IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received October 11, 2020, accepted October 21, 2020, date of publication October 30, 2020, date of current version November 12, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3034875

River Flooding Forecasting and Anomaly Detection Based on Deep Learning

SCOTT MIAU AND WEI-HSI HUNG
Department of Management Information Systems, National Chengchi University (NCCU), Taipei City 11605, Taiwan (R.O.C.)
Corresponding author: Scott Miau (scottm@mitac.com.tw)

ABSTRACT Pluvial floods are rare and dangerous disasters that have a small duration but a destructive impact in most countries. In recent years, the deep learning model has played a significant role in operational flood management areas such as flood forecasting and flood warnings. This paper employed a deep learning-based model to predict the water level flood phenomenon of a river in Taiwan. We combine the advantages of the CNN model and the GRU model and connect the output of the CNN model to the input of the GRU model, called Conv-GRU neural network, and our experiments showed that the Conv-GRU neural network could extract complex features of the river water level. We compared the predictions of several neural network architectures commonly used today. The experimental results indicated that the Conv-GRU model outperformed the other state-of-the-art approaches. We used the Conv-GRU model for anomaly/fault detection in a time series using open data. The efficacy of this approach was demonstrated on 27 river water level station datasets. Data from Typhoon Soudelor in 2015 were investigated by our model using the anomaly detection method. The experimental results showed our proposed method could detect abnormal water levels effectively.

INDEX TERMS Floods, river water level, forecasting, deep learning, anomaly detection.

I. INTRODUCTION

Floods are one of the most common and destructive natural disasters. They cause massive damage to human life, infrastructure, and socioeconomic systems. In 2018, the Reviews of Disaster Events by Emergency Events Database (EM-DAT) [1] reported that floods have affected more people than any other type of natural hazard in the 21st century. Floods occur almost everywhere in the world, resulting in massive damage and the loss of countless lives. In 2019, the National Science and Technology Center for Disaster Reduction (NCDR) [2] indicated that 361 major disaster events happened worldwide in that year, of which floods were the largest disaster event with a total of 170 incidents, accounting for 47% of the total. These events affected around 3 billion people and caused 5100 deaths.

Floods occur in many types [3]. Over the past years, climate change caused by global warming has been increasing the intensity of the occurrence of floods. The temperature in 2019 was 0.95°C higher than the historical global average temperature [2]. As the temperature rises, the atmosphere holds more water vapor, which leads to heavy rain and increases the risk of flooding in rivers [4].

Located in a subtropical region and the path of typhoons in the western Pacific, Taiwan has been severely affected by extreme weather in recent years. The natural environment formed by Taiwan's geographical location and climatic conditions has caused heavy rain and floods [5]. Its short and steep rivers with wide river channels are not conducive to the release the floods [6]. In addition, annual heavy rainfall and typhoons in Taiwan are not only highly intensive but also alarming. On average, 4.7 typhoons occur each year in Taiwan [7] and there is an annual rainfall average of about 2500 millimeters, of which about 80% is concentrated in the rainy and typhoon season [5]. When a typhoon approaches, typhoon brings heavy rainfall, which can easily cause the river water level to soar and cause floods.

Typhoon Soudelor occurred in August 2015 and caused a severe disaster in Taiwan. Its main affected area was the Greater Taipei area, where the capital is located. The estimated property loss was around 22 billion dollars (TWD). Further, it was the worst typhoon on record: eight people died, four went missing, 437 people were injured, 7,000 roadside trees dumped, and 4.5 million households suffered power

The associate editor coordinating the review of this manuscript and approving it for publication was Emre Koyuncu.

outages [8]. Since this area has low terrain downstream and drainage is difficult, an effective flood prediction measure in this area is particularly urgent and important. As a result, the Tamsui River was the first river to be used in research on flood forecasting systems in Taiwan [9]. In 1993, the Taiwan Water Resource Agency cooperated with the Foundation of River & Basin Integrated Communications (FRIC) to carry out improvements to the flood forecasting system [10].

The experiences of flood prevention in recent years indicate that no engineering structure can eliminate the risk of flood disasters. By combining with appropriate non-engineering measures (non- structural measures), floods can be reduced more efficiently [11]. A water flow prediction with high accuracy is a crucial non-structural measure needed to decrease casualties and property damage. Previous stages of water flow prediction were built on mathematical models and generally required a large amount of input data, which was sometimes difficult to obtain or insufficient. In addition, process-based approaches often provided delayed flood warnings, as long periods were needed to test carefully and evaluate the parameters. Traditional hydrological methods could not predict the increase of downstream flows if there were sudden fluctuations of upstream flows resulted from dam and reservoir releases. Such conditions can often be seen in countries with large river basins [12].

As the traditional methods of flood prediction did not perform well, in 2000, Toth *et al.* [12] compared the models of time-series analysis techniques, including the autoregressive moving average model (ARMA), artificial neural networks (ANN), and the K-nearest neighbor (KNN). The result showed a better accuracy of the ANN model in short-term rainfall forecasting. Although the ANN model requires a large amount of data, it is capable of handling both linear and non-linear systems without making any assumptions. Over the past two decades, the ANN model has been widely used in various fields of science and engineering. The better performance of ANN applied to rainfall prediction can also be seen in the research of Thirumalaiah *et al.* [13], Tsai *et al.* [5], and Kenabatho *et al.* [14].

In recent years, ANN, RNN, and CNN have been used for hydrology time series forecasting [15]–[17] and have achieved different levels of success. Previous studies have not presented the comparison among these deep-learning techniques, as these studies were applied to data from different river stations and evaluated with different metrics. Most previous studies mainly focused on the comparison with linear models and with ordinary ANN models. In this paper, we proposed a framework of management using river water level data for anomaly prediction. The data collection was based on the records of 27 water level stations in the Tamsui River Basin during Typhoon Soudelor. The goal was to provide a more accurate predictive model based on deep neural networks for river water levels, in the hope that it could provide timely warnings of anomalies.

This paper makes several contributions. Firstly, five deep learning models (Conv-GRU, ANN, CNN, LSTM, and Seq2seq) were compared, which has not been conducted in previous studies. Secondly, we proposed a model that combines CNN and GRU that is feasible and more capable of grasping patterns in a time series. We believed it to be more efficient than other models on flood forecasting. Thirdly, we applied the Conv-GRU network into anomaly/failure detection in the time series. The network became a predictor in multiple time steps through being trained on the non-anomalous data of the water levels at each water station. The result of the prediction error was modeled as a multivariate Gaussian distribution that could evaluate the possibility of abnormal behavior. Finally, we used the Mahalanobis Distance to present clear results on the degree and probability of anomalies detected by the Conv-GRU model.

II. RELATED WORK

A. DEEP LEARNING

Deep learning is a subfield in machine learning. It is an algorithm inspired by the structure and function of the brain, which is called an artificial neural network (ANN) [18]. This term was first introduced to the machine learning community by Dechter [19]. In 2000, Aizenberg *et al.* [20] used this term in the field of artificial neural networks, also called Deep Neural Networks (DNN). Concerning the definition of the term ''deep'', Hinton [21], as a pioneer in the ANN field, described a fast, greedy algorithm that can learn representations of data with multiple levels of abstraction. Therefore, it has been considered to introduce this phrase as referring to the development of large artificial neural networks. Feature learning of hierarchies is the core element in deep learning. Bengio [22] emphasized that automatically learning features at multiple levels of abstraction can allow a system to learn complex functions by mapping the input to the output directly from the data without depending on human-crafted features. In 2009, Goodfellow *et al.* [23] also stated that the concepts of a hierarchy allow computers to learn complicated concepts by building them out of one concept. Therefore, with more data, bigger models, and more computation, better results can be provided by deep learning [24]. Nowadays, deep learning has been widely applied in various fields for image recognition, speech recognition, natural language processing, recommendation systems, and biomedical information, etc. Several models of deep learning and their applications in flood forecasting systems are introduced in the following section.

B. CONV-GRU

The Convolutional Gated Recurrent Unit (Conv-GRU) was firstly introduced by Bellas *et al.* [25] and was considered as an extension of the GRU-RCN model. In their model, GRU-extension encodes the locality and temporal smoothness prior to videos directly in the model structure, to take advantage of ''percepts'' from different spatial resolutions [25]. In the beginning, the purpose of developing the Conv-GRU model was to determine spatio-temporal features from

videos, and it was used for video captioning and action recognition. Bellas *et al.* used recurrent convolutional units on pre-trained CNN convolutional maps to extract temporal patterns from visual percepts with different spatial sizes [25]. In the paper, they decided to adopt GRU networks, as GRU showed similar performance to Long Short Term Memory (LSTM) but with a lower memory requirement. GRU was firstly introduced by Cho *et al.* [26], [27] as a simplified model of LSTM. Their GRU model has an identical role in the network but has only two gates and fewer parameters than LSTM. Because it lacks an output gate, there is no control over the memory content. GRU has similar performance to LSTM but has a reduced number of gates and therefore fewer parameters. GRU is comparable to LSTM [28].

C. ARTIFICIAL NEURAL NETWORKS

An Artificial Neural Network (ANN) is a mathematical model generated by the biological neural networks that constitute animal brains [29]. Hebbian theory [30], proposed in 1949, is its earliest theoretical basis. Rosenblatt's Perceptron [31] established the prototype of ANN by using the structure of nerve cells to set up a model from a mathematical structure. The first successful application of ANN, handwritten font recognition, was performed using back-propagating neural calculus, proposed by LeCun *et al.* [32]. With the successful development of multiple hidden layers in the neural network model, which greatly improved the prediction and unveiled the deep learning approach by elaboration was carried out by Hinton *et al.* [21]. Over the past few years, ANN has been widely used to solve a large number of tasks from different domains, including computer vision, speech recognition, machine translation, social network filtering, playing board and video games, medical diagnoses, and even painting.

The tremendous success of ANN is determined by its ability to learn from past examples. ANN starts to learn after receiving a representative set of examples, and it finds and extracts the structure and features of the data automatically. Another success factor is its capability to handle the dependencies of complex nonlinear relationships between input and output data sets, which may occur more in real life [33].

Over the past years, ANNs have been applied in the hydrological field for items such as water flow (stream) modeling, water quality assessment, and suspended sediment load prediction. The first application of ANNs in hydrology was in the early 1990s [34], where it was found that ANN has better performance for hydrological forecasts. Afterward, ANN was successfully validated in many hydrological applications and also used to set up models of a different hydrological dataset, such as river water level, water quality, rainfall, etc. The multi-layer perceptron (MLP) model, which is optimized by backpropagation algorithms, has been improved by short-term hydrological forecasts. MLP has now become one of the most commonly-used ANN algorithms. The applications of ANN in hydrology are discussed in the following paragraph [14], [35].

In [36], Rani *et al.* used ANN to predict the water level of the Sukhi reservoir. The results showed that the ANN model is a better predictor of real-time water forecasting. In [15], the ANN model was given form to simulate flows at certain parts of the river reach according to the flow at upstream locations. Different procedures were applied to predict flooding by the ANN. In [37], Biswas *et al.* attempted to predict water levels with a lead-time of one and two days in the Surma River at the Sylhet gauging station by artificial neural networks with feed-forward multilayer perceptron (MLP). The supervised learning and error backpropagation algorithm was based on records of rainfall and water levels. According to the summarized results, it was concluded that it is possible to forecast river water levels continuously in a real-time sense through the use of neural networks. A good correlation between observations and the corresponding network output indicated that the prediction was adequately close to the observation. In the paper of Bustami *et al.* [38], ANN models provided highly accurate predictions of the water level in the Bedup River in Kota Samarahan due to their reliability in estimating missing precipitation. The ANN model developed by Bustami *et al.* [38] successfully estimated the missing precipitation data of a recorder in the Bedup River, Sarawak, with 96.4% accuracy. In the paper of Arbain *et al.* [39], ANN had better accuracy in water level prediction than the SARIMA method due to its excellent ability to recognize time-series patterns and nonlinear features. In [40], Tiwari *et al.* explored the potential of wavelet and bootstrapping techniques for the development of an accurate and reliable ANN model applied to hourly flood forecasting.

D. CONVOLUTIONAL NEURAL NETWORK

Convolutional neural networks (CNN) are neural networks that use convolution, a mathematical operation, to replace general matrix multiplication in at least one of the layers [41]. It originated from the concept of the receptive field through the study of a cat's brain proposed by Hubel and Wiesel [42], who discovered the layered processing mechanism of information in the visual cortical pathway. Fukushima [43] introduced the concept of neocognitron, which is regarded as the first implementation of a convolutional neural network. LeCun *et al.* [32] introduced a model applying back-propagation into a convolution neural network, and the results showed better performance than other methods. He concluded that this network has many connections but relatively few free parameters.

