SEMARAK ILMU PUBLISHING

Journal of Advanced Research in Applied Sciences and Engineering Technology

Journal homepage:
https://semarakilmu.com.my/journals/index.php/applied_sciences_eng_tech/index
ISSN: 2462-1943

Water Level Optimization of Sungai Bunus via Artificial Neural Network

H. Ghazali[1,2], N. S. Khusaini[1], M. H. M. Ramli[1], N. Aziz[1], M. Eilyas[1], Z. Mohamed[1,*], F. Mohamad[1], H. Rusdin[1]

[1] School of Mechanical Engineering, College of Engineering, Universiti Teknologi MARA Shah Alam, Selangor, Malaysia
[2] Mechanical and Electrical Services, Department of Irrigation and Drainage, Kuala Lumpur, Malaysia

ARTICLE INFO

Article history:
Received 6 April 2026
Received in revised form 20 May 2026
Accepted 5 July 2026
Available online 18 July 2026

Keywords:
Artificial Neural Network; Machine Learning; Google Colab; Python

ABSTRACT

Every part of Malaysia was recently hit by flash floods, and it caused a lot of havoc. This disaster happened because of the lack of capacity of the reservoir to hold water during heavy rains. In this project, we aim to help the Malaysian Drainage and Irrigation Department (DID) by providing artificial intelligence solutions. The solution is to optimise each value of the water level in each Bunus River catchment. DID will have access to monitor and operate the pump gate system in selected areas for specific conditions. The catchment's mechanical infrastructure will be optimised using machine learning, which is an artificial neural network (ANN) method. This approach uses the Python language and the Google Colab platform. The catchment's mechanical infrastructure will be optimised using machine learning, which is an artificial neural network (ANN) method. This approach uses the Python language and the Google Colab platform. An algorithm will be developed to estimate the optimal water level, and the data will be used in real-life situations. The result is that all pump gates for each online storage system can connect to each other. This is to ensure safety around the catchment so that water can be released from upstream to downstream very smoothly.

1. Introduction

Floods, along with other natural disasters, frequently occur in Malaysia and can cause significant damage and impact millions of people. A study by DID estimate a 10% chance of flooding across a large area of the country, affecting millions of people and causing billions of ringgits in annual flood losses [1]. Flooding is most common during the northeast monsoon season, which occurs from November to January, especially in Peninsular Malaysia [2].

The Department of Irrigation and Drainage (DID) in Malaysia uses both structural and non-structural measures to manage floods sustainably. Structural measures involve constructing engineering buildings such as reservoir ponds and regulating water flow with sluice gates and pump houses. Non-structural measures include implementing a flood forecasting and warning system and providing awareness initiatives, guidelines, laws, and flood maps to change human behaviour.

* *Corresponding author.*
E-mail address: zulkifli127@uitm.edu.my

https://doi.org/10.37934/araset.61.5.274285

Kuala Lumpur is at risk of flash floods due to heavy rain and urban development. It is located in the Klang River Basin and is divided into eight subdistricts. The northern part includes Mukim Batu, Mukim Setapak, Hulu Klang subdistrict, and Kuala Lumpur subdistrict, while the southern part includes Mukim Ampang, Mukim Cheras, and Mukim Petaling.

Sungai Bunus is a river that connects to Kuala Lumpur's main river, Sungai Klang. It is not safe for recreational activities because it is classified as class IV. Sungai Bunus is about 9 kilometres long, flowing from Wangsa Maju to Kampung Baru and passing through Setapak and Kampung Boyan Detention Pond.

This research will partially focus on the Klang River, which is the basin where Sungai Bunus drains into. The study area is situated in the northeastern part of the Federal Territory of Wilayah Persekutuan Kuala Lumpur, as shown in Figure 1. The chosen catchment area includes several reservoirs such as Online Storage Sri Rampai, Online Storage Air Panas, Online Storage Titiwangsa, Kampung Boyan Pond, and Lencongan Sungai Bunus, which serve to contain excess water caused by heavy rainfall and two monsoon seasons: the Southwest Monsoon (May–September) and the Northeast Monsoon (October–March) [3][4][5][6][7]. Proper management of these reservoirs can protect the well-being of the community, minimise flood-related infrastructure damage, and ensure the long-term survival of the community [8][9][10][11].

