Water Resources Management (2019) 33: 2471–2484
https://doi.org/10.1007/s11269-019-02255-2

Lake Level Prediction using Feed Forward and Recurrent Neural Networks

Bahrudin Hrnjica[1] · **Ognjen Bonacci**[2]

Received: 19 October 2018 / Accepted: 1 April 2019 /
Published online: 30 April 2019

Abstract
The protection of high quality fresh water in times of global climate changes is of tremendous importance since it is the key factor of local demographic and economic development. One such fresh water source is Vrana Lake, located on the completely karstified Island of Cres in Croatia. Over the last few decades a severe and dangerous decrease of the lake level has been documented. In order to develop a reliable lake level prediction, the application of the artificial neural networks (ANN) was used for the first time. The paper proposes time-series forecasting models based on the monthly measurements of the lake level during the last 38 years, capable to predict 6 or 12 months ahead. In order to gain the best possible model performance, the forecasting models were built using two types of ANN: the Long-Short Term Memory (LSTM) recurrent neural network (RNN), and the feed forward neural network (FFNN). Instead of classic lagged data set, the proposed models were trained with the set of sequences with different length created from the time series data. The models were trained with the same set of the training parameters in order to establish the same conditions for the performance analysis. Based on root mean squared error (RMSE) and correlation coefficient (R) the performance analysis shown that both model types can achieve satisfactory results. The analysis also revealed that regardless of the model types, they outperform classic ANN models based on datasets with fixed number of features and one month the prediction period. Analysis also revealed that the proposed models outperform classic time series forecasting models based on ARIMA and other similar methods .

Keywords ANN · LSTM · Time series prediction · Lake level · Karst hydrology

1 Introduction

ANN is one of the basic methods of machine learning (ML) and artificial intelligence (AI), which uses the concept of supervised learning in solving various problems, such as

✉ Bahrudin Hrnjica
bahrudin.hrnjica@unbi.ba

Ognjen Bonacci
obonacci@gradst.hr

[1] Faculty of Technical Sciences, University of Bihac, Bihac, Bosnia and Herzegovina

[2] Faculty of Civil Engineering, Architecture and Geodesy Croatia, University of Split, Split, Croatia

classification, regression, time series, image processing, robotics, internet search, data mining, etc. Recently, the ANN has been successfully used to solve today's most complex and challenging problems, such as speech recognition (Abdel-Hamid et al. 2014), language translations (Sutskever et al. 2014), image and video processing (Donahue et al. 2017), object detection (Ren et al. 2017), and image captioning (Mao et al. 2014).

However, the prediction of time series events is still one of the most challenging tasks, and it is an active research subject of many engineers and scientists. So many examples of time series events can be identified around us, and the prediction of their future state is of tremendous importance. For example, it is of crucial importance for the world economy to predict stock prices (Sethia and Raut 2019), the price of energy (Panapakidis and Dagoumas 2016), corporate sales (Cantón et al. 2018). Also, time series events can be used to predict the weather, environmental, hydrological and geological events (Cao et al. 2012; Lee 2008).

Trend analysis of the hydro and climatological variables of lake is of constant interest (Nourani et al. 2018). Comparative studies of water level predictions between ANN and related methods were investigated in many papers. There are several studies in which the results of water level predictions calculated by ANN were compared to other powerful and popular methods e.g. support vector machines (SVM), and neuro-fuzzy systems (NFS) (Gong et al. 2016; Modaresi et al. 2018). In addition, Ghorbani et al. conducted a comparative study of the sea level results calculated by GP and ANN and concluded that both methods might provide satisfactory results (Ghorbani et al. 2010). More papers can be found about ANN application developing prediction models for various hydrological properties e.g. surface water level (Altunkaynak 2007), monthly streamflow, reservoir level forecasting (Ondimu and Murase 2007; Rani and Parekh 2014), forecasting surface water level fluctuations (Piasecki et al. 2017), or uses different ANN algorithms in order to develop monthly steamflow predictions (Danandeh Mehr et al. 2015).

Depending of the available data, many different methods and approaches can be used for lake level predictions. Crapper et al. (1996) used water balance models for the prediction of lake level. Seo at al. developed water level forecasting for reservoir inflow using hybrid models based on ANN (Seo et al. 2015). RNN models were developed in order to control floods in the urban area of Taipei City, Taiwan (Chang et al. 2014). Five hours ahead, water level prediction, for the three upstream rivers on the east cost of Malaysia, were successfully calculated by using ANN (Lukman et al. 2017). A hybrid MLP-FFA model was implemented for water level predictions for Egirid Lake in Turkey (Ghorbani et al. 2017).

Using LSTM RNN in the predictions of the hydrological properties such as water level, precipitation, etc. was not often used. Because of the complexity of time series and recent improvements of the LSTM RNN, there are only few papers that use LSTM in the prediction of hydrological properties (Kratzert et al. 2018; Akbari Asanjan et al. 2018). The paper presents, for the first time, the application of LSTM RNN in the prediction of lake levels. So far, the analyses of the Vrana Lake level was preformed using classic regression and statistical methods (Bonacci 2017), that are not powerful enough in the predictions of stochastic events like water level. Furthermore, this is the first application of ANN in building prediction models, and analyzing the Vrana Lake level as a time series event.