The popularity of CNNs is resulted from their success to deal with classification problems such as image recognition and time series classification. CNNs are built by a series of convolutional layers, in which the output can have a connection only to local regions of the input. This can be done through moving a filter or weight matrix over the input and then calculating the point product between the two at each point. This structure enables the model to identify specific patterns in the input data. The abilities to learn and extract features from the raw input data allow CCNs to handle time

series prediction problems. An observation sequence can be considered as a one-dimensional image that the CNN model can read and then extract the most salient elements [44], [45]. It is a technique that allows a machine to automatically detect or extract features from the raw data that are directly relevant to the prediction problem. This can replace manual feature engineering [46].

CNNs have the advantage of the multi-layer perceptron (MLP) in time series prediction by supporting multivariate inputs and multivariate outputs. It can also learn arbitrary but complex functional relationships without requiring the model to learn directly from lagged observations. So the CNN model can learn a representation from a large sequence of inputs that is most relevant to the prediction problem [47].

The filters which act for certain repetitive patterns in the sequences, and which are used for future value prediction, generate the idea of applying CNNs to time series prediction. Because of their hierarchical structure, CNNs can work better in noise sequences. The noise can be discarded from each subsequent layer to keep the meaningful patterns [45]. One of the main reasons to use the convolutional layer instead of the fully connected layer is that we would end up with a massive number of parameters requiring a large number of resources if using the fully connected layer. The same issue also occurs in LSTM, where a large number of parameters results in large computational complexity. Unlike LSTM, 1D CNN can be used in low-resource environments. Also, the lower parameters in the CNN will result in the avoidance of overfitting.

Assem *et al.* [17] proposed a better method for the long-term prediction of water flow and water level parameters in Ireland's Shannon River during 1983-2013. The framework was composed of three phases: urban scale analysis, data fusion, and domain knowledge data analytics phase which is the main focus of the paper that employs a machine learning model based on Deep Convolutional Neural Networks (Deep CNNs). The result showed that Deep CNNs could perform better than other well-known time series prediction models (such as ANNs, SVMs, WANNs and WSVMs).

E. LONG SHORT-TERM MEMORY

Long Short-Term Memory (LSTM) is a type of recurrent neural network (RNN) that is capable of learning order dependence in prediction [48]. Generally speaking, it extends the memory of RNN [49]. It was firstly proposed by Hochreiter *et al.* [50] in 1997. LSTM solves the problem of vanishing gradients in RNN and bridges minimal time lags over 1000 discrete-time steps. In tasks with complex and artificial long-time lags, it outperforms RNN. LSTM has been successfully used in various fields, such as speech recognition [51], machine translation [52], [26], language modeling [53], tourism fields [54], [55], stock prediction [56], and rainfall-runoff simulation [16], [57], [58].

In the paper of Le *et al.* [59], LSTM was constructed to carefully predict flood flows for one to three days at Hoa Binh Station on the Da River based on the data-driven method. The LSTM model has learned long-term dependencies between sequential data series and demonstrated reliable performance in flood forecasting. Liang *et al.* [60] used LSTM to predict the daily water level of Dongting Lake. The model acquired daily water level data from 2011 to 2013 and used seven variables as typical input factors of water level changes. The results indicated the higher accuracy of the LSTM model compared to the SVM model. The results of Rehman *et al.* [61] proved that both the Recurrent Neural Network (RNN) and LSTM based on ANNs outperformed the machine learning and other conventional algorithms in the prediction of stream water levels. LSTM was firstly used by Bowes *et al.* [62] to create hourly forecasts in coastal cities, and the results indicated that it is appropriate for running operational forecasts in reality. In the paper of Zhang *et al.* [63], LSTM was proposed as a well-performing method for the prediction of water level saturation, as it is quicker and more stable.

F. SEQUENCE TO SEQUENCE (Seq2seq)

The Seq2seq model is Google's mainstream machine translation architecture. It was originally applied in the field of machine translation [27], which is suitable for sequence-to-sequence applications. Multi-step time series forecasting can be expressed as a sequence-to-sequence supervised prediction problem, a framework amenable to modern neural network models. The description of Seq2seq by Wadhwa [64] stated Seq2seq as a method of inputting a sequence of words (or sentence) and generating an output sequence of words. It does so by using the recurrent neural network (RNN) or more advanced versions of LSTM or GRU.

In the study done by Liu *et al.* [65], a better method based on the integration of a stacked auto-encoder (Seq2seq) and a back propagation neural network (BPNN) was proposed to compare with benchmark models (the support vector machine, SVM; the BP neural network model; the RBF neural network model; and the extreme learning machine model, ELM). The SAE-BP algorithm combines the strong feature representation of SAE and superior prediction of BPNN and has better performance than other models [65]. Lugt *et al.* [66] provided a new encoder/decoder architecture (ED-RNN) that could be used for the conditional prediction of RNNs and compared it with other time series prediction models. The results showed that the performance of ED-RNN is comparable to the feed-forward ANN, and that in the short-term fluctuations of water heights it could even perform better accurately captures short-term fluctuations.

G. MOTIVATION

Most of the past studies compared the accuracy of different input variables with the prediction accuracy, such as comparisons of the CNN model with the ANN model, the RNN model with the ANN model, and the ANN model with the traditional linear prediction model. However, as for data and model parameters, it is unknown which model has more efficient prediction. Therefore, we hoped to compare

the differences in the prediction performance among these models. Furthermore, we proposed a prediction model that combined CNN and GRU for comparison with other models. Finally, we used our prediction model, together with the judgment of anomaly analysis, to provide a complete water level data analysis and management process.

III. METHODS

A. PROPOSED WORKFLOW

This paper focused on a comparison of five models, including ANN, CNN, LSTM, Seq2seq, and Conv-GRU, in the prediction of water levels, so as to identify the model with the smallest prediction error. As shown in figure 1, the experimental workflow was divided into five main sections: 1) data pre-processing, 2) setting up the optimal parameter of each model, 3) model training and testing, 4) predictive results evaluation of each model with three indicators, and 5) anomaly detection.

In the first section, data pre-processing, including resampling and filling empty values, was conducted. Subsequently, wavelet transform was employed to remove noise from the original water level data. In the second section, in order to obtain the same benchmarking, the optimal parameters of each model were found according to three configurations: input and output sequence size, number of neurons, and number of hidden layers. Once the configurations of the optimal parameters for each model were confirmed, the following step was to use data gathered from 2012 to 2018 as the training data and that from 2019 as the test data. In the third section, the training data were brought into the models for training and learning. After completing all models, the test data were added in for backtesting to produce the prediction results.

In the fourth section, the predictive results of five models were evaluated by three error indicators: RMSE, MAE and MAPE, to see which model had the smallest predictive error. In the fifth section, the consequent prediction errors were modeled as a multivariate Gaussian distribution, which was used to evaluate the possibilities of anomalous behavior. For the test of anomalous detection, the previously proposed model was applied to predict the trend of rising water levels caused by Typhoon Soudelor, the worst natural disaster in Taiwan of 2015. Whether the proposed model Conv-GRU was capable of learning multiple patterns and could successfully detect all the anomalous patterns was then determined.

B. THE CALCULATION FORMULAS OF THE SELECTED MODELS

In this study, five models (Conv-GRU, ANN, CNN, LSTM and Seq2Seq) were selected for the comparison of water level prediction. The below-listed calculation formulas of these five models, which were used by other researchers, were applied in this study to compute the predictive results of the water levels of the Danshui River in the 3rd section of our workflow – the model training and testing.

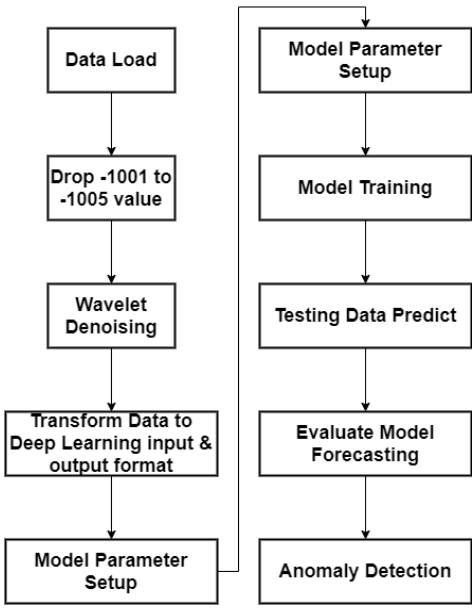

FIGURE 1. The experimental workflow of the comparison of five models combined with anomaly detection.

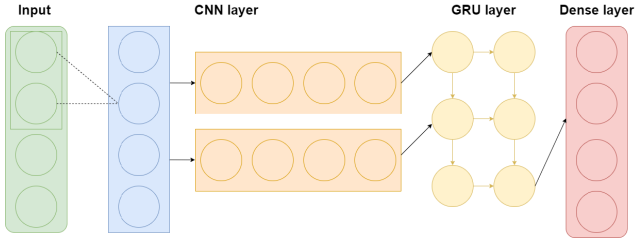

FIGURE 2. The general architecture of the Conv-GRU model.

1) THE PROPOSED CONV-GRU MODEL

Figure 2 illustrates the general architecture of the Conv-GRU model, which combines CNN with GRU. The proposed model shown in Figure 2 indicates that extraction of the input variables for the dataset at the water level station was performed by the CNN layer in the first model and was then delivered to the GRU layer in the second model for information analyzing and time series predicting. Through the last model, which was a fully connected layer, the water level could be predicted by the proposed method led by CNN-GRU.

CNN is easier to train than MLP [67]. For the calculation formulas of CNN in the proposed Conv-GRU model, studies done by Swapna [68] and Le *et al.* [69] were referenced. In their studies, they stated that several neurons in a CNN layer consist of weights and bias values. The training process enables the values to be learned, and several input variables in these models are delivered to each neuron. Afterward, a dot product operator is conducted and an optional nonlinear function follows after. Models in both studies used a convolutional 1D layer, a pooling 1D layer and a fully connected layer. In detail, CNN acquires one-dimensional time series data in which the data are arranged according

to the time series. The input vector of the one-dimension as $x = \{x_1, x_2, \ldots, x_n\}$ wherein $x_n \in R^d$ are the variables in the dataset. A feature map f_m of the 1D convolution is built up by using the convolution operator of the input data with a filter, $w \in R^{fd}$, of which f presents the features constituted in the input data generated at its output, which is a new set of features that is transferred to the input of the next block in line. The following equation shows a new feature map f_m given from a set of features f [68], [69]:

$$hl_i^{fm} = tanh(w^{fm} x_{i:i+f-1} + b) \tag{1}$$

Each set of features f in the input data by $\{x_{1:f}, x_{2:f+1}, \ldots, x_{n-f+1}\}$. The filter hl in Equation (1) is maximized to deliver a feature map presented by $hl = [hl_1, hl_2, \ldots, hl_{n-f+1}]$. $b \in R$ is a bias term, and $hl \in R_{n-f+1}$.

Le *et al.* [69] considered that the output of the convolutional layers is the total of the weighted inputs constructed by multilinear transformations. However, the linear transformation is unable to catch the complex structures of the data; hence, a non-linear activated layer must be used after the convolutional layers for superior data learning in the training step. In the study of Le *et al.* [69], the ReLU activation function was chosen, which applies $max(0, x)$ to each of the inputs. In the next step, the output of the convolutional layer as a down-sampling operator is transferred to the pooling layer. The max-pooling layer is applied to each feature map $\vec{hl} = max\{hl\}$. This is a selection process to find the most meaningful features with the highest values. The following is the formula for the output of the max-pooling layer [69].

$$x_i' = CNN(x_i) \tag{2}$$

The input vector of the CNN network with the energy consumption as x_i, as presented above. In this study, the output of the CNN network x_i' is connected to the input of the GRU network instead of the Bi-LSTM network previously used by Le *et al.* In Equation (2), a new vector x_I' was obtained by the proposed model, which firstly delivered the input vector to CNN.