This study aims to prevent flooding caused by heavy rainfall during monsoons by creating a machine-learning algorithm [12][13][14]. The algorithm will predict the best water level for each upstream storage (Online Storage Sri Rampai, Online Storage Air Panas, and Online Storage Titiwangsa) before releasing water to Lencongan Sungai Bunus and the Klang River. By analysing real-world data and identifying the best outputs, the system can be operated with a lower risk of flooding. The focus of this research is on constructing an algorithm to estimate the optimal water level for each sub-catchment using machine learning, specifically for Lencongan and Pond Kampung Boyan. More details will be provided in the next section.

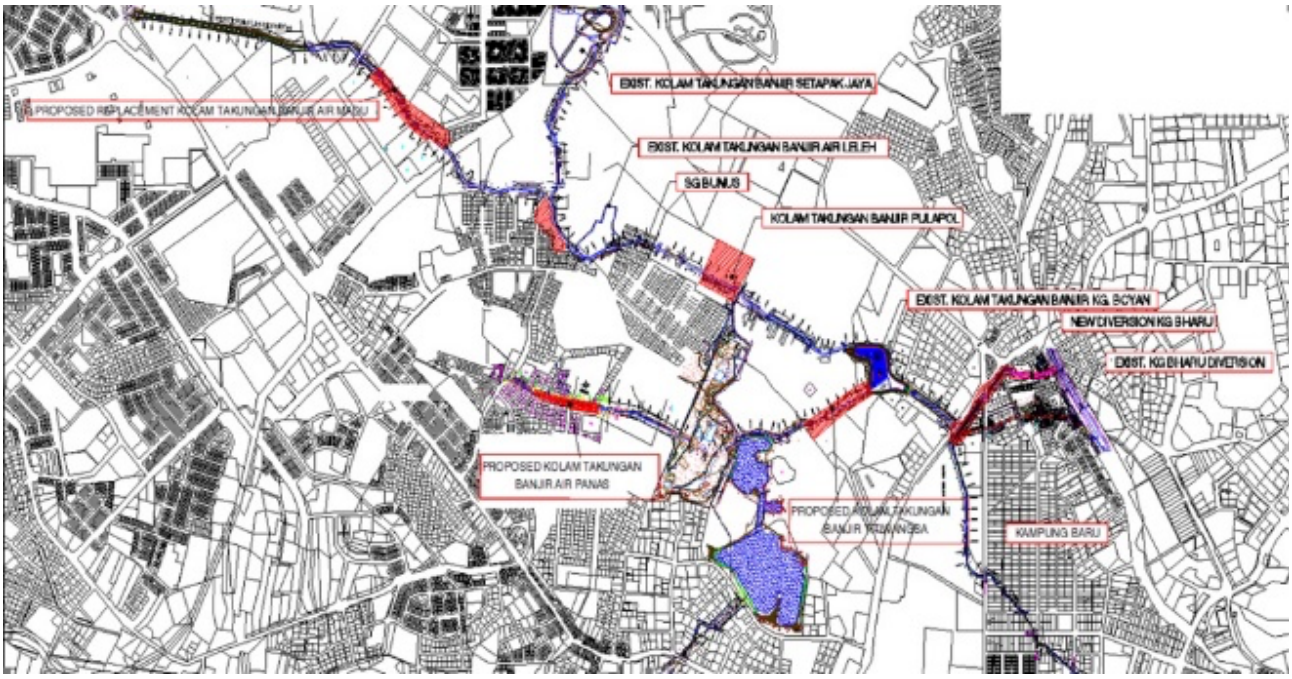

Fig. 1. Sungai Bunus Flood Mitigation.

2. Project Methodology

This research paper utilises a fully simulated approach, modelled in the Google Colab online software. The simulation is based on water level data from each online storage, provided by the Department of Irrigation and Drainage Malaysia (DID), Kuala Lumpur, and a report from a consultant. It employs an artificial neural network (ANN) and a machine learning algorithm to predict the optimal water level in each sub-catchment. This section outlines how the input parameters were collected, the role of the ANN in this research, and the underlying theory behind the output process.

2.1 Data Collection

The Obermeyer Pneumatic Spillway Gate System, shown in Figure 2, is used by DID at each online storage. This mechanism involves the use of air compressors to supply air to the rubber bladders behind the steel gates, which prevent upstream water from flowing downstream. The use of pneumatic gates is crucial when employing air-filled rubber bladders to raise bottom-hinged steel gates, as it safeguards the air-filled bladders from debris damage when the gates are lowered, as noted by Hinks and Goff [15].

