

Intelligent flood disaster warning on the fly: developing IoT-based management platform and using 2-class neural network to predict flood status

Salami Ifedapo Abdullahi, Mohamed Hadi Habaebi, Noreha Abd Malik

Department of Electrical and Computer Engineering, International Islamic University Malaysia (IIUM), Malaysia

Article Info

Article history:

Received Jan 7, 2019

Revised Feb 25, 2019

Accepted Mar 30, 2019

Keywords:

2-class Neural Network
Artificial Neural Network
Azure Machine Learning
Azure Web Service
Flood Forecasting
Flood Monitoring System
Internet of Things (IoT)
NodeMCU (ESP8266)
Thingspeak

ABSTRACT

The number of natural disasters occurring yearly is increasing at an alarming rate which has caused a great concern over the well-being of human lives and economy sustenance. The rainfall pattern has also been affected and this has caused immense amount of flood cases in recent times. Flood disasters are damaging to economy and human lives. Yearly, millions of people are affected by floods in Asia alone. This has brought the attention of the government to develop a flood forecasting method to reduce flood casualties. In this article, a flood mitigation method will be evaluated which incorporates a miniaturized flow, water level sensor and pressure gauge. The data from the two sensors are used to predict flood status using a 2-class neural network. Real-time monitoring of the data from the sensor into Thingspeak channel were possible with the use of NodeMCU ESP8266. Furthermore, Microsoft's Azure Machine Learning (AzureML) has built-in 2-class neural network which was used to predict flood status according to predefined rule. The prediction model has been published as Web services through AzureML service and it enables prediction as new data are available. The experimental result showed that using 3 hidden layers has the highest accuracy of 98.9% and precision of 100% when 2-class neural network is used.

Copyright © 2019 Institute of Advanced Engineering and Science.
All rights reserved.

Corresponding Author:

Mohamed Hadi Habaebi,
Department of Electrical and Computer Engineering,
International Islamic University Malaysia (IIUM),
Jalan Gombak, Kuala Lumpur, 53100, Malaysia.
Email: habaebi@iium.edu.my

1. INTRODUCTION

This article is an extension of the work presented during the ICCCE'18 Conference in September 2018. This article provides more comprehensive details on the previous work as the study developed further by advancing the flood prediction system using Azure Web Service. One of the methods to reduce the damage caused by flood is flood forecasting and it can predict the proneness of flood 6 hours to 2 days in advance [1]. Terengganu, Pahang and Kelantan are one of the first Malaysian states that flood forecasting are being actively implemented in flood prone areas [1]. Terengganu, Pahang and Kelantan are attacked by flood yearly during monsoon seasons and the aftermath of the flood last normally until the consecutive incoming monsoonal season [2]. Artificial neural network (ANN), weather radar image or hydrological flood mapping needs up to date historical data readily available for an effective flood forecasting system to be effective. Flood mapping technique to map flood prone areas was used in [3] and radar image with the inclusion of numerical weather prediction technique were adopted in [2,4] to implement flood forecasting. Machine learning is one of the branches of ANN and it can be useful in predictive measure involved in flood forecasting.

One of the few ways to attain data are through open data and sensor data readings which have been carried out by most researchers in [5, 6, 11, 14, 21]. The deployment of an embedded system which adopts IoT and Neural network has been critically analysed in [6]. A flow rate sensor and a water level sensor were utilized in the process of data acquisition while prediction model was done to make a system that can predict flood occurrence. Researchers have used statistical means to analyse flood data in [5, 7] while the use of ANN and other hybrid system integrating fuzzy system and neural network have been heavily explored in [8-14]. The use of Boosted regression tree for analysis was carried out in [8] to analyze reservoir inflow. Wavelet Analysis was used to clean the data from unwanted errors that might exist in the data from the reservoir inflow. Boosted regression tree is a neural network model which reduces error in the previous tree network by limiting the mean difference of the data under test [7]. [8] used a neuro-fuzzy inference (ANFIS) hybrid system to modelled rainfall data and the system integrated both fuzzy and neural network. Classification of inputs into member functions is carried out by fuzzy systems and the output can be used to make decision and control purposes. In the case of using neural network in ANFIS, the inputs of the member functions are tuned which reduced processing time and boost performance [15].

Data set derived from sensor implanted on dikes were modelled using neural cloud (NC) classification as well as k-means clustering algorithm to evaluate the dike susceptibility [10]. In [10], Cluster of training data are divided according to their mean hierarchy and used to predict future observations depending on the k-means clustering settings. However, in [11] Neural Network Autoregressive with Exogenous Input (NNARX) was used on an acquired data set for a neural network model. NNARX uses time series modelling that relates present input values in a time series to both previous value of the same series along with current and past values of the exogenous series. The series which externally affects the input series is called the exogenous series. [12] adopted multiple linear regression, Dvorak technique and adaptive neuro-fuzzy inference system to compute meteorological data which includes pressure, temperature, humidity which went through training process to find the best correlation for configured system. This approach is mandatory in order to find the best alternative to train system to be effective in detecting annual monsoonal floods. [13] accessed satellite data in order to reduce attenuation caused by rainfall. Additionally, this gave an insight in the means that flood prediction data can be useful in aiding link reliability in Earth-space satellite communication.

In the other paper, the combination of Neural network regression, Bayesian linear regression, Boosted Decision Tree and Decision Forest algorithm to classify flood data from upstream and downstream of river was carried out in [14]. [14] also used linear regression as the control experiment as well as neural network regression modelling for the application of supervised learning with adaptive weights to accustomed non-linear function of the inputs. Bayes theorem is mainly used in Bayesian linear regression as it uses probability distribution function to find the best possible linkage for linear regression [14]. Finally, [14] used boosted decision tree and decision forest as they both are similar in their data processing technique. They earlier used reduction algorithm as it decreases the gap between mean of data set and trained data while the later fit a large number of decision algorithm on a sample data set and used their mean to make an efficient prediction [14].