2 Study Area

The study area (Fig. 1) is located in the border zone between Mediterranean and sub-Mediterranean climates, characterized by a warm climate with hot and relatively dry summers, and rainy autumns. The temperature rarely drops below 0 °C, so the lake surface

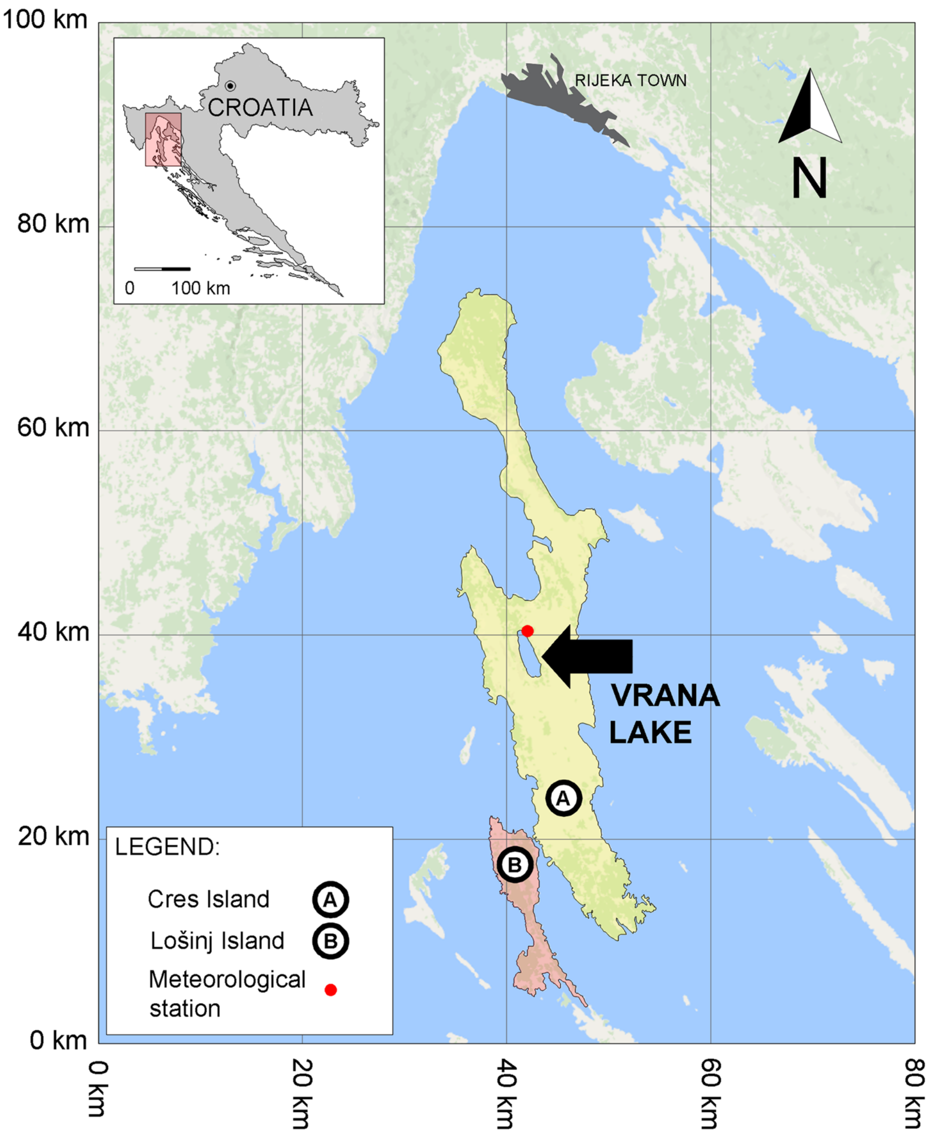


Fig. 1 Map of Cres and Losinj Islands and Vrana Lake

did not freeze during the recording period. Table 1 shows the main annual hydrologic and climatologic characteristics of Vrana Lake and its catchment.

Vrana Lake is a very complex hydrological and hydrogeological system, which is also under strong anthropogenic influences due to water pumping from the lake for water supply. The lake catchment area is strongly protected and can only be visited with special permission. It is an elongated lake that extends in a NW-SE direction, 5 km long, with a maximum width of 1.45 km. A direct measurement of the lake's inflow and outflow is not possible, which makes definition and control of its water budget difficult. Vrana Lake has

Table 1 Main annual hydrologic and climatologic characteristics of Vrana Lake and its catchment

	H (m a.s.l.)	P (mm)	T (°C)
Minimum	9.38	685.7	13.51
Average	12.59	1074.6	14.62
Maximum	15.84	1743.0	16.73
STD	1.48	225.72	0.604
Range	6.47	1057.3	3.22

an endorheic catchment. This kind of lake discharges inland, into a closed lake basin or catchment of internal drainage. Vrana Lake does not have surface outflow. The water budget method was used to define the lake catchment area at approximately 25 km2.

In the last few decades, the demand for water has been rapidly increasing with tourism and industry becoming the leading consumers of the fresh water. Since tourism is mostly limited to the period from May to September, the consumption of water during the tourist season increases three to six times higher than the rest of the year. The pumping of lake water started in 1967, and became the main water source on the island. The continuous increase of pumping is very harmful, and will have many dangerous ecological, as well as social consequences.