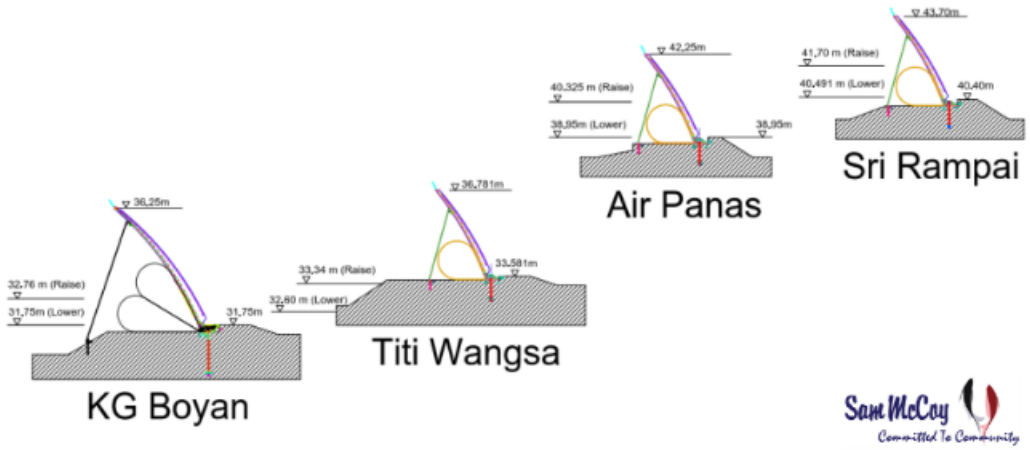

Fig. 2. Gate System for Each Catchment of Sungai Bunus

The Department of Irrigation and Drainage Malaysia (DID) collects data using programmable logic controller (PLC) sensors. These sensors measure physical quantities such as water levels both upstream and downstream and are installed before and after the pneumatic gates to monitor changes in river flow. In order to operate the Obermeyer pneumatic spillway gate system, certain conditions must be met. This project used water level data from various online storages and ponds, which is provided by the DID, and a consultant report once a year. A year's worth of data consists of 48 sets recorded four times per month.

For this study, a total of 240 data sets spanning five years were utilized. The data covers the period from January 2016 to December 2020. A sample of the first eight rows and the last eight rows of water level data can be found in Table 1. These files were uploaded to Google Colab for optimisation.

Table 1
Water level for each catchment of Sungai Bunus River from 2016 - 2020

Month	Sri Rampai (m)	Air Panas (m)	Titiwangsa (m)	Boyan (m)	Lencongan (m)
Jan-16	41.72	40.34	34.32	32.84	28.21
	41.63	40.15	34.21	32.65	28.04
	41.44	40.09	34.05	32.43	27.85
	41.32	39.78	33.99	32.21	27.53
Feb-16	41.05	39.83	33.87	32.09	27.13
	41.02	39.65	32.53	31.79	26.85
	41.11	39.44	31.67	31.36	26.52
	40.99	38.92	30.98	29.88	26.22
...	...	...	...	...	...
Nov-20	45.80	44.81	39.72	38.94	36.37
	45.67	44.62	39.57	38.61	36.21
	45.49	44.44	39.34	38.38	35.78
	44.86	44.14	38.82	38.07	35.39
Dec-20	44.04	43.67	38.13	37.87	35.06
	44.27	43.85	38.34	37.81	35.15
	43.79	43.59	38.56	37.89	35.34
	43.62	43.33	38.78	38.03	35.57

2.2 Data Analyzation

To simulate the optimal water level for each sub-catchment of the Sungai Bunus River, an artificial neural network (ANN) was employed. An ANN is a network of interconnected nodes that imitates the way neurons connect and communicate in the human brain.

The basic structure of an artificial neural network (ANN) is illustrated in Figure 3. It consists of three layers: the input layer, which accepts input variables and external signals; one or more hidden layers, which perform nonlinear input transformations; and the output layer, which produces the desired target outcome. The signal flows from the input layer to the output layer through the hidden layers, with each layer processing the signal before passing it on to the next layer. This process is analogous to the way neurons in the human brain communicate through association, while in ANN, neurons communicate through connection weights.

