Time-Series Prediction in Nodal Networks Using Recurrent Neural Networks and a Pairwise-Gated Recurrent Unit Approach

Yanjie Tong[1] and Iris Tien, Ph.D., M.ASCE[2]

Abstract: Characteristics at nodes in a network, such as values of demand, evolve over time. To make time-dependent decisions for a network, making time series predictions at each node in the network over time is often necessary. Typical time series prediction approaches are based on historical information. However, these fail to account for network-level factors that might affect nodal values. This paper proposes an approach for the time series prediction in nodal networks that accounts for both time history information and nodal characteristics in the prediction. The approach is based on recurrent neural networks and, in particular, gated recurrent units (GRU), creating a new GRU structure called a Pairwise-GRU to include the influence of both historical data and neighboring node information to predict values at each node in the network. The result is a more accurate and confident time series prediction. The performance of the proposed approach is tested using an electricity network in the southeastern United States. The results indicate that the proposed Pairwise-GRU outperforms existing methods in terms of increased accuracy and decreased uncertainty in the prediction. The approach performs particularly well for long-term, multiple-time-steps ahead predictions and anomalous hazard conditions in addition to normal operating scenarios. **DOI: 10.1061/AJRUA6.0001221.**

Author keywords: Recurrent neural networks; Gated recurrent units (GRU); Time series prediction; Nodal networks; Data-driven prediction; Predictions with uncertainty.

Introduction

Systems, including infrastructure systems, are often modeled as nodal networks composed of nodes representing separate assets or locations and links representing connections between them. Because these networks operate and evolve over time, they are characterized by time-varying values at the nodes. For example, an electrical network is composed of nodes representing substations and links the connections between them, and values of electricity generation or consumption at the nodes indicate levels of service for the system. Time series data capture these nodal characteristics of the network to evaluate the continued operations and evolution of the system over time. To facilitate time-dependent network planning or scenario evaluation, making time series predictions at each node in the network over time is desired. Typical time series prediction approaches are based on historical information. However, these fail to account for network-level factors that might affect nodal values. For example, a hazard event could simultaneously affect varying areas of a network, or an emerging factor might affect a subset of nodes in the network. In these cases, relying solely on historical information does not fully capture the new network characteristics for nodal time series predictions.

In this paper, a new method for time series prediction in nodal networks is proposed. This prediction includes both the history of the time series data and the interactions of a node with its neighbors for the prediction. The approach is based on recurrent neural networks (RNNs) and specifically gated recurrent units (GRUs). To include the influence of both historical data and neighboring nodes, a novel Pairwise-GRU approach is proposed. The definition of neighboring nodes beyond physical connections is considered to include measures of similarity between nodes. Assessing nodes by similarity enables the approach to capture network-level effects that influence the states of nodes beyond their physical connections. The results indicate that including both historical and neighboring nodal information within the proposed approach achieves more accurate predictions with higher confidence compared with existing approaches.

Background and Related Work

A RNN processes a temporal sequence through a directed graph. Its applications include handwriting recognition, speech recognition, and time series prediction. The simplest form of a RNN is indicated in Fig. 1, in which inputs and outputs are connected by the hyperbolic tangent transformation. X_t is the input at time step t, Y_t is the output at time step t, and h_t represents the hidden state information at time step t. Note that the term time step is used throughout the paper in the time series sense. When able to predict outputs based on historical information, allowing previous outputs as inputs and accounting for hidden states, the performance of naïve RNNs is often limited by gradient vanishing and long-term recognition because of the limited number of parameters (Li et al. 2018).

Thus, to avoid the long-term dependency problem and to accommodate lags between data points, particularly in time series prediction problems, a Long Short-Term Memory network (LSTM) was

[1]Ph.D. Student, School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0355 (corresponding author). Email: yjtong@gatech.edu

[2]Williams Family Associate Professor, School of Civil and Environmental Engineering, Georgia Institute of Technology, Atlanta, GA 30332-0355. Email: itien@ce.gatech.edu

Note. This manuscript was submitted on June 16, 2021; approved on December 1, 2021; published online on January 17, 2022. Discussion period open until June 17, 2022; separate discussions must be submitted for individual papers. This paper is part of the ***ASCE-ASME Journal of Risk and Uncertainty in Engineering Systems, Part A: Civil Engineering***, © ASCE, ISSN 2376-7642.

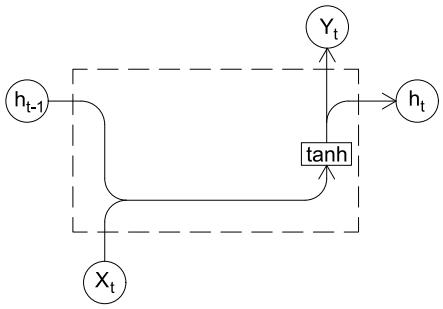

Fig. 1. Naïve RNN.

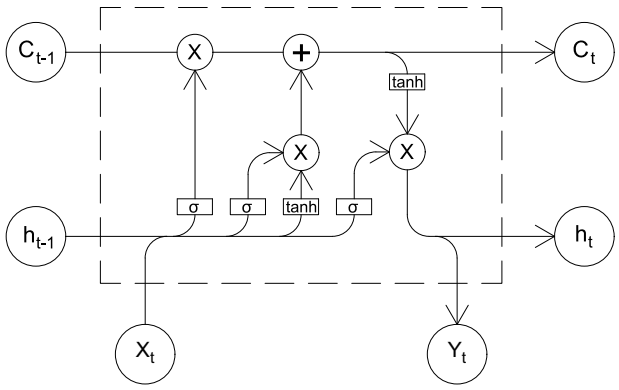

Fig. 2. LSTM RNN.

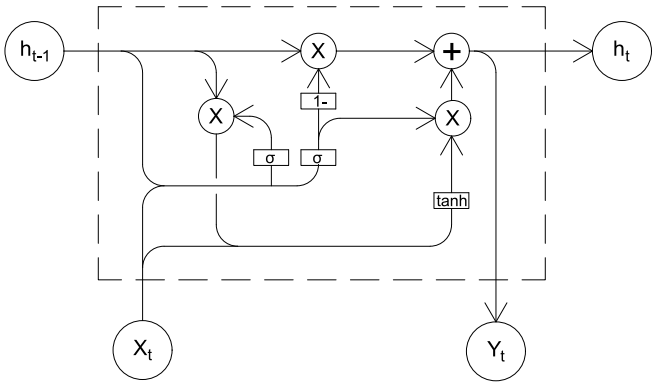

Fig. 3. Gate recurrent unit (GRU).

developed (Hochreiter and Schmidhuber 1997). Compared with the naïve RNN in Fig. 1, an additional layer of cell states, denoted as C_t in Fig. 2, is introduced. C_t is designed to preserve long-term information. A typical LSTM network is indicated in Fig. 2. A LSTM can be improved by reducing the number of parameters, facilitating computational efficiency.