To make an effective flood forecasting, real-time monitoring to aid early warning is a must in any flood monitoring system. Several research articles have designed and integrate monitoring of sensor data to flood monitoring system and one of them is [16] which used a GSM module. Data from sensors are transmitted to server via GPRS network in a GSM module. TCP communication is necessary to notify the receiving end of incoming data and then the server sends alerts to the numbers stored in the database and at the same time the computer will give information to the user interface. [17] tested the effectiveness of Long Range (LoRa) Low Power Wide Area Network (LPWAN) in increasing quality of communication between nodes in an Internet of Things (IoT) ecosystem. LPWAN is beneficial for Monitoring systems has it benefits in long lasting real-time monitoring with less power usage between nodes and sensors.

Research in [18] extensively accessed the Medium Access Control (MAC) layer involved in an IoT system. The interoperability of the protocol layer was access along with the means to reduce energy consumption. [18] efforts in reducing the protocol overheads and energy consumption would greatly help IoT implementations for monitoring systems. Datalog V1 is used to send sensor data to a Web Interface in [19]. GPRS server enables Supervisory Control and Data Acquisition System (SCADA) to use UDP/IP protocol to allow the inflow of data gathered from sensor. On-site flood prediction was being monitored by a dedicated web service in [8, 10]. Azure Web service was used in [8] for prediction of future reservoir out-flow using data recorded from sensors and [10] used Common Information Space (CIS) to monitor flood prone areas embedded on a neural cloud.

In a monitoring system, continuous measurement of data is very important. However, the sensors play a critical role in making a robust system that does not breakdown at important moments of a monitoring system. Therefore, [20] did an extensive test on the reliability and efficiency of a fabricated printed circuit

board water level sensor. The electrode sensor is very useful in a system that requires a continuous measurement of water level with the usage of less power. Water level sensors are crucial in a flood monitoring system and thus [20] had tested the electrode water level sensor in a flood monitoring system and the results are very encouraging. Furthermore, another important sensor would be a flow sensor. Flow sensors are very important in a way that they keep constant monitoring of water flow either to a dam or in a canal. [22] explicitly discussed the criteria needed to design a flow sensor with a simulation to guide the mathematical theory. The flow sensor was 3-D printed in [22] and the performance of the sensor was tested using two different means of software enablers. This showed that flow rate sensors are crucial for a flood prediction system. [21] created Save Water Overflow Detection (SWOD) web service which utilized an ESP8266 WiFi chip for communication control.

This paper developed a state-of-the-art flood monitoring system which integrated a miniaturized flow and water level sensor. The data from the sensor are stored in real time in a webservice called Thingspeak. The hardware enabler in this development is NodeMCU ESP8266 and it enabled transmission of sensor data to Thingspeak channel that comes with interactive charts and graphs. Azure Machine Learning (Azure ML) built in package of two-class neural network module was used to predict severity of flood from the flow rate and water level readings. Then, the system is implemented as Web services on Azure ML, which enables predictions as new data are obtained. The next section includes the research methodology, followed by result analysis section and then the conclusion section.

2. RESEARCH METHOD

2.1. System design principles

The experimental procedure and set-up are outlined in this section. Hydraulic pressure gauge meter that has 0 to 200PSI with a 1/8-inch National Pipe Thread (NPT) was used to measure the in-flow pressure of water from a single-hole faucet tap that has mouth opening diameter of 1.7 cm. The braided polymer faucet adapter used is 30cm in length, 1.4 cm in diameter and has a compression connector thread of 3/8-inch. The braided polymer faucet adapter connector was carefully installed on the opening of the tap and at the other end the pressure gauge meter was attached together with the female compression thread connector of the braided polymer faucet adapter. The water pressure coming out from the tap opening pressure gauge meter was measured before it flows into the inlet of the reservoir as shown in Figure 1. The transparent plastic container is referred to as the reservoir. The reservoir has diameter of 25 cm, height of 18cm and a capacity of 5.8 litre. The flow sensor is attached to the Polyvinyl chloride (PVC) pipe which has height of about 20 cm and diameter of 2.8 cm. the flow rate measures the flow rate of the inflowing water. The water level sensor has 5 float switch sensors with 5 cm spacing interval between each sensor and it was attached on a PVC pipe which diameter of about 2.8 cm and length of 34 cm. the water level sensor detects the level height of the water in the reservoir as shown in Figure 1 [6].



Figure 1. Emulated laboratory set up [6]

The mentioned processes and sensors can be seen in Figure 1. The functional programming code was written such that the water level measured in centimetres is output in meters using a multiplicative of 10. The flow sensor gets the reading of the flow rate passing through the inlet of the flow sensor using the expression in (1). The water level sensor gets the reading of the water level in the reservoir using the expression in (2).

$$Q = V \times A \quad (1)$$

where Q is the volume flow, V is the velocity and A is the cross-sectional area.

$$\text{water level} = \text{sensor_state} \times 0.5 \quad (2)$$

where water level is in metres, sensor_state is from the first flow switch sensor at the bottom of the PVC pipe to the fifth float switch sensor at the top of the PVC pipe.

