korea
- <sup>5</sup> Clean Groundwater Tech, 239 Daedeok-daero, Seo-gu, Daejeon 35299, Republic of Korea
- <sup>6</sup> Groundwater Environment Research Center, Korea Institute of Geoscience and Mineral Resources, Daejeon 34132, Republic of Korea
- \* Correspondence: sun@cnu.ac.kr

**Abstract:** Multiscale forecasting of groundwater levels (GWLs) is essential for ensuring the sustainable management of groundwater resources, particularly considering the potential impacts of climate change. Such forecasting requires a model that is not only accurate in predicting GWLs but also computationally efficient, ensuring its suitability for practical applications. In this study, a temporal convolutional network (TCN) is implemented to forecast GWLs for 17 monitoring wells possessing diverse hydrogeological characteristics, located across South Korea. Using deep learning, the influence of meteorological variables (i.e., temperature, precipitation) on the forecasted GWLs was investigated by dividing the input features into three categories. Additionally, the models were developed for three forecast intervals (at 1-, 3-, and 6-month lead times) using each category input. When compared with state-of-the-art models, that is, long short-term memory (LSTM) and artificial neural network (ANN), the TCN model showed superior performance and required much less computational complexity. On average, the TCN model outperformed the LSTM model by 24%, 21%, and 25%, and the ANN model by 24%, 37%, and 47%, respectively, for 1-, 3-, and 6-month lead times. Based on these results, the proposed TCN model can be used for real-time GWL forecasting in hydrological applications.

**Keywords:** groundwater level forecasting; artificial neural networks; long short-term memory; temporal convolutional network; computational time



**Citation:** Haider, A.; Lee, G.; Jafri, T.H.; Yoon, P.; Piao, J.; Jhang, K. Enhancing Accuracy of Groundwater Level Forecasting with Minimal Computational Complexity Using Temporal Convolutional Network. *Water* **2023**, *15*, 4041. <https://doi.org/10.3390/w15234041>

Academic Editors: Bandita Mainali and Muhammad Kashif Shahid

Received: 13 October 2023

Revised: 6 November 2023

Accepted: 17 November 2023

Published: 22 November 2023



**Copyright:** © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

## 1. Introduction

Groundwater often serves as a primary source of water in many parts of the world. According to a study, groundwater accounts for 30.1% of the freshwater resources available on our planet and it fulfills 50% of domestic demand, about 40% of industrial usage, and 20% of irrigation supply globally [1]. Due to the expansion of cities and rapid population growth, water demand and consumption have drastically increased over the years and groundwater is often tapped as a convenient and accessible resource to meet these needs [2]. Furthermore, unprecedented droughts and altered rainfall patterns have exacerbated the pressure on groundwater resources [3]. Also, the over-exploitation of groundwater due to increased urbanization has led to a water sustainability challenge in developing countries [4]. As a result, water demand is likely to increase further in the future, which requires efficient utilization and management of groundwater in conjunction with surface

water resources [5–8]. Furthermore, the disparity between groundwater extraction and recharge rate has caused adverse environmental side effects such as water level depletion, low well yields, increased pumping rates, and recession in surface water levels [4]. In order to ensure the sustainability of groundwater resources, accurate long-term groundwater level (GWL) forecasting is critical. Hydrogeologists and water policymakers rely on these forecasts in making effective strategies for the domestic, agricultural, and industrial water supplies and preventing excessive water loss [9,10]. However, groundwater level modeling is critical due to its highly complex and non-linear nature as the changes in groundwater level are affected by various factors such as hydrogeological, i.e., groundwater discharge and recharge, anthropogenic, i.e., climate change, and meteorological, i.e., temperature and precipitation [11]. In this regard, over the years, a wide range of methods have been explored to accurately forecast groundwater levels.

In the last two decades, numerical and physically based modeling has often been used for GWL forecasting, with MODFLOW [12] being the most widely implemented method. The physically based models are highly accurate in predicting GWLs; however, they are time-consuming, and their performance is constrained by the significant amount of required hydrological data and, in addition, the understanding of the underlying water system mechanism [13–16]. In recent years, data-driven-based models have been increasingly adopted to forecast GWL; the details can be studied in [14,17,18]. The key advantage of such models (e.g., artificial neural network, ANN) is that they are computationally efficient and can learn from sufficient data to model any dynamic non-linear function [1,19,20]. Thereby, ANNs have been implemented in various GWL forecasting studies [18]. For instance, ref. [21] simulated weekly GWL to test the performance of ANN with three different training algorithms for 18 wells located in East India. The evaluation showed that ANN trained with the Levenberg–Marquardt algorithm had the best results, with the prediction interval extended up to 4 weeks ahead. Ref. [22] found the optimal temporal input length and the impact of exogenous input variables on GWL simulation with ANNs and multi-layer perceptron (MLP). Ref. [13] reported that ANN often fails to capture long-range temporal dependencies and subsequently does not perform well.

To address the limitations of ANNs, recurrent neural networks (RNNs) were specifically designed for sequential and time-series data (e.g., text, audio, video). The key characteristic of RNNs is their cyclic structure, which enables them to maintain the flow of data back and forth between recurrent cells and update the internal state of cells based on both the current input and the previous hidden state. The model incorporates past information stored in the hidden state to make predictions for the future state, particularly when the input data exhibit certain dependencies [8]. The standard RNN has limited memory capacity, which means that it cannot process long sequences and suffers from vanishing gradient problem. Thereafter, long short-term memory (LSTM) was developed to solve the shortcomings of standard RNN, and it has been successively employed in many research areas, including natural language processing [23,24] and machine translation [25]. Most of the groundwater studies in the literature have implemented recurrent neural networks and LSTM [26]. Ref. [27] trained LSTM models for two monitoring wells using the proposed cost function, namely, least trimmed squares (LTS) with symmetric weighting (AW) and the Whittaker smoother (WS), which is robust against noise, and outliers in the data [28]. The results obtained by employing the developed cost function outperformed the results of models trained using other cost functions in effectively identifying and rejecting outliers. In Normandy, France, LSTM has been utilized to reconstruct GWLs, and generate missing values, of 31 piezometers over 50 years of recorded data [29].

For a long time, RNNs have been employed as state-of-the-art models for sequential tasks. Even though LSTM is more efficient and has better complex function fitting ability than traditional RNNs, it can still suffer from vanishing gradient problem when processing extremely long sequences. Furthermore, due to its inherent sequential nature and excessive hyperparameter settings, it requires high computational resources and takes longer to train [30]. To solve these computational complexities, recently, researchers have been

exploring temporal convolutional networks (TCNs) which are one-dimensional CNNs (convolutional neural networks). The TCNs are specially designed for sequence processing tasks, and they have shown performance comparable to RNNs while providing computational advantages [31]. The TCN has higher arithmetic intensity and fewer trainable parameters due to the weight sharing property of the convolutional layers, and consequently, uses less memory and time to develop [32]. Ref. [33] presented a general temporal convolutional network architecture for sequence modeling tasks. Ref. [34] implemented TCN for multivariate forecasting and achieved superior performance compared to recurrent neural networks. Furthermore, TCN integrated with discrete wavelet transform (DWT) and random forest (RF) feature selection method outperformed RF-DWT-RNN and RF-DWT-LSTM in predicting daily urban water demand [34].

The fluctuations in groundwater level are characterized by multiple highly non-linear, non-stationary, and complex variables. This intricacy makes it challenging to achieve a high degree of prediction accuracy for the forecasted GWLs. Through detailed experiments, the impact of external factors on the changes in GWLs was thoroughly examined. Given the complexity in the dataset of each monitoring well, it is important for the forecasting model to be both accurate and computationally efficient for practical usability. Therefore, in this study, the TCN model has been introduced for short and long-term GWL forecasting (at 1-, 3-, and 6-month lead time). The main contributions of the study are as follows:

1. To the authors' knowledge, this is the first study to implement TCN for short and long-term groundwater level forecasting. This study provides valuable insights into its superior performance and less computational complexity compared to the widely used state-of-the-art models.
2. The studied wells are situated in different regions of South Korea and thereby possess diverse hydrogeological properties. The influence of meteorological variables (i.e., temperature and precipitation) on the forecasted GWLs was investigated by conducting intensive experiments using three input feature categories.
3. The results of the proposed model were compared with those of LSTM and ANN for 1-, 3-, and 6-month lead time forecasting. Additionally, the computational time complexity of the TCN is discussed in comparison with the respective models.

The rest of this paper is organized as follows: Section 2 provides information on the study area and describes the adopted methodology, Section 3 discusses the results, and Section 4 makes the concluding remarks.