Formulated Equation (3) below, which is the gate that controls the information flow from the pass activation. The new information was added to Equation (5), while the reset gate in Equation (4) was put into the candidate activation. In Equation (5), the notation $\odot$ is an operator of the element-wise product. Comprehensively, they concluded that GRU and LSTM are similar in performance and that there is neither lowliness nor nobleness between them for a given problem. The same conclusion was made by Chung [28]. For comparison, see [26]. The formulation is denoted by:

$$z_t = sigmoid(W_z x_t + U_z h_{t-1} + b_z) \tag{3}$$

$$r_t = sigmoid(W_r x_t + U_r h_{t-1} + b_r) \tag{4}$$

$$h_t = z_t \odot h_{t-1} + (1 - z_t) \odot tanh(W_h x_t + U_h(r_t \odot h_{t-1}) + b_h) \tag{5}$$

2) ARTIFICIAL NEURAL NETWORKS (ANN)

For the calculation formulas of ANN, Borovykh *et al.* [45] illustrated the basic structure of a feed forward neural network, which is composed of L layers with M_l hidden nodes in each layer $l = 1, \ldots, L$. They considered that if input $x(1), \ldots, x(t)$ are given in assumption, the forecasted values in the following step $\hat{x}(t+1)$ will be output by using the multi-layer neural network. In the first layer, M_1 linear combinations of the input variables in the form w^1 are constructed as below:

$$a^1(i) = \sum_{j=1}^{t} w^1(i,j)x(j) + b^1(i), \quad for\ i = 1, \ldots., M_1$$

$w^1 \in R^{M1xt}$ in the equation above is defined as the weights and $b^1 \in R^{M1}$ is the biases. Each output $a^1(i)$, $i = 1, \ldots, M_1$ is then changed to use a differentiable nonlinear activation function $h(\cdot)$ to formulate the following equation:

$$f^1(i) = h(a^1(i)), \quad \text{for } i = 1, \ldots, M_1$$

The ANN model can learn nonlinear relations between data points through nonlinear function. Moreover, in subsequent layer each $l = 2, \ldots, L-1$, the outputs of the previous layer f^{l-1} are linearly combined once again and passed through the nonlinearity, which is shown below:

$$f^l(i) = h(\sum_{j=0}^{M_{l-1}} w^l(i,j) f^{l-1}(j) + b^l(j)), \text{ for } i = 1, ., M_1.$$

They next provided the following equation, in which $b^1 \in R_l^M$ and $w^1 \in R^{M1xMl-1}$. $\hat{x}(t+1)$, the forecasted value, computed by using the formulation below in the final layer $l = L$ from the neural network:

$$\hat{x}(t+1) = h(\sum_{j=0}^{M_{L-1}} w^L(j) f^{L-1}(j) + b^L)$$

with $b^1 \in R$ and $w^1 \in R^{1 \times Ml-1}$.

3) CONVOLUTION NEURAL NETWORK (CNN)

The idea of developing convolutional neural networks was based on local connectivity at the beginning in which each node is connected only to a local region of the input [45].

The input in a convolutional layer is usually regarded as having three-dimensions: the number of channels, the height, and the weight. In the study of Borovykh *et al.* [45], the input in the first layer was convolved with a set of M_1 which is a three-dimensional filter used over all channels of the input for making the feature map of the output. Following is the equation, in which the one-dimensional input is considered by $x = (x_t)_{t=0}^{N-1}$ of size N with no zero padding. Each filter w_h^l for $h = 1, \ldots, M_1$ with the input below is convolved, and the output is the feature map from the first layer:

$$a^1(i,h) = (w_h^l * x)(i) = \sum_{j=-\infty}^{\infty} w_h^l(j)\, x(i-j)$$

where $w \in R^{1xkxl}$ and $a^1 \in R^{1xN_{l-1}xM_{l-1}}$. Borovykh *et al.* noted that the weight matrix has only one channel because the number of input channels is also one in this case. Similar to the feed forward neural network, the output is given as $f^1 = h(a_1)$ by the nonlinearity $h(\cdot)$.

In each subsequent layer $l = 2, \ldots, L$ of their model, where $1 \times N_{l-1} \times M_{l-1}$ is the size of the output filter map from the previous convolution with $N_{l-1} = N_{l-2} - k + 1$ is the feature map of the input, $f^{l-1} \in R^{1 \times N_{l-1} x M_{l-1}}$. It is then convolved with a set of M_l filters $w_h^l \in R^{1xkxM_{i-1}}$, $h = 1, \ldots, M_l$ to create feature map $a^l \in R^{1 \times N_t x M_t}$:

$$a^l(i,h) = \left(w_h^l * f^{l-1}\right)(i) = \sum_{j=-\infty}^{\infty} \sum_{m=1}^{M_{l-1}} w_h^l(j,m) f^{l-1}(i-j,m)$$

Through the non-linearity, the output of this is then transferred to give $f^l = h(a^l)$. Hence, the receptive field of each output node is controlled by parameter k of the filter size. Without zero padding, the output of convolution from each layer has width $N_l = N_{l-1} - k + 1$ for $l = 1, \ldots, L$. The features can be found in a time-invariant manner because of the same weights shared from all elements of the feature map. Meanwhile, the number of trainable parameters is reduced. After L convolutional layers, the output of the network will be matrix f^L and its size can be determined by the number of filters and the size used in the last layer.

4) LONG-SHORT TERM MEMORY (LSTM)

There are four units in the memory cell of LSTM: an input gate, an output gate, a forget gate and a self-recurrent neuron. Figure 3 illustrates how the value in each gate is updated as well as the process of the full network into which the LSTM model is unrolled. According to Bao *et al.* [70], the definitions of the mathematical symbols in Figure 3 are shown as below:

1. x_t is the input vector to the memory cell at time t.
2. W_i, W_f, W_c, W_o, U_i, U_f, U_c, U_o and V_o are weight matrices.
3. b_i, b_f, b_c and b_o are bias vectors.
4. h_t is the value of the memory cell at time t.
5. i_t is the value of the input gate and $\tilde{C}_t$ is the candidate state of the memory cell at time t, which can be given as:

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$\tilde{C}_t = tanh(W_c x_t + U_c h_{t-1} + b_c)$$

6. f_t is the value of the forget gate and C_t is the state of the memory cell at time t, which can be calculated by:

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$C_t = i_t * \tilde{C}_t + f_t * C_{t-1}$$

7. o_t is the value of the output gate and h_t is the value of the memory cell at time t, which can be formulated as:

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + V_o C_t + b_o)$$
$$h_t = o_t * tanh(C_t)$$

Bao *et al.* defined the input vector as x_t and the output as h_t representing the result of the memory cell at time t. The value of the memory cell at time t is also h_t. At time t, the values of the input gate and forget gate, and that the output gate, are f_t and o_t respectively. $\tilde{C}_t$ is value of the candidate state of the memory cell at time t.

5) SEQUENCE TO SEQUENCE (Seq2seq)

Although RNN is poor to learn long-term dependencies [71], it provides a framework of the feed forward network in dealing with sequential data [72], [73]. In the experiment of this study, the equation of the sequence to sequence model in the study of Sutskever *et al.* was applied [74]. $p(y_1, \ldots, y_N | x_1, \ldots, x_N)$ in the equation below presents the conditional probability, where the input sequence is $(x_1, \ldots, x_N)$ and the output sequence with the same length is $(y_1, \ldots, y_N)$. LSTM was applied to evaluate this conditional probability through acquirement of the fixed-dimensional representation v in the input sequence $(x_1, \ldots, x_N)$ provided by the last hidden state of LSTM. LSTM was then used to calculate the probability of $(y_1, \ldots, y_N)$. Below is a standard LSTM-LM equation [74], in which its initial hidden state is set to the representation v of $(x_1, \ldots, x_N)$:

$$p(y_1, \ldots., y_N | x_1, \ldots., x_N) = \prod_{t=1}^{N} p(y_t | v, y_1, \ldots., y_{t-1})$$

Given input $X = \{x^1, x^2, \ldots, ^{\mathrm{n}}\}$, $\mathrm{c_t}$ is the intermediate state of the encoder at step t, where $\mathrm{c_t} \in R^{\mathrm{m}}$ and m is the number of neurons in the encoder. The decoder decodes $\mathrm{c_t}$ into the target sequence $Y = \{y^1, y^2, \ldots, y^{\mathrm{n}}\}$.

C. EVALUATION CRITERIA

After applying these calculation formulas for model training and testing, the results of the water level prediction from our selected five models were obtained. Next, three measures were adopted to evaluate the performance of each model: the Mean Absolute Error (MAE), Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (RMSE). This was the fourth section of the proposed workflow - the evaluation of the predicted results for each model with three indicators. The following are the introduction of these three error indicators.

1) ROOT-MEAN-SQUARE ERROR

The root-mean-square error (RMSE) is a measure commonly used for evaluating the differences between the values of a sample or population that have been predicted by a model or an estimator and the observed values [75].

$$\mathrm{RMSE} = \sqrt{\frac{1}{n} \Sigma_1^n (y - \hat{y})^2}$$

2) MEAN ABSOLUTE PERCENTAGE ERROR

The mean absolute percentage error (MAPE) is also known as the mean absolute percentage deviation (MAPD). It is a measure that applies statistics to evaluate the accuracy of a forecasting method. A loss function is also used by MAPE for solving regression problems in machine learning [76]. It uses the percentage to show the accuracy level and is formulated

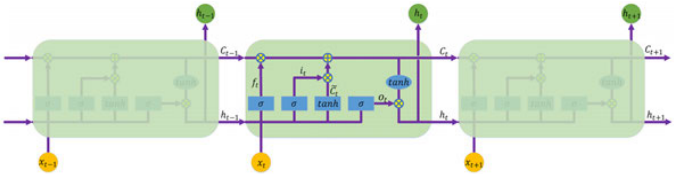

FIGURE 3. The repeating module in LSTM [70].

as:

$$\text{MAPE} = \frac{100\%}{n}\Sigma_1^n\left|\frac{y-\hat{y}}{y}\right|$$

3) MEAN ABSOLUTE ERROR

The mean absolute error (MAE) is a measure to evaluate the difference between continuous variables. Examples of Y versus X include comparisons of predicted versus observed, subsequent time versus initial time, and one technique of measurement versus an alternative technique of measurement. It is thus an arithmetic average of the absolute errors $|e_i| = |y_i - x_i|$, where y_i is the prediction and x_i the true value. From each point to the identity line, MAE can also be deemed as the average vertical and horizontal distance if a scatter plot of n points is considered, where point i has the coordinates (x_i, y_i) . . . The mean absolute error is formulated as below [77], [78]:

$$\text{MAE} = \frac{\Sigma_{i=1}^n|y_i - x_i|}{n} = \frac{\Sigma_{i=1}^n|e_i|}{n}$$

D. DEEP LEARNING-BASED ANOMALY DETECTION

In this section, the LSTM-based Anomaly Detection of Malhotra *et al.* [79] was applied in the following experiment. In their approach, $X = \{x^{(1)}, x^{(2)}, \ldots, x^{(n)}\}$ is considered as a time series, in which an m-dimensional vector $\{x_1^{(t)}, x_2^{(t)}, x_m^{(t)}\}$ is each point $x^{(t)} \in R^m$, and their elements correspond to the input variables. They believed that through learning, the predictive model can predict the next l values for d of the input variables *s.t.* $1 \leq d \leq m$. Malhotra *et al.* [79] proposed four parts in the normal sequence(s), which were normal validation-1 (v_{N1}), normal validation-2 (v_{N2}), normal train (s_N) and normal test (t_N). The anomalous sequence(s) has two sets: anomalous validation (vA) and anomalous test (tA). At the beginning, they developed a predictive model using deep learning-based neural networks and then calculated the distribution of the prediction errors using anomaly detection.

E. ANOMALY DETECTION BY APPLYING THE DISTRIBUTION OF THE PREDICTION ERROR

In the study of Malhotra *et al.* [79], every selected d dimension of $x^{(t)} \in X$ for $l < t \leq n - l$ was predicted l times with a predictive length of l. An error vector $e^{(t)}$ for point $x^{(t)}$ as $e^{(t)} = e_{11}^{(t)}, \ldots, e_{1l}^{(t)}, \ldots, e_{d1}^{(t)}, \ldots, e_{dl}^{(t)}$ was calculated, in which $e_{ij}^{(t)}$ was different between $x_i^{(t)}$ and its value predicted at time $t - j$.

They mentioned that in the test sequences and validation, the error vectors for each point could be computed by using the predictive model trained on s_N. The error vectors are formed into a model to match a multivariate Gaussian distribution $N = N(\mu, \Sigma)$. Moreover, the value of N at $\mathrm{e}^{(t)}$ provides the possibility $p^{(t)}$ of detecting an error vector $\mathrm{e}^{(t)}$ (similar to the normalized innovations squared (NIS) using the Kalman filter-based dynamic prediction model for novel detection [80]). Malhotra *et al.* [79] considered that through the application of the error vectors for the points from v_{N1}, the parameters μ and Σ, which use the Maximum Likelihood Estimation, can be estimated. If $p^{(t)} < \tau$, an observation $\mathrm{x}^{(t)}$ is detected as being anomalous, and the rest of the observations are detected as being normal. By maximizing F_β-score, sets v_{N2} and v_A can be adopted to learn τ (where the anomalous points are classed as positive and the normal points are classed as negative) [79].

IV. EXPERIMENT RESULT & ANALYSIS

A. DATASET PRE-PROCESSING

Firstly, the water level data were downloaded from the 10th River Management Office website (https://www.wra10.gov.tw/13264/13282/13352/13356/), which is in charge of three major streams: the Danshui River, the Keelung River, the Xindian River and the Daha River. There are a total of 118 water systems in Taiwan. The reason that we decided to collect information of the Danshui River is that it covers the entire Taipei area. Taipei is the political and economic center of Taiwan and has more than 30% of its total population. Also, water level prediction for the Danshui River was the first to be conducted in Taiwan, therefore it has complete and abundant data.