For improved computational efficiency, a variational form of LSTM—GRUs—was introduced by Cho et al. (2014). The structure of a GRU is indicated in Fig. 3. GRUs solve the long-term dependency problem and gradient vanishing problem with fewer parameters than LSTMs. As with the previous networks, GRUs feature a chain-like structure. In Figs. 1–3, the complete naïve RNN, LSTM RNN, and GRU are repetitions of Figs. 1–3, respectively, connecting heads to tails as the system evolves from time step $t-1$ to t.

In this paper, an approach based on GRUs is proposed. The focus is on the structure of the hidden units in the neural network. The proposed Pairwise-GRU connects two GRUs with a transition box in between. By doing so, the approach is able to account for the effect of neighboring nodes in addition to the historical data in time series prediction. The result is more accurate and more confident predictions across the nodal network in a computationally efficient manner.

The remainder of this paper is organized as follows. The next section describes related work in time series prediction, including limitations to existing approaches. Then, the proposed approach for time series prediction in nodal networks that is based on GRUs, called Pairwise-GRU, is described. The Pairwise-GRU approach is then tested on a real-world application of an electricity network. The performance of the proposed approach and the effects of including neighboring nodes are analyzed. The impact of the type of neighboring node information in terms of time gap to the prediction time step is investigated in terms of both accuracy and uncertainty. The contributions of this paper are then concluded.

Aside from RNN and its variational forms, many methods have been proposed for time series prediction. Although not an exhaustive description, the following describes major approaches to the problem. The relevance vector machine (RVM) first proposed by Tipping (2001) features sparse Bayesian learning. Similar to the support vector machine (SVM), RVM has a kernel function, and the parameter learning is based on Bayes' rule. Using RVM, a multiscale relevance vector regression approach is proposed by Bai et al. (2014) to forecast daily urban water demand. However, the performance of RVM depends on the choice of bandwidth in the kernel function, which might lead to overfitting problems. SVM can be applied to time series predictions as well. In Sapankevych and Sankar (2009), SVM is applied in various scenarios, such as financial market forecasting and control system processes. Due to the highly nonlinear aspect of the data, similar to the RVM approach, the choice of a kernel function largely influences the performance of the prediction.

The group method of data handling (GMDH) is a family of inductive algorithms for modeling multiparametric datasets. GMDH automatically generates model structures and performs parametric optimization of models. The auto layer generation stops when a preset criterion is met. In Nikolaev and Iba (2003) and Shelekhova (1995), time series data are broken down into harmonic forms, in which the harmonic model parameters are learned through GMDH. Because no limitations exist on the number of layers that can be generated, the overfitting problem exists in GMDH as well.

The idea of grey systems is described in Deng (1982) to address the problem of incomplete information. In a grey system, part of the system information is known, and part of the information is unknown. With this definition, information quantity and quality form a continuum from a total lack of information to complete information—from black through grey to white. Applications of grey system theory to time series prediction are found in forecasting economic (Kayacan et al. 2010) and traffic volume (Xu and Zhang 2010) growth. However, when the underlying governing equation fails to depict the growth rules, the prediction becomes unreliable.

For a multivariate time series prediction, Althelaya et al. (2018) use RNN with unidirectional and bidirectional stacked structures to perform stock market forecasting. The nodal network problem addressed in this study can be thought of as a multivariate time series prediction problem, with predictions on the node of interest and the neighboring nodes. In Wei et al. (2021), RNN is used for the time series prediction on pore water pressure. The performances of RNN,

LSTM, and GRU are compared, with the authors finding that the LSTM and GRU models provide more precise and robust predictions than the standard RNN. Investigations are made into the structure of the hidden units in the RNN to improve the performance of time series predictions. To mitigate the overfitting problem, a dropout technique following a Bernoulli distribution is used. In Hu and Zheng (2019), additional operators are added to capture the short-term dependencies in the dataset. However, in these studies, a network of connected nodes and accompanying variables for prediction is not considered. Existing structures of the hidden units in a RNN cannot include the influence from neighboring nodes in the prediction. Thus, this paper proposes a new Pairwise-GRU approach to include the influence of neighboring nodes in the nodal prediction for the nodal networks of interest in this study.

For RNN applications in the civil engineering area, in Wei et al. (2021), a case study of pore water pressure prediction in the Hong Kong area is conducted based on the GRU. A comparison over different activation forms of GRU is performed, including linear, hard sigmoid, exponential, tanh, and others. However, for simplicity, in Wei et al. (2021), two representative measurement stations are selected from ten available installed piezometers, which exclude the neighboring effects in making time series predictions. In the Pairwise-GRU approach proposed in this paper, neighboring effects are considered by connecting the two separate GRU networks. In addition, predictions are made on all stations, enabling more comprehensive predictions over a system. In Jang et al. (2019), three input variables (accounting, construction market, and macroeconomic variables) are used to predict the business performance of a construction company. Based on the performance results, the most accurate prediction is given by considering all three variables as inputs. The output of the model given by Jang (2019) is targeted at predicting the success rate of a single construction project. In a more general scenario of multiple ongoing construction projects, the prediction model fails to include the dependencies between different construction projects. The proposed Pairwise-GRU approach is not limited to a single output and focuses on the challenge of capturing the interaction between different output components. Likewise, in Shahin (2014), the load-settlement response of a single drilled shaft is predicted based on RNN. Because the RNN model given by Shashin (2014) works on each pile individually, the prediction model fails to consider the group pile effect. In the proposed Pairwise-GRU approach, the influence between different target output components is considered by the proposed RNN structure.

Finally, to address the gradient vanishing and explosion problem, another approach is to set a window size on the training data. In Bai et al. (2014), the prediction of the daily urban water demand is limited to a training window size of seven days (one week). The selection of the size of the window is determined by the autocorrelation and saturated correlation dimension methods (Holzfuss and Mayer-Kress 1986). In Frank et al. (2001), time series prediction by neural networks is applied to Lorenz data, voice traffic demand, and tree ring data. Discussions are had on the selection of the size of a sliding window. In that study, the optimal window size is chosen by the false nearest neighbor method and singular-value analysis. The test application in this paper performs time series prediction based on hourly data of electricity consumption. The data indicate a strong periodic characteristic of 24 h. Therefore, in the test application part of this paper, a moving window size for prediction is selected as 48 h, accounting for the nearest 48-h dataset. By doing so, the periodic nature of the data is represented while accounting for some error between periods in consideration of both the computational efficiency and the prediction performance.

Proposed Method

Target Network

The network of interest consists of nodes with time series data information at each node. The goal is to make a prediction at each node based on the given historical data. To account for the influence of neighboring nodes in the time series prediction, the Pairwise-GRU approach is proposed.

Pairwise-GRU

The proposed method originates from the traditional GRU. The structure of the proposed Pairwise-GRU is indicated in Fig. 4 and consists of two separate GRUs connected by the box in the middle. The connection ensures that one node can share information with its neighboring node. By adding the middle box, the influence of neighboring nodes is considered. Although the proposed Pairwise-GRU connects two GRUs, it can be extended to include multiple GRUs in the prediction. The choice will be a tradeoff between computational cost and accuracy, with increased parameters if several GRUs are involved. The pairwise structure is selected to demonstrate the approach and is indicated to result in increased accuracy with efficient computational times.