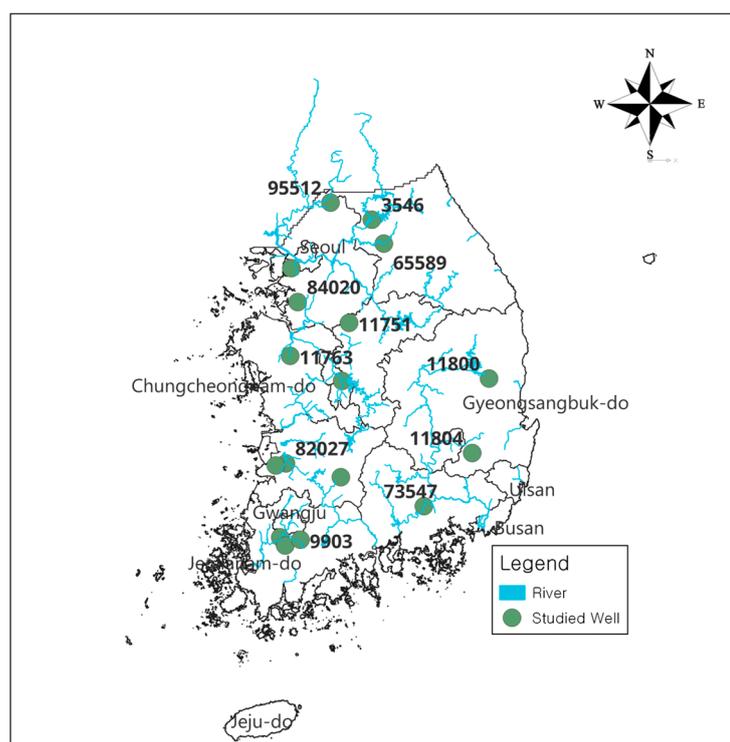
## 2. Study Area and Method

### 2.1. Study Area and Dataset

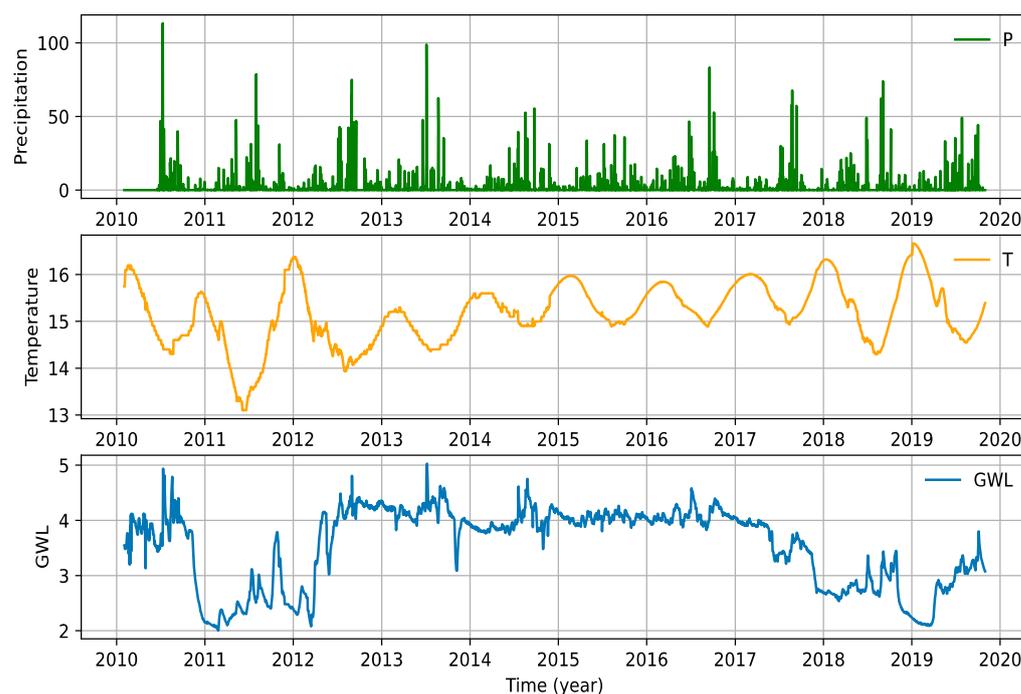
South Korea is located in the southern half of Korean Peninsula bordered by North Korea to the North and the Yellow Sea and the Sea of Japan to the West and East, respectively. The country encompasses a territory of 100,266 km<sup>2</sup> at 35.90° N latitude, and 127.77° E longitude. The landscape of Korea is characterized by diverse range of rock types including granite, gneiss, schist, limestone, metamorphic, and volcanic rocks [35]. North Pacific and continental weather patterns greatly influence the climate of South Korea. The eastern regions of Korea receive less rainfall than western regions, with more than half of it during summer season which continues for 30 days. The mean annual precipitation of the country is about 1277 mm and temperature varies greatly with time and in different regions, from −20 °C in winter to 35 °C in summer. Nearly every six years, South Korea experiences severe droughts due to the drastic changes in seasonal and annual rainfall patterns caused by climate change [36].

In South Korea, there are mainly two types of aquifers: shallow alluvial and deep bedrock aquifers. The shallow alluvial aquifer consists of unconsolidated sediments, and is commonly found near major rivers, with a water yield capacity of 30–800 m<sup>3</sup> day<sup>-1</sup> in a single well [35]. This aquifer has been the primary source of water supply for irrigation in many rural areas since the 1950s and is mostly recharged through the infiltration of rainwater during the wet season [35,37]. In many regions of Korea, the drinking water comes from bedrock aquifer wherein a well can yield water up to 10–5000 m<sup>3</sup> day<sup>-1</sup>. The bedrock aquifer is composed of faulty, porous, and different types of fractured rocks [37]. The link to the dataset is provided in the Data Availability Statement section.

Figure 1 shows the locations of the studied 17 groundwater wells located across South Korea. The dataset was obtained from the National Groundwater Information and Management Network's (NGMN) website maintained by the South Korean national government. In Korea, the groundwater monitoring wells are operated under NGMN to keep a national record of data and those wells which are affected by high pumping rates. To ensure the accurate measurement of groundwater levels, the wells are installed in low pumping areas. The dataset of each monitoring well consists of groundwater level (GWL), precipitation (P), and temperature (T) variables. Each variable in the dataset contains observations recorded on a daily basis over a decade, from 2010 to 2019. This offers an insight into the variables' patterns and trends over the period, enabling one to understand the behavior of variables and potentially predict future patterns based on past observations. Missing values were filled by taking the average of neighboring observations. Figure 2 shows the time-series plots of GWL, T, and P variables for well #11774.



**Figure 1.** Locations of studied groundwater wells situated in various regions of South Korea.



**Figure 2.** Plots of precipitation (P), temperature (T), and groundwater level (GWL) variables data over a span of 10 years (2010–2019) for groundwater well #11774.

## 2.2. Methodology

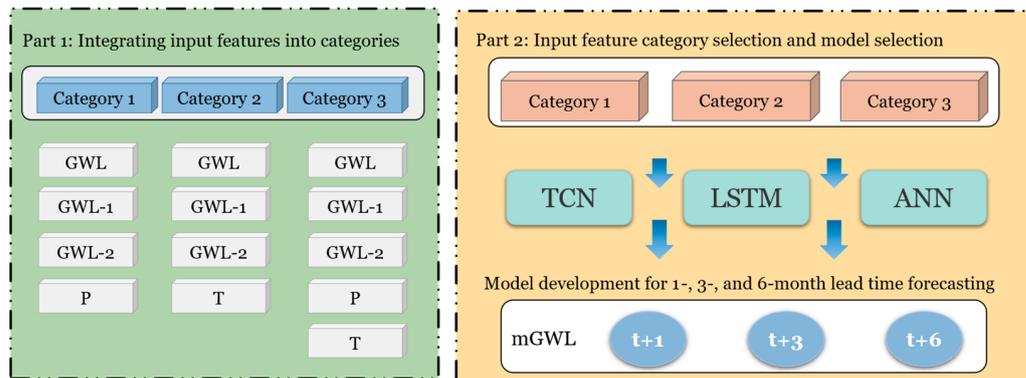
This section provides details on the implementation and development of the models (TCN, LSTM, and ANN) for GWL forecasting under different configurations. Daily observation of GWLs was used to develop forecasting models for each of the monitoring wells, aiming to forecast the average monthly GWL (mGWL). Generating a dataset for each well with a set of input features is the first step in constructing the models. As illustrated in Figure 3 (Part 1), the input features were grouped into three categories, each with a different set of meteorological variables to assess their impact on the forecasted GWLs. The input lengths for the GWL, P, and T variables were set at 30 days for 1-month lead time, 60 days for 3-month lead time, and 90 days for 6-month lead time forecasting. The ACF (autocorrelation function) was used to find the autocorrelation and determine the lagged variables. Figure 4 displays the correlation between the GWL input parameter with lags up to 30 days for two monitoring wells. A correlation above 0.75 is considered a strong correlation, and the GWL exhibits this with lags of up to 4 days for both the wells. However, during the experiments, no significant difference was observed between the results obtained using 4 lag variables (GWL-1, . . . , GWL-4) and those with 2 lag variables (GWL-1, GWL-2). Therefore, only GWL-1, GWL-2 were included as explanatory features in each input features category to help the models in identifying the recurring patterns. And with each category input, the mGWL at 1-, 3-, and 6-month lead times were set as the target variable of the models (as depicted in Figure 3 (Part 2)). For each monitoring well, 60% (train set) of the dataset was used to develop the models, 20% (validation set) was used in fine-tuning the hyperparameters, and the last 20% served as a test set for making inferences. In the datasets, the distribution of input feature values varied significantly. This disparity could lead to model overfitting, so the datasets were normalized using the StandardScaler method. It adjusts the values of variables with a mean of 0 and a standard deviation of 1, based on Equation (1).

$$x' = \frac{x - \mu}{\sigma} \quad (1)$$

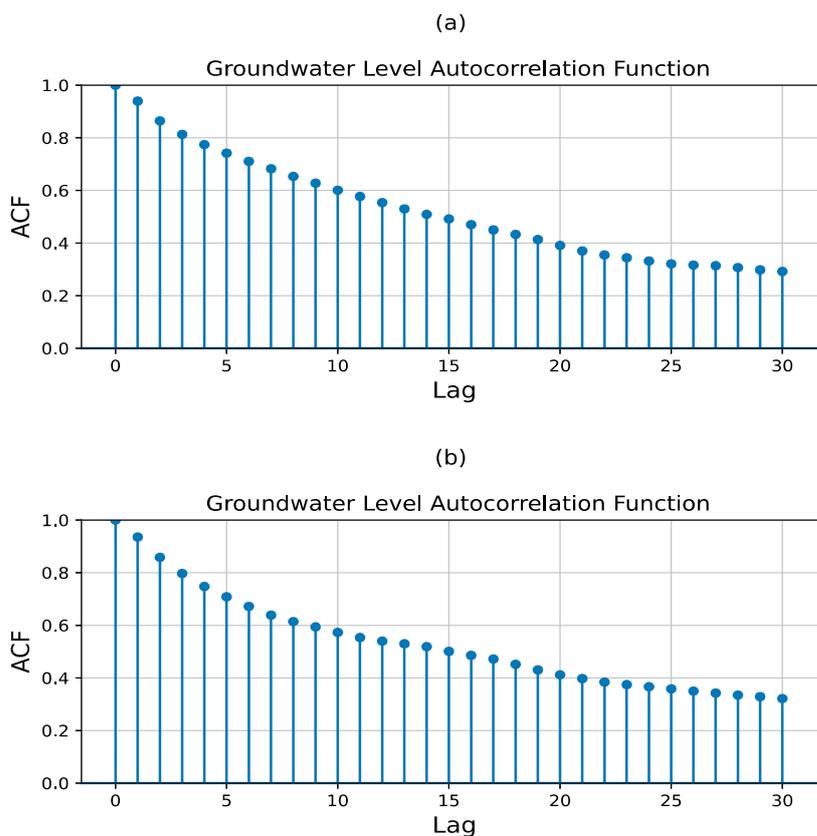
$$\mu = \frac{1}{N} \sum_{i=1}^N (x_i) \quad (2)$$

$$\sigma = \frac{1}{\sqrt{\frac{1}{N} \sum_{i=1}^N (x_i - \mu)^2}} \tag{3}$$

where  $x'$  is the standardized value with  $x$  as the original feature value and  $N$  is the length of values.  $\mu$  derives the mean of the feature values and  $\sigma$  gives the standard deviation.



**Figure 3.** Graphical representation of the study process. Part 1 illustrates the integration of the input features into categories. Part 2 depicts the development of models for three forecast intervals using each category variable.