From the website, 27 out of 31 water level stations were selected because they were relatively closer to the Great Taipei area (Table 1). Data from these 27 water level stations for 2012 to 2018 were used as training data and that from 2019 was used as testing data. Data were recorded every 10 minutes from each water level station, and the fields for the alerting of abnormal water levels are shown in Table 2.

However, the data from the 10th River Management Office were incomplete, as some were not recorded every 10 minutes, and were listed using the codes −1001 and −1005. In order to make the data for each year remain the same, these codes were deleted and replaced with the prior water level value before the vacant data.

TABLE 1. The data from 27 water level stations in the Danshui River Basin.

No.	Watershed	Water level station	Water time	Current water level	Alert value			Alert
					First level	Second level	Third level	
1	Tamsui River	Lan-Sheng	01-19 09:30	109.80	114.5	112.5	--	NO
2	Tamsui River	Kamigasan Bridge	01-19 09:30	57.36	62.4	59.5	--	NO
3	Tamsui River	Sanyingqiao Bridge	01-19 09:30	28.98	44.6	43.6	--	NO
4	Tamsui River	Dingnei	01-19 09:30	20.61	28.1	25.1	--	NO
5	Tamsui River	Ganyuan Bridge	01-19 09:30	12.33	20.3	17.9	--	NO
6	Tamsui River	Bitan Bridge	01-19 09:30	12.23	21.1	19	--	NO
7	Tamsui River	Wanfu Bridge	01-19 09:30	12.02	22.9	20.3	--	NO
8	Tamsui River	Dahua Bridge	01-19 09:30	9.15	20.7	17.3	--	NO
9	Tamsui River	Wudu	01-19 09:30	4.72	17.4	14.4	--	NO
10	Tamsui River	Ankeng Bridge	01-19 09:30	2.56	15.1	12.4	10.7	NO
11	Tamsui River	Chang'an Bridg	01-19 09:30	2.51	13.5	10.5	--	NO
12	Tamsui River	Xiulang Bridg	01-19 09:30	1.97	11.3	9.1	5.9	NO
13	Tamsui River	Jiangbei Bridge	01-19 09:30	1.65	12.7	9.7	--	NO
14	Tamsui River	Shehou Bridge	01-19 09:30	0.91	11.5	8.5	--	NO
15	Tamsui River	Nanhu Bridge	01-19 09:30	0.6	11.6	9.8	6.4	NO
16	Tamsui River	Chenglin Bridge	01-19 09:30	0.48	15.7	14	--	NO
17	Tamsui River	Xinhai Bridge	01-19 09:30	0.34	10.2	7.7	2.8	NO
18	Tamsui River	Dazhi Bridge	01-19 09:30	0.16	9.8	8	3.3	NO
19	Tamsui River	Zhongshan Bridge	01-19 09:30	0.1	9.2	7.1	--	NO
20	Tamsui River	Shihzihtou	01-19 09:30	-0.04	4.5	3	--	NO
21	Tamsui River	Tudigongbi	01-19 09:30	-0.17	5	3.1	2.1	NO
22	Tamsui River	Taipei Bridge	01-19 09:30	-0.4	8.5	6.7	2.2	NO
23	Tamsui River	Sanxia(2)	01-19 09:30	-987.0	32.5	31.3	--	NO
24	Tamsui River	Baoqiao	01-19 09:30	-987.0	18	15	11.6	NO
25	Tamsui River	Zhongzheng Bridge	01-19 09:30	--	10.5	8.3	5.4	NO
26	Tamsui River	Quchi	01-19 09:30	--	50.5	--	--	NO
27	Tamsui River	Wengzaitan Bridge	01-19 09:30	--	--	90	--	NO

Next, a wavelet transform was employed to remove unnecessary noise. This method is called wavelet coefficients thresholding. After the signal is wavelet transformed, the noise becomes a smaller signal (low scale), so the scale turns smaller. The noise is removed by removing the signal. The general method is to set a threshold, discard the signals below this threshold, and keep the signals above the threshold. We applied DWT by using Symlet 5 to the data, and the decomposition order was set at level 6.

In reference to the experimental environment of Cho *et al.* [81], we built a river water level forecasting and anomaly detection system using Keras, a framework of deep

learning, and all experiments were carried out on a 64-bit Ubuntu 18.04 system using an INTEL i7-4790 CPU with a GEFORCE RTX 2080 GPU and 16 GB RAM. We used Python version 3.5.

B. CONFIGURATIONS OF OPTIMAL PARAMETERS

After data pre-processing, we began searching for the configurations of the optimal parameters for each model in order to have a comparison for the same benchmark in the following sections. These configurations included the configurations of the input sequence and out sequence size, configuration of the number of neurons, and that of the hidden layer. Once the optimal parameter configurations were confirmed for each model, the model training and testing could be carried out.

1) CONFIGURATION OF THE INPUT SEQUENCE AND OUTPUT SEQUENCE SIZE

First of all, we had to determine the length of the input sequence and output sequence. The input sequence referred to the length of each sample's time sequence, while the output sequence was the length of the time sequence to be predicted.

As shown in Table 2 below, we used LSTM to find the best parameter configuration for the best input sequence and output sequence, in which number 120 represents the length of 120∗10 minutes and 6 represents the output sequence of 6∗10 minutes. We tested various input information sequence size settings (120, 180, 240, and 300) to optimize our deep learning models. Theoretically, LSTM can be trained to bridge time lags in excess of 1000 discrete time steps [50]. However, in our case, the water level data was non-stationary and nonlinear [82]. In long-range time steps, increased uncertainty leads to larger errors. Moreover, various predictive output step size settings were tested, including 6 units, 12 units, and 18 units (1 unit equal to 10-minutes), to optimize the deep learning models. In general, the predictions were prone to offsets; however, extending the prediction time led to better predictions of future trends and effective decisions. In Table 2, the best parameters for each model are shown in bold. To conclude, the LSTM model performed with the lowest error under $180 \times 32 * 3 \times 6$ parameters, in which the input sequence length is 180, $32 * 3$ refers to three hidden layers with 32 neurons in each, and the number 6 is the output sequence length. Hence, we chose 180 for the input sequence length and 6 for the output sequence length.

TABLE 2. Comparison of the input sequence & output sequence parameters.

LSTM	RMSE	MAPE	MAE
120 x 32*3 x 6	1.456	35.968	1.022
120 x 32*3 x 12	1.487	35.724	0.789
120 x 32*3 x 18	1.444	35.711	0.791
180 x 32*3 x 6	**1.032***	**31.035***	**0.620***
180 x 32*3 x 12	1.446	36.679	0.783
180 x 32*3 x 18	1.275	31.946	1.000
240 x 32*3 x 6	1.720	38.009	1.171
240 x 32*3 x 12	1.652	42.257	0.691
240 x 32*3 x 18	1.796	38.911	1.203
300 x 32*3 x 6	1.235	31.549	0.542
300 x 32*3 x 12	1.441	38.840	0.580
300 x 32*3 x 18	1.575	38.625	0.595

TABLE 3. Comparison of different neuron parameters.

	RMSE	MAPE	MAE
LSTM-180 x 8*3 x 6	1.460	44.056	0.636
LSTM-180 x 16*3 x 6	1.382	36.244	1.016
LSTM-180 x 24*3 x 6	1.415	33.692	0.679
LSTM-180 x 32*3 x 6	**1.032***	**31.035***	**0.620***
CNN-180 x 8*3 x 6	1.408	57.154	1.145
CNN-180 x 16*3 x 6	1.614	84.630	1.255
CNN-180 x 24*3 x 6	**1.334***	**44.346***	**0.902***
CNN-180 x 32*3 x 6	1.542	77.922	1.510
Seq2seq-180 x 8*3 x 6	1.822	93.297	1.358
Seq2seq-180 x 16*3 x 6	2.059	180.123	2.892
Seq2seq -180 x 24*3 x 6	1.863	143.226	1.469
Seq2seq -180 x 32*3 x 6	**1.712***	**75.476***	**1.281***
ANN-180 x 8*3 x 6	**1.948***	**91.298***	**1.798***
ANN -180 x 16*3 x 6	2.059	94.567	2.892
ANN -180 x 24*3 x 6	2.180	95.539	3.140
ANN -180 x 32*3 x 6	2.092	94.020	2.608
Conv-GRU-180 x 8*3 x 6	0.892	35.001	0.717
Conv-GRU-180 x 16*3 x 6	**0.788***	**34.035***	**0.575***
Conv-GRU -180 x 24*3 x 6	0.914	36.756	0.670
Conv-GRU -180 x 32*3 x 6	0.836	35.211	0.622

2) CONFIGURATION OF NUMBER OF NEURONS

We initially set the number of hidden layers in each model as 3. The three sets of numbers in the first column on the left of Table 3 respectively represent the best input sequence parameter, the number of hidden layers ∗ the number of neurons in each layer, and the best output sequence parameter. Various numbers of neuron settings, including 8, 16, 24 and 32, were tested to optimize our deep learning models. The optimal number of neurons in hidden layers for different tasks is still an unsettled question in neural network research. An insufficient number of hidden units results in high errors due to under-fitting, while having too many hidden units also leads to high errors due to over-fitting [83]. However, from Table 3, it could be seen that LSTM had the best effect with 32 neurons in each hidden layer and that the best effect on CNN was 24 neurons in each hidden layer. The best result for Seq2seq was 32 neurons in each hidden layer, and that for ANN was eight neurons in each hidden layer. Lastly, in our proposed model, Conv-GRU had the best effect when configuring 16 neurons in each hidden layer.

3) CONFIGURATION OF NUMBER OF HIDDEN LAYERS

Currently, there is no rule of thumb for selecting the best number of hidden layers [71]. Although we knew that increasing the depth of the network would allow the network to learn more complex features, having too many layers could

TABLE 4. Comparison of different hidden layer parameters.

	RMSE	MAPE	MAE
LSTM-180 x 32*2 x 6	1.355	37.870	0.665
LSTM-180 x 32*3 x 6	**1.032***	**31.035***	**0.620***
LSTM-180 x 32*4 x 6	1.185	33.172	0.653
LSTM-180 x 32*5 x 6	1.416	39.838	0.759
CNN-180 x 24*2 x 6	1.265	40.485	0.854
CNN-180 x 24*3 x 6	1.334	44.346	0.902
CNN-180 x 24*4 x 6	**1.144***	**37.154***	**0.745***
CNN-180 x 24*5 x 6	1.242	37.922	0.810
Seq2seq-180 x 32*2 x 6	1.811	77.060	1.349
Seq2seq -180 x 32*3 x 6	1.712	75.476	1.281
Seq2seq -180 x 32*4 x 6	1.628	61.235	1.249
Seq2seq -180 x 32*5 x 6	**1.431***	**49.903***	**0.933***
ANN-180 x 8*2 x 6	1.766	76.297	1.310
ANN -180 x 8*3 x 6	1.948	91.298	1.798
ANN -180 x 8*4 x 6	1.733	73.570	1.280
ANN -180 x 8*5 x 6	**1.527***	**55.215***	**0.984***
Conv-GRU-180 x 16*2 x 6	0.963	41.140	0.742
Conv-GRU -180 x 16*3 x 6	0.788	34.035	0.575
Conv-GRU -180 x 16*4 x 6	**0.774***	**30.684***	**0.567***
Conv-GRU -180 x 16*5 x 6	0.885	31.307	0.627

cause the gradient to disappear. Even after various numbers of hidden layer settings were tested (2, 3, 4 and 5), while optimizing our deep learning models, no clear pattern emerged to suggest the best number of hidden layers. The increment of hidden layers could also lead to additional computation time and the danger of over-fitting, which would cause the poor performance of out-of-sample forecasts. In Table 4 below, it can be seen that the LSTM model performed the best when the number of hidden layers was 3, the CNN and Conv-GRU models performed the best when the number of hidden layers was 4, and Seq2seq and ANN performed the best with five hidden layers. The number of hidden layers in each neural network could be increased or reduced. Inputting different parameter values trained the model and generated the predicted values.