2.2. Monitoring system architecture

To practicalized Internet of Things (IoT), emerging hardware enablers like NodeMCU integrates both a microcontroller and a WiFi module (ESP8266) in a single device to enhance the capability of transmitting sensory data online [21]. The Architecture of the NodeMCU is illustrated in Figure 2(a). Multiple sensor interfaces such as Digital inputs, Analog input, UART interface are available on NodeMCU board which enable them to take quite a number of measurement from sensors and actuators. A Printed Circuit Board (PCB) antenna placed on top of the NodeMCU board aids in long distance communication even with a low signal strength WIFI. The NodeMCU supports IEEE802.11b/g/n with frequency in the range between 2.4 to 2.5GHz. The ESP8266 microprocessor has WIFI module and STM8S003 microcontroller that is mounted in a certified metal shield approved by Federal Communication Commission (FCC) against strong interference from and to other devices. Micro-USB and USB conversion is possible with a USB to UART Bridge implanted on top of the NodeMCU while sensors running on 3.3 volts can be connected to the NodeMCU as the voltage regulator is used to step down 5 volts input from the power source to 3.3 volts from the sensors. Powering up of two sensors which needs input voltage would not be a problem as the NodeMCU packs two voltage output pins. Furthermore, a pull up resistor value of 4.7kilo-ohms is necessary in between the connection of the NodeMCU and float switch level sensors to enable bias of the ESP8266 digital pin state to be able to differentiate between voltage up and down of the float switch sensor as shown in Figure 2(b). The flow sensor can only be connected to the analog input of the ESP8266 to detect the water inflow rate from the tap. After all this procedure, Hypertext Transfer Protocol (HTTP) enables sensory data to be sent and received via Representational State Transfer (REST) API [23]. REST API uses calls which are GET, POST and PUT to create channel, read and write data in Thingspeak channel. The server acts as the gateway which Thingspeak sends a request to and get response with the requested original data. The miniaturized flow and water level sensor enabled by NodeMCU node highlighted in red colour circle and displayed in Figure 2(c) before deployment.

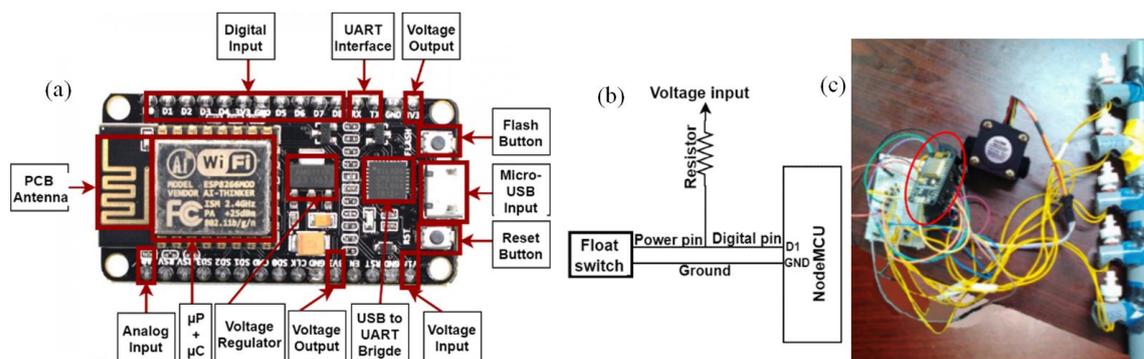


Figure 2. (a) NodeMCU architecture, (b) Pull-up resistor connection, (c) miniaturized sensor node [6]

2.3. Data processing using artificial neural network

The use of advanced statistical methods to improve the learning and performance of specific input data source is known as Machine Learning [25]. Microsoft Azure ML Studio is a web service with an intuitive drag and drop user interface that helps analyse data through readily available modules [24]. Nodes like structures that connects neural algorithm with data transformation routes are known as modules as they also contain saved models and user define codes [24]. The laboratory experimental data were interpolated into Azure ML to conduct neural network modelling. Before data are uploaded into Azure ML datasets, the file type must be changed into CSV file format. 2-class neural network is used for training data in Azure ML. The neural network flowgraph was programmed in Azure ML and the modules connection

resembles a cyclical graph that is similar to a tree. The parameters that can be manipulated to achieve desire outcome are number of hidden nodes, size of learning rate and number of learning iterations. Visualization of the trained model performance can be accessed inside Azure ML evaluation model module. Table 1 provides information on the rules followed for the flood status and prediction guidelines. OR logic gate is an important guideline to provide a basic system for the classification of flood prediction and a logic output of 0 is classified as safe while a logic output of 1 is classified as unsafe. OR logic table is showed in Table 2. The system workflow diagram is presented in Figure 3.

Table 1. Rules of flood status prediction

Status Category	Flow Rate	Water Level
Safe	$\leq 8\text{L/min}$	$\leq 1.5\text{m}$
Unsafe	8.1-12L/min	1.6-2.5m

Table 2. OR logic table

Input 1	Input 2	Output
0	0	0
0	1	1
1	0	1
1	1	1

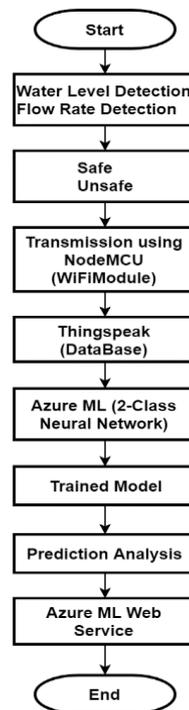


Figure 3. System workflow diagram

2.4. Deployment of web service

Deploying an Azure Web Service is one of the critical aspects of building a flood forecasting system. In this article, the Web Service that was deployed in this system is a continuation from the Azure Machine Learning work station that the 2-class network was modelled. To begin with, the train model, evaluate model and unused score model is removed from the flow graph [24]. Then the conversion of the training experiment will need to be changed to predictive experiment. The defined places where the web service will accept and generate input would have to be known as well. This step can be accomplished by clicking set up web service at the bottom of the experiment user interface which is then followed by pressing on the Predictive Web Service option. After this step, the trained model is converted to a single Trained model module and stored in the left section module of the experiment user interface [24]. The flow graph is

further edited and the remaining parts which includes the web service input, metadata, score model and web service output are used for the necessary prediction algorithm. Then, the web service can be deployed based on Azure Resource Manager [24]. The display name is entered and the deploy button is clicked. The web service is then tested and ready to be used for flood forecasting.