The necessity of the establishment of lake level predictions is especially enhanced by the fact that the lake level has been decreasing over the last few decades. The first significant drop in the lake level was recorded in 1983. This threatening trend has been caused by natural (global warming) and anthropogenic (water overexploitation) drives (Bonacci 2017).

3 Methodology

3.1 Introduction to ANN

The concept of ANN can be compared to the human brain and its neurons. The artificial neuron in ANN acts similarly to the neuron in a human bran. It can be defined as a set of input parameters x_i (i = 1,..n), a set of weight factors w_i $(i = 1, , n)$, the dot product, $\sum \mathbf{w} \cdot \mathbf{x}$, and an activation function $a(.)$ (Mehrotra et al. 1997). The first concept of the artificial neuron is called the perceptron which was introduced by Rosenblatt (1960), and can produce only two values: -1 and 1.

Let the x_n represent the input vector with n components, the associated weight w_n, and a bias value b_0 and the activation function $sign$. The output y of the perceptron is expressed as:

$$y = f\ (net) = f\ (\mathbf{w} \cdot \mathbf{x}) = sign\ \left(\sum_{i=1}^{n} x_i w_i + b_0 \right), \tag{1}$$

where $sign$ represents the activation function defined as:

$$sign\ \ (net) = \{+1\ if\ \ net \geq \mathbf{w} \cdot \mathbf{x} \geq 0;\ -1\ if\ \ net < \mathbf{w} \cdot \mathbf{x} < 0 \tag{2}$$

Besides $sign$, there are many other activation functions which can produce different kinds of outputs e.g., $Tanh$, $ReLU$, $Sofmax$, etc. Simple FFNN may have at least three layers: input, hidden and output layer. The number of neurons in each layer may vary depending on the complexity of the problem. In the input layer, each neuron corresponds to the input parameter, while the output layer is related to the output result. In the middle of

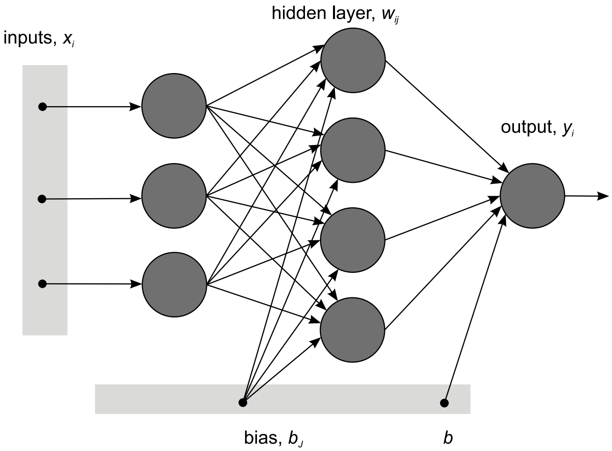


Fig. 2 Simple FFNN with three interconnected layers

the input and the output layer there can be one or more hidden layers with arbitrary numbers of neurons. Figure 2 shows the simple FFNN with the three layers.

The ANN is created by connecting many neurons into a network. Since each neuron can produce a simple decision, the ANN must consist of many neurons in order to establish a network that can find solutions to real world problems.

3.2 Introduction to RNN

FFNN models are usually trained so that data do not have any order when entering into the network. In case when data is recorded in time, or when the data are represented as sequences, the FFNN cannot manage it. The FFNN should be enriched and upgraded in order to support these kinds of data, as well as the network output that depends on inputs and the previous states of the network. One of the approaches is to develop the RNN, which was first introduced by Hopfields in the 1980s, and later popularized when the backpropagation algorithm was improved (Pineda 1987). The concept of the RNN is based on cycles showing that the current state of the network relies on the current data, and also on the data produced by the previous network outputs. The RNN is a special type of the ANN that provides two kinds of inputs: the output of the previous time, $\mathbf{h_{i-1}}$ and the current input $\mathbf{x_i}$. Furthermore, the RNN has a special kind of internal memory which can hold long-term information history (Sak et al. 2014).