4) CONFIRMING THE CONFIGURATIONS OF THE OPTIMAL PARAMETER IN EACH MODEL

As shown in sections 4.2.1∼4.2.3 above, the optimal parameter configuration was found for each model. The best parameters were determined for each model by using the information from the past 180 ∗ 10 minutes (30 hours) as input and predicting the 6 ∗ 10 minutes (one hour) as output. In addition, in Figure 4 and 5 below, it can be seen that ANN performed the best when there were five hidden layers and eight hidden neurons in each layer. In the CNN model, the best performance occurred when there were four hidden layers, with 24 hidden neurons in each layer. The LSTM performed the best with three hidden layers and 32 neurons in each layer. In the Seq2seq model, there were five hidden layers with 32 hidden neurons in each layer. Finally, the CNN + GRU model performed best when there were four hidden

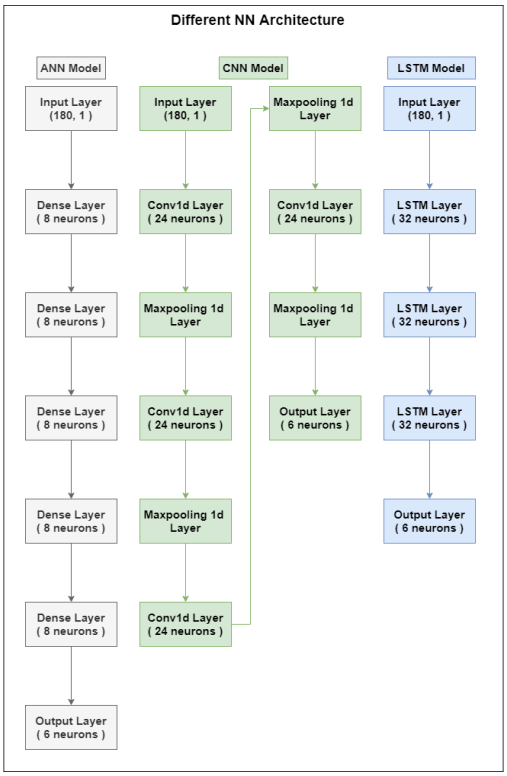

FIGURE 4. Complete architectures for ANN with five hidden layers and eight hidden neurons in each layer (gray), CNN with four hidden layers and 24 hidden neurons in each layer (green) and LSTM with three hidden layers and 32 hidden neurons in each layer (blue).

layers with 16 hidden neurons in each layer. In this study, the Mean Absolute Error (MAE) loss function and the efficient Adam version of the stochastic gradient descent were used. Figure 4-5 show the complete architectures of the final five models.

C. COMPARISON RESULTS

After the optimal parameter configurations were confirmed, each model was placed at the same benchmark for comparison. The training data from 2012∼2018 were then brought into these five models, and the data from 2019 were used for model testing. Figure 6 ∼ 11 show the results of the time series predictions of water level found by LSTM (in red), Seq2seq (in black), CNN (in pink), ANN (in orange) and Conv-GRU (in blue). The green line presents the actual water levels at six water level stations: Taipei Bridge (No. 22), Zhongshan Bridge (No. 19), Zhongzheng Bridge (No. 25), Quchi (No. 26), Nanhu Bridge (No. 15) and Chang'an Bridge (No. 11). In this paper, we demonstrate six results out of 27 stations were shown, and that the Conv-GRU model (the blue line) was the closest to the actual water level data (presented in green). Therefore, the outperformance of Conv-GRU over the other neural network (NN) methods in the prediction of water level characteristics could be acquired here. The method could also detect local features very well. As shown in Figure 6∼11, the proposed model was proficient

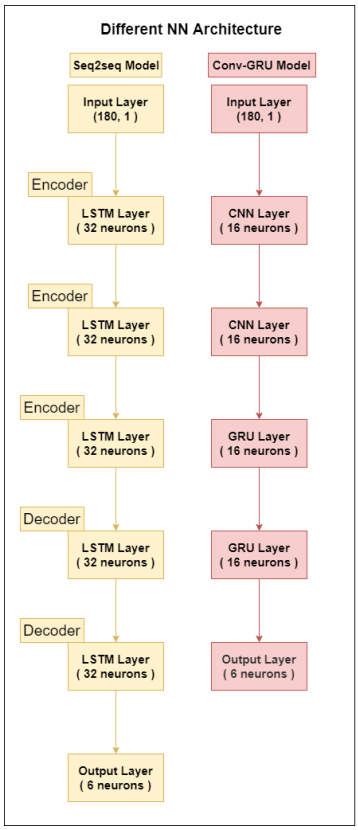

FIGURE 5. Complete architectures for Seq2seq with five hidden layers and 32 hidden neurons in each layer (yellow), and Conv-GRU with four hidden layers and 16 hidden neurons in each layer (red).

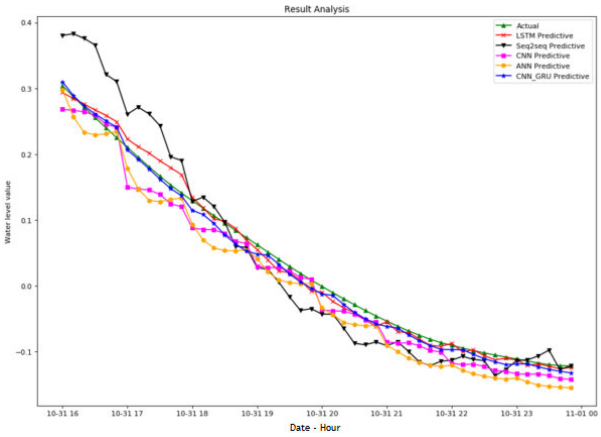

FIGURE 6. The water level predicted by LSTM/Seq2seq/CNN/ANN/Conv-GRU at Taipei Bridge station.

at simulating irregular water level trends compared to the other four models. The results show that the predicted value of the Conv-GRU model and the real value has the smallest error. Moreover, similar performance in the prediction of complex time series could be confirmed, particularly the performance of the peak prediction that occurs often in river water levels.

D. ERROR EVALUATION

The results of the time series predictions performed by these five models in Section 4.3 were evaluated by three indicators,

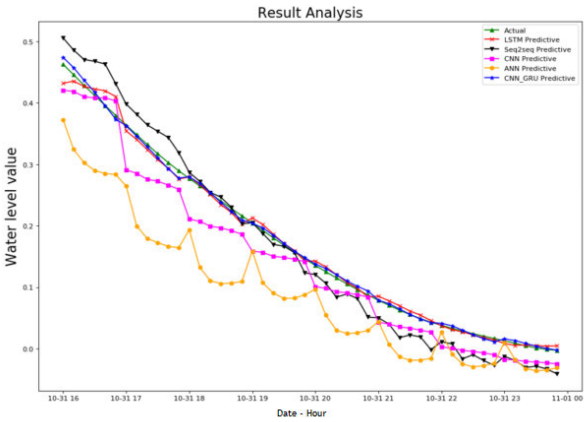

FIGURE 7. The water level predicted by LSTM/Seq2seq/CNN/ANN/Conv-GRU at Zhongshan Bridge station.

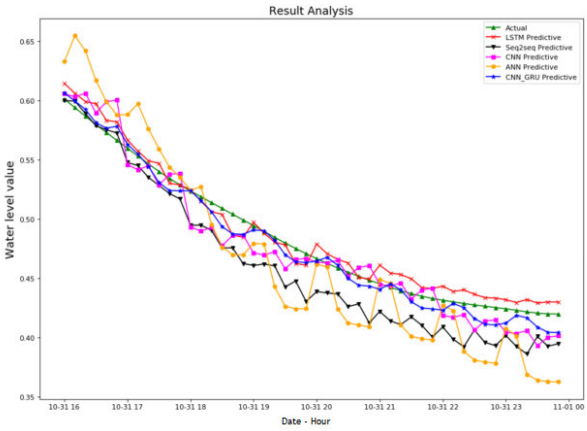

FIGURE 8. The water level predicted by LSTM/Seq2seq/CNN/ANN/Conv-GRU at Zhongzheng Bridge station.

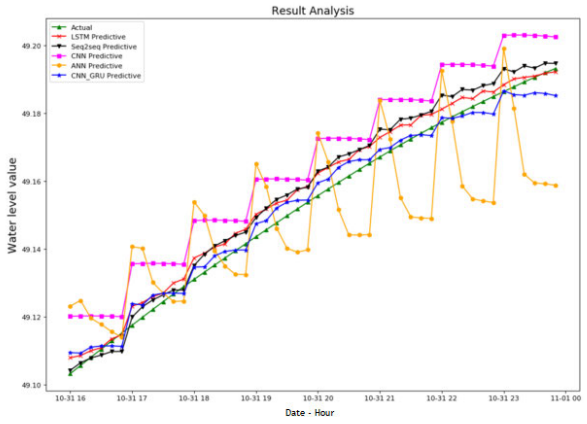

FIGURE 9. The water level predicted by LSTM/Seq2seq/CNN/ANN/Conv-GRU at Quchi station.

as described in this section. The three indicators were RMSE, MAPE and MAE, which were used to calculate the average error values of the five models between their predicted water level values and their actual water level values, as shown in Table 5. It was evident that Conv-GRU had the least error

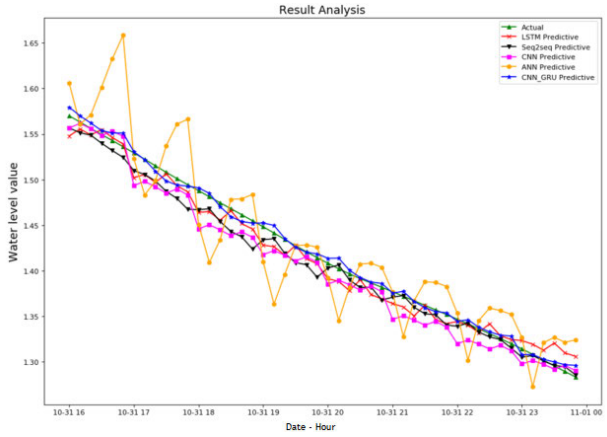

FIGURE 10. The water level predicted by LSTM/Seq2seq/CNN/ANN/Conv-GRU at Nanhu Bridge station.

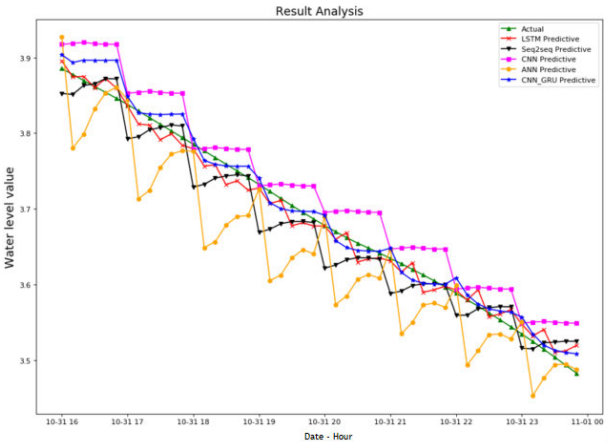

FIGURE 11. The water level predicted by LSTM/Seq2seq/CNN/ANN/Conv-GRU at Chang'an Bridge station.

across all intervals. Therefore, we could confirm that the proposed Conv-GRU model was superior to the other four models. In Table 5, the evaluation results show that the proposed Conv-GRU model performed better in terms of RMSE (0.774), MAPE (30.684), and MAE (0.567) in the river level prediction. The LSTM model (RMSE-1.032, MAPE-31.035, MAE-0.620) and the CNN model (RMSE-1.144, MAPE-37.154, MAE-0.745) both showed a slightly higher prediction error than the Conv-GRU model. In this experiment, we confirmed that the proposed Conv-GRU model has a considerable ability to learn long-term water level characteristics in terms of predicting river water level.

E. ANOMALY DETECTION

In this experiment, we used 27 water level stations for anomaly detection verification, combined with the previous Conv-GRU prediction method, and calculated the error value, and finally used Gaussian statistical distribution to capture the time point of the water level anomaly. The data sets had different kinds of patterns and natural anomalies. The river water level dataset contained noise and had the longest dependency, and it delivered different patterns at different time

TABLE 5. The results of the time series prediction for each model evaluated by three error indicators.

	RMSE	MAPE	MAE
ANN -180 x 8*5 x 6	1.527	55.215	0.984
CNN-180 x 24*4 x 6	1.144	37.154	0.745
LSTM-180 x 32*3 x 6	1.032	31.035	0.620
Seq2seq -180 x 32*5 x 6	1.431	49.903	0.933
Conv-GRU -180 x 16*4 x 6	**0.774***	**30.684***	**0.567***

scales. Although there were no labels, we could distinguish between normal and abnormal patterns by using the time of day as the context. For the anomaly detection, we focused on our proposed model (Conv-GRU) for prediction due to it outperforming the other four models, as described in Section 4.5. We firstly combined the best predictive model, Conv-GRU (proposed earlier), to learn the data from 2012 to 2018, and fitted an M-dimensional Gaussian distribution to the error vectors. We assumed the parameters of an M dimensional Gaussian distribution could be estimated as follows:

$$p(x|Data) = N(x|\hat{\hat{\mu}}, \hat{\Sigma})$$

We next used the water level data in 2019 to conduct an anomaly detection test. Finally, we plotted the Mahalanobis distance for each error vector and corresponding water level station data. The Mahalanobis distance is defined as:

$$a(x) = (x - \hat{\mu})^T \hat{\Sigma}^{-1} (x - \hat{\mu})$$

We could measure the rarity of the event with $a(x)$, which is represented by the blue line in the lower figures of Figure 12∼17.