2.5. Performance of ANN classification model

Connectionist system is part of an architecture for supervised learning that comprises of intertwined networks called nodes. ANN goes through supervised learning provided there are exemplary inputs which includes correct prediction or classification for the data that is to be trained [26]. ANN uses feedforward neural network in which connection between nodes includes a layer of input units, varying layer of hidden units and one output unit layer [26]. During training, feedforward neural network propagates in backward direction with an adjusted weight that reduces the difference between original output and desired output while feedforward neural network learns in forward direction at normal period. Figure 4 showed a general model of a feedforward neural network. To begin with neural network learning, feedforward propagation starts by multiplying the input values (x_j), j moving from 1 to p input units, carrying together their bias unit (x_0) which corresponds to their respective weights (w_0) [19]. The input layer values (x_j) is the output product along with the weights (w_{kj}) which are analysed and taken in to the non-bias units in the hidden layer (z_k), where k is moving from 1 to K hidden units, as expressed in (3) [26].

$$z_h = \sum_{j=0}^p w_{kj}x_j \tag{3}$$

(4) expresses the general form of sigmoid function which is an activation function for expression below.

$$sigmoid(a) = \frac{1}{1+e^{-a}} \tag{4}$$

Expression (5) below is the sigmoid function which squashes the sum of their input values to a value close to either 0 or 1.

$$z_h = sigmoid(\sum_{j=0}^p w_{kj}x_j) = \frac{1}{1+e^{-\sum_{j=0}^p w_{kj}x_j}} \tag{5}$$

The outputs of individual hidden layer (z_k) unit which encompasses the bias unit (z_0) multiplied by another set of respective weights (v_{ik}) and processed out to the output unit (i) [26]. Finally, the output unit (i) is analyse with the weighted sum of its input values expressed in (6).

$$o_i = \sum_{k=0}^K v_{ik}z_k \tag{6}$$

Then expression (6) is subjected to the same sigmoid function in (4) to produce a predicted output (y), expressed in (7) below which falls between 0 and 1 [26].

$$y = sigmoid(o) = sigmoid(\sum_{k=0}^K v_kz_k) = \frac{1}{1+e^{-\sum_{k=0}^K v_kz_k}} \tag{7}$$

The network is trained with backpropagation algorithm and the use of gradient descent is employed to refresh the weights which reduces the squared error between the network output values and desired output values [19].

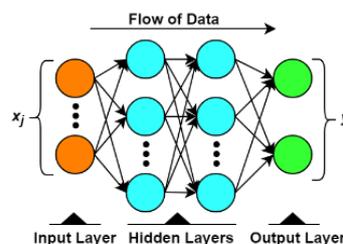


Figure 4. General model of a feedforward neural network [6]

3. RESULTS AND ANALYSIS

3.1. Laboratory experimental result

Four different pressure values were measured which are 30, 40, 50 and 55 pounds force per square inch (psi) while the inflow rate which is in litre per minute (L/min) were recorded for the different pressure values mentioned above. The laboratory experiment was able to gather 367 observed readings. The measured pressure data, flow data and water level data were tabulated in Table 3. Figure 5 showed the plot of the flow rate against water level alongside the measured pressure. Average flow rate for 30 psi, 40 psi, 50 psi and 55 psi were 4.93 L/min, 9.12 L/min, 11.17 L/min and 11.19 L/min accordingly. Through the observations, it can be said that as the pressure increases, the flow rate also increases. Coefficient of adjusted multiple correlation was tested using Minitab 18 and 36.75% correlation factor was realised which means that the data are significant enough to relate to each other. The possibility of the model being able to predict new observations is known as coefficient of predicted multiple correlation which also showed that the data gathered is not as complicated as it is depicted. However, the coefficient of predicted multiple correlation is 34.12% for the data gathered from the experiment.

Table 3. Sample of recorded data [6]

30 psi		40 psi		50 psi		55 psi	
Flow Rate (L/min)	Water Level (m)						
4.6	0.0	8.22	0.00	10.5	0.0	11.2	0.0
5.1	0.0	8.00	0.17	11.1	0.5	8.7	0.5
5.1	0.1	8.11	0.33	11.1	0.6	10.6	0.6
5.0	0.1	8.00	0.50	11.0	0.7	10.2	0.8
5.1	0.1	8.22	0.53	11.0	0.8	10.6	0.9
5.1	0.1	8.33	0.55	10.9	0.9	10.9	1.0
5.1	0.2	8.56	0.58	10.9	1.0	11.1	1.2
5.1	0.2	8.55	0.60	10.8	1.1	8.6	1.3
5.1	0.2	8.55	0.63	10.7	1.1	11.4	1.4
5.0	0.2	8.78	0.65	10.7	1.2	11.8	1.5
5.0	0.3	9.11	0.68	10.7	1.2	11.7	1.6
4.9	0.3	9.66	0.70	10.9	1.3	11.6	1.7
5.0	0.3	8.77	0.73	11.1	1.3	11.8	1.8