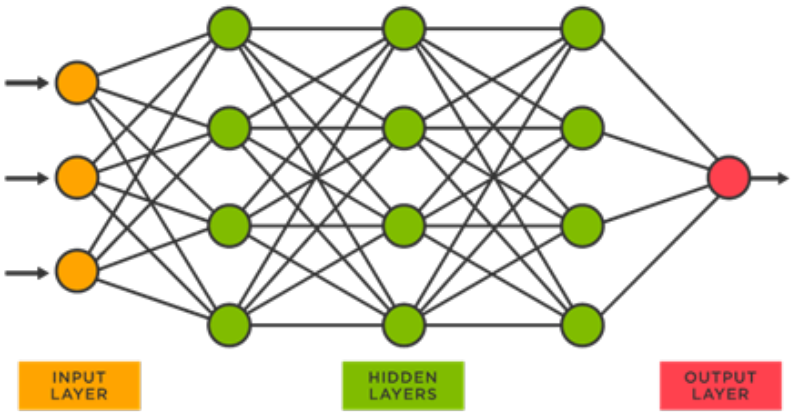

Fig. 3. The Illustration of Neural Networks

For this project, the data used will be water level observations obtained from each watershed in Sungai Bunus. These observations were collected by the Malaysian Department of Irrigation and Drainage (DID) and a consultant study. The data will be employed to develop a neural network application that will use machine learning techniques to provide feedback to the user. The goal is to achieve high accuracy in predicting whether the catchment will flood or not, which will be discussed in more detail in the upcoming section titled Results and Discussion.

2.3 Conceptualization

In Figure 4, there is an empty square that represents a node. Three signals, x1, x2, and x3, enter this node, and the output is represented by Y. The weights associated with each signal are w1, w2, and w3 respectively, while the bias associated with storing information is 'b'. The neural network stores its information using these weights and biases. The equation for the node shows that the input signals are multiplied by their respective weights before entering the node, and the output is the weighted sum of these signals:

$$\boldsymbol{v} = (w_1 - x_1) + (w_2 - x_2) + (w_3 - x_3) + b \quad (1)$$

Prior to being processed by the node, the input signals are multiplied by their respective weights. The resulting output is the sum of the weighted signals. This equation shows that the effect of the signal is directly proportional to the weight assigned to it. The neural network node's output can be mathematically represented as:

$$y = \varphi(v) = \varphi(wx + b) \quad (2)$$

The output of the node is processed by the activation function, φ , which determines the behavior of the node.In this case, the collected data will serve as the input signals for the neural network. The expected outputs that are already known will be the result. As a result, the neural network will train the input signals by adjusting the weights and biases and decreasing the error during the training phase. This will result in more accurate and efficient outcomes.

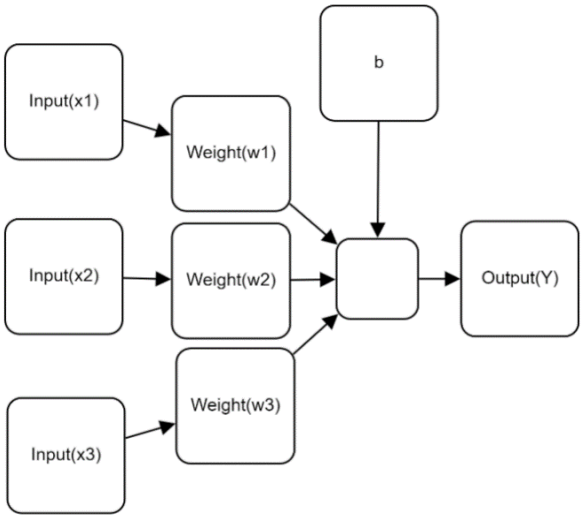

Fig. 4. The Diagram of Neural Networks with One Hidden Layer (Single Layer NN)

2.4 Development of Machine Learning

To begin with, import the csv data obtained from the Department of Irrigation and Drainage (DID) into Google Colab. Then, check the data files by running the command '! head (file name.csv)' as shown in Figure 5. This step is crucial to ensure that the files are properly formatted.

```
! head waterlevel-boyan2.csv

Sri Rampai,Air Panas,Titiwangsa,Boyan
41.72,40.34,34.32,32.84
41.63,40.15,34.21,32.65
41.44,40.09,34.05,32.43
41.32,39.78,33.99,32.21
41.05,39.83,33.87,32.09
41.02,39.65,32.53,31.79
41.11,39.44,31.67,31.36
40.99,38.92,30.98,29.88
40.94,38.49,30.34,29.53
```

Fig. 5. Check the Data Set

To ensure that the data is in the correct format, use the 'head' function followed by the file name in csv format after loading it from the Department of Irrigation and Drainage (DID) into Google Colab. To check the number of rows and columns, use 'print(dataset.shape)' and preview the dataset by using '[0:5,:]' to view the first five rows with all columns. To view the entire dataset, use ':' instead of '0:5'. As this programming is a supervised learning neural network, prepare the output by setting limits for the water level of Kampung Boyan and Lencongan Sungai Bunus based on past flood reports. A limit has been set for the water level of Kampung Boyan, as shown in Figure 6 below. If the last column is less than 38.0, it will be set to 0, indicating no flood; otherwise, it will be set to 1, indicating a flood has occurred.

```
[ ]  dataset[dataset[:,-1]<38.0, -1]=0
     dataset[dataset[:,-1]>=38.0, -1]=1
```

