





Chapter 7

Flood Forecasting Using Edge AI and LoRa Mesh Network



Mau-Luen Tham , Xin Hao Ng , Rong-Chuan Leong ,
and Yasunori Owada 

Abstract Remote flood forecasting has exponentially grown over the past decade together with the unprecedented expansion of Internet of Things (IoT) network. This is feasible with the use of long-range wireless communication technology such as LoRa. Ideally, each LoRa device shall process the sensor data locally and trigger warnings to the remote server based on prediction results. However, conventional prediction methods rely on highly computational artificial intelligence (AI) algorithms, which are not suitable for low-powered LoRa network. In this paper, the LoRa device is integrated with an edge AI model, which is based on long short-term memory (LSTM) neural network. OpenVINO is adopted to optimize the LSTM model before executing the solution on a Raspberry Pi 4 in combination with Intel Movidius Neural Computing Stick 2 (NCS2). Experimental results demonstrate the feasibility of deployment of the customized model on low-cost and power-efficient embedded hardware.

Keywords Edge AI · LSTM · Flood forecasting · LoRa Mesh Network · IoT

7.1 Introduction

Flood forecasting models have been researched in the hydrological engineering area for many years. Recently, there has been increased research interest in river flood prediction and modeling, defined as data-driven approaches. The artificial neural network (ANN) model is the most famous usual data-driven approach. Most

M.-L. Tham (✉) · X. H. Ng · R.-C. Leong
Department of Electrical and Electronic Engineering, Universiti Tunku Abdul Rahman, Kajang,
Malaysia
e-mail: thamml@utar.edu.my; lrongchuan@1utar.my

Y. Owada
Resilient ICT Research Center, National Institute of Information and Communications
Technology (NICT), Tokyo, Japan
e-mail: yowada@nict.go.jp

conventional statistical methods require a lot of data for their models, and they can generate no assumptions for both linear and nonlinear systems. Hence, the data-driven approach, ANN, is an alternative to hydrological flood forecasting instead of the existing methods [1].

Artificial intelligence (AI) has made essential development in modeling hydrological forecasting and dynamic hydrological issues. With the advancement of information technology, the application of ANN models in many aspects of science and engineering is increasingly becoming common due to its simplicity of structure. Diverse neural network modeling approaches have been applied, like implementing the model approaches individually or combining process-based approaches to minimize mistakes and increase the models' forecasting accuracy. The study in [2] applied AI model to forecast river flow for 15 years starting from 2000.

However, there are some limitations of the ANN model. One of them is lacking understanding of watershed processes. Furthermore, the limitation of memory in calculating sequential data exposes the disadvantages of the ANN model. The breakthrough in computational science has recently increased the interest in deep neural network (DNN) approaches. In addition, the most recent DNN applications, such as the long short-term memory (LSTM) [3] and gated recurrent unit (GRU) [4] neural networks, have been efficiently implemented in diverse areas and fields, such as time sequence problems. Those models can apply to machine translation, speech recognition, tourism field, language modeling, rainfall-runoff simulation, stock prediction, and river flow forecasting.

On 11th March 2011, around 29,000 cellular towers were damaged in the East Japan Great Earthquake. These damages have restricted the broadcast of evacuation notices and the collection of historical information for disaster forecasting. Hence, it can be known that the resilience of a network remains an open issue in the deployment of the fault-tolerant network during an emergency disaster. Fortunately, a disaster-resilient mesh-topological network called NerveNet was developed by Japan NICT. Each NerveNet node is independent and tolerant to system failure and link disconnection due to its mesh structure.

In this paper, a flood forecasting model is proposed. In the study area, rainfall and river water levels collected at hydrological stations serve as dataset for the training and testing process of the AI models. Then, the forecasted flood water level will be processed to generate the flood warning message. It will be sent through the NerveNet LoRa mesh network. Note that the proposed solution facilitates edge computing, which is one of goals of the ASEAN IVO project titled "Context-Aware Disaster Mitigation using Mobile Edge Computing and Wireless Mesh Network."