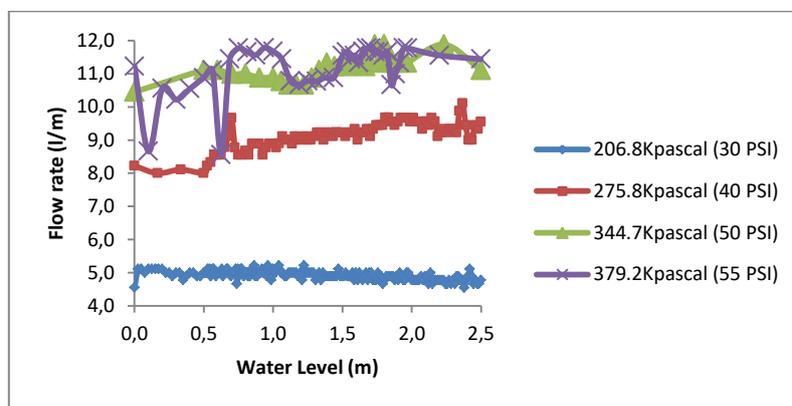


Figure 5. Multiple plot of varying flow rate vs water level in respect to inflow pressure [6]

3.2. Monitoring system

Sensor reading were transmitted into Thingspeak channel and each sensor has its own designated recording chart box for every 15 seconds interval. API key from Thingspeak needs to be input into the functional programming of the NodeMCU to enable communication between Thingspeak and the web applet. Mobile application of the chart displayed in Thingspeak through web can be achieved by viewing ThingView charts available in Android platform smartphones. ThingView needs the channel ID of the web Thingspeak channel for on the go flood monitoring status updates. Azure ML requires the CSV file format of the data available in web Thingspeak for 2-class neural network modelling. Thingspeak provides an interactive graph of the flood status, flow rate and water level. The available fields in Thingspeak channel is shown in Figure 6.

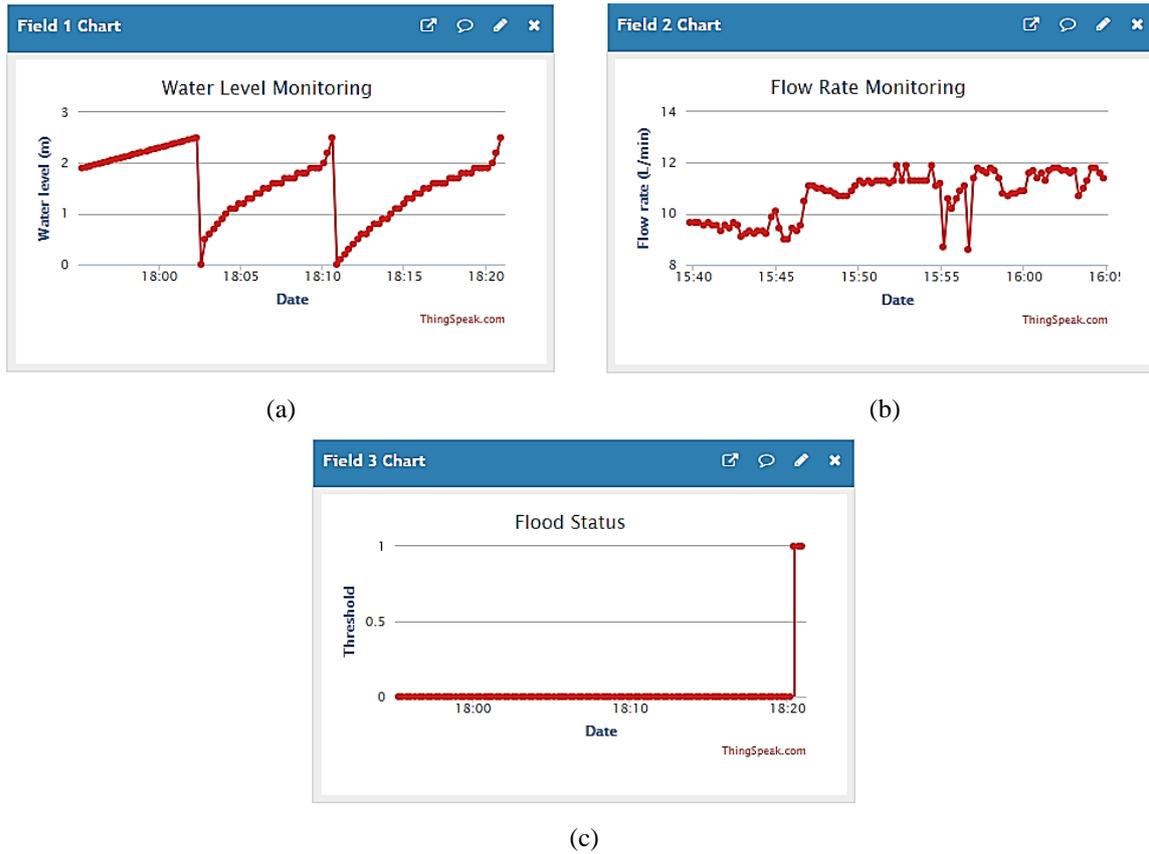


Figure 6. (a) Water level monitoring, (b) Flow rate monitoring, (c) Flood status

3.3. ANN prediction model analysis and web service interface

Figure 7 shows the 2-Class Neural Network model associated with Azure ML Studio. Azure ML links together the data workflow by joining modules as depicted in Figure 7. The CSV file block “Flood Prediction.csv” was the data taken from the Thingspeak database while the block “Select Column in Dataset” was used to select column used for model prediction. The block “Split Data” was used to split the data that will be used in connection for training and testing purposes. The block “Two-Class Neural Network” was where the number of hidden nodes were set at 100 and the number of hidden layers were maintained at 1, 2 and 3. The number of learning rate was pre-selected to be 100 and at the same time, the learning rate was fixated at 0.1 as it is most appropriate value to train the model. In order to achieve highest possible performance, 0.1 is chosen as the initial learning rate diameter while the value 0 was maintained for the momentum. The block “Train Model” is involved with the training of the dataset whereas the block “Score Model” carried out the testing of the model. The prediction model performance done by the block “Evaluation Model” and it shows the overall neural network analysis.