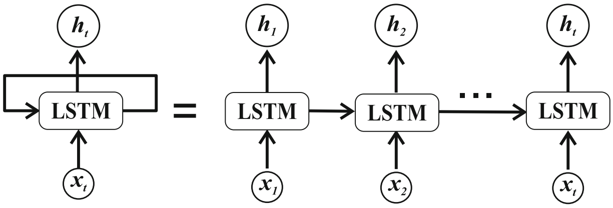


Fig. 3 Schematic representation of the folded and unfolded RNN

Figure 3 shows two kinds of representations of the RNN. On the left side of the equal sign, the RNN is presented in classic feed forward like mode, where the three layers are presented: input, hidden and output layer. The hidden layer contains the recursion for handling the previous outputs to behave as the input in the current time. The RNN can also be shown in an unrolled state in time t (the right side of the equal sign in the Figure). The RNN is presented with t interconnected FFNN, where t indicates the past steps, thus far. The RNN has several variants that can produce significant results, and one of the most used is the LSTM (Hochreiter and Schmidhuber 1997). The LSTM is a special RNN which can provide a constant error propagation through the network due to a special network design. LSTM consists of memory blocks with self-connections defined in the hidden layer, which has the ability to store the temporal state of the network. Besides memorization, the LSTM cell has special multiplicative units called gates, which control the information flow. Each memory block consists of the input gate – which controls the flow of the input activation into the memory cell, and the output gate which controls the output flow of the cell activation. In addition, the LSTM cell also contains the forget gate, which filters the information from the input and previous output and decides which one should be remembered or forgotten and dropped out. With such selective information filtering, the forget gate scales the LSTM cell's internal state, which is self-recurrently connected by previous cell states. Beside gating units, the LSTM cell contains the cell state. The cell state allows constant error flow. The gates of the LSTM are adaptive, since each time the content of the cell is out of date, the forget gate learns to reset the cell state. The input and the output gates control the LSTM input and output, respectively.

Figure 4 shows the LSTM cell with activation layers: input, output, forget gates, and cell.

The LSTM network is expressed as ANN where the input vector $x = (x_1, x_2, x_3, \ldots x_t)$ in time t, maps to the output vector $y = (y_1,\ y_2,\ \ldots,\ y_m)$, through the calculation of:

- forget gate, f_t:

$$f_t = \sigma\left(W_f \cdot \left[h_{t-1}, x_t\right] + b_f\right), \tag{3}$$

- input gate, i_t:

$$i_t = \sigma\left(W_i \cdot \left[h_{t-1}, x_t\right] + b_i\right), \tag{4}$$

- cell state, C_t:

$$C_t = f_t \ \otimes C_{t-1} \oplus \ i_t \otimes \tanh\left(\ W_C \cdot \left[h_{t-1}, x_t\right] + b_C\right), \tag{5}$$

- output gate, o_t:

$$o_t = \sigma\left(\ W_0 \cdot \left[h_{t-1}, x_t\right] + b_0\right). \tag{6}$$

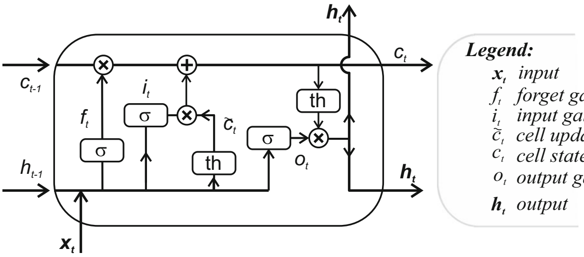


Fig. 4 LSTM cell with its internal structure

The final stage of the LSTM cell is the output calculation h_t in time t. The current output h_t is calculated with the multiplicative operation $\otimes$ between output gate layer and *tanh* layer of the current cell state C_t.

$$h_t = o_t \otimes \tanh(C_t). \tag{7}$$

The output, h_t, has passed through the network as the previous state for the next LSTM cell, or as the input for the ANN output layer.

4 FF and LSTM Models for Lake Level Prediction

The paper proposes a time series forecasting models based on LSTM and FFNN. First, the times series data were prepare and transformed into sequences. Then the training datasets were created based on the prediction periods. Each dataset is getting into one LSTM and one FF model by set of sequences. The network of each model consist of several multilayered subnetworks configured to accept a sequence of data. The output of the models is the prediction of the lake level of 6 or 12 months ahead.

4.1 Proposed Data Preparation

The training data represent univariate time series, where each row value presents the lake level for a certain month in the period from January 1978 to December 2016, as presented at

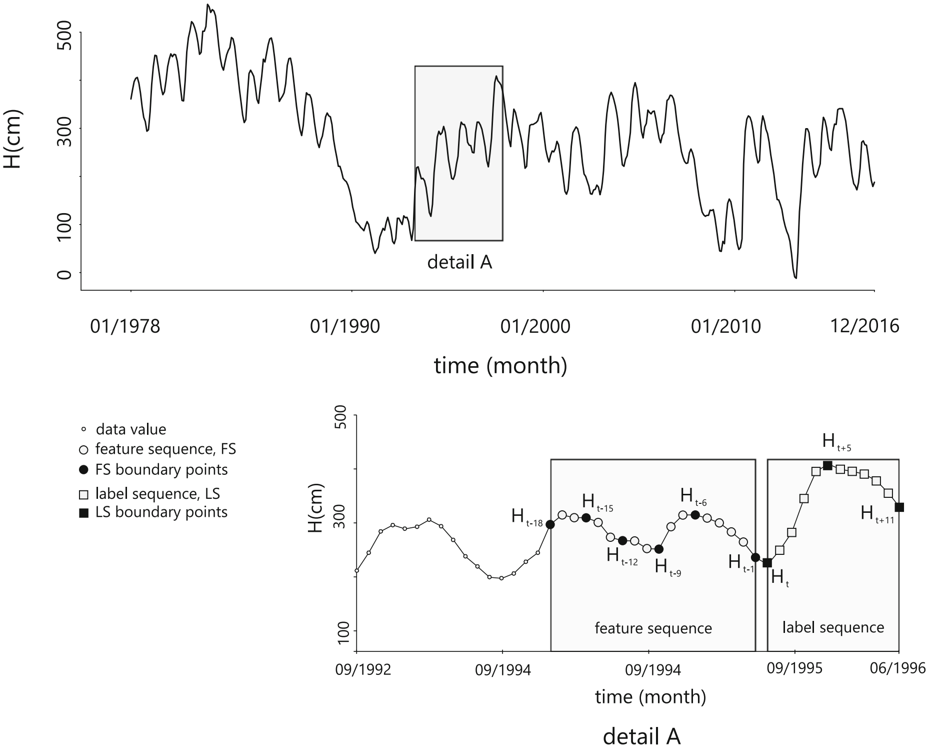


Fig. 5 Vrana lake level time series with detail how the feature and label sequences are generated for the selected time *t*