Fig. 6. Output Preparation

In order to make the results more dependable, randomization will be used for optimization. This involves shuffling the rows of the dataset, which will improve the outcome during the training and validation process. Although this can make it difficult to reproduce results, it ensures that the training values are not entirely positive, and the validation data does not contain solely negative values.

After shuffling, the dataset will be split into two parts: the training and validation sets. This code will divide the randomised dataset into 80% training data and 20% validation data as in Figure 7. The dataset contains a total of 240 rows, which means the training set will contain 192 rows and the validation set will contain 48 rows.

Figures 8–11 display bar charts representing the training and validation data sets obtained after the splitting process. Figure 8 reveals that there are roughly 160–170 zero values and 20–30 one values. For the validation data set of Lencongan, as seen in Figure 9, there are about 40 zeros and 7

ones, indicating that the likelihood of not experiencing any floods at Lencongan is high. Figure 10 illustrates that the training data set for Boyan comprises about 120 zeros and 60 ones, whereas Figure 11 shows that the validation data set for Boyan consists of around 30-35 zeros and 15 ones. Based on these bar charts, it can be concluded that the probability of Boyan experiencing a flood is over 50%.

```
[ ]  index_20percent = int(0.2*len(dataset[:,0]))
     print(index_20percent)

     48

[ ]  XVALIDATION = dataset[:index_20percent, :-1]
     YVALIDATION = dataset[:index_20percent, -1]

     XTRAIN = dataset[index_20percent:, 0:-1]
     YTRAIN = dataset[index_20percent:, -1]
```

Fig. 7. Split Data into Training (80%) and Validation (20%) Data Set

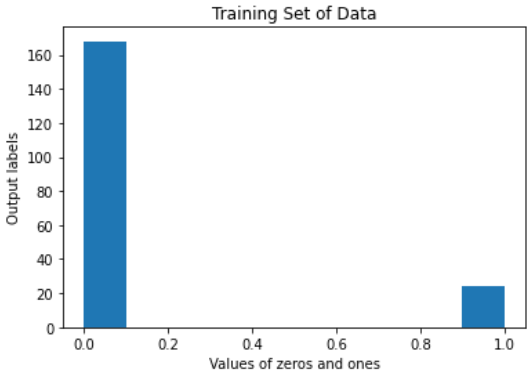

Fig. 8. Training Set of Data (Lencongan)

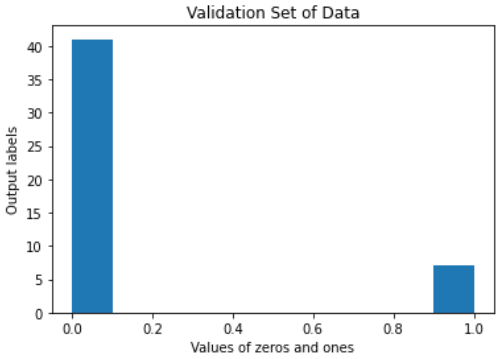

Fig. 9. Validation Set of Data (Lencongan)

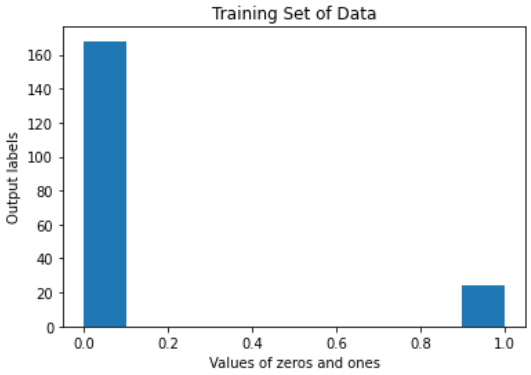

Fig. 10. Training Set of Data (Boyan)

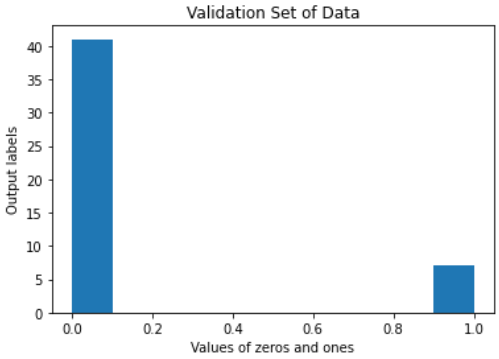

Fig. 11. Validation Set of Data (Boyan)

Subsequently, the data will undergo normalization. This is necessary because the water level values vary significantly over time due to various factors. The data must be standardised (normalised) since the flood record is only available from the beginning of August 2016. Standardisation is necessary to ensure accuracy and precision.