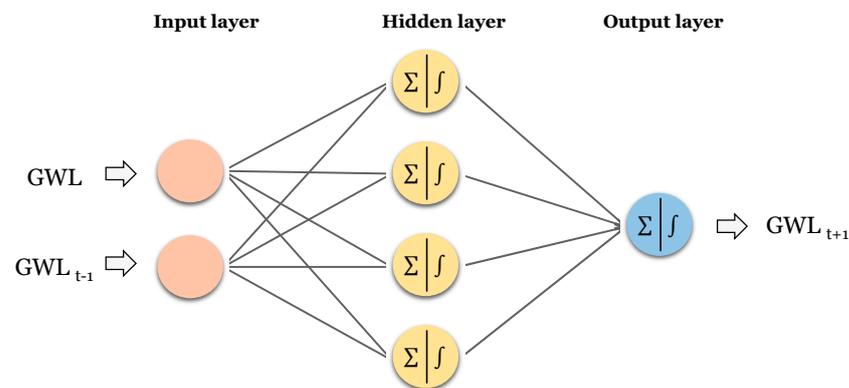
**Figure 4.** Autocorrelation diagram of (a) #65589 well and (b) #11800 well with lag up to 30 days.

In all experiments, the models were trained for 50 epochs using the Adam optimizer with a learning rate of 0.0001. To avoid overfitting on the training dataset, the early stopping method was implemented with a patience of 7 epochs to monitor the validation loss. During the training process, if the model did not show improvements in validation loss beyond the patience, then it stopped training. This saves the computational time and resources

and prevents overfitting. All models were developed using PyTorch and experiments were conducted using three NVIDIA RTX A5000 GPUs.

### 2.2.1. Artificial Neural Network (ANN)

ANNs are regarded as universal approximation functions, inspired by the structure and function of the biological neural network of the human brain. A typical feedforward ANN consists of three layers, namely, input, hidden, and output layers, where neurons (nodes) in one layer are connected to all neurons in the following layer and the strength between connections is determined through learnable parameters (weight vectors) across all layers [38]. During the training phase, the network iteratively performs an optimization algorithm with the goal of minimizing the prediction error and adapting such parameters that can accurately map input data from outside the training set to the desired output. Non-linear activation functions (e.g., Sigmoid, Tanh, ReLU) are used to model non-linearity present in the data, which helps ANN to solve large-scale complex functions with good generalizability [39]. In this study, the adaptive moment estimation (ADAM) algorithm was used to minimize the cost function and ReLU as an activation function. Figure 5 presents a graphical representation of a standard ANN.



**Figure 5.** The structure of a standard ANN with one hidden layer.

Consider a model with  $H$  input neurons and  $N$  neurons in a hidden layer. The  $n$ th neuron in the hidden layer would be equal to the weighted sum of the input layer neurons and a bias value. In a three-layer ANN, the values for the hidden layer neuron and output neuron are derived by using Equations (4)–(6):

$$u_j = \sum_{i=1}^H w_{ji} \times x_i + w_{j0} \quad (4)$$

where  $u_j$  is the input value of the  $j$ th neuron in the hidden layer,  $w_{ji}$  is a weight between the  $j$ th neuron in the hidden layer and the  $i$ th neuron in the input layer.  $x_i$  is the  $i$ th input variable of the input layer and  $w_{j0}$  is the bias term of the  $j$ th neuron in the hidden layer. Different activation functions are used to transform the output of the hidden layer neurons.

$$s_j = f(u_j) \quad (5)$$

where  $s_j$  is the output of the  $j$ th neuron in the hidden layer, and  $f$  is the activation function such as Sigmoid, Tanh, and ReLU.

$$y_k = \sum_{j=1}^M w_{kj} \times s_j + w_{k0} \quad (6)$$

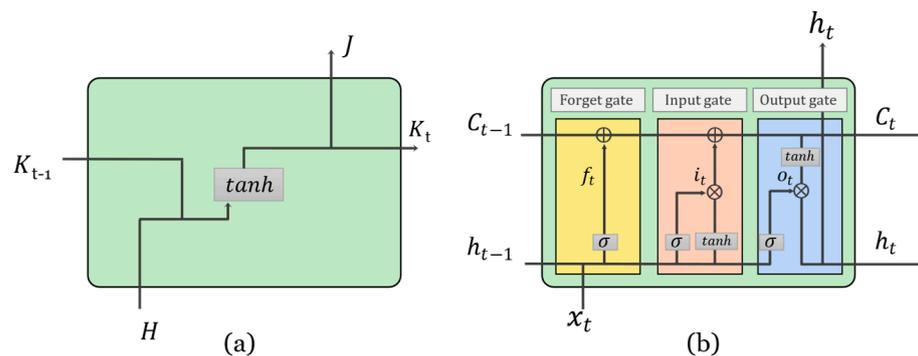
where  $y_k$  is the output neuron, derived by taking the weighted sum of hidden-layer neurons ( $w_{kj}$ ) and bias term ( $w_{k0}$ ) of the output neuron. Based on a trial-and-error method, 128 hidden nodes with one hidden layer were used in the development of ANN models.

### 2.2.2. Long Short-Term Memory (LSTM)

LSTM is a special network that was developed to solve the vanishing gradient and exploding problems of recurrent neural networks (RNNs). A simple RNN has a stack of non-linear cells that form a chain-like structure, with at least one connection between cells, enabling information to flow forward and backward through the network (Figure 6a). A recurrent cell is a mathematical function that takes input and generates output by maintaining the hidden state. The hidden state acts as a memory cell to process, store, and remember previous information over a long time period, which enables the network to capture and learn complex temporal dependencies within the data. The recursive nature of the network makes it capable of modeling sequential data, and it has achieved state-of-the-art performance on tasks such as machine translation, speech recognition, and language modeling. Forward propagation usually begins with the hidden state  $h_t = 0$ , and then, the following equations are used to update the hidden state and generate the output:

$$h_t = \tanh(b_h + Wh_{t-1} + Px_t) \tag{7}$$

$$o_t = b_o + Vh_t \tag{8}$$



**Figure 6.** (a) Graphical representation of RNN memory cell. (b) Schematic diagram of LSTM network.

In Equation (7),  $h_t$  denotes the hidden state at time  $t$ , obtained by applying the non-linear function to the sum of the input vector  $x_t$ , previous hidden state  $h_{t-1}$ , and bias term  $b_h$ , wherein  $P$  and  $W$  are learnable weight matrices associated with hidden-state and input vectors, respectively. The output vector  $o_t$  is derived by adding up the dot product of the hidden-state vector  $h_t$  and weight matrix  $V$  and bias term of the output vector  $b_o$  as Equation (8).

In RNNs, the gradients of the weights are computed with a back propagation through time (BPTT) algorithm, and subsequently, the weights are modified to minimize the total error of the network. However, with long input sequences, the gradients become smaller and smaller, and therefore, the standard RNN fails to propagate them all the way back from output node to the first layer nodes. Therefore, standard RNNs are difficult to train on large-scale data and, hence, cannot map long-term dependencies. The LSTM network was designed to address the vanishing gradient problem of RNNs [38,40,41]. Like traditional RNNs, LSTMs consist of recurrent cells (LSTM cells) wherein each LSTM cell has a relatively complex structure with specially designed memory block compared to the RNN memory cell. A memory block was invented inside the LSTM cell to efficiently store and control the flow of information over a long time period to capture the long-range dependencies.

Inside the LSTM memory block, a cell state runs straight through the network, allowing the information to flow without being subject to vanishing gradients and also serving as

a “memory cell”. In addition, the information flux is controlled by three gates, namely, the input gate, the forget gate, and the output gate with specified functionality. The input gate determines the extent to which new data and the previous hidden state flow into the cell state, while the forget gate decides what data are irrelevant and redundant, and subsequently removes them from the memory cell. The output gate controls how much information from the cell state should be used to generate output at each time  $t$ , as well as the final output of the network. The gates in the LSTM network use sigmoid as an activation function, which maps the output of the gates to a value between 0 and 1, making the flow of information easy, and being differentiable prevents vanishing gradients. A schematic diagram of memory block and LSTM cell is shown in Figure 6b. Equations (9)–(14) are used in LSTM to map an input sequence  $x$  to an output sequence.

$$i_t = \sigma(W_i x_t + P_i h_{t-1} + b_i) \quad (9)$$

$$f_t = \sigma(W_f x_t + P_f h_{t-1} + b_f) \quad (10)$$

$$o_t = \sigma(W_o x_t + P_o h_{t-1} + b_o) \quad (11)$$

$$\tilde{C}_t = \tanh(W_c x_t + P_c h_{t-1} + b_c) \quad (12)$$

$$C_t = f_t \times C_{t-1} + i_t \times \tilde{C}_t \quad (13)$$

$$h_t = o_t \times \tanh(C_t) \quad (14)$$

where  $W_i$ ,  $W_f$ , and  $W_o$  are the weight matrices of the input, forget, and output gates associated with input  $x$  at time  $t$ . Likewise,  $P_i$ ,  $P_f$ , and  $P_o$  represent the weight matrix for the input, forget, and output gates to the hidden state  $h$  at time  $t - 1$ .  $b_i$ ,  $b_f$ , and  $b_o$  are bias vectors for the input, forget, and the output gates, respectively.  $\sigma$  (logistic sigmoid) is the element-wise activation function and  $\times$  denotes the element-wise multiplication of two vectors. The input, forget, output, and cell-state vectors at time  $t$  are denoted by  $i_t$ ,  $f_t$ ,  $o_t$ , and  $C_t$ , respectively, and are the same in size as the cell output vector  $h_t$  at time  $t$ .

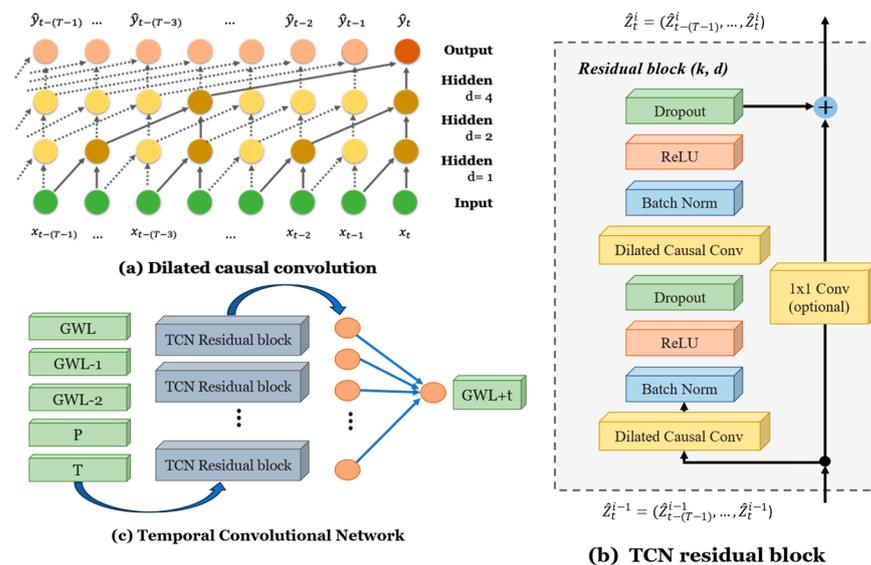
### 2.2.3. Temporal Convolutional Network

Like standard convolutional neural networks (CNNs), the building blocks of a TCN consist of convolutional, pooling, and fully connected layers, etc. However, the convolutional layers used in TCN are 1-dimensional (1D), and to make them adapt to time-series-processing tasks, causality and dilation are applied on these 1D convolutional layers. Like RNNs, the network can take an input sequence of any length and map it to an output sequence of equal length [32].