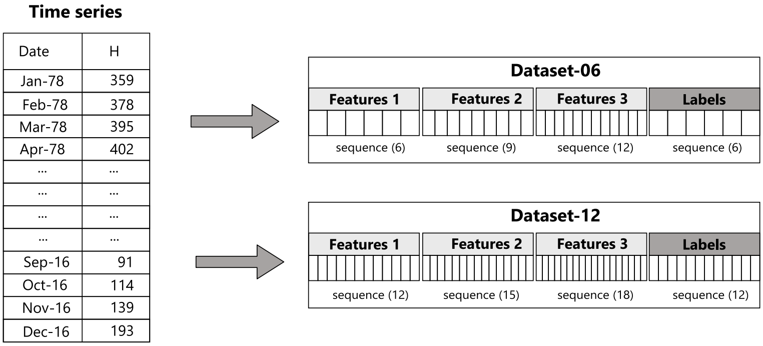
Fig. 5. Prior to training the models, the data were prepared and transformed into a data frame suitable for training the ANN. At first, the time series were transformed into sequences of features and labels. The feature sequence represents a n values in the past, while the label sequence represents the m values ahead, with the respect to the current time t. Figure 5 shows the monthly lake level time series with details how sequences of the features and labels were generated for the selected time t.

Once the features and labels sequences were defined, the time series data were transformed into the data frame consisting of a 4 different sequences. The first 3 sequences are feature sequences. The sequences were generated based on the 3 different values of the time lag. The fourth sequence was generated based on a prediction period of 6 or 12 months ahead, and it represents the label sequence. Depending of the prediction period, the feature sequences were generated with different length. The sequences of 6, 9 and 12 length were used to create the data frame for the 6 months testing period, and the sequences of 12, 15 and 18 length were used in order to create the data frame for 12 months of testing period (Fig. 6). As the result of the data preparation, two datasets were created: the *dataset-06* for training models for the prediction of 6 months ahead, and *dataset-12* for training models for the predictions of 12 months ahead.

Data preparation, model training and evaluations were performed by using ANNdotNET, a deep learning tool on the .NET platform (Hrnjica 2018). The ANNdotNET implements the ML Engine, based on Cognitive Toolkit (CNTK), a deep learning framework developed by Microsoft Research (Yu et al. 2014).

4.2 Proposed Models

The prediction models used in this paper were trained on the previously defined datasets. For each dataset, one LSTM and one FF models were defined. Prior to data transformation, the time series were scaled using log-transformation and further decomposed in order to remove the seasonal component from the data. The de-seasonalized data were used to create training datasets. Once the training process has been completed, the model evaluation and testing were performed by adding seasonal component on the model output, and then rescaled by inverse log operation. The proposed model workflow of data preparation and



Date	H
Jan-78	359
Feb-78	378
Mar-78	395
Apr-78	402
...	...
...	...
...	...
...	...
Sep-16	91
Oct-16	114
Nov-16	139
Dec-16	193

Fig. 6 Time series transformation into two data sets. Both datasets are created by generating 3 feature sequences and 1 label sequence

transformation, network configuration, network training, model evaluation and testing are shown in Fig. 7.

During the training process, each network configuration accepts the input of the three feature sequences. Each feature sequence trains one subnetwork independently of the other two. At the end, the three subnetworks are aggregated by summing the outputs.

The proposed LSTM model consists of three LSTM subnetworks. Each LSTM subnetwork contains three bidirectionally connected LSTM layers. One feature sequence is the input for the corresponded LSTM subnetwork, so that three feature sequences are the input for the three corresponded subnetworks. Similarly, proposed FF model consist of three FF subnetworks, and they are fed on the same way as LSTM subnetworks. One FF subnetwork consists of the three dense layers with TanH activation function. At the end of each subnetwork, dropout layer was added in order to prevent overfitting. For both model configurations, aggregation layer was added by summing the outputs for all three subnetworks