The top figures of Figure 12∼17 are the comparison of the predicted value by Conv-GRU (green) and the actual values (red) from six water level stations: Taipei Bridge, Zhongshan Bridge, Zhongzheng Bridge, Quchi, Nanhu Bridge and Chang'an Bridge. The bottom figures are the Mahalanobis distance, which was a statistical representation of the anomaly score (blue).

We could see that when the peak of the blue curve in the top figure was higher, the level of abnormal events in the lower figure would be higher. It could be proved that, through careful maintenance of the training on the non-anomalous water level data, the Conv-GRU model was capable of learning multiple patterns and detected all the anomalous patterns. In addition, the score of the anomaly also presented the degree of probability of the anomaly. In the next section, we describe the processing of the data for Typhoon Soudelor in 2005 in our proposed Conv-GRU model to test its performance in anomalous detection and verify whether it could predict abnormal water levels in advance.

1) EFFICACY OF CONV-GRU IN THE ANOMALY DETECTION OF TYPHOON SOUDELOR

Typhoon Soudelor, which occurred in August 2015, is the worst natural disaster to occur in Taiwan. Figure 18 is the

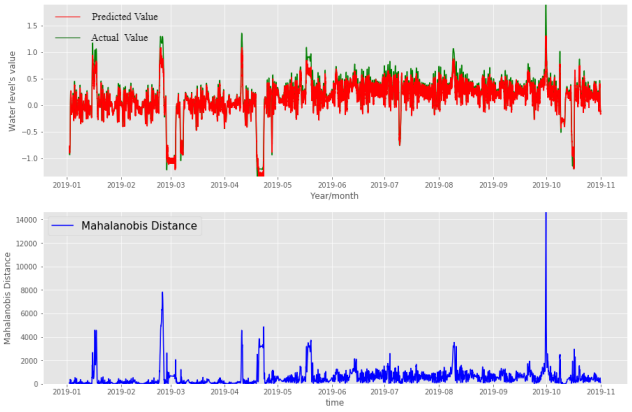

FIGURE 12. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Taipei Bridge; lower: the Mahalanobis distance.

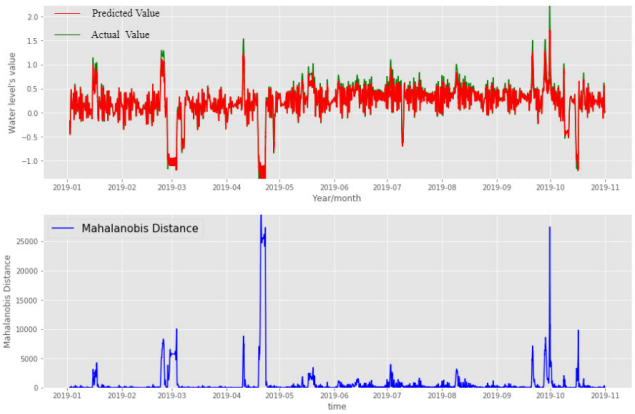

FIGURE 13. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Zhongshan Bridge; lower: the Mahalanobis distance.

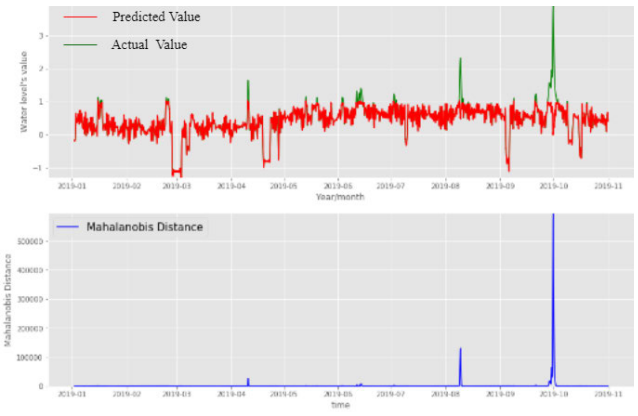

FIGURE 14. Upper: The prediction by Conv-GRU (red) and the actual water level value of (green) at Zhongzheng Bridge; lower: the Mahalanobis distance.

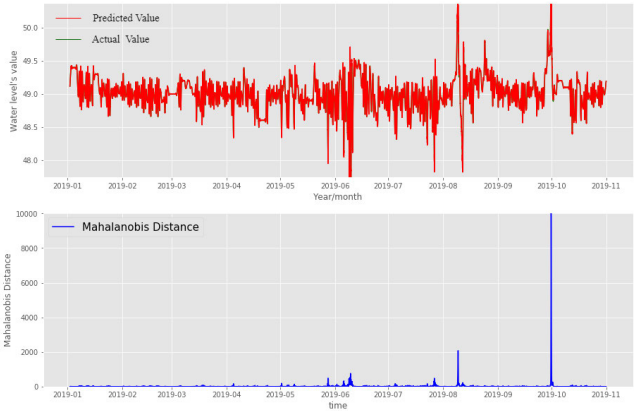

FIGURE 15. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Quchi Bridge; Lower: the Mahalanobis distance.

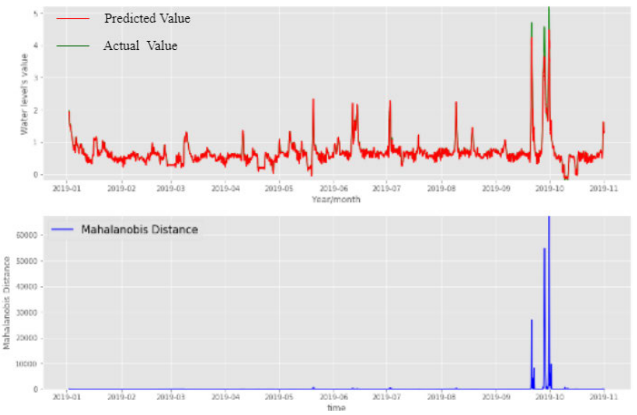

FIGURE 16. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Nanhu Bridge; lower: the Mahalanobis distance.

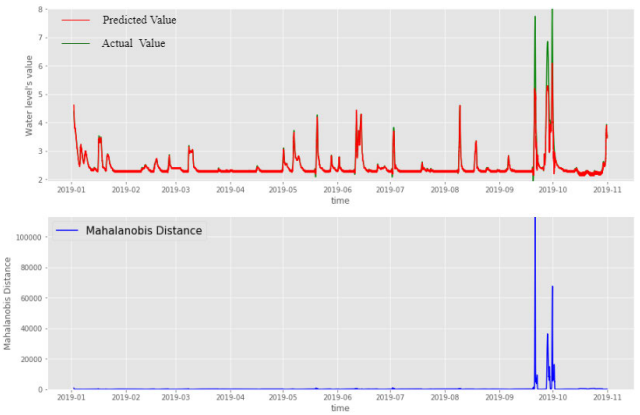

FIGURE 17. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Chang'an Bridge; lower: the Mahalanobis distance.

rainfall distribution map during Typhoon Soudelor. The preliminary results showed that in the northern region, especially Wulai District, the return periods of 3, 6, 12, and 24 hours delayed rainfall all have signals of more than 100 years of frequency. Therefore, we believed that the water level data during Typhoon Soudelor was suitable for testing the performance of our proposed model on anomalous detection in order to more accurately provide alerts in advance before future disasters occur.

In this experiment, we collected water level data from 27 water level stations from January 1st, 2012 to January 1st, 2015 for training data, and used the data from

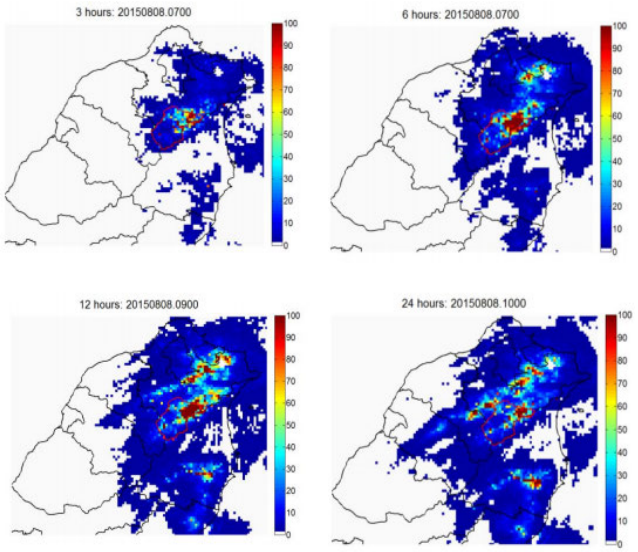

FIGURE 18. Frequency analysis results: upper left - 3 hour delay; upper right - 6 hour delay; lower left - 12 hour delay; bottom right - 24-hour delay.

January 2nd, 2015 to August 8th, 2015 as testing data. We then imported these data into our proposed model Conv-GRU for anomaly detection. Figure 19∼27 show the results of the water level prediction by Conv-GRU for nine water level stations during Typhoon Soudelor in 2015, including Sanxia, Xiulang Bridge, Kamigasan Bridge, Quchi, Zhongzheng Bridge, Wanfu Bridge, Zhongshan Bridge, Taipei Bridge and Chenglin Bridge.

In Figure 19∼27, the top figures are the comparisons of the predicted value by Conv-GRU (red) and the real values (green), and the lower figures are the Mahalanobis distance, which statistically represent an anomaly score (blue line). It could be seen that the blue line in the lower figures could effectively predict the trend of rising water levels caused by the typhoon at time corresponding to the top figures. The efficient predictions of our proposed model could help to create warnings of the existence of abnormalities before a flood strikes. Based on this result, we believed that Conv-GRU also had better performance in anomalous detection.

F. DISCUSSION

Deep learning methods, such as CNN and LSTM, are widely used in many fields. We used deep learning models for multiple time step forecasting. In past studies, different comparisons indicated that each NN has the ability to capture the time-dependent characteristics of river water levels, but the optimal parameter configuration for each model was still unspecified. Therefore, the first step of our experiment was to find out the optimal parameter configuration of each model through comparisons among different parameters, which was followed by a comparison of the predictive performance of each model.

In this study, the results provided by CNN and LSTM had a smaller prediction error than ANN. In terms of the LSTM

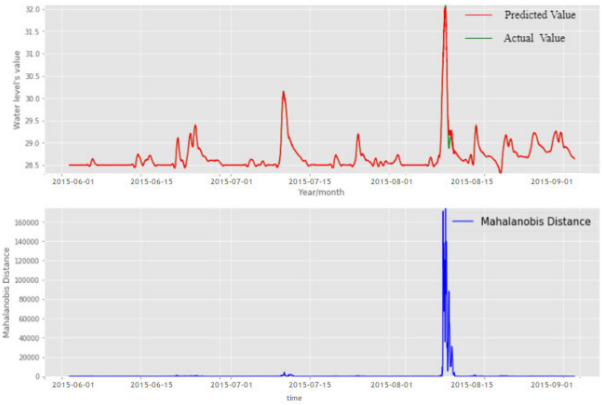

FIGURE 19. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Sanxia in 2015; lower: the Mahalanobis distance.

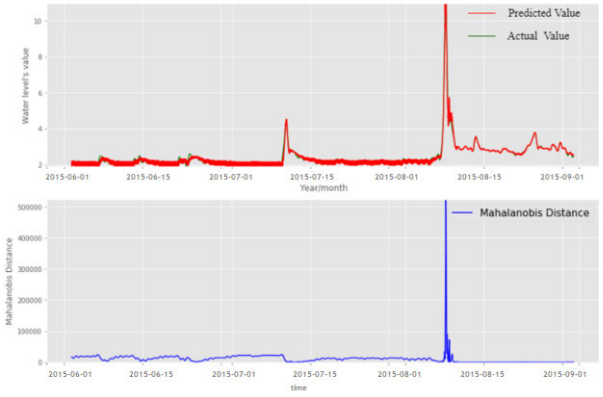

FIGURE 20. Upper: The prediction by Conv-GRU (red) and the actual value of water level at Xiulang Bridge (green) in 2015; lower: the Mahalanobis distance.

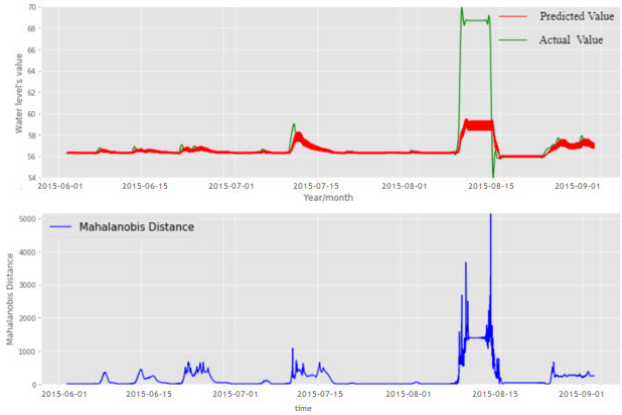

FIGURE 21. Upper: The prediction by Conv-GRU (red) and the actual value of water level (green) at Kamigasan Bridge in 2015; Lower: the Mahalanobis distance.

model, it could forget useless information through the forget gate and learn useful information from historical data over time through memory units. Based on this result, the LSTM model could make use of long-term dependencies to produce accurate water level predictions. As for the CNN model, due to its abilities to extract the patterns of local trends and catch

FIGURE 22. Upper: The prediction by Conv-GRU (red) and the actual value of water level (green) at Quchi in 2015; lower: the Mahalanobis distance.