Causality in a temporal convolutional network ensures that the output of convolution operations is computed using a finite set of past inputs and does not look ahead into the sequence [32]. Dilation is used to exponentially enlarge the receptive field of convolutions on the time axis, so the wider range of inputs contribute to represent an output [33]. The dilated causal convolution is shown in Figure 7a. Figure 7b shows the TCN residual block, a stack of these residual blocks makes a temporal convolutional network. Within the residual block, four different layers are used, namely, Dilated Causal Convolution, Batch Norm, ReLU, and Dropout. Batch Norm and Dropout layers enforce the smooth training process and prevent overfitting. Non-linearity is introduced into the network by ReLU layers to learn complex patterns present within the data. The TCN takes an input sequence and passes it through these layers twice (i.e., Dilated Causal Convolution, Batch Norm, ReLU, and Dropout) as shown in Figure 7b. A  $1 \times 1$  convolution is added elementwise when the input and output of the last Dropout layer have different dimensions and is then processed

through a fully connected layer to obtain the final outputs as shown in Figure 7c. The 1D dilated convolution operation is performed in the TCN using Equation (15) [32].

$$y_n^h = \sum_{i=-}^{K-1} \sum_{l=0}^{C_{in}-1} x_{ns-di}^l \cdot W_i^{l,h}, \forall m \in [0, C_{out} - 1], \forall x \in [0, T - 1] \tag{15}$$



**Figure 7.** Elements of the TCN: (a) Dilated causal block with dilations  $d = 1, 2, 4$  and filter  $k = 2$ . It inserts a fixed step between inputs processed by each convolution filter, larger dilations effectively expand the receptive fields. (b) the TCN residual block (c) TCN network with input variables and producing an output.

In Equation (15),  $x$  and  $y$  compute the input and output activations,  $T$  is the length of the output sequence.  $W$  gives the array of filter weights,  $C_{in}$  and  $C_{out}$  are the number of input and output channels of the filter,  $K$  is the filter size, and  $s$  is the stride. The receptive field of the layer is defined as  $F = d \cdot (K - 1) + 1$ . In the experiments, with  $K = 3$ ,  $d = 2$ , and  $C_{in}$  and  $C_{out} = 25$ , we obtain the best results.

### 2.2.4. Performance Evaluation Metrics

The predictions of the models were assessed against observed values using three evaluation metrics: Pearson correlation coefficient ( $PR$ ), Nash–Sutcliffe efficiency ( $NSE$ ), and root mean square error ( $RMSE$ ).  $PR$  measures the linear correlation between the predicted and observed values, its value ranges between 0 and 1, based on Equation (16). A value close to 1 means a good fitness of the model, whereas a value close to zero indicates poor performance.  $NSE$  quantifies the magnitude of residual variance in relation to the variance in observed data, Equation (17).  $RMSE$  determines the magnitude of errors by taking the square root of the average of squared differences between observed and predicted values and facilitates in the performance comparison of various models, Equation (18). Based on  $NSE$  and  $RMSE$ , the prediction accuracy of the models can be interpreted as follows [30]: very good ( $0.75 < NSE \leq 1$ ;  $0.00 \leq RMSE < 0.50$ ), good ( $0.65 < NSE \leq 0.75$ ;  $0.50 \leq RMSE < 0.60$ ), satisfactory ( $0.50 < NSE \leq 0.65$ ;  $0.60 \leq RMSE < 0.70$ ), and unsatisfactory ( $0.50 < NSE$ ;  $RMSE \geq 0.70$ ).

$$PR = \frac{\sum_{i=1}^n (o_i - \bar{o})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^n (o_i - \bar{o})^2} \sqrt{\sum_{i=1}^n (y_i - \bar{y})^2}} \tag{16}$$

$$NSE = 1 - \left[ \frac{\sum_{i=1}^n (o - y)^2}{\sum_{i=1}^n (o - \bar{o})^2} \right] \quad (17)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (o_i - y_i)^2} \quad (18)$$

where  $o_i$  and  $y_i$  are the observed and predicted values, respectively, and  $\bar{o}$  and  $\bar{y}$  are their corresponding mean values. The size of data samples is denoted by  $n$ .

### 3. Results and Discussion

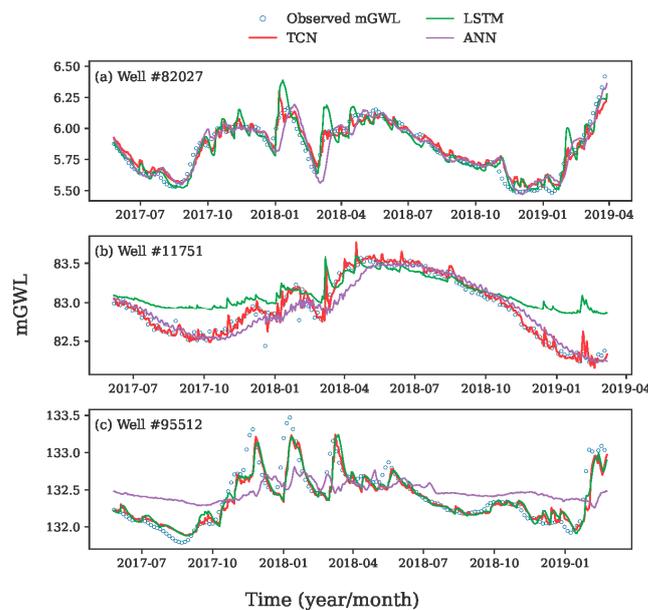
This section presents the detailed analysis and findings from the experiments conducted in this research. This study aimed to formulate a methodology for both short-term and long-term forecasting of groundwater levels and tried to develop a computationally efficient model, which sets this study apart from other works. Ref. [42] aimed to find the best predictors and enhance the prediction accuracy by optimizing the hyperparameters of the models (LSTM, GRNN). The models were developed to forecast 1-day ahead GWL. In a similar study, ref. [43] carried out correlation analysis between piezometer and meteorological variables to find the most informative input features. Moreover, the proposed method (ANN) forecasts one-month-ahead groundwater levels without necessarily requiring pumping rates as an input parameter. Ref. [30] performed a study to determine the most influential meteorological explanatory variables and assess their effectiveness on the observed groundwater levels. Furthermore, hybrid models (WT-LSTM, WT-RF, and WT-XGB) were developed using the best-performing WT (wavelet transform) method for 1-, 2-, and 3-month lead times. The scope of our study differs from that of the studies mentioned above in terms of the study area, input parameters, and forecasting intervals. While most of the studies aim at daily scale and 1-month-ahead predictions, in our study, experiments were conducted for 1-, 3-, and 6-month lead times. Moreover, our study investigated the computational efficiency (development and inference time) of the models, which past studies did not address.

#### 3.1. Assessing the Forecasting Accuracy of the Proposed and Other Models

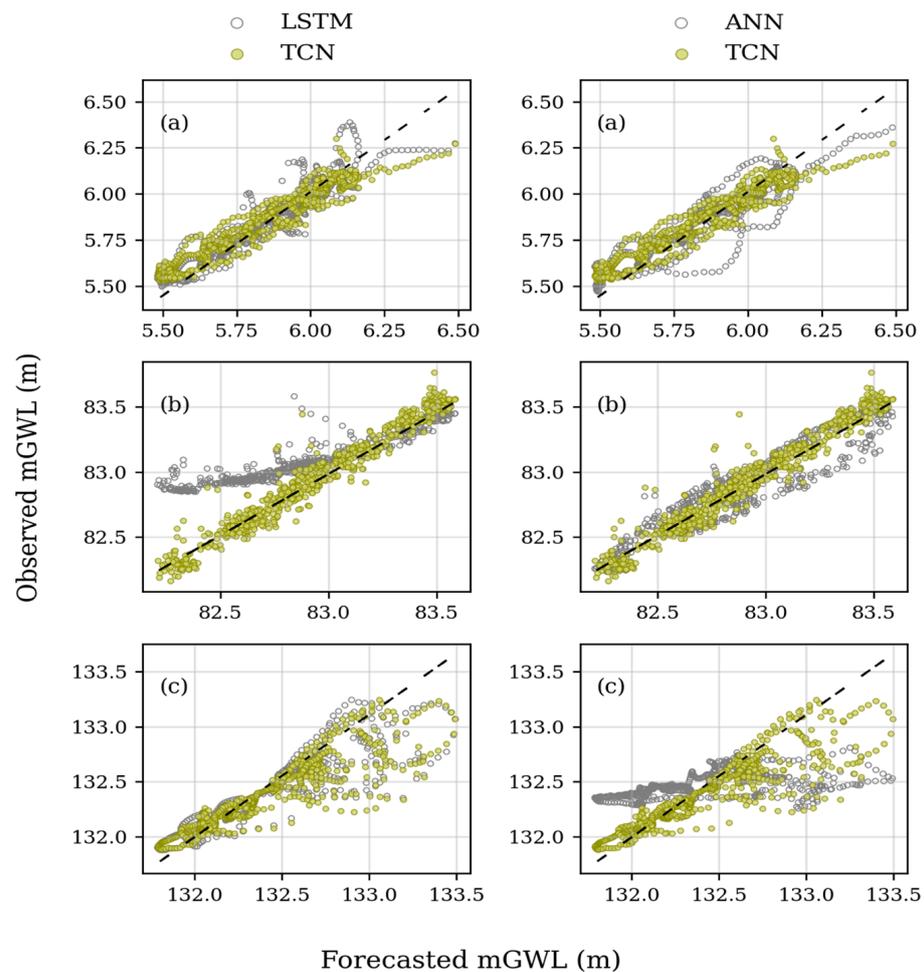
The prediction performance of the models was evaluated on the test dataset of each groundwater well using three statistical metrics. The comprehensive results of the models developed using category-1 input features at 1-month lead time are shown in Table 1, and the detailed results for 3- and 6-month lead times are provided in Tables A1 and A2. For brevity, the predicted and observed measurements of three monitoring wells are shown in Figure 8. The plots clearly indicate that there is no discernible pattern or trend in the observed groundwater levels of the wells during the period from 2017 to 2019 (test dataset). The TCN model predicted the observations with high accuracy for a 1-month lead time while capturing the rapid fluctuations in the GWLs, except for a few extreme values (Figure 8a–c), whereas the LSTM demonstrated the best fit for wells #82027 and #95512 with less accuracy than the TCN, as illustrated in the scatter plots (Figure 9a,c), but failed to do so for well #11751 for reasons unknown (Figure 9b). Similarly, the ANN model poorly forecasted the groundwater levels for well #95512 (Figure 8c); however, its performance was good for wells #82027 and #17751, with significant differences in prediction accuracy compared to the TCN model, as shown in Figure 9a,b. The Pearson correlation measures the degree of variability between the model's predicted values and observed measurements. For the 1-month lead time, the models for 12 monitoring wells exhibit a correlation coefficient greater than 0.80, indicating a strong correlation between the models' predictions and the actual values (Table 1). Similarly, the predictions of 11 models at the 3-month lead time and 14 wells at the 6-month lead time demonstrate a strong correlation with the observed values.