The rest of the paper is organized as follows. Section II discusses the related works. Section III describes the system architecture. Section IV presents the experimental results and discussions. Section V concludes the article.

7.2 Related Work

7.2.1 *Edge AI*

Several existing works [5, 6] explored the potential of edge AI for various applications. The authors in [5] focused on real-time apple detection with the implementation of YOLOv3-tiny algorithm on various embedded platforms. However, they did not consider the communication aspects. Recognizing the importance of LoRa, the authors in [6] proposed an edge AI in LoRa-based fall detection system with fog computing and LSTM. The processing burden is placed on a LoRa-based edge gateway, where the collected sensor information is transmitted from an edge node via Bluetooth Low Energy (BLE). Differently, our solution integrates both edge AI and LoRa functionalities into one single device, which simplifies the deployment effort.

7.2.2 *NerveNet*

NerveNet is a resilient network developed by Japan National Institute of Information and Communications Technology (NICT) [7]. NerveNet is a specially developed network for the regional area to provide reliable network access and a stable, resilient information-sharing platform in emergencies, even if the base station is destroyed in a disaster. The base stations of NerveNet are interconnected by the Ethernet-based wired or wireless transmission systems such as satellite, Wi-Fi, LoRa, and so on. They will form a mesh-topological network.

Nowadays, the current trend of the common network infrastructures uses the tree topology. As compared to it, NerveNet has the characteristic that it is more tolerant to the faults such as node failures, disconnections, destruction of the base station, and so on. Since the base station in the NerveNet supports basic services such as SIP proxy, DNS, and DHCP, the NerveNet can also continuously provide connectivity services to the devices.

7.3 System Architecture

7.3.1 *Dataset*

The dataset we employ is the Abashiri River watershed [8], located northeast of Hokkaido, Japan. The area of the watershed is around 1380 km². It has a 115 km long main river to the North Pacific and a range of elevation from 0 to 978 m [9]. All AI models are trained and tested using the datasets observed at the downstream

Table 7.1 Training and testing period for the dataset

Dataset	Training	Test
Hongou (Jan 2019 to Dec 2020)	Jan 2019 to May 2020	Jun 2020 to Dec 2020

Table 7.2 Hyperparameter settings for LSTM model

Hyperparameter	Value
Sequence length	24
Optimization algorithm	Root mean squared propagation
Epoch	50
Batch size	64

stations called “Hongou.” The used datasets are hourly datasets with the water level and rainfall variables from first January 2019 to 31st December 2020.

During data preprocessing, the rainfall and water level data undergo a train-test split, separated into 70% of the data as training dataset and 30% as a testing dataset, as listed in Table 7.1. The training data calculates the training process error and estimates the AI models’ parameters. The testing data provides an independent performance evaluation of the AI models after training.

Next, the hydrological dataset has also undergone data standardization where the values’ distribution is rescaled to a mean value of 0 and a standard deviation value of 1. Data scaling is essential to fasten the training process of the AI model because the AI models can converge more rapidly if the dataset features are closer to the normal distribution. Prior to the AI model training, the time series dataset is converted into sequential data with 24-time steps as the sequence length. The model performs equally well when the sequence length is between 5 and 15 or more. Therefore, in this paper, the sequence length value of 24 is used in the model to represent 24 h in 1 day.

7.3.2 AI Model Training in Google Colab

In this paper, four types of AI models, namely, Random Forest, SVM, LSTM, and GRU, are trained and tested on the dataset to benchmark the performance of the system in terms of flood water level forecasting. Trained in in Google Colab platform, the best AI model will be selected as the edge AI.

For Random Forest, the parameter “max_depth” represents each tree’s depth in the forest. Here, we set the max_depth value to 2. There are several hyperparameters in the LSTM model-building process. Firstly, the optimization algorithm is the stochastic gradient descent procedure’s extension to update the weights iterative of the network according to the training dataset. Secondly, an epoch is defined as the whole dataset transferring forward and backward across the model’s neural network once. Thirdly, the batch size is the number of samples propagating throughout the entire neural network. Table 7.2 demonstrates the hyperparameter settings of the

LSTM model. For fair comparison, the same hyperparameters are adopted to train the GRU models.