In Fig. 4, the upper script of the variables, that is, 1 in X_t^1 and 2 in X_t^2, refers to the first (1)/second (2) time series sequence. The upper script also indicates the top/bottom GRU represented in a dashed

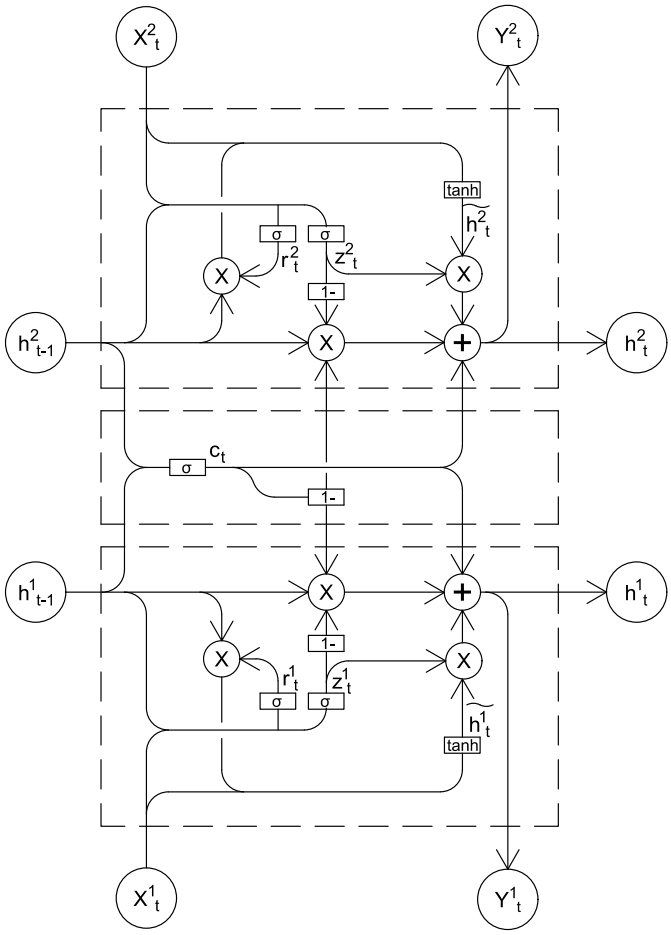

Fig. 4. Proposed Pairwise-GRU.

box in Fig. 4. In each time series sequence, Y follows X. As indicated in Fig. 4, the structure has one hidden layer and two hidden neurons. The operation in the top/bottom dashed box follows the rules given by the traditional GRU. The calculation process is as follows:

$$z_t^1 = \sigma(w_z^1 \cdot [h_{t-1}^1, X_t^1]) \tag{1}$$

$$z_t^2 = \sigma(w_z^2 \cdot [h_{t-1}^2, X_t^2]) \tag{2}$$

$$c_t = \sigma[w_c \cdot [h_{t-1}^1, h_{t-1}^2]] \tag{3}$$

$$r_t^1 = \sigma(w_r^1 \cdot [h_{t-1}^1, X_t^1]) \tag{4}$$

$$r_t^2 = \sigma(w_r^2 \cdot [h_{t-1}^2, X_t^2]) \tag{5}$$

$$\tilde{h_t^1} = \tanh(w^1 \cdot [r_t^1 \otimes h_{t-1}^1, X_t^1]) \tag{6}$$

$$\tilde{h_t^2} = \tanh(w^2 \cdot [r_t^2 \otimes h_{t-1}^2, X_t^2]) \tag{7}$$

$$h_t^1 = (1 - z_t^1) \otimes (1 - c_t) \otimes h_{t-1}^1 + z_t^1 \otimes c_t \otimes \tilde{h_t^1} \tag{8}$$

$$h_t^2 = (1 - z_t^2) \otimes (1 - c_t) \otimes h_{t-1}^2 + z_t^2 \otimes c_t \otimes \tilde{h_t^2} \tag{9}$$

$$Y_t^1 = h_t^1 \tag{10}$$

$$Y_t^2 = h_t^2 \tag{11}$$

Eqs. (1) and (2) refer to the update gates, denoted by z as the update gate parameter. They deal with the historical information from the collected dataset X and the hidden layer h and are the prerequisite for determining the information to remember during the processing. Eqs. (4) and (5) refer to the reset gates, denoted by r as the reset gate parameter. They also consider the historical information from the collected dataset X and the hidden layer h and are used to decide how much of the past information to forget in the model processing. Eqs. (3), (6), and (7) combine the information from the neighboring node, denoted by the parameters c and $\tilde{h}$. Here, for simplicity, the sigmoid and one minus functions are used to consider the neighboring effect. In Eqs. (1) to (7), w, w_z, w_c, and w_r are the weight matrices corresponding to h, z, c, and r, respectively. All weight matrices are learned based on the details indicated in the *learning* section of the process. Eqs. (8) to (11) are output equations that calculate the predictions and hidden layer information used for the next time step calculation.

Preprocessing

The performance of the prediction from the GRU depends on the training data input. In practice, the GRU is found to perform better when the input is monotonically increasing. This is due to the calculation equations between each time step being monotonically increasing, such as tanh and sigmoid, and the difficulty in curve fitting the training set to learn the parameters if the input is oscillating or varying heavily. Therefore, because parameter learning occurs over multiple steps, a monotonically increasing input facilitates smoother parameter learning and is more suitable for the multiple step ahead prediction objective of this study. Additionally, for parameter learning, the more recent historical data are found to be more predictive of future datapoints. Therefore, the more recent data are selected as the training data for parameter learning and are used for the prediction. A moving window size of 48 h is chosen, and the data series is transformed into a normalized monotonically increasing data series using Eq. (12). Due to the characteristics of the operators in Eqs. (1) to (11), for example, tanh and sigmoid, the function varies between –1 and 1. To efficiently train the parameters, the data are preprocessed to be normalized between 0 and 1. Alternative normalization approaches, such as the min-max scaling used in Hu and Zheng (2019) or the min-max normalization to fit the ranges of the tanh and sigmoid functions in Wei et al. (2021), can also be used to normalize the original data. In Eq. (12), let the training data series be X_i and the transformed data series be $\widehat{X}_i$, where $i = 1, 2, \ldots, 48$. The relation between $\widehat{X}_i$ and X_i is established as

$$\widehat{X}_i = \frac{X_i}{a_i \max(\{X_i\})} \tag{12}$$

where $\max(\{X_i\})$ refers to the maximum element in set $\{X_i\}$, and a_i is a linearly decreasing scale factor ranging from 4.5 to 1.5. Although not guaranteed, by this transformation, the original input training time series data are likely to be transformed into a monotonically increasing time series data ranging from 0 to 1.