**Table 1.** Accuracy of the proposed TCN model of 17 groundwater wells for 1-month lead time developed using category-1 inputs (GWL, GWL-1, GWL-2, P). Tr-In Time is the cumulative training and inference time of the model measured in seconds (s).

Well	Model	1-Month Lead Time				Well	Model	1-Month Lead Time			
		NSE	RMSE	PR	Tr-In Time(s)			NSE	RMSE	PR	Tr-In Time(s)
11751	TCN	0.957	0.095	0.979	59.338	73547	TCN	0.717	0.409	0.879	146.967
	LSTM	0.422	0.349	0.933	187.100		LSTM	0.749	0.386	0.882	194.166
	ANN	0.943	0.108	0.970	175.203		ANN	0.674	0.426	0.835	170.758
11763	TCN	0.815	0.475	0.920	111.880	82027	TCN	0.910	0.170	0.961	60.054
	LSTM	0.768	0.532	0.892	253.125		LSTM	0.888	0.190	0.943	276.868
	ANN	0.769	0.531	0.880	143.874		ANN	0.856	0.202	0.925	258.042
11774	TCN	0.847	0.184	0.927	147.684	82029	TCN	0.594	0.548	0.858	92.794
	LSTM	0.463	0.345	0.759	105.126		LSTM	0.292	0.724	0.822	327.219
	ANN	0.470	0.332	0.780	89.676		ANN	0.525	0.551	0.811	209.062
11800	TCN	0.631	0.560	0.779	150.621	84020	TCN	0.814	0.270	0.944	115.216
	LSTM	0.604	0.580	0.775	288.318		LSTM	0.822	0.263	0.939	296.460
	ANN	0.531	0.629	0.778	153.215		ANN	0.727	0.320	0.887	159.707
11804	TCN	0.520	0.640	0.820	89.956	95512	TCN	0.779	0.419	0.886	149.382
	LSTM	0.372	0.732	0.796	131.759		LSTM	0.779	0.419	0.895	296.448
	ANN	0.237	0.798	0.723	226.187		ANN	0.563	0.566	0.747	74.888
3546	TCN	0.562	0.290	0.789	38.425	9858	TCN	0.780	0.264	0.887	45.714
	LSTM	0.449	0.325	0.708	103.738		LSTM	0.511	0.393	0.780	65.374
	ANN	0.641	0.231	0.865	108.125		ANN	0.462	0.422	0.687	85.778
65051	TCN	0.616	0.418	0.898	94.159	9903	TCN	0.288	0.617	0.601	164.170
	LSTM	0.262	0.580	0.835	269.069		LSTM	0.271	0.624	0.568	262.447
	ANN	0.476	0.496	0.809	102.218		ANN	0.001	0.731	0.347	123.112
65053	TCN	0.859	0.165	0.936	68.660	9908	TCN	0.544	0.405	0.548	185.003
	LSTM	0.858	0.166	0.930	193.528		LSTM	0.496	0.426	0.581	171.245
	ANN	0.780	0.207	0.886	151.506		ANN	0.375	0.483	0.576	248.357
65589	TCN	0.478	0.526	0.698	127.097						
	LSTM	0.418	0.556	0.688	156.549						
	ANN	0.391	0.579	0.666	292.196						

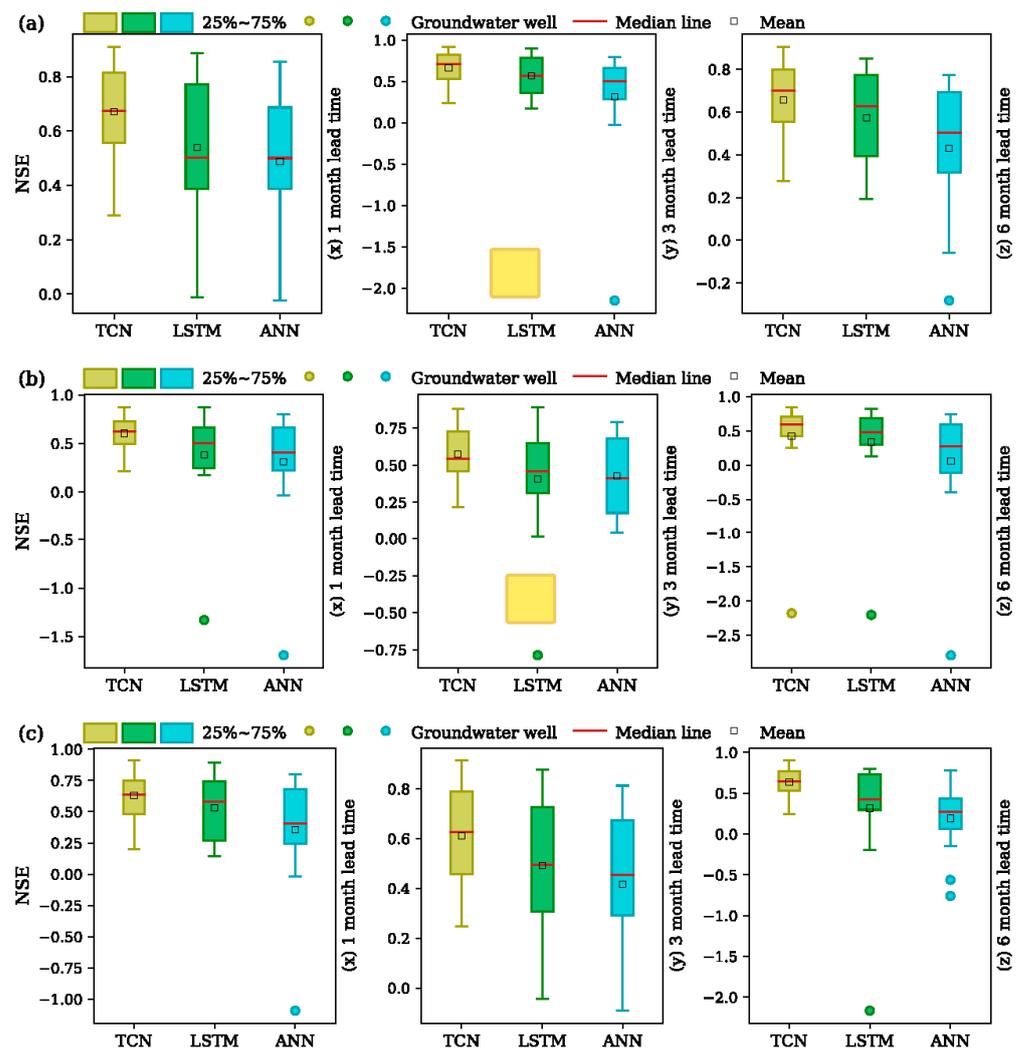


**Figure 8.** Predicted and observed measurements of TCN, LSTM, and ANN models for (a) well #82027, (b) well #11751, and (c) well #95512 at 1-month forecast period.



**Figure 9.** Scatter plots between the observed and predicted values of the TCN, LSTM, and ANN models for (a) well #82027, (b) well #11751, and (c) well #95512 for 1-month lead time.

Figure 10a shows the distribution of the best achieved NSE values under all lead time forecasts for the developed models. The TCN model, with its high median and mean values, outperforms LSTM, while ANN yields the least accurate results due to extremely inaccurate GWL predictions for certain wells. For 3- and 6-months-ahead forecasts, the average NSE values were 0.681 and 0.672 for TCN, 0.560 and 0.537 for LSTM, and 0.497 and 0.455 for ANN, respectively. In addition, it was found that the prediction efficacy of LSTM and ANN decreases as the forecast interval lengthens. Notably, the performance of TCN significantly surpasses that of LSTM by 21% and 25% based on the average NSE values under 3- and 6-month lead times, respectively. Similarly, TCN achieves 37% and 47% improved results compared to those of ANN at 3- and 6-months-ahead forecasts. The statistical significance of the proposed model was assessed with *t*-test, which determines the magnitude of difference between predicted and observed measurements.



**Figure 10.** Boxplots displaying the distribution of average NSE values of models developed using three input feature categories: (a) category 1, (b) category 2, and (c) category 3. The GWLs were forecasted at 1-, 3-, and 6-month lead times (x, y, and z) using each category input.

### 3.2. Evaluating the Impact of Meteorological Variabls on the Prediction Accuracy of the Models

As discussed in Section 3 (methodology), we also aimed to examine the influence of external factors (such as temperature and precipitation) on the forecasted GWLs for all the groundwater wells. Since these variables are highly non-linear and complex in nature, feature selection methods often fail to measure their influence on GWLs. Thereby, empirical experiments were conducted to achieve the best results using optimal features. Accordingly, we developed models for each groundwater well using three different input feature categories at all the forecasting lead times (1-, 3-, and 6-month). Figure 10a shows the distribution of NSE values of all the models developed using category-1 input features (GWL, GWL-1, GWL-2, P). Figure 10b shows the distribution of values when developed using category-2 inputs (GWL, GWL-1, GWL-2, T). Similarly, Figure 10c shows the distribution of values when developed using category-3 input variables (GWL, GWL-1, GWL-2, P, T) at all the forecast intervals. For 1-month-ahead forecasting, the average NSE values of the proposed model for category-2 and category-3 inputs were 0.618, and 0.635, respectively, whereas the model yielded a mean value of 0.689 when developed using category-1 features. At 3- and 6-month lead times, the mean NSE values using category-2 inputs were 0.592 and 0.452, respectively, while with category-3, they were 0.621 and 0.650. However, when trained using category-1 inputs, the model demonstrated superior performance and recorded values at 0.681 and 0.672 for the respective lead times. From Figure 10a–c, it

is evident that models developed using category-1 inputs consistently outperform those developed with the two other input feature categories.