7.3.3 AI Model Optimization Using OpenVINO

The immediate output format of the LSTM model is .h5, which will be converted to pb format. The intention is to utilize the OpenVINO toolkit [10], which enables the faster running of the AI model in edge device. There are two main components in the OpenVINO toolkit, which are the model optimizer and inference engine. Firstly, when the trained model in pb format is fed into the model optimizer, it converts them to the IR format. At the same time, it optimizes the performance, space, and hardware-agnostic with conservative topology transformations. The outputs of the model optimizer are .xml and .bin.

Secondly, the AI inferencing process is performed at the edge device by setting the inference engine to Intel Neural Compute Stick 2 (NCS2), which is a hardware accelerator. Before feeding to the inference engine, the data is scaled using the scaler.gz exported from the training process. The scaled data is then reframed. The historical time series data representing the last 24 h is extracted from the scaled dataset by retrieving the top 24 values of the rainfall and water level data. After that, the sequence data and the trained model in IR format are fed into the inference engine to generate the water levels ahead of 1 hour in text form and the result graph in image form.

7.3.4 Evaluation Metrics

The mean absolute error (MAE) is the mean of the differences between the original value with the forecasted value. On an excellent flood forecast, the MAE should be smaller. Mathematically, it can be expressed as:

$$\text{MAE} = \frac{1}{n} \sum_{i=1}^n |e_i| \quad (7.1)$$

The mean absolute percentage error (MAPE) is the percentage of the mean of the total error. On an excellent flood forecast, the MAPE should be smaller. It is written as:

$$\text{MAPE} = \frac{1}{n} \sum_{i=1}^n \left| \frac{e_i}{y_i} \right| \times 100 \quad (7.2)$$

The root mean squared error (RMSE) is the square root of the mean of the squared deviation of the forecasted flood water level value. On an excellent flood forecast, the RMSE should be smaller. It is written as:

$$\text{RMSE} = \sqrt{\frac{1}{n} \sum_{i=1}^n e_i^2} \quad (7.3)$$

R^2 is the coefficient of determination and goodness of fit. With an excellent flood forecast, the R^2 should be larger.

$$R^2 = 1 - \frac{\text{sum squared regression (SSR)}}{\text{total sum of squares (SST)}} \quad (7.4)$$

The NerveNet LoRa data transmission performance is evaluated by calculating the packet delivery ratio (PDR) of LoRa packets.

$$\text{PDR} = \frac{\text{number of packets received}}{\text{number of packets sent}} \quad (7.5)$$

7.4 Results and Discussions

Table 7.3 compares the water level forecasting performance of the aforementioned five AI model types on the testing dataset. Theoretically, the deep learning methods outperform the conventional machine learning methods when the big data comes into its input. This is consistent with the result, where the LSTM and GRU models have a lower value of MAE, MAPE, and RMSE than the random forest and SVM models. This indicates that the deep learning models have a lower deviation of the forecasted results from the ground truth and a lower error percentage. A higher R^2 value indicates a more excellent time series forecasting performance from the deep learning models.

From the table, it can be observed that the LSTM model has more excellent performance than the GRU model, since it has lower MAE, MAPE, RMSE, and

Table 7.3 Benchmarking performance for prediction

AI model	MAE	RMSE	MAPE	R^2
Random forest	0.0656	0.078	0.0972	0.7807
SVM	0.0541	0.0632	0.0763	0.8562
GRU	0.0138	0.0154	0.0217	0.9915
LSTM (Keras)	0.0088	0.0092	0.0126	0.997
LSTM (OpenVINO)	0.0593	0.0907	0.0899	0.704

higher R^2 . This finding is consistent with the findings in [11], where the LSTM model performs better than the GRU model in the case of short text processing and large-size datasets. In this paper, there is a huge amount of rainfall and water level dataset where both types of variables are short integers. They act as the inputs to the LSTM and GRU models. Therefore, it can be seen that the LSTM is more appropriate than the GRU models in these scenarios.