Neighbor Definition

Neighbors in a network can be defined in numerous ways. For example, neighbors can be defined through physical connections. Alternatively, neighbors can be defined based on their characteristics. Defining a neighboring node based on the similarity of its time series data has been found to provide the best performance in terms of both computational efficiency and prediction accuracy. Iglesias and Kastner (2013) and Gonzalez-Abril et al. (2014) proposed multiple methods to identify the distance (similarity) between two sets of time series data.

The similarity measure adopted in this study quantifying the similarity between sets of time series data uses the percentiles of the data. For time series data X, let

$$Q_X = \{p_{1X}, \ldots, p_{qX}\} \tag{13}$$

be a set of q percentiles of the dataset X in ascending order and $q \geq 2$. Intervals are defined as

$$I_{iX} = (p_{iX}, p_{(i+1)X} \tag{14}$$

with $i = 1, 2, 3, \ldots, q-1$.

The similarity score between two datasets A and B are then denoted by $K(A, B)$ defined as

$$K(A, B) = \frac{1}{1 + \frac{1}{q-1}\sum_{i=1}^{q-1} d(I_{iA}, I_{iB})} \tag{15}$$

where $d(I_{iA}, I_{iB})$ = Euclidean distance. A higher score results in two datasets that are more similar. For two identical datasets, the similarity score is 1. Using these definitions, if data series A is the closest (most similar) one to data series B, then node A is defined as the neighboring node of node B. A data-driven similarity-based approach is used here to facilitate the data-driven prediction. Alternative methods for defining neighbors in the network can also be used based on the specific characteristics of the network of interest if such information is available.

Learning

The loss function for the GRU is defined as the sum of the squared errors. The efficiency of the learning depends on the initial values set on the parameters. In the proposed approach, the learning process is divided into two parts. First, the parameters for the two traditional GRUs are learned separately. Then, the parameter values

learned in the previous two separate GRUs are taken as the initial values in the Pairwise-GRU. The learning process in the Pairwise-GRU then finds the parameters minimizing the loss function. Let the untransformed outcome value be $\breve{Y}_i$ and the actual collected data after transformation be $\hat{Y}_i$ at time step i. Then, the loss function L is defined as

$$L(w, w_z, w_c, w_r) = \sum_i \{\breve{Y}_i(w, w_z, w_c, w_r) - \hat{Y}_i\}^2 \tag{16}$$

Let the weight matrices prior to updating be w, w_z, w_c and w_r. Then, the weight matrices after updating denoted $\tilde{w}$, $\widetilde{w_z}$, $\widetilde{w_c}$, and $\widetilde{w_r}$ are found as indicated in Eqs. (17) to (20). α is the defined learning rate that is taken as 0.01 in the test application. For the test application, gradient descent is used to optimize the parameters, the training window size is 48 h, the batch size is 48, and the epoch number is 1

$$\tilde{w} = w - \alpha L / \frac{\partial L}{\partial w} \tag{17}$$

$$\widetilde{w_z} = w_z - \alpha L / \frac{\partial L}{\partial w_z} \tag{18}$$

$$\widetilde{w_c} = w_c - \alpha L / \frac{\partial L}{\partial w_c} \tag{19}$$

$$\widetilde{w_r} = w_r - \alpha L / \frac{\partial L}{\partial w_r} \tag{20}$$

Predicting

During the preprocessing, the original input data are transformed into a monotonically increasing data series. Thus, once an outcome is obtained, the transformation needs to be undone to result in the data series prediction. Let the untransformed outcome be $\breve{Y}$ and the final prediction be $\bar{Y}$. Then, the transformation from $\breve{Y}$ to $\bar{Y}$ is

$$\bar{Y} = \breve{Y} * a_y * \max(\{X_i\}) \tag{21}$$

where a_y = scale factor depending on the time step of the prediction.

Testing and Evaluation

The performance of the proposed Pairwise-GRU is evaluated in terms of two measures: accuracy and uncertainty. The accuracy of the prediction, denoted p_j, is evaluated by comparing the predicted value to the actual collected data value. Let the predicted value be $\overline{Y_j}$ and the actual collected data be Y_j at time step j. Then, the prediction accuracy at time step j is

$$p_j = 1 - \left| \frac{\overline{Y_j} - Y_j}{Y_j} \right| \tag{22}$$

The coefficient of variation (CoV) is used to measure the confidence level or uncertainty in the predicted data. The CoV for a prediction at time step j is determined by the performance of the prediction, including uncertainty in the training set.

In summary, the workflow of the proposed Pairwise-GRU method is as indicated in Fig. 5.

Test Application

Electricity Network

The electricity grid network in Florida is taken as the example network for testing the proposed approach. Electricity consumption data are collected on an hourly basis from 10 stations in August and December of 2019. Each month has 744 consecutive time series data points collected for each station (U.S. Energy Information Administration 2020). This period is chosen to include data from normal operating and hazard conditions, as well as data from different seasons. The stations are connected, as indicated in Fig. 6. Each station is treated as a node in the network, and the physical connections between the nodes are kept as links. The objective is to predict the time series data for electricity consumption at each station using just the historical time series information over the network without information from any variables external to the network. Analyses for this test application are completed on a 16 GB RAM computer in MATLAB_R2017b.

Neighbor Identification

Based on the previous discussion, the closest (most similar) datasets are considered neighbors. Eqs. (13) to (15) are used to calculate similarity scores based on the first 100 time series datapoints in December. After comparing the similarities between all pairs, the neighbors in this test application are defined as indicated in Table 1. From the results in Table 1, the highest similarity score is not necessarily associated with the closest physical connection. Station 7 (SEC-7) is most dissimilar to the other datasets. A neighbor is still chosen for the Pairwise-GRU, and the method is able to provide relatively accurate bounded prediction results by taking into account and learning weights between the historical data and neighboring node information, as indicated in the results following.

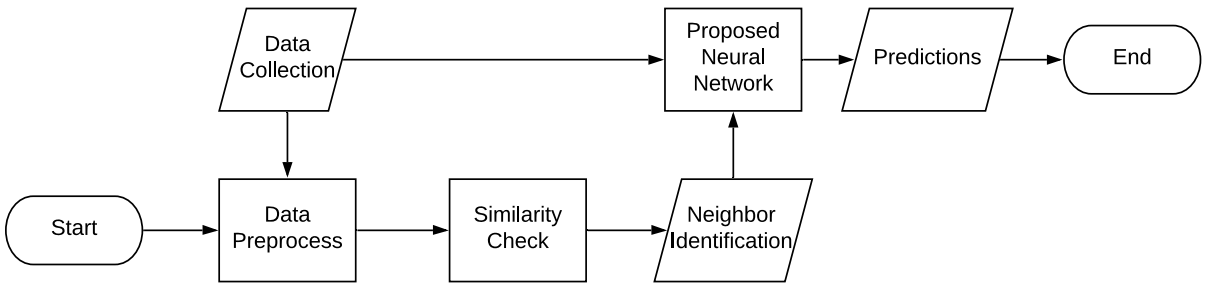

Fig. 5. Workflow for building proposed Pairwise-GRU.