The models developed using category-2 inputs, which include the T variable, consistently produced the lowest mean NSE values for all lead time forecasts, as shown in Table 2. Furthermore, the performance of the models for each groundwater well decreased relative to those developed using category-1 input variables (Table A3). There could be several reasons why the inclusion of the temperature variable produced poor results and failed to enhance the performance of the models. In the study area, temperature might not have a direct impact on the fluctuations of GWLs in monitoring wells. And the recorded data might contain noise and their poor quality could diminish the model’s efficiency. Importantly, it has been observed that the results obtained using category-3 inputs were superior to those derived using category-2 features. The improved results suggest that precipitation may exert a strong influence on GWLs, while temperature could have a diminishing effect. This assertion is further supported by the superior prediction accuracy of the models developed using category-1 features, which excludes the temperature feature, and the obtained results were the best achieved across all lead times.

**Table 2.** Average results of the models over 17 groundwater wells under all forecast intervals (for 1-, 3-, and 6-month lead times) developed using category-1, category-2, and category-3 input features. Tr-In Time is the cumulative training and inference. Italicized values are the best achieved results.

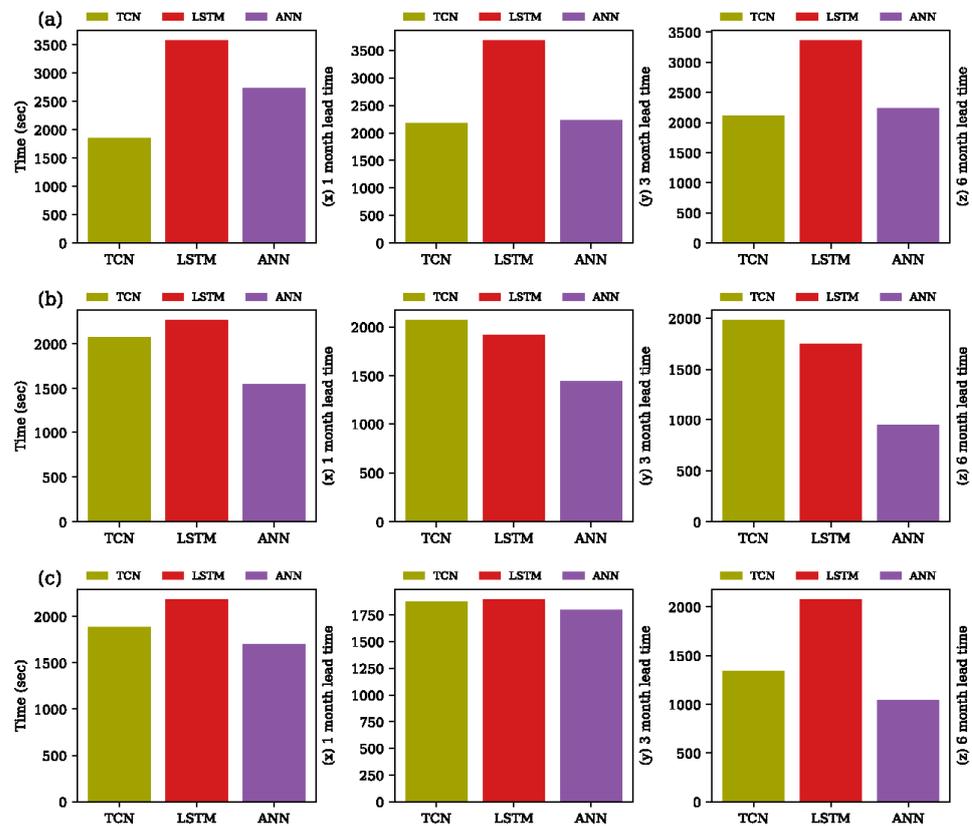
Forecast Periods	Model	Input Feature Categories											
		Category 1				Category 2				Category 3			
		NSE	RMSE	PR	Tr-In Time (s)	NSE	RMSE	PR	Tr-In Time (s)	NSE	RMSE	PR	Tr-In Time (s)
1 month	TCN	<i>0.689</i>	0.380	0.842	1847.119	0.618	0.420	0.828	2078.103	0.636	0.409	0.836	1883.596
	LSTM	0.554	0.447	0.807	3578.540	0.380	0.509	0.805	2263.363	0.515	0.468	0.816	2186.131
	ANN	0.554	0.448	0.775	2771.901	0.346	0.519	0.691	1553.834	0.385	0.502	0.729	1702.720
3 month	TCN	<i>0.681</i>	0.377	0.858	2174.110	0.592	0.423	0.809	2066.418	0.621	0.406	0.818	1870.937
	LSTM	0.560	0.436	0.834	3680.599	0.405	0.508	0.753	1918.383	0.476	0.477	0.780	1896.407
	ANN	0.495	0.467	0.749	2015.670	0.450	0.483	0.705	1441.908	0.441	0.485	0.729	1798.172
6 month	TCN	<i>0.672</i>	0.387	0.863	2117.902	0.452	0.446	0.819	1979.350	0.650	0.395	0.859	1341.820
	LSTM	0.537	0.444	0.815	3363.503	0.354	0.518	0.775	1745.894	0.320	0.527	0.775	2077.749
	ANN	0.455	0.489	0.736	2591.320	0.099	0.605	0.637	946.950	0.226	0.580	0.634	1040.722

Using category-2 inputs, LSTM and ANN yielded mean NSE values of 0.508 and 0.450 under a 3-month lead time, respectively. And with category-3 variables, the mean values were obtained as 0.476 and 0.441. However, the best achieved values were recorded as 0.560 and 0.497 with category-1 input features for the respective models. Likewise, at 6-months-ahead forecasting, the optimal values obtained by both models were 0.537 and 0.455, respectively, with category-1 inputs. Notably, the LSTM and ANN models present higher outlier values for certain wells at 3- and 6-month lead times with category-2 and category-3 input features. This also proves the effectiveness of our proposed model.

### 3.3. Discussing the Training and Inference Time Complexity of the Models

The architecture of RNNs, specially designed for sequence processing tasks, greatly improved the prediction accuracy compared to traditional machine learning and ANN models. However, due to the sequential nature, the RNNs lack parallelism in data processing, and this makes them require high computational resources and increases the time complexity. With limited computational resources, sometimes, it requires hours or days to develop such a model. Thus, it is crucial to minimize the amount of time a model takes to train and make inferences. Unlike ANNs, the architecture of TCN leverages the advantage of a 1D dilated causal convolutional layer, where each neuron is not connected to all neurons of the previous layer, which leads to faster parameter convergence. In addition, unlike LSTM, the use of multiple kernels (filters) in the convolutional layer computes different features from the input data simultaneously and speeds up convergence, whereas LSTM requires longer

to converge due to its step-by-step data processing. The bar plots in Figure 11 depict the accumulated training and inference time of each model on 17 groundwater wells, evaluated under three forecast intervals, and developed using three input feature categories. The results show that the proposed TCN model achieved superior performance with minimal cumulative training and inference time compared to LSTM and ANN.



**Figure 11.** Bar plots show the accumulated training and inference time of each model for all groundwater wells developed using three input feature categories (a–c) at three lead times (x, y, and z).

For 1-month-ahead forecasting using category-1 input features, the proposed TCN model takes approximately half an hour (1850 s) in total to develop models and inferencing on the test dataset of each groundwater well. In contrast, LSTM took over an hour (more than 3500 s) which is twice the time taken by TCN, and ANN required more than 2500 s for a similar task. At 3- and 6-month lead times, 3680 and 3363 s were required by LSTM, and 2000 and 2590 s were required by the ANN to develop models and make inferences, respectively. However, the TCN achieved higher prediction accuracy than the LSTM and ANN models, with accumulated training and inference times of 2170 and 2110 s, respectively (Figure 11a (y and z)). As discussed in Section 3, if the validation loss of the model stops decreasing, early stopping function is applied to stop further training. This not only leads to less accuracy but also reduces both training and inference times. For this reason, the LSTM and ANN models needed less time for training and inference than the proposed TCN model at 3- and 6-month lead times using category-2 input features (Figure 11b (y and z)). By using category-1 and category-2 input features, the average RMSE values increased from 0.436 to 0.508 for LSTM, and from 0.466 to 0.483 for ANN under 3-month forecast interval, respectively. The decline in performance may be attributed to the influence of meteorological variables (T and P) on the prediction efficacy of the models as discussed in Section 3.2. Nonetheless, this demonstrates that when a model uses less training time, consequently, the accuracy of the model also decreases. However, using the most effective input features, higher prediction accuracy can be achieved while reducing training and inference times.

Using category-3 input variables, the proposed TCN model outperformed the LSTM one in accurately predicting GWLs. It took marginally less training time at 1- and 3-month lead times and only half the time of LSTM for 6-months-ahead forecasts. However, while the ANN needed slightly less training time compared to TCN, the average RMSE values of ANN models were 0.501, 0.484, and 0.579. In contrast, the TCN models yielded 0.635, 0.620, and 0.650, for the respective lead times, respectively. Based on the results, the proposed TCN model offers the best trade-off between time efficiency and prediction accuracy compared to the LSTM and ANN models.

#### 4. Conclusions

This work presents a groundwater forecasting model using the data obtained from 17 monitoring wells with deep learning models. The studied wells are located in different regions of South Korea and possess diverse hydrogeological properties, making the forecasting of GWLs a challenging task. Considering the diversity in the dataset, a robust methodology was established to study the influence of meteorological variables on the forecasted GWLs. Accordingly, the experiments were performed using three input feature categories to find the best set of features which produces the optimal results. In this study, the TCN model was proposed for GWL forecasting due to its ability to efficiently capture long-range dependencies using dilated and causal convolutions. For each monitoring well, the predictions were made for 1-, 3-, and 6-month lead times. For comparison, the state-of-the-art models LSTM and ANN were developed to assess the prediction accuracy of the proposed model. The obtained results for the groundwater wells showed that the proposed model outperformed LSTM and ANN in 1-month lead time forecasting. Moreover, it achieved significantly higher prediction accuracy, by 21% and 37% for the 3-month lead time and by 25% and 47% for the 6-month lead time, compared to the respective models. These results demonstrate the robustness of the TCN model for long-term forecasting (at 3- and 6-month lead times). In addition, the best results were achieved from the models developed using category-1 input features, namely, GWL, GWL-1, GWL-2, and P. The inclusion of the temperature variable as input feature consistently resulted in poorer performance. Importantly, the proposed model was computationally time-efficient in both development and inference as compared to the LSTM and ANN models. Remarkably, for the top performing models, it took only half the time of the LSTM when forecasting at a 1-month lead time. This further strengthens the applicability of the model introduced in this study for multiscale GWL forecasting in real time and advances the sustainability of groundwater resources. Future research will be focused on optimizing the hyperparameters of the proposed model to achieve optimal results for each groundwater well.

While the TCN model yielded greater prediction accuracy, it has its own limitations. For instance, increasing the depth of the network can significantly increase the model's complexity, compared to LSTM and ANN. Based on the success of very deep CNN networks for image analysis, it is stated that the deeper the network, the better the accuracy [44]. However, this is not necessarily the case for time-series analysis. Therefore, the structure of the network must be carefully designed, considering the nature of input variables alongside optimal selection of hyperparameters, in order to achieve the best results.