All in all, the LSTM has the best performance in the AI water level forecasting, since it has the lowest MAE, MAPE, and RMSE while the highest R^2 among all the proposed AI models. Therefore, LSTM is chosen as the AI water level forecasting model. Specifically, OpenVINO is used to convert the .h5 model to .xml and .bin format. It can be seen that there is a performance degradation of the converted model in all aspects.

Figure 7.1a displays the prediction versus ground truth for test dataset by using LSTM variations. As expected, the prediction using Keras model is close to the actual values. To reveal more insights, Fig. 7.1b compares the inference time between these two LSTM models. It can be seen that the LSTM (OpenVINO) is 28× slower than the Keras version. The reason is that the Keras model was using the Intel® Xeon® CPU at 2.20Ghz provided by the Google Colab. This hardware has more computational power than the NCS2, which consumes only around 1.5 W.

Figure 7.2 shows the actual deployment of LoRa nodes. For the LoRa parameters, we adopted spreading factor of 12, transmission power of 20 mW, and bandwidth of 500 kHz. Three NerveNet LoRa nodes serve as MQTT subscriber, whereas one NerveNet LoRa node acts as MQTT publisher. The publisher publishes the MQTT message at three different locations. At each location, a total of 11 LoRaMesh packets are transmitted. The quality of service (QoS) level is set to zero, which guarantees best-effort message delivery. In other words, the publisher only transmits each packet once, and LoRa message packets may be lost during the transmission process. Node 208 is located inside the building in such a way that nodes 203 and 204 can act as relay node. We implement subscriber and publisher nodes using Intel next unit computing (NUC) and Raspberry Pi 4, respectively. The latter is chosen due to its high portability and low cost, which is suitable for massive deployment of flood monitoring.

Figure 7.3 depicts the overall performance of NerveNet LoRaMesh. It can be observed from Fig. 7.3a that only extra hops are needed at location 3. This is reasonable since the distance between 204/208 and location 3 is at least 1200 m. In this case, node 203 which is closer to location 3 acts as relay node. For LoRaMesh packet to arrive at node 208, the packet initially sent by node 214 at location 3 is passed to 203, through 204 to 208. For other two locations, only one hop transmission is needed. This is because there are less obstacles, such as trees and buildings. The multi-hop transmission is affected by the received signal strength indicator (RSSI), as reported in Fig. 7.3b. All RSSI values are measured with respect to the publisher node 214, except the last two columns. Specially, 204 and 208 measurements are based on their relay nodes 203 and 204, respectively.