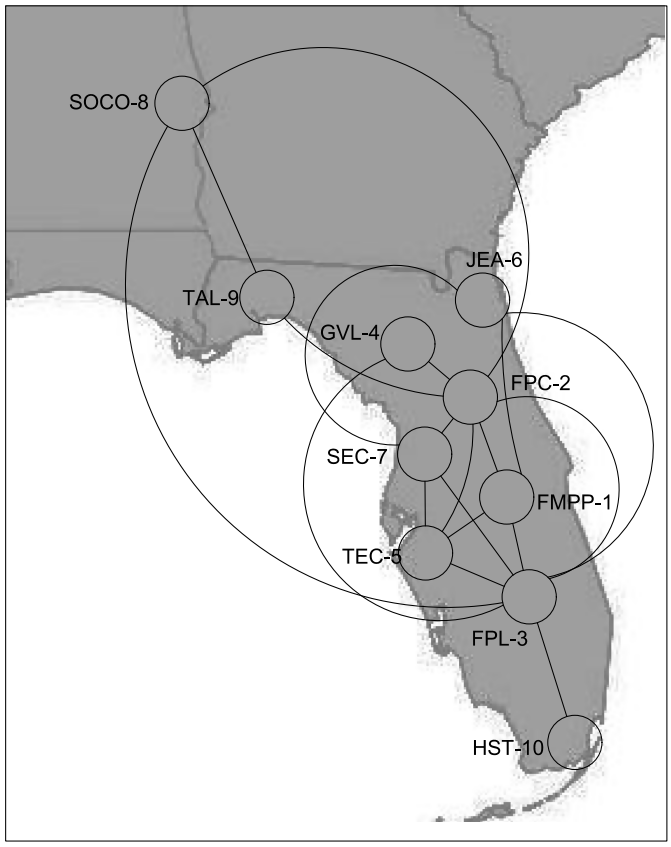

Fig. 6. Network configuration.

Table 1. List of neighbors

Node of interest	Neighboring node	Similarity score
FMPP-1	TEC-5	0.9746
FPC-2	TEC-5	0.9759
FPL-3	TEC-5	0.9827
GVL-4	SOCO-8	0.9729
TEC-5	FPL-3	0.9827
JEA-6	TAL-9	0.9712
SEC-7	GVL-4	0.2120
SOCO-8	GVL-4	0.9729
TAL-9	JEA-6	0.9712
HST-10	SOCO-8	0.9691

Methods Comparison

To assess the performance of the proposed method, five time series prediction methods (grey systems, RVM, GMDH, GRU, and the proposed Pairwise-GRU) with a single variable input are compared. Fig. 7 indicates the accuracy of the prediction across these methods as a function of the number of steps ahead of the prediction. Predictions are made from December 23 to 24 in the dataset. The results for additional periods in the analyzed dataset are provided later in this section. System-level accuracy is assessed as the average accuracy over all stations, with results indicated from one-step-ahead prediction to 24-steps-ahead prediction, that is, one-hour-ahead to one-day-ahead predictions. The multiple time steps ahead prediction is achieved sequentially using the newly predicted values. For example, for a two-steps-ahead prediction, a one-step-ahead prediction is first generated. That prediction is then used generate a two-steps-ahead prediction, and so on. Based on the results indicated in Fig. 7, all methods perform relatively well for short-term predictions. However, for long-term predictions, the GRU and proposed Pairwise-GRU approaches perform significantly better than the other three methods. The proposed Pairwise-GRU approach indicates the highest accuracy, particularly for multiple time steps ahead predictions.

Performance: Accuracy and Uncertainty

Given GRU and Pairwise-GRU as the top performing methods, this section performs a detailed comparison between the two approaches under three cases, including both normal operating and hazard conditions. All comparisons are made based on a three-step-ahead prediction. Looking chronologically through the dataset, the three cases for prediction are as follows: from August 21 to 22, as a dataset for the system operating under normal conditions; from December 23 to 24, where a flooding hazard impacted node FPL-3 and node HST-10 [flooding information from the National Weather Service (2020)]; and from December 29 to 30, as a second dataset for normal operating conditions. The results comparing prediction performance using the GRU versus the proposed Pairwise-GRU are summarized in Table 2. The GRU includes information only from the time history. The proposed Pairwise-GRU takes into account both the time history and information from neighboring nodes. To further the comparison, the traditional GRU is analyzed under two scenarios: under a single variable input scenario that separately processes information for the ten stations, and under a multiple variables input scenario that treats the entire network as a single unit and takes information across the 10 stations at once as input.

In Table 2, performance is assessed in terms of prediction accuracy calculated by Eq. (16). System-level accuracy is indicated. Prediction uncertainty is measured in terms of the CoV. From Table 2, across normal operating and hazard conditions, the Pairwise-GRU approach results in increased accuracy and decreased uncertainty (CoV) in the prediction compared with the traditional GRU. Even adding variables with the traditional GRU does not improve the results and, in fact, degrades them, as indicated with the multiple variables analysis. The results indicate improved performance in terms of both accuracy and CoV across all three periods, supporting the generalizability of the performance results of the proposed method. The findings suggest that by combining time history information with neighboring node information, the proposed Pairwise-GRU is able to more flexibly adapt to data and result in more accurate time series predictions across both normal and anomalous data conditions.

To provide a more detailed assessment of the time series prediction performance at individual stations, Figs. 8–10 compare prediction results from the GRU under a single variable input and Pairwise-GRU at each node in the network. Bounds are obtained based on CoV and indicated for a 95% confidence level following a normal distribution for each station. The confidence level is obtained through the performance of the training set, that is, with the window size of 48 h and one realization or one run of the Pairwise-GRU to obtain the confidence level. Assuming the prediction follows a normal distribution, the CoV of the training set is applied to the predictions to find the upper and lower bounds. Alternate bounds can be found based on different underlying distributions if desired as long the distributions can be defined based on mean and standard deviation values. The three periods considered in Figs. 8–10 are the same as those investigated in Table 2. The upper (lower) bound 1 represents the results from the traditional GRU. The upper (lower) bound 2 are the results from the proposed Pairwise-GRU. It can be easily seen that a narrower bandwidth is achieved by the proposed Pairwise-GRU approach for all stations and across all three prediction periods. For the most part, the actual data lie within the 95%