Figures 12 and 13 display bar charts for the 0th Column (Sri Rampai) before standardization for the Boyan and Lencongan models. These charts demonstrate that the values before standardization

are distinct, even though both models have the same dataset. For instance, for Boyan, a water level of 45 m is slightly above 10, whereas for Lencongan, a water level of 45 m is approximately 10. This indicates that the shuffling process of the data has been effective, and the outcomes will be dependable.

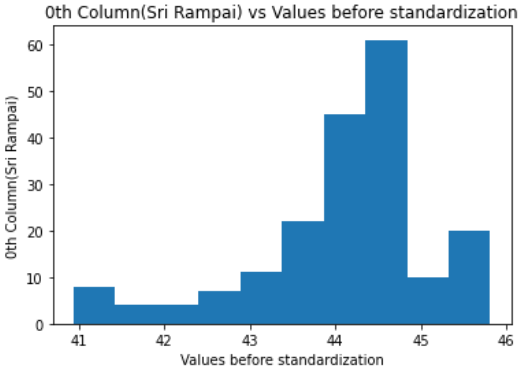

Fig. 12. Values of Sri Rampai Water Level Before Standardization (Lencongan).

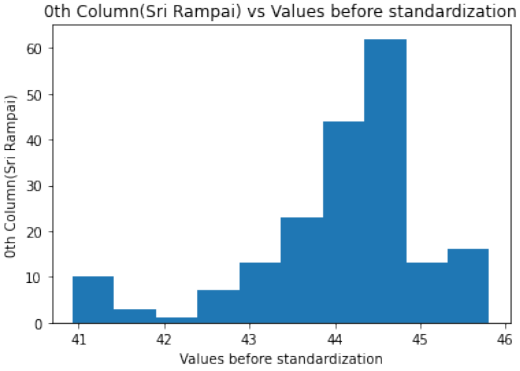

Fig. 13. Values of Sri Rampai Water Level Before Standardization (Boyan).

The bar chart in Figure 14 and Figure 15 illustrates the Sri Rampai water level values after being standardized. The standardization process is done by using the mean and standard deviation to reduce the range of the water level. This results in a reduction of the data distribution in the bar chart for both catchments (Lencongan and Boyan) from the original range of 41-46 to the new range of -3 to 1.5.

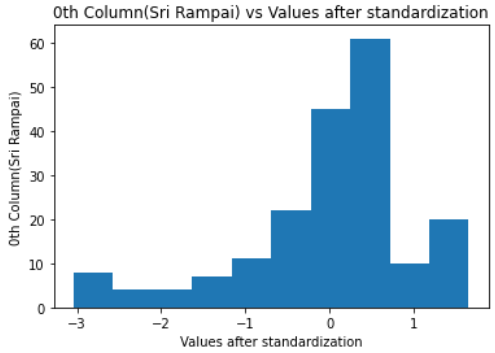

Fig. 14. Values of Sri Rampai Water Level After Standardization (Lencongan)

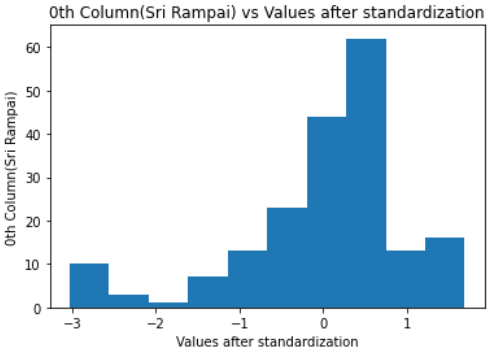

Fig. 15. Values of Sri Rampai Water Level After Standardization (Boyan)

2.5 Simulation Implementation

In order to model the optimised and normalised data set, a neural network was constructed with a specific architecture. The network has 8 neurons in the first layer, 4 neurons in the second layer, and 1 neuron in the output layer. The activation function used for both the first and second layers is "relu," while "sigmoid" is used for the output layer. The ReLU (Rectified Linear Unit) function helps the model converge quickly during training, while the sigmoid function is suitable for probability outputs as it constrains values between 0 and 1.