FIGURE 23. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Zhongzheng Bridge in 2015; lower: the Mahalanobis distance.

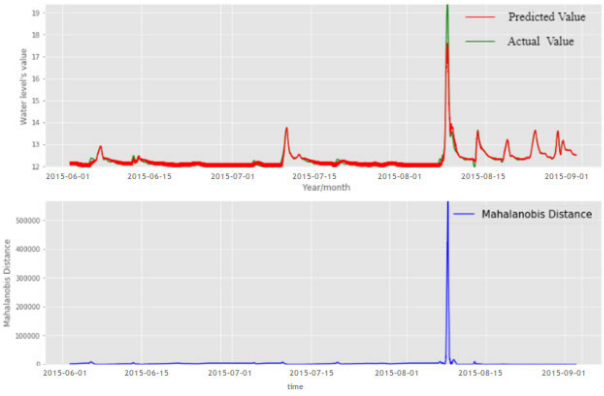

FIGURE 24. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Wanfu Bridge in 2015; lower: the Mahalanobis distance.

the same patterns occurring in different regions, it performed well in the water level prediction of our experiment. For further improvements in the accuracy and stability of water level prediction, we proposed a new deep neural network framework that integrated CNN modules with GRU modules. GRU is relatively new, and its performance is comparable to LSTM but has higher computational efficiency (due to having a more complex structure, as noted previously) [28].

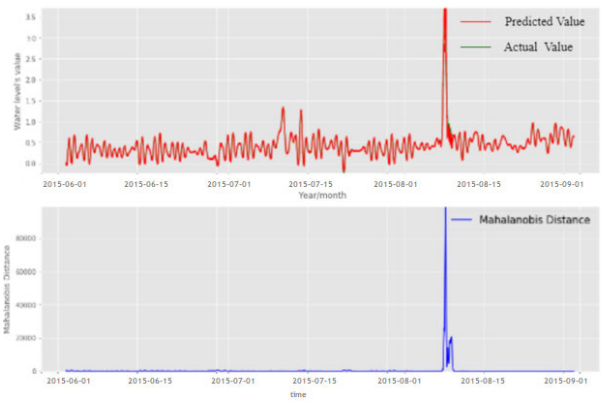

FIGURE 25. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Zhongshan Bridge in 2015; lower: the Mahalanobis distance.

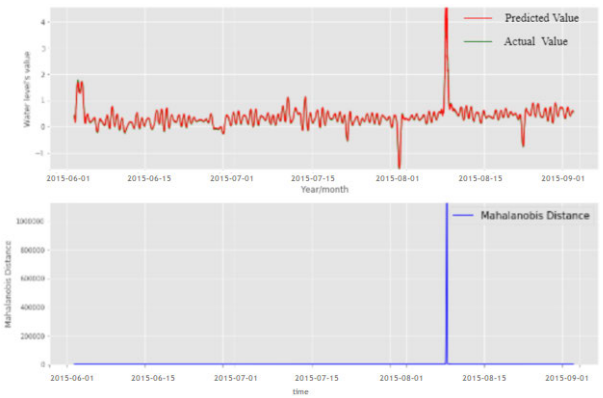

FIGURE 26. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Taipei Bridge in 2015; lower: the Mahalanobis distance.

FIGURE 27. Upper: The prediction by Conv-GRU (red) and the actual water level value (green) at Chenglin Bridge in 2015; lower: the Mahalanobis' distance.

In the experiment, our proposed Conv-GRU model had outperformed the other four models. Individual single CNN and GRU models are not very good in predicting performance, but if the advantages of the two models are combined, we propose an integrated model, the prediction performance is better than individual single network models. Because of the integration

of the hidden features from CNN and GRU, the proposed model could be more effective and stable in water level prediction. This result demonstrated that the integrative model had better stability than the stand-alone model and also gave a new research direction for timing prediction based on the fusion of GRU and CNN. In the future, more studies could focus on the improvement of accuracy in short-term river level predictions by integrating the hidden features of GRU and CNN more efficiently.

Furthermore, we used the probability distributions of the prediction errors from these deep learning models to indicate normal or abnormal behavior. One advantage of using neural networks is that the water level can be fed directly into the network without the elaborate pre-treatment required by other technologies. At the same time, networks do not need prior information about anomalous signals, because they are trained only on normal data. Based on our experiments, we believed Conv-GRU to be an effective time anomaly detector. It was possible to tune the Conv-GRU for different feature sets and detect various types of anomalous signals. The memory of the Conv-GRU allowed us to make predictions while using context and input data. For anomaly detection, a one-time step prediction was adequate. The multi-time step prediction demonstrated the ability of Conv-GRU to be a time series modeler. The prediction of multiple time steps could provide an early indication of anomalous behavior. It could also serve as an important first step in the detection of an anomaly for IoT sensor data maintenance. Finally, it could be seen that anomalous could be detected by our proposed model during Typhoon Soudelor in August 2015 (the example used for data collection). After conducting experiments based on the data from several river water level stations, it could be seen that our model could effectively predict changes in river water levels.

V. CONCLUSION

In this study, we proposed a method for river water level prediction and anomaly detection by combining the Conv-GRU model and the multivariate Gaussian distribution method. In this powerful model for time series prediction, CNN was used for the feature acquisition of time series data and the GRU model was used to learn the long-term dependent features in a time series. The combined CNN and GRU model was applied to predict water levels based on the data sets of the water level stations. Finally, the resulting prediction error was modeled as a multivariate Gaussian distribution and was used to assess the probability of anomalous water level behavior.

The first step of our experiment was to compare different models (including ANN, CNN, LSTM, Seq2seq and Conv-GRU) based on deep learning to predict water levels. In order to have the same benchmark comparison, we compared each model and found the best configuration of network parameters. The experimental results showed that compared to the competitive methods of deep learning, such as ANN, CNN, LSTM and Seq2seq, Conv-GRU performed well on several performance metrics, such as RMSE, MAE and MAPE, for water level prediction.

It was verified that Conv-GRU networks could learn higher-level temporal patterns with unknown pattern durations. In addition, it may be possible to use Conv-GRU networks to simulate normal time series behavior and detect anomalies. Our proposed Conv-GRU combined Gaussian statistical anomaly detection method, which produced good results using real-world datasets that involved short-term temporal and long-term temporal dependence modeling, especially when it was not known in advance whether the normal behavior involved the long-term dependence or not. In the future, the performance of the river water level prediction model could be improved by applying evolutionary algorithms so as to optimize the Conv-GRU model. Furthermore, more data sets on river level predictions could be collected to validate our proposed model. We could also use relevant information from upstream and downstream of a river to analyze their relationships and water level changes caused by time. In addition, we may try to apply Conv-GRU to test time series data gathered from different IoT sensors (such as electricity and PM2.5, etc.). In the future, the application of our forecast and anomaly analysis system in river level stations could be extended to other areas of Taiwan, especially the central and southern regions, where large disasters often occur during the typhoon season (July-September), to establish disaster prevention strategies and reduce casualties.

REFERENCES

[1] *Disasters 2018: Year in Review. Issue No. 54*, CRED Crunch, Apr. 2019.
[2] *National Science and technology Center for Disaster Reduction*, NCDR, New Delhi, India, Mar. 2020.
[3] (ToughCo Homepage). *The Types of Flood Events and Their causes*. Accessed: May 12, 2020. [Online]. Available: https://www.thoughtco.com/the-types-of-flood-events-4059251
[4] S. Ahile, "Household perception and preparednessagainst flooding in makurdi town, benue state, nigeria," *IOSR J. Environ. Sci., Toxicol. Food Technol.*, vol. 8, no. 11, pp. 1–6, 2014.
[5] M. Tsai, M. Hsu, J. Fu, L. Lin, and A. Wang, "Radar-based quantitative precipitation estimation for flood forecast in rivers," *Taiwan Agricult. Eng. Soc.*, vol. 8, p. 2, May 2011.
[6] Z. T. Guo. *Flood Disaster in Taiwan*. Accessed: May 12, 2020. [Online]. Available: https://web.fg.tp.edu.tw/~earth/learn/esf/magazine/980902.htm
[7] L. H. Xie. *Newtalk Home Page*. Accessed: May 12, 2020. [Online]. Available: https://newtalk.tw/news/view/2019-05-22/249788
[8] Z. Zhang, "The survey of typhoon Soudelor disaster in 2015," National Science and Technology Center for Disaster Reduction, Xindian, New Taipei, Tech. Rep., Nov. 2015.
[9] *Flood Forecasting in the Year When The Team of a Flood Forecasting Was Set Up*. Accessed: May 12, 2020. [Online]. Available: http://epaper.wra.gov.tw/Article_Detail.aspx?s=FB41F1D9FF8DD41E
[10] *The Improvement of the Real-Time Flood Forecasting System in Tamsui River*. Accessed: May 12, 2020. [Online]. Available: http://epaper.wra.gov.tw/Article_Detail.aspx?s=9616E7D735D18E8C
[11] R. Krzysztofowicz, K. S. Kelly, and D. Long, "Reliability of flood warning systems," *J. Water Resour. Planning Manage.*, vol. 120, no. 6, pp. 906–926, Nov. 1994.
[12] E. Toth, A. Brath, and A. Montanari, "Comparison of short-term rainfall prediction models for real-time flood forecasting," *J. Hydrol.*, vol. 239, nos. 1–4, pp. 132–147, Dec. 2000.
[13] K. Thirumalaiah and M. C. Deo, "Hydrological forecasting using neural networks," *J. Hydrologic Eng.*, vol. 5, no. 2, pp. 180–189, Apr. 2000.

[14] P. K. Kenabatho, B. P. Parida, D. B. Moalafhi, and T. Segosebe, "Analysis of rainfall and large-scale predictors using a stochastic model and artificial neural network for hydrological applications in southern Africa," *Hydrol. Sci. J.*, vol. 60, pp. 1943–1955, Dec. 2015, doi: 10.1080/02626667.2015.1040021.

[15] S. H. Elsafi, "Artificial neural networks (ANNs) for flood forecasting at Dongola station in the river Nile, Sudan," Tech. Rep., 2014.

[16] F. Kratzert, D. Klotz, C. Brenner, K. Schulz, and M. Herrnegger, "Rainfall-runoff modeling using long short-term memory (LSTM) networks, Hydrol," *Earth Syst. Sci.*, vol. 22, pp. 6005–6022, Dec. 2018, doi: 10.5194/hess-22-6005-2018.

[17] H. Assem, S. Ghariba, G. Makrai, P. Johnston, L. Gill, and F. Pilla, "Urban water flow and water level prediction base," *Deep Learn.*, vol. 10536, pp. 317–329, Dec. 2017.

[18] J. Brownlee. (Aug 2019). *What is Deep Learning Machine Learning Mastery Home Page*. Accessed: May 12, 2020. [Online]. Available: https://machinelearningmastery.com/what-is-deep-learning/

[19] R. Dechter, "Learning while searching in constraint-satisfaction problems," Comput. Sci. Dept., Univ. California, Berkeley, CA, USA, Tech. Rep., 1986.

[20] I. Aizenberg, N. Aizenberg, and J. Vandewalle, *Multi-Valued and Universal Binary Neurons: Theory, Learning and Applications*. Cham, Switzerland: Springer, 2000.

[21] G. Hinton, S. Osindero, and Y. W. Teh, "A fast learning algorithm for deep belief nets," Massachusetts Inst. Technol., Cambridge, MA, USA, Tehc. Rep., 2006.

[22] Y. Bengio, *Learning deep architectures for AI. Journal of Foundations and Trends in Machine Learning*. New York, NY, USA: Now, 2009, p5.

[23] I. Goodfellow, Y. Bengio, and A. Courville, *The Deep Learning Textbook*. Cambridge, MA, USA: MIT Press, 2009, pp. 1–2.

[24] J. Dean, *Large-Scale Deep Learning for Intelligent Computer Systems. Google Brain Team in collaboration with many other teams*. San Francisco, CA, USA: BayLearn, 2015, p. 11.