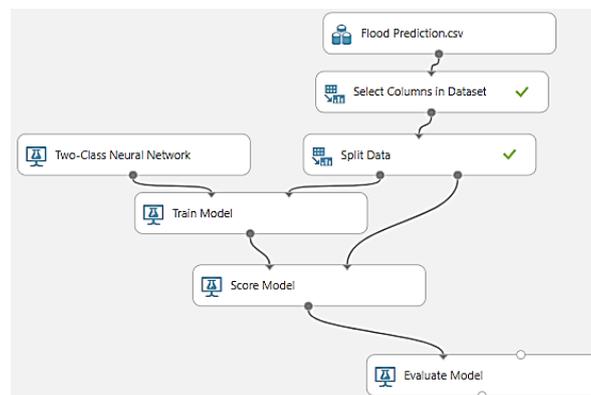


Figure 7. Flood status prediction using Azure ML workflow

The neural network results were analysed using classifiers which are accuracy, precision and recall rate. Expression (6) which showed how accuracy calculated is a ratio of correctly predicted observation to the total observation.

$$Accuracy = \frac{(TP+TN)}{(TP+FP+TN+FN)} \quad (6)$$

Expression (7) which showed the precision rate equation relating the ratio of correctly predicted positive observation to the total predicted positive observation.

$$Precision = \frac{TP}{(TP+FP)} \quad (7)$$

Expression (8) is recall rate formula which is the ratio of correctly predicted positive observation to all the observation in the actual class.

$$Recall = \frac{TP}{(TP+FN)} \quad (8)$$

True positive (TP), true negative (TN), false positive (FP) and false negative (FN) were the means AzureML assess classification of data based on the usage of confusion matrix. The result of varying the number of hidden layers gave different measurement for the recall rate, precision and accuracy of the model and Figure 8 showed this plot. The Receiver Operative Characteristic (ROC) curve achieved through AzureML showed that the curve is closer to the upper left corner for the highest accuracy. As observed from Figure 9, three-hidden layer yielded highest accuracy of 98.9% from the matrix calculation shown above. Recall rate for one-hidden layer, two-hidden layers and three-hidden layer is at 98.3%. The highest precision performance for three-hidden layer is 100%. Thus, showing that this model correctly predicts the flood status with little error in the classification of predictive measures.

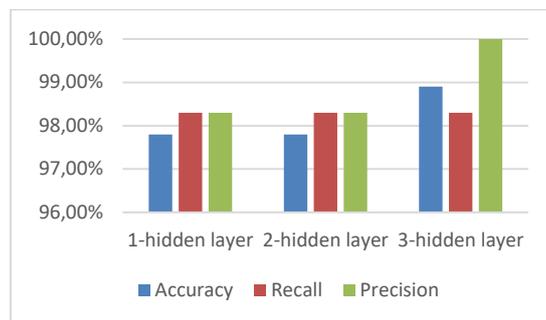


Figure 8. Evaluation model result

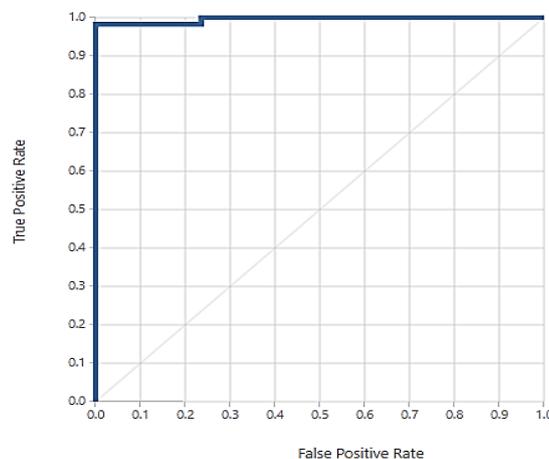


Figure 9. ROC curve of true positive rate against false positive rate

3.4. Web service interface and prediction testing

Furthermore, the AzureML Web service used the web service input and output module to connect to the built workflow. The built workflow was published using “Deploy Web Service” function. The Uniform Resource Locator (URL) and API Web Service keys were generated. The URL and API keys are used to communicate between the built workflow in AzureML and Excel Online. Azure Machine Learning add-in is downloaded in Excel Online which enables the Worksheet to be the web interface to make prediction. The interface shows the columns containing the pressure, flow rate, water level, warning level numeric, warning level status and scored label. The scored label shows the prediction based on the input of the pressure, flow rate, water level, warning level numeric and warning level status. The flood prediction could predict flood status and Figure 10 shows the Flood Prediction interface in Excel Online. It can also be seen from Figure 10 that one of the sample prediction was wrongly predicted in compare with warning level numeric (4th row) but the other predictions were correct. Excel Online allows more rows to be added for larger volume of data set to be predicted.