**Author Contributions:** A.H.: conceptualization, investigation, writing—original draft preparation, and writing—review and editing. G.L.: conceptualization, methodology. T.H.J. and K.J.: supervision, conceptualization, investigation, writing—original draft preparation, and writing—review and editing. P.Y.: data curation, investigation. K.J. and J.P.: writing—review and editing. All authors have read and agreed to the published version of the manuscript.

**Funding:** This work was supported by an Institute of Information & Communications Technology Planning & Evaluation (IITP) grant funded by the Korean government (MSIT) (Grant No. RS-2022-00155857, Artificial Intelligence Convergence Innovation Human Resources Development (Chungnam National University)).

**Data Availability Statement:** The dataset of the monitoring wells is available via the following link: <https://www.gims.go.kr/en/brochure.do> (accessed on 19 November 2023).

**Conflicts of Interest:** The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

### Appendix A

**Table A1.** Accuracy of the proposed TCN model of 17 groundwater wells for 3-month lead time developed using category-1 inputs (GWL, GWL-1, GWL-2, P). Tr-In Time is the cumulative training and inference time of the model measured in seconds.

Well	Model	3-Month Lead Time				Well	Model	3-Month Lead Time			
		NSE	RMSE	PR	Tr-In Time(s)			NSE	RMSE	PR	Tr-In Time(s)
11751	TCN	0.960	0.086	0.981	58.205	73547	TCN	0.742	0.378	0.878	78.461
	LSTM	0.466	0.313	0.944	130.925		LSTM	0.732	0.384	0.873	155.065
	ANN	0.815	0.172	0.901	48.473		ANN	0.665	0.428	0.829	180.744
11763	TCN	0.809	0.489	0.916	196.223	82027	TCN	0.920	0.153	0.966	64.783
	LSTM	0.776	0.530	0.896	232.629		LSTM	0.899	0.172	0.950	182.546
	ANN	0.794	0.521	0.892	211.557		ANN	0.779	0.244	0.887	115.892
11774	TCN	0.808	0.194	0.917	188.141	82029	TCN	0.565	0.537	0.861	181.326
	LSTM	0.359	0.355	0.703	111.162		LSTM	0.243	0.708	0.785	295.124
	ANN	0.529	0.296	0.801	152.329		ANN	0.559	0.547	0.800	166.849
11800	TCN	0.645	0.555	0.859	179.653	84020	TCN	0.833	0.251	0.948	175.166
	LSTM	0.610	0.581	0.756	270.098		LSTM	0.823	0.259	0.932	274.569
	ANN	0.396	0.726	0.603	50.485		ANN	0.635	0.381	0.853	97.944
11804	TCN	0.334	0.751	0.790	90.558	95512	TCN	0.818	0.353	0.918	191.902
	LSTM	0.176	0.835	0.787	247.692		LSTM	0.801	0.369	0.894	291.118
	ANN	0.138	0.834	0.718	152.667		ANN	0.636	0.497	0.817	53.559
3546	TCN	0.533	0.297	0.792	195.984	9858	TCN	0.820	0.249	0.909	127.637
	LSTM	0.467	0.318	0.763	214.415		LSTM	0.751	0.293	0.868	125.706
	ANN	−0.007	0.404	0.556	52.728		ANN	0.560	0.398	0.759	179.028
65051	TCN	0.681	0.390	0.915	101.667	9903	TCN	0.242	0.645	0.548	83.606
	LSTM	0.402	0.534	0.865	299.583		LSTM	0.285	0.626	0.603	309.726
	ANN	0.524	0.473	0.831	91.620		ANN	−0.029	0.742	0.358	87.016
65053	TCN	0.883	0.155	0.942	60.056	9908	TCN	0.520	0.430	0.754	130.383
	LSTM	0.855	0.173	0.935	219.955		LSTM	0.522	0.429	1.000	235.154
	ANN	0.818	0.191	0.906	189.019		ANN	0.321	0.514	0.679	128.786
65589	TCN	0.457	0.496	0.694	70.358						
	LSTM	0.359	0.539	0.626	85.132						
	ANN	0.281	0.569	0.535	63.216						

**Table A2.** Accuracy of the proposed TCN model of 17 groundwater wells for 6-month lead time developed using category-1 inputs (GWL, GWL-1, GWL-2, P). Tr-In Time is the cumulative training and inference time of the model measured in seconds.

Well	Model	6-Month Lead Time				Well	Model	6-Month Lead Time			
		NSE	RMSE	PR	Tr-In Time(s)			NSE	RMSE	PR	Tr-In Time(s)
11751	TCN	0.939	0.095	0.977	66.553	73547	TCN	0.767	0.367	0.896	75.937
	LSTM	−0.043	0.393	0.867	247.127		LSTM	0.771	0.365	0.893	163.170
	ANN	0.877	0.135	0.944	95.110		ANN	0.608	0.476	0.813	95.915
11763	TCN	0.824	0.502	0.918	184.696	82027	TCN	0.906	0.142	0.954	248.170
	LSTM	0.751	0.597	0.885	208.385		LSTM	0.852	0.178	0.932	219.407
	ANN	0.773	0.570	0.879	289.275		ANN	0.731	0.240	0.863	198.411

**Table A2.** *Cont.*

Well	Model	6-Month Lead Time				Well	Model	6-Month Lead Time			
		NSE	RMSE	PR	Tr-In Time(s)			NSE	RMSE	PR	Tr-In Time(s)
11774	TCN	0.797	0.204	0.918	140.097	82029	TCN	0.602	0.529	0.875	195.457
	LSTM	0.657	0.265	0.864	308.295		LSTM	0.416	0.640	0.816	192.243
	ANN	0.361	0.362	0.755	95.533		ANN	0.526	0.577	0.784	209.325
11800	TCN	0.634	0.597	0.872	96.493	84020	TCN	0.809	0.291	0.945	107.710
	LSTM	0.599	0.625	0.810	160.500		LSTM	0.833	0.272	0.935	299.926
	ANN	0.653	0.581	0.709	192.022		ANN	0.695	0.367	0.864	70.034
11804	TCN	0.276	0.802	0.820	130.049	95512	TCN	0.793	0.383	0.903	72.768
	LSTM	0.320	0.777	0.743	124.559		LSTM	0.784	0.391	0.881	366.572
	ANN	−0.282	1.067	0.707	165.872		ANN	0.690	0.469	0.826	214.996
3546	TCN	0.568	0.255	0.764	76.797	9858	TCN	0.806	0.269	0.900	98.521
	LSTM	0.463	0.284	0.813	190.978		LSTM	0.722	0.322	0.851	120.095
	ANN	0.327	0.318	0.766	144.934		ANN	0.482	0.439	0.715	260.219
65051	TCN	0.635	0.399	0.901	198.751	9903	TCN	0.288	0.620	0.582	109.746
	LSTM	0.224	0.582	0.831	155.154		LSTM	0.194	0.660	0.566	245.625
	ANN	0.353	0.531	0.755	93.793		ANN	−0.058	0.756	0.279	51.344
65053	TCN	0.793	0.184	0.919	69.162	9908	TCN	0.519	0.444	0.864	120.703
	LSTM	0.849	0.158	0.923	148.065		LSTM	0.447	0.476	0.643	150.363
	ANN	0.733	0.209	0.897	258.734		ANN	−0.023	0.648	0.377	62.733
65589	TCN	0.466	0.498	0.667	126.293						
	LSTM	0.294	0.572	0.610	63.040						
	ANN	0.294	0.573	0.576	93.069						

**Table A3.** Accuracy of the proposed TCN model of 17 groundwater wells for 1-month lead time developed using category-2 inputs (GWL, GWL-1, GWL-2, T). Tr-In Time is the cumulative training and inference time of the model measured in seconds.

Well	Model	1-Month Lead Time				Well	Model	1-Month Lead Time			
		NSE	RMSE	PR	Tr-In Time(s)			NSE	RMSE	PR	Tr-In Time(s)
11751	TCN	0.870	0.166	0.980	62.926	73547	TCN	0.648	0.457	0.805	68.429
	LSTM	0.319	0.379	0.969	206.709		LSTM	0.656	0.452	0.824	161.278
	ANN	0.899	0.143	0.950	57.688		ANN	0.640	0.447	0.816	109.145
11763	TCN	0.851	0.427	0.925	107.874	82027	TCN	0.871	0.204	0.940	88.263
	LSTM	0.746	0.557	0.888	117.113		LSTM	0.874	0.201	0.940	193.384
	ANN	0.789	0.507	0.893	141.819		ANN	0.797	0.239	0.903	122.978
11774	TCN	0.774	0.223	0.935	200.742	82029	TCN	0.589	0.552	0.854	157.761
	LSTM	−1.330	0.718	0.736	44.206		LSTM	0.271	0.734	0.803	187.854
	ANN	0.278	0.388	0.593	43.440		ANN	0.538	0.543	0.796	199.000
11800	TCN	0.543	0.623	0.797	79.067	84020	TCN	0.740	0.318	0.928	167.283
	LSTM	0.494	0.656	0.690	59.688		LSTM	0.698	0.343	0.906	123.860
	ANN	0.501	0.649	0.685	53.197		ANN	0.745	0.309	0.891	116.712
11804	TCN	0.421	0.702	0.763	92.032	95512	TCN	0.653	0.525	0.840	136.935
	LSTM	0.175	0.839	0.720	111.337		LSTM	0.571	0.584	0.825	224.514
	ANN	0.227	0.804	0.722	136.251		ANN	0.191	0.770	0.510	30.312
3546	TCN	0.460	0.322	0.772	41.872	9858	TCN	0.720	0.297	0.861	251.934
	LSTM	0.466	0.320	0.828	191.021		LSTM	0.616	0.349	0.786	96.990
	ANN	0.098	0.416	0.567	56.181		ANN	0.429	0.435	0.667	37.401
65051	TCN	0.624	0.414	0.885	171.605	9903	TCN	0.214	0.648	0.495	129.426
	LSTM	0.181	0.611	0.814	187.370		LSTM	0.182	0.661	0.491	154.957
	ANN	−1.693	1.123	0.565	29.811		ANN	−0.032	0.743	0.243	50.446

Table A3. Cont.