The next step is to compile the model developed in Figure 16 using the Adam (abbreviation for Adaptive Moment Estimation) optimizer. Although other optimizers such as RMSProp (Root Mean Square Propagation), Momentum, and AdaGrad (Adaptive Gradient) exist, Adam is considered the

best since it performs well in practice and has become the preferred choice for machine learning and deep learning problems. [16][17][18]

```
[ ]  from tensorflow.keras.models import Sequential
     from tensorflow.keras.layers import Dense

     model = Sequential()
     model.add(Dense(8, input_dim=len(XTRAIN[0, :]), activation='relu'))
     model.add(Dense(4, activation='relu'))
     model.add(Dense(1, activation='sigmoid'))
     print(model.summary())
```

Fig. 16. Creating A Neural Network.

Then, the model is trained using XTRAIN and YTRAIN. Each epoch involves the model scanning through all the rows in XTRAIN and calculating errors using YTRAIN. Increasing the number of epochs usually leads to higher accuracy. During training, callback functions are used to stop the epoch if the accuracy values remain the same for over 20 epochs, resulting in optimal learning curves.

After training, the model is evaluated using the same training data, although this evaluation is technically meaningless. Its purpose is to ensure that the model is functioning correctly. The model is then evaluated on the validation set, which is an unknown dataset, and its accuracy is compared to that of the training data.

Finally, the neural network model trained on the training dataset is used for prediction and compared to the validation dataset. The results are evaluated using precision and recall, in addition to accuracy.

The appendix section of the research documentation will contain all the computer code and scripts used in this study. It will provide detailed explanations, short pieces of code, and any extra information needed to understand and recreate the programming methods and techniques used. Sharing this programming information in the appendix helps make the research transparent and reproducible, allowing other researchers and interested people to explore and analyse the study's findings more easily.

3. Results and Discussion

3.1 Algorithm Performance Testing

The algorithm's effectiveness was evaluated by analysing the learning curves of both the training data and the validation data. The training data is illustrated by the blue straight line, while the validation data is represented by the orange dotted line. Learning curves are essentially the output of the neural network's training model.

The graph in Figure 17 displays the learning process of the Lencongan model. The model is being trained and improved over time, with the best weights being saved whenever a better model is found. As the model progresses, the accuracy gradually increases. At the start, the training accuracy was 59.9% and the validation accuracy was 75.0%. However, after the first epoch, the training accuracy improves to 75.0%, while the validation accuracy improves to 77.08%. As the model continues to train, the accuracy for training reaches 87.5%, and the accuracy for validation reaches 85.42% after 10 epochs. The model stops training at 220 epochs, where the accuracy values for both variables remain the same for over 20 times. At the end of training, the training accuracy reaches 94.27%, while the validation accuracy reaches 95.83%.

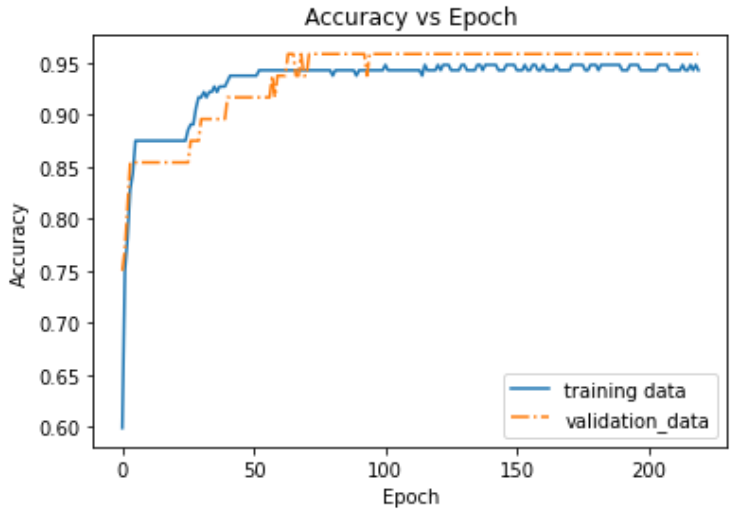

Fig. 17. Accuracy vs Epoch (Lencongan).

The learning curve for the Boyan model is displayed in Figure 18. The accuracy of the training data begins at 66.07%, while the accuracy of the validation data begins at 75.0%. After the first epoch, the training accuracy increases to 75.52%, and the validation accuracy increases to 81.25%. Over the course of 10 epochs, the training accuracy reaches 81.77%, and the validation accuracy reaches 83.33%. The model stops after 220 epochs when the accuracy of both variables remains the same for more than 20 times. Ultimately, the training accuracy is 89.06%, and the validation accuracy is 89.58%.