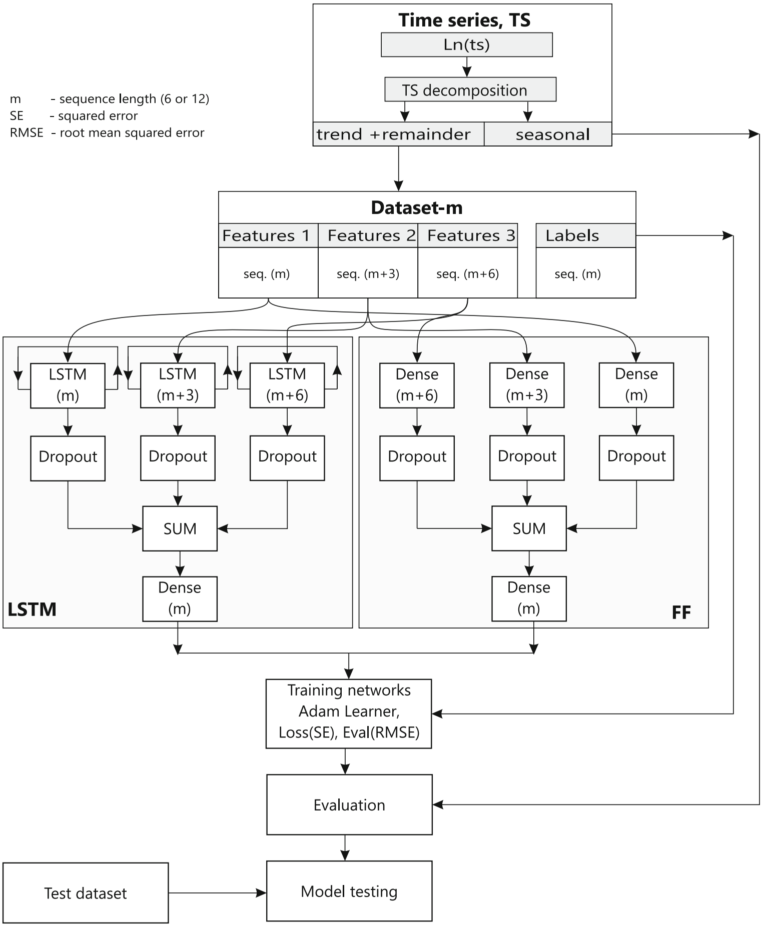


Fig. 7 Proposed model workflow for Vrana Lake level prediction

Table 2 Network configuration and training parameters used for training FF and LSTM models

FFNN		LSTM	
Parameter	Value	Parameter	Value
Network type	FFNN	Network type	LSTM RNN
Layers per subnetwork	3	Layers per subnetwork	3
Neurons per layer	6, 9, 12, 15, 18	LSTM cells per layer	6, 9, 12, 15, 18
Learner	Adam	Learner	Adam
Learning rate	0.1	Learning rate	0.1
Momentum	0.9	Momentum	0.9
Loss function	SE	Loss function	SE
Evaluation function	RMSE	Evaluation function	RMSE
Number of epochs	5000	Number of epochs	5000
Minibatch size	75	Minibatch size	75
Used early stopping	yes	Used early stopping	yes

into one result. The output layer of the models is dense layer with dimension equal to the length of the label sequence (Fig. 7).

4.3 Network and Training Parameters Settings

The training process consist of training 4 different models: LSTM-6, LSTM-12, FF-6 and FF-12. The models are built for two prediction periods (6 and 12 months) with two different networks (LSTM and FF). The model name indicates the network type, while the number in the model name indicates the prediction period. All models were trained with the same training parameters shown in Table 2. The number of iteration was 5000. The early stopping was used by saving models for every 100 epochs. After the training process was completed the best model was selected.

4.4 Model Results and Performance Analysis

The summary of the training process for the models is shown in Table 3. In the training phase, the performance parameters were roughly equal for all models. The minimum RMSE reached the LSTM-06 model, and the best correlation coefficient value was reached by the LSTM-12 model.

Table 3 Summary of the training process

Models	LSTM-06	FF-06	LSTM-12	FF-12
RMSE	81.7	66.2	78.3	83.4
R	0.898	0.882	0.897	0.852

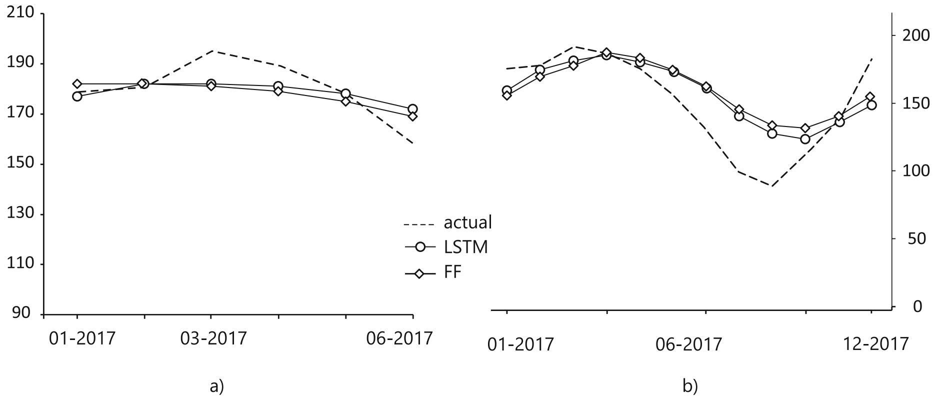


Fig. 8 Test result of the models for 6 and 12 months ahead

In order to test the models, test datasets were created based on the feature sequences from the training datasets. The feature sequences were created according to the procedures described in the data preparation section. The predicted values represent sequences of the lake level for the next 6 and 12 months in 2017. The models were evaluated for the test dataset, and the evaluation results are shown in Fig. 8

During the testing periods, the models gained the accuracy which can follow the trend of the lake level in a certain amount. Particularly, in the prediction period of 12 months, the models followed the "sinus shape" of the level. However, the obtained low level values were predicted with lower accuracy compared to the other predicted values. This may be the indication of the changes in the lake level which cannot be thoroughly extracted from the data due to climate changes and human activities.