Well	Model	1-Month Lead Time				Well	Model	1-Month Lead Time			
		NSE	RMSE	PR	Tr-In Time(s)			NSE	RMSE	PR	Tr-In Time(s)
65053	TCN	0.629	0.268	0.886	49.990	9908	TCN	0.505	0.422	0.739	145.578
	LSTM	0.737	0.226	0.901	42.892		LSTM	0.511	0.420	0.981	100.441
	ANN	0.771	0.212	0.887	115.237		ANN	0.380	0.481	0.457	124.940
65589	TCN	0.399	0.565	0.669	126.387						
	LSTM	0.296	0.611	0.578	59.748						
	ANN	0.327	0.608	0.597	129.278						

## References

- 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]
- Nordin, N.F.C.; Mohd, N.S.; Koting, S.; Ismail, Z.; Sherif, M.; El-Shafie, A. Groundwater quality forecasting modelling using artificial intelligence: A review. *Groundw. Sustain. Dev.* **2021**, *14*, 100643. [CrossRef]
- Essahlaoui, F.; Elhajrat, N.; Halimi, M.; El Abbassi, A. New approach to monitoring a wastewater irrigation system controlled by the artificial neural network (ANN). *Groundw. Sustain. Dev.* **2023**, *23*, 100999. [CrossRef]
- Adamowski, J.; Chan, H.F. A wavelet neural network conjunction model for groundwater level forecasting. *J. Hydrol.* **2011**, *407*, 28–40. [CrossRef]
- Shahid, M.K.; Choi, Y. CO<sub>2</sub> as an Alternative to Traditional Antiscalants in Pressure-Driven Membrane Processes: An Experimental Study of Lab-Scale Operation and Cleaning Strategies. *Membranes* **2022**, *12*, 918. [CrossRef]
- Chen, C.; He, W.; Zhou, H.; Xue, Y.; Zhu, M. A comparative study among machine learning and numerical models for simulating groundwater dynamics in the Heihe River Basin, northwestern China. *Sci. Rep.* **2020**, *10*, 3904. [CrossRef]
- Nourani, V.; Alami, M.T.; Voutsoughi, F.D. Wavelet-entropy data pre-processing approach for ANN-based groundwater level modeling. *J. Hydrol.* **2015**, *524*, 255–269. [CrossRef]
- Gharehbaghi, A.; Ghasemlounia, R.; Ahmadi, F.; Albaji, M. Groundwater level prediction with meteorologically sensitive Gated Recurrent Unit (GRU) neural networks. *J. Hydrol.* **2022**, *612*, 128262. [CrossRef]
- Rahman, A.T.M.S.; Hosono, T.; Quilty, J.M.; Das, J.; Basak, A. Multiscale groundwater level forecasting: Coupling new machine learning approaches with wavelet transforms. *Adv. Water Resour.* **2020**, *141*, 103595. [CrossRef]
- Suryanarayana, C.; Sudheer, C.; Mahmood, V.; Panigrahi, B.K. An integrated wavelet-support vector machine for groundwater level prediction in Visakhapatnam, India. *Neurocomputing* **2014**, *145*, 324–335. [CrossRef]
- Hasda, R.; Rahaman, M.F.; Jahan, C.S.; Molla, K.I.; Mazumder, Q.H. Climatic data analysis for groundwater level simulation in drought prone Barind Tract, Bangladesh: Modelling approach using artificial neural network. *Groundw. Sustain. Dev.* **2020**, *10*, 100361. [CrossRef]
- Harbaugh, A.W.; Banta, E.R.; Hill, M.C.; McDonald, M.G. MODFLOW-2000, The U.S. Geological Survey Modular Ground-Water Model—User Guide to Modularization Concepts and the Ground-Water Flow Process. CO<sub>4</sub> McDonald Morrissey Associates. Available online: <https://pubs.usgs.gov/of/2000/0092/report.pdf> (accessed on 19 November 2022).
- Yin, W.; Fan, Z.; Tangdamrongsub, N.; Hu, L.; Zhang, M. Comparison of physical and data-driven models to forecast groundwater level changes with the inclusion of GRACE—A case study over the state of Victoria, Australia. *J. Hydrol.* **2021**, *602*, 126735. [CrossRef]
- Rajae, T.; Ebrahimi, H.; Nourani, V. A review of the artificial intelligence methods in groundwater level modeling. *J. Hydrol.* **2019**, *572*, 336–351. [CrossRef]
- Khan, J.; Lee, E.; Balobaid, A.S.; Kim, K. A Comprehensive Review of Conventional, Machine Learning, and Deep Learning Models for Groundwater Level (GWL) Forecasting. *Appl. Sci.* **2023**, *13*, 2743. [CrossRef]
- Roy, D.K.; Biswas, S.K.; Mattar, M.A.; El-Shafei, A.A.; Murad, K.F.I.; Saha, K.K.; Datta, B.; Dewidar, A.Z. Groundwater level prediction using a multiple objective genetic algorithm-grey relational analysis based weighted ensemble of anfis models. *Water* **2021**, *13*, 3130. [CrossRef]
- Liu, W.; Yu, H.; Yang, L.; Yin, Z.; Zhu, M.; Wen, X. Deep learning-based predictive framework for groundwater level forecast in arid irrigated areas. *Water* **2021**, *13*, 2558. [CrossRef]
- Di Salvo, C. Improving Results of Existing Groundwater Numerical Models Using Machine Learning Techniques: A Review. *Water* **2022**, *14*, 2307. [CrossRef]
- Moghaddam, H.K.; Moghaddam, H.K.; Kivi, Z.R.; Bahreinimotlagh, M.; Alizadeh, M.J. Developing comparative mathematic models, BN and ANN for forecasting of groundwater levels. *Groundw. Sustain. Dev.* **2019**, *9*, 100237. [CrossRef]
- Shakya, C.M.; Bhattacharjya, R.K.; Dadhich, S. Groundwater level prediction with machine learning for the Vidisha district, a semi-arid region of Central India. *Groundw. Sustain. Dev.* **2022**, *19*, 100825. [CrossRef]

21. Mohanty, S.; Jha, M.K.; Kumar, A.; Sudheer, K.P. Artificial neural network modeling for groundwater level forecasting in a river island of eastern India. *Water Resour. Manag.* **2010**, *24*, 1845–1865. [[CrossRef](#)]
22. Lallahem, S.; Mania, J.; Hani, A.; Najjar, Y. On the use of neural networks to evaluate groundwater levels in fractured media. *J. Hydrol.* **2005**, *307*, 92–111. [[CrossRef](#)]
23. Sundermeyer, M.; Schlüter, R.; Ney, H. LSTM Neural Networks for Language Modeling. Available online: <http://www.isca-speech.org/archive> (accessed on 19 November 2022).
24. Tom, T.; Mikolov, T. Overview Introduction Model description ASR Results Extensions MT Results Comparison Main outcomes Future work Recurrent neural network based language model. *Interspeech* **2010**, *2*, 1045–1048.
25. Google, I.S.; Google, O.V.; Le Google, Q.V. Sequence to Sequence Learning with Neural Networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*.
26. 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](#)]
27. Jeong, J.; Park, E.; Chen, H.; Kim, K.Y.; Han, W.S.; Suk, H. Estimation of groundwater level based on the robust training of recurrent neural networks using corrupted data. *J. Hydrol.* **2020**, *582*, 124512. [[CrossRef](#)]
28. Graf, R.; Zhu, S.; Sivakumar, B. Forecasting river water temperature time series using a wavelet–neural network hybrid modelling approach. *J. Hydrol.* **2019**, *578*, 124115. [[CrossRef](#)]
29. Vu, M.T.; Jardani, A.; Massei, N.; Fournier, M. Reconstruction of missing groundwater level data by using Long Short-Term Memory (LSTM) deep neural network. *J. Hydrol.* **2021**, *597*, 125776. [[CrossRef](#)]
30. Zhang, Q.; Li, P.; Ren, X.; Ning, J.; Li, J.; Liu, C.; Wang, Y.; Wang, G. A new real-time groundwater level forecasting strategy: Coupling hybrid data-driven models with remote sensing data. *J. Hydrol.* **2023**, *625*, 129962. [[CrossRef](#)]
31. Wei, A.; Li, X.; Yan, L.; Wang, Z.; Yu, X. Machine learning models combined with wavelet transform and phase space reconstruction for groundwater level forecasting. *Comput. Geosci.* **2023**, *177*, 105386. [[CrossRef](#)]
32. Risso, M.; Burrello, A.; Conti, F.; Lamberti, L.; Chen, Y.; Benini, L.; Macii, E.; Poncino, M.; Pagliari, D.J. Lightweight Neural Architecture Search for Temporal Convolutional Networks at the Edge. *IEEE Trans. Comput.* **2023**, *72*, 744–758. [[CrossRef](#)]
33. Bai, S.; Kolter, J.Z.; Koltun, V. An Empirical Evaluation of Generic Convolutional and Recurrent Networks for Sequence Modeling. 2018. Available online: <http://arxiv.org/abs/1803.01271> (accessed on 19 November 2022).
34. Wan, R.; Mei, S.; Wang, J.; Liu, M.; Yang, F. Multivariate temporal convolutional network: A deep neural networks approach for multivariate time series forecasting. *Electronics* **2019**, *8*, 876. [[CrossRef](#)]
35. Lee, J.Y.; Yi, M.J.; Yoo, Y.K.; Ahn, K.H.; Kim, G.B.; Won, J.H. A review of the National Groundwater Monitoring Network in Korea. *Hydrol. Process.* **2007**, *21*, 907–919. [[CrossRef](#)]
36. Lee, J.Y.; Kwon, K.D. Current status of groundwater monitoring networks in Korea. *Water* **2016**, *8*, 168. [[CrossRef](#)]
37. Kim, D.; Jang, C.; Choi, J.; Kwak, J. A Case Study: Groundwater Level Forecasting of the Gyorae Area in Actual Practice on Jeju Island Using Deep-Learning Technique. *Water* **2023**, *15*, 972. [[CrossRef](#)]
38. Lee, E.; Khan, J.; Son, W.J.; Kim, K. An Efficient Feature Augmentation and LSTM-Based Method to Predict Maritime Traffic Conditions. *Appl. Sci.* **2023**, *13*, 2556. [[CrossRef](#)]
39. Wunsch, A.; Liesch, T.; Broda, S. Forecasting groundwater levels using nonlinear autoregressive networks with exogenous input (NARX). *J. Hydrol.* **2018**, *567*, 743–758. [[CrossRef](#)]
40. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)]
41. Jozefowicz, R.; Zaremba, W. An Empirical Exploration of Recurrent Network Architectures. *Int. Conf. Mach. Learn.* **2015**, *37*, 2342–2350.
42. Chu, H.; Bian, J.; Lang, Q.; Sun, X.; Wang, Z. Daily Groundwater Level Prediction and Uncertainty Using LSTM Coupled with PMI and Bootstrap Incorporating Teleconnection Patterns Information. *Sustainability* **2022**, *14*, 11598. [[CrossRef](#)]
43. Derbela, M.; Nouiri, I. Intelligent approach to predict future groundwater level based on artificial neural networks (ANN). *EuroMediterr J. Environ. Integr.* **2020**, *5*, 51. [[CrossRef](#)]
44. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. September 2014. Available online: <http://arxiv.org/abs/1409.1556> (accessed on 19 November 2022).

**Disclaimer/Publisher’s Note:** The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.