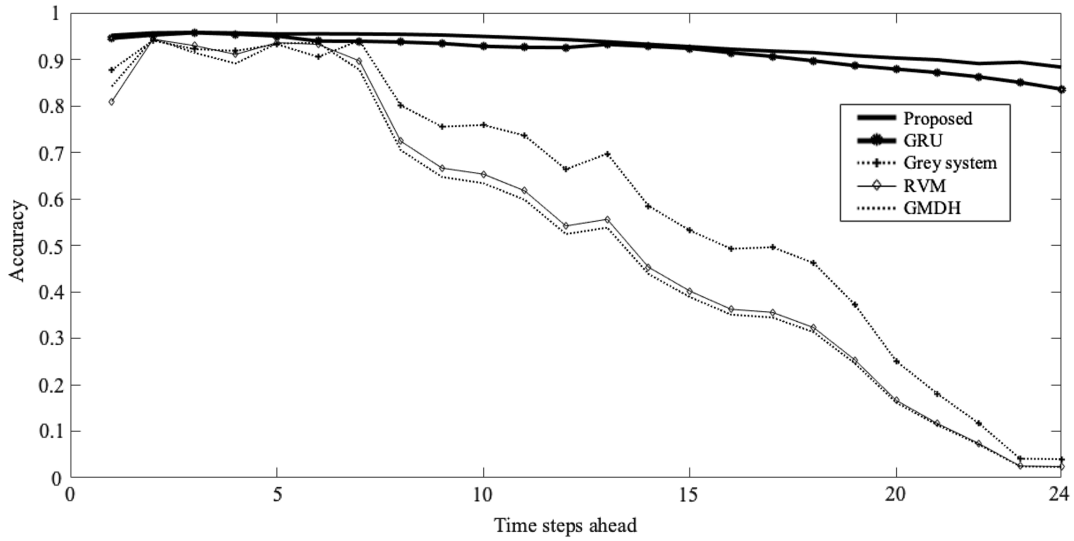

Fig. 7. Accuracy comparison across time series prediction methods.

Table 2. Accuracy and uncertainty performance comparison between GRU and proposed Pairwise-GRU methods

		History only (GRU)					
		Single variable		Multiple variables		Proposed (pairwise-GRU)	
Condition	Time range	Average accuracy	Average CoV	Average accuracy	Average CoV	Average accuracy	Average CoV
Normal	August 21–22	94.61%	0.0690	90.02%	0.0773	97.20%	0.0495
Flood	December 23–24	94.19%	0.0713	93.91%	0.0736	95.78%	0.0535
Normal	December 29–30	92.89%	0.0727	90.40%	0.0866	95.83%	0.0653

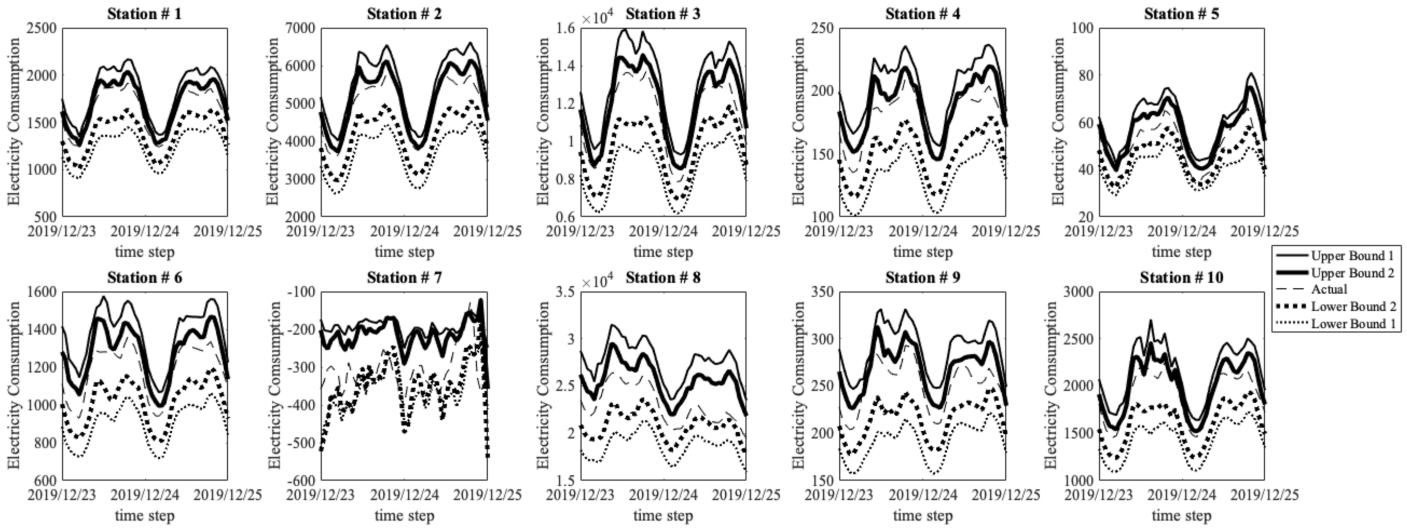

Fig. 8. Prediction at each station under normal operating conditions (August 21–22).

confidence bounds for the Pairwise-GRU. Even for station 7, which had a low similarity score relative to the other nodes, as indicated in Table 1, the predicted results using the Pairwise-GRU are improved from the traditional GRU. This is because the Pairwise-GRU is able to learn weights when considering the influence of the historical data compared with the neighboring node information in the prediction. The structure of the Pairwise-GRU given in Fig. 4 considers both neighboring node and historical effects. The learning weight matrices in Eqs. (1) to (11) determine the influence of both factors. In the extreme case in which the neighboring matrix parameter is set as 0, the influence of the neighboring node will not be included, and the same could occur for the historical data. In general, the Pairwise-GRU will learn some non-zero weight for the two factors, taking into account both historical data and neighboring node information to result in more accurate and confident time series predictions.

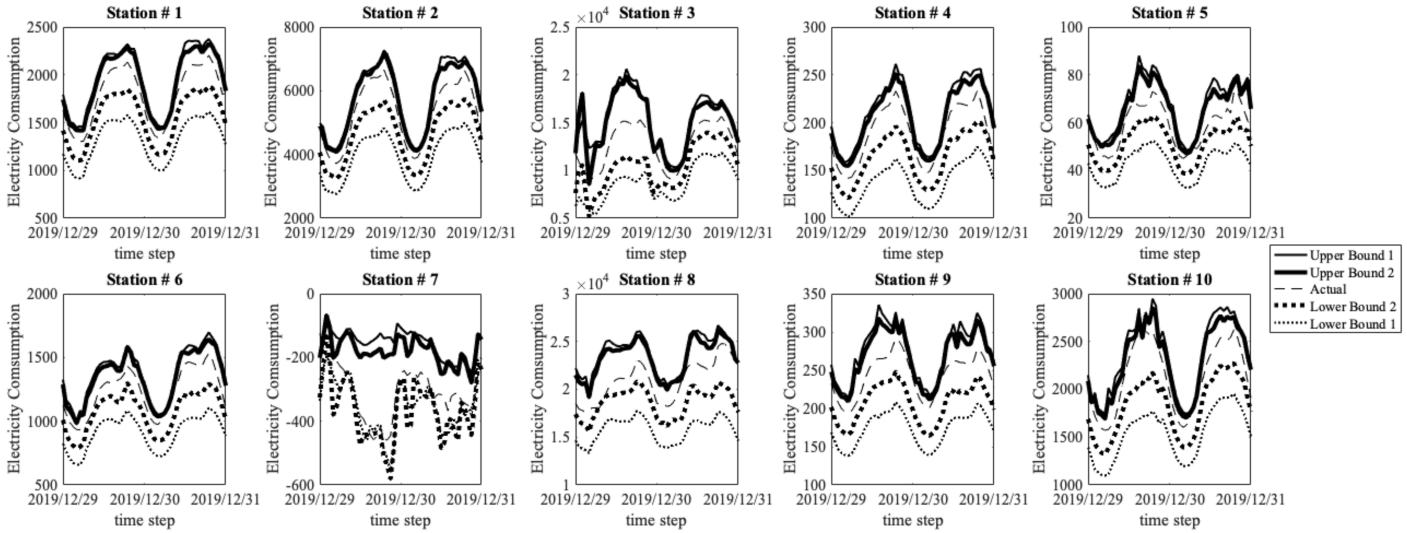

Fig. 9. Prediction at each station under flooding hazard conditions (December 23–24).