In order to show how proposed models outperform the models based on the lagged dataframe, two ANN models with similar numbers of layers and neurons were created. During transformation of the time series into dataset, the classic approach was used by utilizing 2 and 3 lagged features to build the models. The models were trained with the same datasets and similar training parameters as the proposed models.

In order to show how proposed models outperform the forecasting models based on ARIMA and other classic regression methods, two forecasting models were created. The first model was created using classic ARIMA procedure, and the second by using different regression and exponential smoothing methods implemented in the R package called prophet (Taylor and Letham 2018).

The summary of the performance for the proposed models, the lagged ANN models and the models based on different regression methods, are shown in Tables 4 and 5.

Table 4 The performance of prediction models for Jan-June 2017

Models	**LSTM-06**	FF-06	Lag2	Lag3	ARIMA	Prophet
RMSE	**8.47**	8.69	28.45	71.06	100.58	84.71
R	**0.907**	0.770	-0.882	0.213	-0.212	-0.415

Table 5 The performance of prediction models for Jan-Dec 2017

Models	**LSTM-12**	FF-12	Lag2	Lag3	ARIMA	Prophet
RMSE	**22.84**	25.08	64.67	72.86	141.89	147.33
R	**0.808**	0.790	-0.672	0.797	-0.393	-0.595

Performance results presented in tables 4 and 5 shows that the proposed LSTM and FF models outperform classic models. Moreover, the performance of the LSTM models for 6 and 12 months prediction periods gained the best values and as shown in bold font in the tables.

5 Discussion

During the evaluation and comparison analysis of the forecasting models, several conclusions about the predictions can be identified. The first is that the prediction accuracy of the LSTM models was better than for the FF models, but not as high as expected. The reason for this can be found in the small dataset, but also from the fact that sequences in the time series are far complex than sequences the LSTM has proven state of the art results. However, the LSTM models achieved satisfactory results, with good prediction performances, specially for the 6 months test period. Regardless of the network type, the proposed models achieved state of the art results in comparison with classic lagged time series models created by ANN.

The models based on the fixed number of time lags with one months prediction period are not able to provide accurately predictions for more than one month in advance, because of the error multiplication for every further prediction.

6 Conclusion

This paper presents proposed models for Vrana Lake level predictions using FFNN and LSTM RNN, which proved to be very effective in the prediction of time-based events. However, time series predictions is still one of the most challenging task, forcing engineers to constantly try optimize existing solutions in order to get more accurate results. The results presented in the paper show that proposed forecasting models obtain good performance results and outperform ANN models built with classics approach, as well as the models based on traditional time series methods (ARIMA, nonlinear regression, exponential smoothing etc.). Due to the increased climate changes and the importance of Vrana Lake, it is crucial to establish an accurate system of the level predictions. The paper is an attempt to accomplish this.

References

Altunkaynak A (2007) Forecasting surface water level fluctuations of lake Van by artificial neural networks. Water Resour Manag 21(2):399–408

Abdel-Hamid O, Mohamed A, Jiang H, Deng L, Penn G, Yu D (2014) Convolutional neural networks for speech recognition. IEEE/ACM Transactions on Audio, Speech, and Language Processing 22(10):1533–1545. https://doi.org/10.1109/TASLP.2014.2339736

Akbari Asanjan A, Yang T, Hsu K, Sorooshian S, Lin J, Peng Q (2018) Short–term precipitation forecast based on the PERSIANN system and LSTM recurrent neural networks. J Geophys Res-Atmos 123(22):12,543-12,563. https://doi.org/10.1029/2018JD028375

Bonacci O (2017) Preliminary analysis of the decrease in water level of Vrana Lake on the small carbonate island of Cres (Dinaric karst, Croatia). Geological Society, London, Special Publications, 466. https://doi.org/10.1144/SP466.6

Cao Q, Ewing BT, Thompson MA (2012) Forecasting wind speed with recurrent neural networks. Eur J Oper Res 221(1):148–154. https://doi.org/10.1016/j.ejor.2012.02.042

Chang F-J et al (2014) Real-time multi-step-ahead water level forecasting by recurrent neural networks for urban flood control. J Hydrol 517:836–46

Cantón R, Gibaja M, Romero DE, Caballero Morales SO (2018) Sales prediction through neural networks for a small dataset, International Journal of Interactive Multimedia and Artificial Intelligence, https://doi.org/10.9781/ijimai.2018.04.003

Crapper PF, Fleming PM, Kalma JD (1996) Prediction of lake levels using water balance models. Environ Softw 11(4):251–258. https://doi.org/10.1016/S0266-9838(96)00018-4

Danandeh Mehr A, Kahya E, Sahin A, Nazemosadat MJ (2015) Successive-station monthly streamflow prediction using different ANN algorithms. Int J Environ Sci Technol 12(7):2191–2200

Donahue J, Hendricks LA, Rohrbach M, Venugopalan S, Guadarrama S, Saenko K, Darrell T (2017) Long-term recurrent convolutional networks for visual recognition and description. IEEE Trans Pattern Anal Mach Intell 39(4):677–691. https://doi.org/10.1109/TPAMI.2016.2599174