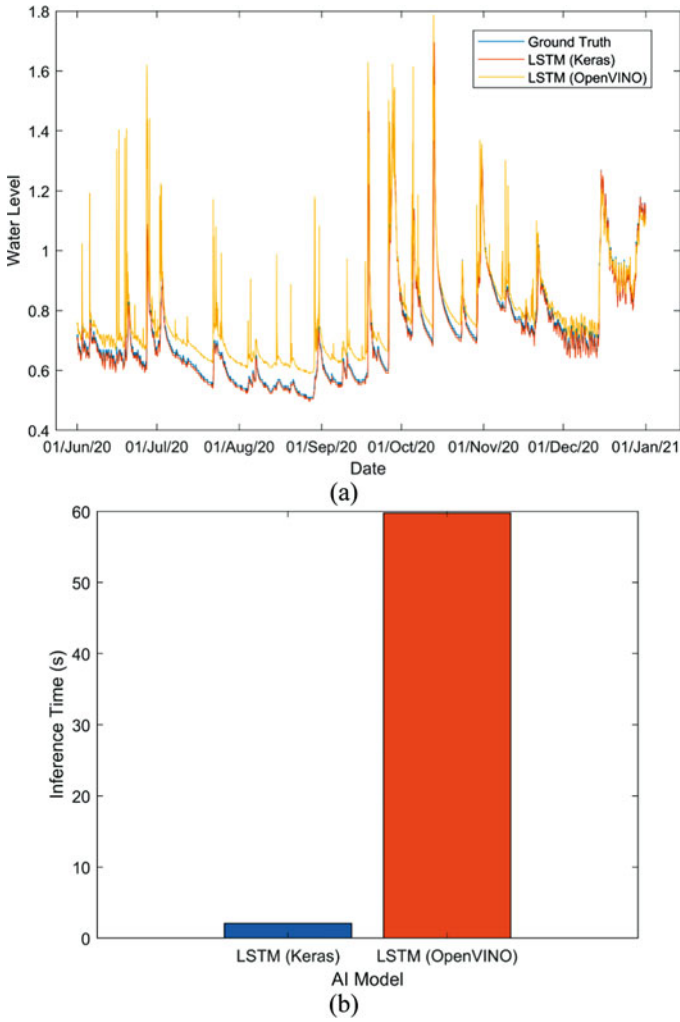


Fig. 7.1 LSTM performance benchmarking. (a) Prediction vs. ground truth. (b) Inference time

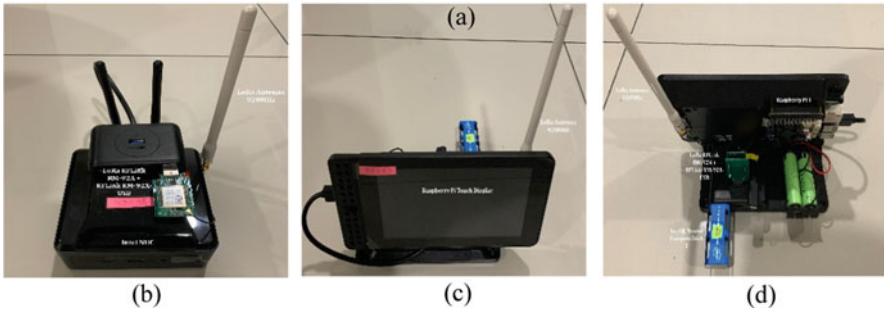


Fig. 7.2 System deployment. (a) The location of three subscriber nodes (203, 204, and 208) and one publisher node (214). (b) Subscriber node (Intel NUC). (c) Publisher node (front view). (d) Publisher node (rear view)

As shown in Fig. 7.3c, all LoRaMesh packets are received when the publisher transmits messages at locations 1 and 2. For location 3, 2 out of 11 packets are lost during the transmission for nodes 204 and 208. Specifically, when node 204 does not receive the packets from 203, it could not forward them to 208. Figure 7.3d compares the time on air. In LoRaMesh, time on air defines the elapsed time on air for a LoRaMesh packet between publisher and subscriber. As expected, the further the distance, the longer time needed to transmit the LoRaMesh packets.