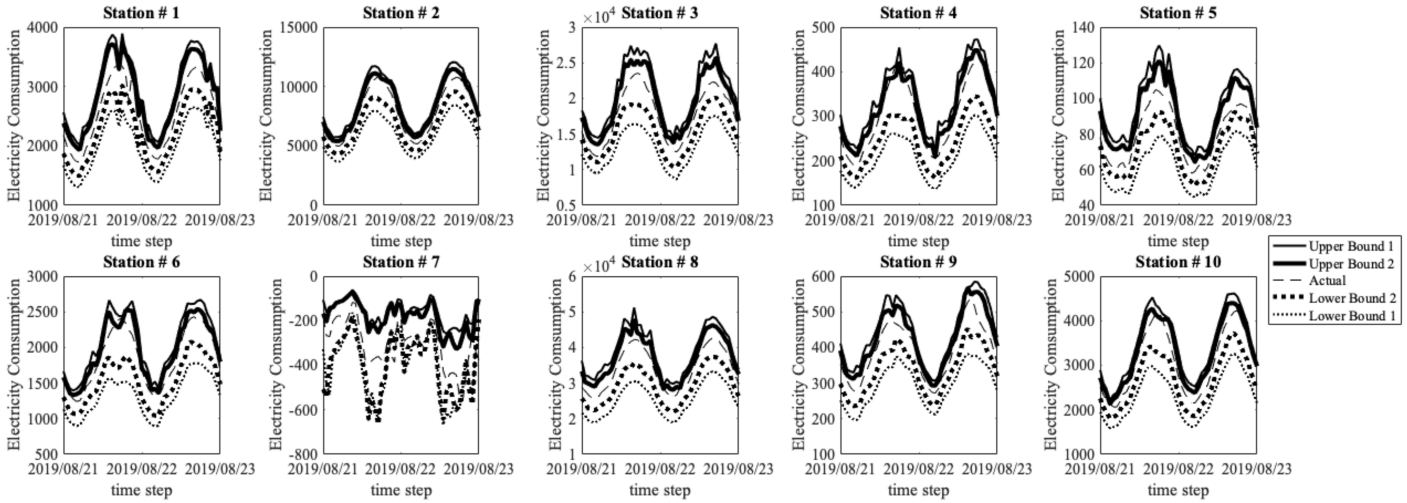

Fig. 10. Prediction at each station under normal operating conditions (December 29–30).

Effect of Neighboring Node Information

To more closely evaluate the effect of neighboring node information on the prediction, this section investigates the influence of the type of neighboring node information on the prediction in terms of the time gap to the prediction time step. This is applicable for the case in which the data collection among all stations is not synchronized, or different nodes in the network have varying levels of information. Fig. 11 indicates the accuracy and uncertainty of the prediction as the neighboring node information changes. For the observation time gap indicated on the x-axis in Fig. 11, zero indicates that the data collection is synchronized on the node of interest and the neighboring node; that is, data are collected for the same time step between the prediction and the neighboring nodes. A positive number on the x-axis, such as $+3$, indicates that the data collection on the neighboring node is three hours ahead of or three hours more up to date than the node of interest. Similarly, a negative number, such as -3, indicates that the data on the neighboring node is three hours behind compared with the time step of the node of interest. Fig. 11 indicates the prediction performance for a range of time differences between the neighboring node and the node of interest, from –6 to 6. For each observation time gap value, 100 points are sampled across the system, including different stations and different periods. All sample points are plotted in Fig. 11 for accuracy and CoV for three-hour ahead predictions, including median and 5 and 95 percentile values.

Fig. 11 makes it clear that the prediction improves when the neighboring node offers more information that is more up to date. The accuracy increases, and the uncertainty (CoV) decreases as the neighboring node information becomes more up to date or even ahead of the node of interest. The percentile curves also become tighter, indicating decreasing variability in the results in terms of both accuracy and uncertainty as the neighboring node data improve.

Finally, in terms of computational efficiency, Table 3 provides the computational time required for the GRU compared with the

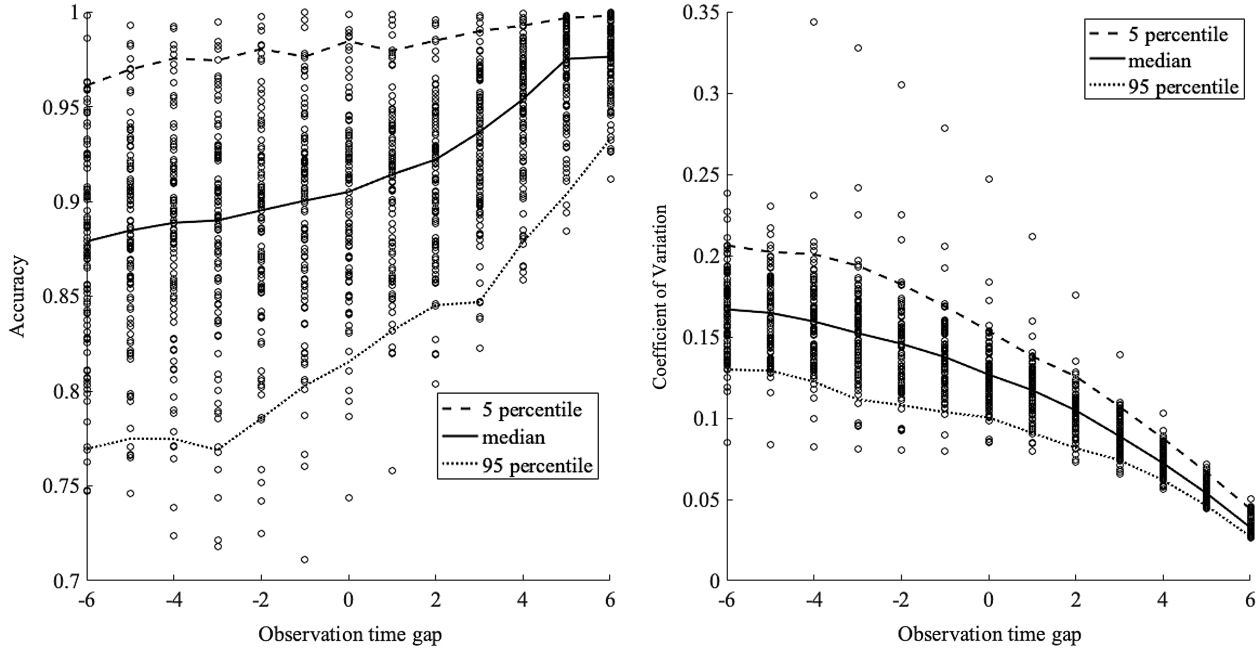

Fig. 11. Influence of neighboring node information on accuracy and uncertainty in prediction.