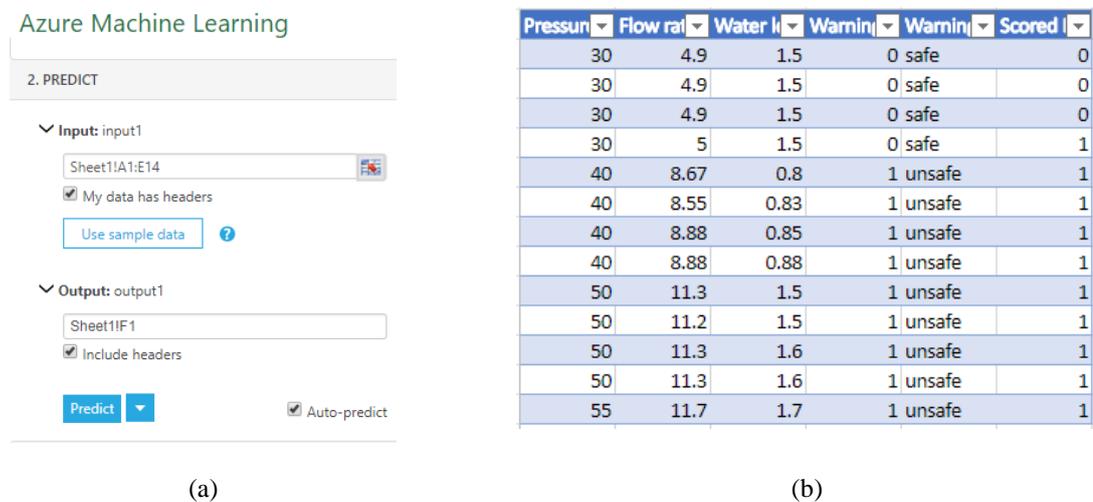


Figure 10. Screenshot of excel online interface for flood status prediction

In comparison to the work done in [8], it can be conclusively drawn that the flood forecasting done in this article made use of two-class neural network while [8] used three prediction models. However, with the work achieved in this article, it can be concluded that 2-class neural network modelling is enough to serve as a prediction model for this scope of work.

4. CONCLUSION

The use of miniaturized flow sensor and water level sensor has made this experiment to successfully build a flood monitoring system that can be useful for flood forecasting as well. The use of pressure gauge meter was to ensure that increasing pressure correlates with increasing flow rate. The experiment was able to prove that there is a strong correlation between increasing water pressure and inflow rate of water. The coefficient of multiple correlation which was tested to be 36.75% corresponds to a strong relationship between flow rate and the pressure. A real-time transmission of data from the sensor was possible as data were logged in Thingspeak channel for virtual visualization of data recorded. A dedicated in-built package available in Azure ML was used to predict flood status according to the data attained from the flow and water level sensor. AzureML was efficient in implementing the whole workflow as well as publishing the workflow as Web Service. Furthermore, the web service could predict flood status of new input parameters fed into the Excel Online interface. Prediction analysis result showed that the highest accuracy of 98.9% and precision of 100% while using 3 hidden layers. In future work, it would add to the robustness of this experiment if more sensors are added which can increase the sensitivity of the neural network prediction status. Additionally, modelling the data attained from the sensor with another neural network model will test the learning rate effectiveness of 2-class neural network. Further modification will need to be done on flood monitoring system to enable outdoor deployment and testing.

ACKNOWLEDGEMENTS

This work was conducted at the IOT and Wireless Communication Protocols Lab, and is partially funded by IIUM Publication RIGS grant no. P-RIGS19-003-0003 and the Malaysian Ministry of Education (MOE) research fund No. FRGS16-067-0566.