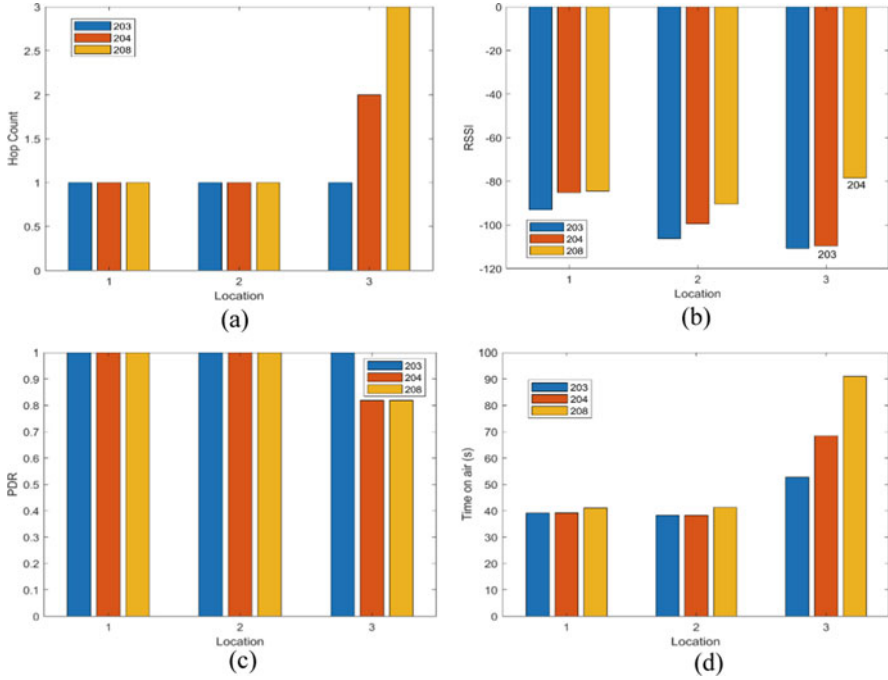


Fig. 7.3 Performance of NerveNet LoRaMesh. (a) Hop count. (b) RSSI. (c) PDR. (d) Time on air

7.5 Conclusion

In this paper, we have proposed an edge AI solution that forecasts flood water level and transmits the packet via LoRa mesh network. The AI model training and the testing dataset are obtained from Japan’s organization. Hence, the AI results may not apply to the local area since the weather, season, humidity, and geographical condition of Malaysia are different from Japan. The local dataset can be requested from the local government to build an AI model that can fit the situation in Malaysia’s local area so that a better understanding of the feasibility of the AI model in disaster detection in Malaysia.

Acknowledgments This work is the output of the ASEAN IVO (http://www.nict.go.jp/en/asean_ivo/index.html) project titled “Context-Aware Disaster Mitigation using Mobile Edge Computing and Wireless Mesh Network” and financially supported by NICT (<http://www.nict.go.jp/en/index.html>).

References

1. O.A. Kisi, A combined generalized regression neural network wavelet model for monthly streamflow prediction. *KSCE J. Civ. Eng.* **15**, 1469–1479 (2011)
2. Z.M. Yaseen et al., Artificial intelligence based models for stream-flow forecasting: 2000–2015. *J. Hydrol.* **530**, 829–844 (2015)
3. S. Hochreiter, J. Schmidhuber, Long short-term memory. *Neural Comput.* **9**(8), 1735–1780 (1997)
4. K. Cho, B.V. Merriënboer, D. Bahdanau, Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches, in *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Association for Computational Linguistics, Doha, Qatar (2014), pp. 103–111
5. V. Mazzia, A. Khaliq, F. Salvetti, M. Chiaberge, Real-time apple detection system using embedded systems with hardware accelerators: An edge ai application. *IEEE Access* **8**, 9102–9114 (2020)
6. J.P. Queralta, T.N. Gia, H. Tenhunen, T. Westerlund. Edge-AI in LoRa-based health monitoring: Fall detection system with fog computing and LSTM recurrent neural networks, in *Proceedings of 2019 42nd International Conference on Telecommunications and Signal Processing (TSP)*. IEEE, Budapest, Hungary (2019), pp. 601–604
7. M. Inoue, Y. Owada, NerveNet architecture and its pilot test in Shirahama for resilient social infrastructure. *IEICE Trans. Commun.* **100**(9), 1526–1537 (2017)
8. Ministry of Land, Infrastructure, Transport, and Tourism in Japan (MLIT Japan). Hydrology and Water Quality Database, <http://www1.river.go.jp/>. Last accessed 30 July 2022
9. N. Kimura et al., Convolutional neural network coupled with a transfer-learning approach for time-series flood predictions. *Water* **12**(96) (2020)
10. OpenVINO Toolkit. <https://software.intel.com/enus/openvino toolkit>. Last accessed 01 July 2022
11. S. Yang, X. Yu, Y. Zhou. Lstm and gru neural network performance comparison study: Taking yelp review dataset as an example, in *Proceedings of 2020 International Workshop on Electronic Communication and Artificial Intelligence (IWEC AI)*. IEEE, Shanghai, China (2020), pp. 98–101