[25] N. Ballas, L. Yao, C. Pal, and A. Courville, "Delving deeper into convolutional networks for learning video representations," 2015, *arXiv:1511.06432*. [Online]. Available: http://arxiv.org/abs/1511.06432

[26] K. Cho, B. van Merrienboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," 2014, *arXiv:1406.1078*. [Online]. Available: http://arxiv.org/abs/1406.1078

[27] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014, *arXiv:1409.1259*. [Online]. Available: http://arxiv.org/abs/1409.1259

[28] J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, "Empirical evaluation of gated recurrent neural networks on sequence modeling," 2014, *arXiv:1412.3555*. [Online]. Available: http://arxiv.org/abs/1412.3555

[29] W. S. McCulloch and W. Pitts, "A logical calculus of the ideas immanent in nervous activity," *Bull. Math. Biophys.*, vol. 5, no. 4, pp. 115–133, Dec. 1943, doi: 10.1007/BF02478259.

[30] R. G. Morris and D. Hebb, *The Organization of Behavior*. New York, NY, USA: Wiley, 1949.

[31] F. Rosenblatt, "The perceptron: A probabilistic model for information storage and organization in the brain," *Psychol. Rev.*, vol. 65, no. 6, p. 386, 1958.

[32] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Comput.*, vol. 1, no. 4, pp. 541–551, Dec. 1989.

[33] M. Kantardzic, *Data Mining: Concepts, Models, Methods, and Algorithms*. Hoboken, NJ, USA: Wiley, 2011, p. 207.

[34] T. Daniell, "Neural networks. Applications in hydrology and water resources engineering," in *Proc. Nat. Conf. Inst. Eng. Australia*, 1991.

[35] P. B. Chattopadhyay and R. Rangarajan, "Application of ANN in sketching spatial nonlinearity of unconfined aquifer in agricultural basin," *Agricult. Water Manage.*, vol. 133, pp. 81–91, Feb. 2014.

[36] S. F. Rani Parekh, "Application of artificial neural network (ANN) for reservoir water level forecasting," *Int. J. Sci. Res.*, vol. 3, no. 7, pp. 1077–1082, 2014.

[37] R. K. Biswas, A. Jayawardena, and P. Hai, "Water levels forecasting using artificial neural networks," Tech. Rep., 2009.

[38] R. Bustami, N. Bessaih, C. Bong, and S. Suhaila, "Artificial neural network for precipitation and water level predictions of bedup river," *IAENG Int. J. Comput. Sci.*, vol. 34, no. 2, pp. 1–15, 2007.

[39] S. Arbain and A. Wibowo, "Time series methods for water level forecasting of Dungun river in Terengganu Malaysia," *Int. J. Eng. Sci. Technol.*, vol. 4, pp. 1803–1811, Apr. 2012.

[40] M. K. Tiwari and C. Chatterjee, "Development of an accurate and reliable hourly flood forecasting model using wavelet–bootstrap–ANN (WBANN) hybrid approach," *J. Hydrol.*, vol. 394, nos. 3–4, pp. 458–470, Nov. 2010.

[41] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. Cambridge, MA, USA: MIT Press, 2016, p. 326.

[42] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, Oct. 1959.

[43] K. Fukushima and S. Miyake, "Neocognitron: A new algorithm for pattern recognition tolerant of deformations and shifts in position," *Pattern Recognit.*, vol. 15, no. 6, pp. 455–469, Jan. 1982.

[44] R. Yamashita, M. Nishio, R. K. G. Do, and K. Togashi, "Convolutional neural networks: An overview and application in radiology," *Insights Into Imag.*, vol. 9, no. 4, pp. 611–629, Aug. 2018.

[45] A. Borovykh, S. Bohte, and C. W. Oosterlee, "Conditional time series forecasting with convolutional neural networks," 2017, *arXiv:1703.04691*. [Online]. Available: http://arxiv.org/abs/1703.04691

[46] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," 2012, *arXiv:1206.5538*. [Online]. Available: http://arxiv.org/abs/1206.5538

[47] H. Ismail Fawaz, G. Forestier, J. Weber, L. Idoumghar, and P.-A. Muller, "Deep learning for time series classification: A review," *Data Mining Knowl. Discovery*, vol. 33, no. 4, pp. 917–963, Jul. 2019, doi: 10.1007/s10618-019-00619-1.

[48] J. Brownlee. *A Gentle Introduction to Long Short-Term Memory Networks by the Experts*. Accessed: May 13, 2020. [Online]. Available: https://machinelearningmastery.com/gentle-introduction-long-short-term-memory-networks-experts/

[49] R. Gall. *What is LSTM Home*. Accessed: May 13, 2020. [Online]. Available: https://hub.packtpub.com/what-is-lstm/

[50] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[51] A. Graves, A. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, Vancouver, BC, Canada, May. 2013, pp. 6645–6649.

[52] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," in *Proc. 27th Int. Conf. Neural Inf. Process. Syst.*, Montreal, QC, Canada, vol. 2, De. 2014, pp. 3104–3112.

[53] T. Mikolov, A. Joulin, S. Chopra, M. Mathieu, and M. Ranzato, "Learning longer memory in recurrent neural networks," 2014, *arXiv:1412.7753*. [Online]. Available: https://arxiv.org/abs/1412.7753

[54] Y. Li and H. Cao, "Prediction for tourism flow based on LSTM neural network," *Procedia Comput. Sci.*, vol. 129, pp. 277–283, 2018.

[55] Y. Duan and F.-Y. Wang, "Travel time prediction with LSTM neural network," in *Proc. IEEE 19th Int. Conf. Intell. Transp. Syst. (ITSC)*, Rio de Janeiro, Brazil, Nov. 2016, pp. 1053–1058.

[56] D. Nelson, A. Pereira, and R. de Oliveira, "Stock market's price movement prediction with LSTM neural networks," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Anchorage, AK, USA, May 2017, pp. 1419–1426.

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

[58] X. Le, V. Ho, G. Lee, and S. Jung, "A deep neural network application for forecasting the inflow into the Hoa Binh reservoir in Vietnam," in *Proc. 11th Int. Symp. Lowland Technol. (ISLT)*, Hanoi, Vietnam, Sep. 2018, pp. 26–28.

[59] Le, Ho, Lee, and Jung, "Application of long short-term memory (LSTM) neural network for flood forecasting," *Water*, vol. 11, no. 7, p. 1387, Jul. 2019.

[60] C. Liang, H. Li, M. Lei, and A. Q. Du, "Dongting lake water level forecast and its relationship with the three gorges dam based on a long short-term memory network," *Water*, vol. 10, no. 10, p. 1389, Oct. 2018.

[61] S. ur Rehman, Z. Yang, M. Shahid, N. Wei, Y. Huang, M. Waqas, S. Tu, and O. ur Rehman, "Water preservation in soan river basin using deep learning techniques," 2019, *arXiv:1906.10852*. [Online]. Available: http://arxiv.org/abs/1906.10852

[62] B. D. Bowes, J. M. Sadler, M. M. Morsy, M. Behl, and J. L. Goodall, "Forecasting groundwater table in a flood prone coastal city with long short-term memory and recurrent neural networks," *Water*, vol. 11, no. 5, p. 1098, May 2019.

[63] Q. Zhang, C. Wei, Y. Wang, S. Du, and Y. Zhou, "Potential for prediction of water saturation distribution in reservoirs utilizing machine Learning Methods," *Energies*, vol. 12, no. 19, pp. 1–21, 2019.
[64] M. Wadhwa. *Seq2seq Model in Machine Learning Home Page*. Accessed: May 13, 2020. [Online]. Available: https://www.geeksforgeeks.org/seq2seq-model-in-machine-learning/
[65] F. Liu, F. Xu, and S. Yang, "A flood forecasting model based on deep learning algorithm via integrating stacked autoencoders with BP neural network," in *Proc. IEEE 3rd Int. Conf. Multimedia Big Data (BigMM)*, Apr. 2017, pp. 1–8.
[66] B. Lugt and A. Feelders, "Conditional forecasting of water level time series with RNNs," Dept. Inf. Comput. Sci., Utrecht Univ., Utrecht, The Netherlands, Tech. Rep., 2018.
[67] Uniqtech, "Multilayer Perceptron (MLP) vs Convolutional Neural Network in Deep Learning," Available online:. [Online]. Available: https://medium.com/data-science-bootcamp/multilayer-perceptron-mlp-vs-convolutional-neural-network-in-deep-learning-c890f487a8f1 (Accessed on 13 May 2020)
[68] G. Swapna, K. Soman, and R. Vinayakumar. *Procedia Computer Science Home Page*. Accessed: May 13, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S1877050918307737
[69] T. Le, M. Vo, B. Vo, E. Hwang, S. Rho, and S. Baik, "Improving electric energy consumption prediction using CNN and Bi-LSTM," *Appl. Sci.*, vol. 9, no. 2, p. 4237, 2019.
[70] W. Bao, J. Yue, and Y. Raol, "A deep learning framework for financial time series using stacked autoencoders and longshort term memory," *PLOS ONE*, vol. 12, no. 7, 2017, Art. no. e0180944.
[71] H. Palangi, R. Ward, and L. Deng, "Distributed compressive sensing: A deep learning approach," *IEEE Trans. Signal Process.*, vol. 64, no. 17, pp. 4504–4518, Sep. 2016.
[72] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, pp. 533–536, Oct. 1986.
[73] P. J. Werbos, "Backpropagation through time: What it does and how to do it," *Proc. IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990, doi: 10.1109/5.58337.
[74] I. Sutskever, O. Vinyals, and Q. V. Le, "Sequence to sequence learning with neural networks," 2014, *arXiv:1409.3215*. [Online]. Available: http://arxiv.org/abs/1409.3215
[75] A. G. Barnston, "Correspondence among the correlation, RMSE, and heidke forecast verification measures; refinement of the heidke score," *Weather Forecasting*, vol. 7, no. 4, pp. 699–709, Dec. 1992.
[76] A. de Myttenaere, B. Golden, B. Le Grand, and F. Rossi, "Mean absolute percentage error for regression models," *Neurocomputing*, vol. 192, pp. 38–48, Jun. 2016.
[77] C. Willmott and K. Matsuura, "Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance," *Climate Res.*, vol. 30, pp. 79–82, 2005, doi: 10.3354/cr030079.
[78] *MeanAbsolute Error-MAE [Machine Learning(ML)]*. Accessed: May 13, 2020. [Online]. Available: https://medium.com/@ewuramaminka/mean-absolute-error-mae-machine-learning-ml-b9b4afc63077
[79] P. Malhotra, L. Vig, G. Shroff, and P. Agarwal, "Long short term memory networks for anomaly detection in time series," in *Proc. 23rd Eur. Symp. Artif. Neural Netw., Comput. Intell. Mach. Learn.*, Belgium, Brussels, Apr. 2015, pp. 90–91.
[80] P. Hayton, S. Utete, D. King, S. King, P. Anuzis, and L. Tarassenko, "Static and dynamic novelty detection methods for jet engine health monitoring," *Phil. Trans. Roy. Soc. A, Math., Phys. Eng. Sci.*, vol. 365, no. 1851, pp. 493–514, Feb. 2007.
[81] C. Cho, G. Lee, Y. Tsai, and K. Lan, "Toward stock price prediction using deep learning," in *Proc. 12th IEEE/ACM Int. Conf. Utility Cloud Comput. Companion*, Auckand, New Zealand, Sep. 2019, pp. 133–135.
[82] C. Giles, S. Lawrence, and A. Tsoi, "Noisy time series prediction using a recurrent neural network and grammatical inference," *Mach. Learn.*, vol. 44, pp. 161–183, Dec. 2001.
[83] S. Xu and L. Chen, "A novel approach for determining the optimal number of hidden layer neurons for FNN's and its application in data mining," in *Proc. 5th Int. Conf. Inf. Technol. Appl.*, Cairns, Australia, 2008, pp. 1–8.

SCOTT MIAU received the master's degree in management from Waseda University, Japan, in 2000. He is currently pursuing the Ph.D. degree with the Department of Management Information Systems, National Chengchi University (NCCU), Taiwan. His current research interests are mainly in financial technologies and blockchain.

WEI-HSI HUNG received the master's and Ph.D. degrees (Hons.) from the Department of Management Systems, University of Waikato, New Zealand. He is currently a Professor of Management Information Systems with the National Chengchi University, Taiwan. His research interests are in the areas of e-commerce, IS alignment, knowledge management, and supply chain management. His research articles appeared in journals such as *Decision Support Systems*, *Communications of the Association for Information Systems*, *Journal of Global Information Management*, *Internet Research*, *Industrial Marketing Management*, *Technology Analysis & Strategic Management*, *Journal of Computer Information Systems*, *Telematics and Informatics*, *Computers in Human Behavior*, *Industrial Management & Data Systems*, *International Journal of Logistics Research and Applications*, *Asia Pacific Management Review*, *International Journal of Web Portals*, *Communications of the ICISA*, *Pacific Asian Journal of Association for Information Systems*, and *Journal of Information Management*.