REFERENCES

- [1] Malay mail, "Minister: Flood forecasting, warning system to be implemented nationwide," October 2017. Retrieved 24 April 2018, <https://www.malaymail.com/news/malaysia/2017/10/25/minister-flood-forecasting-warning-system-to-be-implemented-nationwide/1495305>
- [2] Bin Wan Abdul Majid, W., Brown, E., Bin Osman, S., Bin Asan, G., Bin Osman, A., Bin Samsudin, R., & Ahmad, F., "Flood forecasting and warning for Muar River: non-structural measures for flood mitigation," *In: 37th IAHR World Congress*, Kuala Lumpur, 2017.
- [3] Romali, N. S., Yusop, Z., & Ismail, A. Z., "Application of Hec-Ras and Arc Gis for Floodplain Mapping in Segamat Town," Malaysia. *International Journal of GEOMATE*, vol 14, no. 43, pp. 125-131, 2018.
- [4] Liguori, S., Rico-Ramirez, M. A., Schellart, A. N. A., & Saul, A. J., "Using probabilistic radar rainfall nowcasts and NWP forecasts for flow prediction in urban catchments," *Atmospheric Research*, vol. 103, pp. 80-95, January 2012.
- [5] Abdullahi, S. I., Habaebi, M. H., Gunawan, T. S., & Islam, M. R., "Miniaturized Water Flow and Level Monitoring System for Flood Disaster Early Warning," *In IOP Conference Series: Materials Science and Engineering*, vol. 260, no. 1, pp. 012019.
- [6] Abdullahi, S. I., Habaebi, M. H., & Malik, N. A., "Flood Disaster Warning System on the go," *In 2018 7th International Conference on Computer and Communication Engineering (ICCCCE)*, 2018, pp. 258-263.
- [7] Joel T. de Castro, Gabriel M. Salistre, Jr, Young-Cheol Byun and Bobby D. Gerardo., "Flash Flood Prediction Model based on Multiple Regression Analysis for Decision Support System," *In Proceedings of the World Congress on Engineering and Computer Science*, San Francisco, vol. 2, pp. 1-2, 2013.
- [8] Zhao, T., Minsker, B., Salas, F., Maidment, D., Diev, V., Spoelstra, J., & Dhingra, P., "Statistical and Hybrid Methods Implemented in a Web Application for Predicting Reservoir Inflows during Flood Events," *JAWRA Journal of the American Water Resources Association*, vol. 54, no. 1, pp. 69-89, 2017
- [9] El-Shafie, A., Jaafer, O., & Akrami, S. A., "Adaptive neuro-fuzzy inference system based model for rainfall forecasting in Klang River," Malaysia. *International Journal of Physical Sciences*, vol 6, no. 12, pp. 2875-2888, 2011.
- [10] Krzhizhanovskaya, V. V., Shirshov, G. S., Melnikova, N. B., Belleman, R. G., Rusadi, F. I., Broekhuijsen, B. J., & Pyayt, A. L., "Flood early warning system: design, implementation and computational modules," *Procedia Computer Science*, vol. 4, pp. 106-115, 2011.
- [11] F. A. Ruslan, A. M. Samad, M. Tajjudin and R. Adnan, "7 hours flood prediction modelling using NNARX structure: Case study Terengganu," *2016 IEEE 12th International Colloquium on Signal Processing & Its Applications (CSPA)*, Malacca City, 2016, pp. 263-268.
- [12] Suparta W, Putro WS., "Comparison of Tropical Thunderstorm Estimation between Multiple Linear Regression, Dvorak, and ANFIS." *Bulletin of Electrical Engineering and Informatics*, vol 6, no. 2, pp. 149-58, Jun 2017.
- [13] Rafiqul, I. M., Lwas, A. K., Habaebi, M. H., Alam, M. M., Chebil, J., Mandeep, J. S., & Zyoud, A., "Analysis of Time Diversity Gain for Satellite Communication Link Based on Ku-Band Rain Attenuation Data Measured in Malaysia," *International Journal of Electrical and Computer Engineering*, vol. 8, no. 4, pp. 2608-2613, 2018.
- [14] Noymanee, J., Nikitin, N. O., & Kalyuzhnaya, A. V., "Urban Pluvial Flood Forecasting using Open Data with Machine Learning Techniques in Pattani Basin." *Procedia Computer Science*, vol 119, pp. 288-297, 2017.
- [15] Fuller, R. Neuro-Fuzzy methods for modeling and fault diagnosis. *E" otv" os Lor'and University, Budapest VacationSchool*, 2001, pp. 1-22.
- [16] Pagatpat, J. C., Arellano, A. C., & Gerasta, O. J., "GSM & web-based flood monitoring system. In *IOP Conference Series: Materials Science and Engineering*," vol. 79, no. 1, pp. 012023, 2015.
- [17] Habaebi MH, Chowdhury IJ, Islam MR, Zainal NA., "Effects of Shadowing on LoRa LPWAN Radio Links," *International Journal of Electrical and Computer Engineering* vol. 7, no. 6, pp. 2970-e976, 2017
- [18] Agel, M. M., Habaebi, M. H., Alnahdi, S., & Islam, M. R., "Evaluation of MAC's Distributed Queuing Algorithm for Internet of Things Applications," *Indonesian Journal of Electrical Engineering and Computer Science*, vol. 10, no. 3, pp. 966-97, 2018.
- [19] Marin-Perez, R., García-Pintado, J., & Gómez, A. S., "A real-time measurement system for long-life flood monitoring and warning applications," *Sensors*, vol. 12, no. 4, pp. 4213-4236, 2012.
- [20] Abdullahi, S. I., Habaebi, M. H., & Malik, N. A., "Capacitive Electrode Sensor: Design and Testing," *In 2018 7th International Conference on Computer and Communication Engineering (ICCCCE)*, 2018, pp. 34-37.
- [21] H. Yuliandoko, Subono, V. A. Wardhany, S. H. Pramono and P. Siwindarto, "Design of flooding detection system based on velocity and water level DAM with ESP8266," *2017 2nd International conferences on Information Technology, Information Systems and Electrical Engineering (ICITISEE)*, Yogyakarta, 2017, pp. 396-401.
- [22] S. I. Abdullahi, N. A. Malik, M. H. Habaebi and A. B. Salami, "Miniaturized Turbine Flow Sensor: Design and Simulation," *2018 7th International Conference on Computer and Communication Engineering (ICCCCE)*, Kuala Lumpur, 2018, pp. 38-43.

- [23] Maureira, M. A. G., Oldenhof, D., & Teerstra, L., "ThingSpeak—an API and Web Service for the Internet of Things," 2011. Retrieved 7/11/15, http://mediatechnology.leiden.edu/images/uploads/docs/wt2014_thingspeak.pdf
- [24] Team, A, "AzureML: Anatomy of a machine learning service," *In Conference on Predictive APIs and Apps*, pp. 1-13, June 2016.
- [25] Parker, L. E., "Notes on multilayer, feedforward neural networks", *CS494/594: Projects in Machine Learning*, 2006.
- [26] Leverington, D., "A basic introduction to feedforward backpropagation neural networks," *Neural Network Basics*, 2009.

BIOGRAPHIES OF AUTHORS



Abdullahi Salami recently acquired his Bachelor's degree in Telecommunication Engineering from International Islamic University Malaysia in 2018 and he is pursuing his Master in Telecommunication Engineering in International Islamic University Malaysia. His research interests are towards the fields of Telecommunications, Internet of Things, Data Analytics and Electronics. E-mail address: abdusalife@gmail.com



Mohamed Hadi Habaebi is with the department of electrical and Computer Engineering, International Islamic University Malaysia. His research interests are in IoT, wireless communications and Networking.



Noreha Abdul Malik received her BEng in Medical Electronics from University of Technology Malaysia (2001) and later pursued her MEng in Communication and Computer Engineering at National University of Malaysia (2004). She later received her PhD in Electronics and Electrical Engineering from University of Southampton, United Kingdom (2011). She is currently an assistant professor at International Islamic University Malaysia (IIUM). Her research interests are in biomedical signal processing and biomedical applications. She is a member of Institute of Engineers Malaysia (IEM) and Board of Engineer Malaysia (BEM).