Ghorbani A et al (2010) Mohammad sea water level forecasting using genetic programming and comparing the performance with artificial neural networks. Comput Geosci 36(5):620–27

Ghorbani MA, Deo RC, Yaseen ZM, Kashani H, Mohammadi M, Pan B (2017) Evaporation prediction using a hybrid multilayer perceptron-firefly algorithm (MLP-FFA) model: case study in North Iran. Theor Appl Climatol :1–13. https://doi.org/10.1007/s00704-017-2244-0

Gong Y, Zhang Y, Lan S, Wang H (2016) A comparative study of artificial neural networks, support vector machines and adaptive neuro fuzzy inference system for forecasting groundwater levels near lake Okeechobee, Florida. Water Resour Manag 30(1):375–391. https://doi.org/10.1007/s11269-015-1167-8

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

Hrnjica B (2018) ANNdotNET- deep learning on .NET platform, Zenodo, https://doi.org/10.5281/zenodo.1461722

Kratzert F, Klotz D, Brenner C, Schulz K, Herrnegger M (2018) Rainfall–runoff modelling using Long Short-Term Memory (LSTM) networks. Hydrol Earth Syst Sci. https://doi.org/10.5194/hess-22-6005-2018

Lee TL (2008) Back-propagation neural network for the prediction of the short-term storm surge in Taichung Harbor, Taiwan. Eng Appl Artif Intell 21(1):63–72. https://doi.org/10.1016/j.engappai.2007.03.002

Lukman QA, Ruslan FA, Adnan R (2017) 5 hours ahead of time flood water level prediction modelling using NNARX technique: case study terengganu. In: 2016 7th IEEE control and system graduate research colloquium, ICSGRC 2016 - Proceeding, pp 104–108. https://doi.org/10.1109/ICSGRC.2016.7813310

Mao J, Xu W, Yang Y, Wang J, Huang Z, Yuille A (2014) Deep captioning with multimodal recurrent neural networks (m-RNN). Retrieved from arXiv:1412.6632

Mehrotra K, Mohan CK, Ranka S (1997) Elements of artificial neural networks. Elements of Artificial Neural Networks

Modaresi F, Araghinejad S, Ebrahimi K (2018) A comparative assessment of artificial neural network, generalized regression neural network, least-square support vector regression, and K-nearest neighbor regression for monthly streamflow forecasting in linear and nonlinear conditions. Water Resour Manag 32(1):243–258

Nourani V, Danandeh Mehr A, Azad N (2018) Trend analysis of hydroclimatological variables in Urmia lake basin using hybrid wavelet Mann–Kendall and Sen tests. Environ Earth Sci 77:207. https://doi.org/10.1007/s12665-018-7390

Ondimu S, Murase H (2007) Reservoir level forecasting using neural networks: lake Naivasha. Biosyst Eng 96(1):135–138

Panapakidis IP, Dagoumas AS (2016) Day-ahead electricity price forecasting via the application of artificial neural network based models. Appl Energy. https://doi.org/10.1016/j.apenergy.2016.03.089

Piasecki A, Jurasz J, Skowron R (2017) Forecasting surface water level fluctuations of lake Serwy (Northeastern Poland) by artificial neural networks and multiple linear regression 2017. J Environ Eng Landsc Manag 25(4):379–388

Pineda FJ (1987) Generalization of back-propagation to recurrent neural networks. Phys Rev Lett 59(19):2229–2232. https://doi.org/10.1103/PhysRevLett.59.2229

Rani S, Parekh F (2014) Predicting reservoir water level using artificial neural network. International Journal of Innovative Research in Science Eng Technol 3(7):14489–14496

Ren S, He K, Girshick R, Sun J (2017) Faster r-CNN: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis and Machine Intelligence 39(6):1137–1149. https://doi.org/10.1109/TPAMI.2016.2577031

Rosenblatt F (1960) Perceptron simulation experiments. Proc IRE 48(3):301–309. https://doi.org/10.1109/JRPROC.1960.287598

Sak H, Senior A, Beaufays F (2014) Long short-term memory recurrent neural network architectures for large scale acoustic modeling. Interspeech 2014(September):338–342. arXiv:1402.1128

Seo Y, Kim S, Kisi O, Singh VP (2015) Daily water level forecasting using wavelet decomposition and artificial intelligence techniques. J Hydrol 520:224–43

Sethia A, Raut P (2019) Application of LSTM, GRU and ICA for stock price prediction. Springer, Singapore, pp 479–487. https://doi.org/10.1007/978-981-13-1747-7-46

Sutskever I, Vinyals O, Le QV (2014) Sequence to sequence learning with neural networks. Adv Neural Inf Proces Syst (NIPS) :3104–3112. https://doi.org/10.1007/s10107-014-0839-0

Taylor SJ, Letham B (2018) Forecasting at scale. Am Stat. https://doi.org/10.1080/00031305.2017.1380080

Yu D, Eversole A, Seltzer M, Yao K, Kuchaiev O, Zhang Y, Huang X (2014) An introduction to computational networks and the computational network toolkit. Microsoft Res

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.