Table 3. Computational time comparison

GRU	Computational time (s)
Single variable	11.98
Multiple variables	18.67
Pairwise	14.55

Pairwise-GRU. At the cost of an increased time increment of shorter than three seconds compared with the single variable input GRU approach, the Pairwise-GRU offers a method to potentially increase the accuracy and decrease the uncertainty of the time series prediction.

Conclusion

This paper proposes a new recurrent neural network, called the Pairwise-GRU, to perform time series predictions in nodal networks. Compared with the structure of a traditional GRU, a connection is implemented in which the influence of both historical data and the information from neighboring nodes is considered in the prediction. Despite the fact that double the parameters are involved in the newly proposed RNN, the computational time is not significantly increased with the initial value setting method for parameter learning. Compared with existing methods, the proposed Pairwise-GRU improves the accuracy and confidence level of the prediction. These results are found across nodes in the network and across both normal operating and hazard conditions. The proposed approach works particularly well for multiple time steps ahead prediction. The Pairwise-GRU is able to weigh the effects of the historical data and neighboring node information, such that even if the neighboring node is not highly relevant to the node of interest, the result is not largely influenced. The performance of the prediction in terms of both accuracy and uncertainty improves significantly when the neighboring node has up-to-date information. The proposed Pairwise-GRU provides a way to take into account not only time history information but also the nodal characteristics within a network for time series prediction.

Data Availability Statement

All data, models, or code that support the findings of this study are available from the corresponding author on reasonable request.

References

Althelaya, K. A., M. El-Alfy El-Sayed, and S. Mohammed. 2018. "Stock market forecast using multivariate analysis with bidirectional and stacked (LSTM, GRU)." In *Proc., 21st Conf. of Saudi Computer Society National Computer (NCC)*, 1–7. New York: IEEE.

Bai, Y., P. Wang, C. Li, J. Xie, and Y. Wang. 2014. "A multi-scale relevance vector regression approach for daily urban water demand forecasting." *J. Hydrol.* 517 (Sep): 236–245. https://doi.org/10.1016/j.jhydrol.2014.05.033.

Cho, K., B. Merrienboer, C. Culcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio. 2014. "Learning phrase representations using RNN encoder-decoder for statistical machine translation." Preprint, submitted June 6, 2014. http://arxiv.org/abs/1406.1708.

Deng, J. 1982. "Control problems of grey systems." *Syst. Control Lett.* 1 (5): 288–294. https://doi.org/10.1016/S0167-6911(82)80025-X.

Frank, R. J., N. Davey, and S. P. Hunt. 2001. "Time series prediction and neural networks." *J. Intell. Rob. Syst.* 31 (1): 91–103. https://doi.org/10.1023/A:1012074215150.

Gonzalez-Abril, L., J. M. Gavilan, and F. Velasco Morente. 2014. "Three similarity measures between one-dimensional dataSets." *Revista Colombiana de Estadística* 37 (1): 79–94. https://doi.org/10.15446/rce.v37n1.44359.

Hochreiter, S., and J. Schmidhuber. 1997. "Long short-term memory." *Neural Comput.* 8 (9): 1735–1780. https://doi.org/10.1162/neco.1997.9.8.1735.

Holzfuss, J., and G. Mayer-Kress. 1986. "An approach to error-estimation in the application of dimension algorithms." *Dimens. Entropies Chaotic Syst.* 17 (13): 114–122. https://doi.org/10.1007/978-3-642-71001-8_15.

Hu, J., and W. Zheng. 2019. "Transformation-gated LSTM: Efficient capture of short-term mutation dependencies for multivariate time series

prediction tasks." In *Proc., Int. Joint Conf. on Neural Networks (IJCNN)*, 1–8. Mumbai, India: Tata Consultancy Services.

Iglesias, G., and W. Wastner. 2013. "Analysis of similarity measures in time series clustering for the discovery of building energy patterns." *Energies* 6 (2): 579–597. https://doi.org/10.3390/en6020579.

Jang, Y., I. Jeong, and Y. Cho. 2019. "Business failure prediction with LSTM RNN in the construction industry." In *Proc., Int. Conf. on Computing in Civil Engineering 2019: Data, Sensing, and Analytics*, 114–121. Reston, VA: ASCE.

Kayacan, E., U. Baris, and O. Kaynak. 2010. "Grey system theory-based models in time series prediction." *Expert Syst. Appl.* 7 (2): 1784–1789. https://doi.org/10.1016/j.eswa.2009.07.064.

Li, S., W. Li, C. Cook, C. Zhu, and Y. Gao. 2018. "Independently recurrent neural network (IndRNN): Building a longer and deeper RNN." In *Proc., IEEE Conf., on Computer Vision and Pattern Recognition (CVPR)*, 5457–5466. New York: IEEE.

National Weather Service. 2019. "December 2019 South Florida flooding." Accessed July 1, 2020. https://www.weather.gov/mfl/dec2019flooding.

Nikolaev, N. Y., and H. Iba. 2003. "Polynomial harmonic GMDH learning networks for time series modeling." *Neural Netw.* 16 (10): 1527–1540. https://doi.org/10.1016/S0893-6080(03)00188-6.

Sapankevych, N. I., and R. Sankar. 2009. "Time series prediction using support vector machines: A survey." *IEEE Comput. Intell. Mag.* 4 (2): 24–38. https://doi.org/10.1109/MCI.2009.932254.

Shahin, M. A. 2014. "Load-settlement modeling of axially loaded drilled shafts using CPT-based recurrent neural networks." *Int. J. Geomech.* 14 (6): 06014012. https://doi.org/10.1061/(ASCE)GM.1943-5622.0000370.

Shelekhova, V. Y. 1995. "Harmonic algorithm GMDH for large data volume." *Syst. Anal. Modell. Simul.* 20 (1): 117–126. https://doi.org/10.5555/214610.214620.

Tipping, M. 2001. "Sparse Bayesian learning and the relevance vector machine." *J. Mach. Learn. Res.* 1 (Jun): 211–244. https://doi.org/10.1162/15324430152748236.

U.S. Energy Information Administration. 2020. "Florida state profile and energy estimates." Accessed July 1 2020. https://www.eia.gov.

Wei, X., L. Zhang, H. Yang, L. Zhang, and Y. Yao. 2021. "Machine learning for pore-water pressure time series prediction: Application of recurrent neural networks." *Geosci. Front.* 12 (1): 453–467. https://doi.org/10.1016/j.gsf.2020.04.011.

Xu, N., and X. Zhang. 2010. "Traffic volume prediction based on improved grey self-adaptable prediction formula." In *Proc., 9th Int. Conf., on Machine Learning and Cybernetics, Qingdao*, 11–14. New York: IEEE.