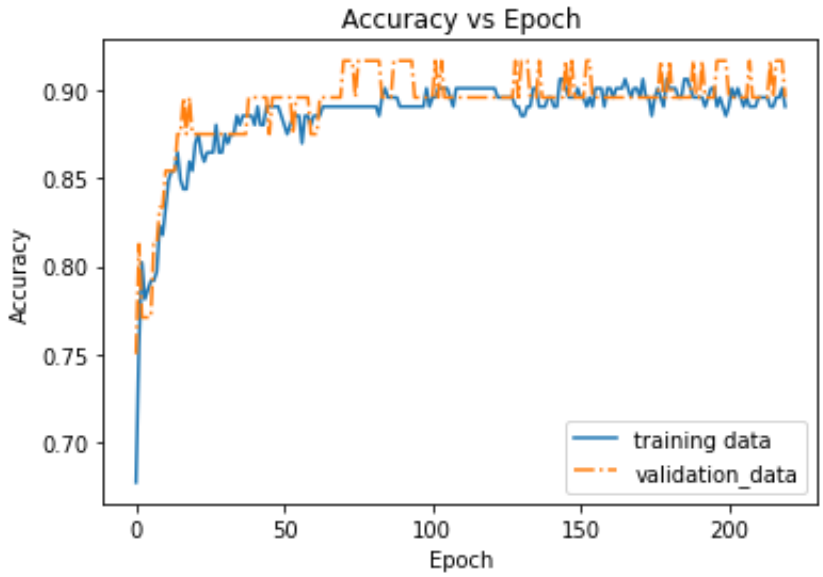

Fig. 18. Accuracy vs Epoch (Boyan).

The most important concepts in machine learning, which are often confusing, are precision and recall. They are crucial performance metrics used for pattern recognition and classification, particularly when dealing with imbalanced classes [19]. Unlike accuracy, which tells us the overall number of correct predictions made by the model, precision and recall provide a more useful measure of prediction success [20]. It is imperative to understand and apply these concepts to create a highly accurate and precise machine learning model. programming information in the appendix

helps make the research transparent and reproducible, allowing other researchers and interested people to explore and analyse the study's findings more easily.

```
[ ]  from sklearn.metrics import accuracy_score, precision_score, recall_score, f1_score

     accuracy = accuracy_score(YVALIDATION, prediction.round())
     precision = precision_score(YVALIDATION, prediction.round())
     recall = recall_score(YVALIDATION, prediction.round())
     f1score = f1_score(YVALIDATION, prediction.round())
     print("Accuracy: %.2f%%" % (accuracy * 100))
     print("Precision: %.2f%%" % (precision * 100))
     print("Recall: %.2f%%" % (recall * 100))
```

Fig. 19. Evaluation of The Model (Boyan and Lencongan).

Figure 19 highlights the final evaluation of both models (Lencongan and Boyan) on the training data, which produced satisfactory outcomes. Boyan achieved an accuracy of 91.67%, a precision of 92.86%, and a recall of 81.25%. On the other hand, Lencongan obtained a higher accuracy score of 95.83%, a precision of 100.00%, but a relatively lower recall of 71.43% compared to Boyan.

4. Conclusion

In conclusion, the study successfully developed an algorithm to predict optimal water levels in two catchments, Lencongan Sungai Bunus and Pond Kampung Boyan. The proposed method's performance was evaluated based on accuracy, precision, and recall score. The study's findings will be shared with the Department of Irrigation and Drainage Malaysia (DID) to prevent flash floods. For future studies, the installation of sensors to measure flow rates at each catchment is recommended, as it can provide additional information to estimate the possibility of flooding. Furthermore, the model's optimisation can be enhanced by increasing the number of epochs and adding neural network layers. Finally, the frequency of water level data can be increased to improve the model's efficiency and precision.

Acknowledgement

This work was supported by “Geran Insentif Penyeliaan” [00-RMC/GIP 5/3 (062/2021)], Universiti Teknologi MARA.

References

[1] Pusat Ramalan dan Amaran Banjir Negara Bahagian Pengurusan Sumber Air Dan Hidrologi and Jabatan Pengairan Dan Saliran Malaysia, “Laporan Banjir Tahunan 2020,” 2021. Accessed: Feb. 26, 2024. [Online]. Available: http://h2o.water.gov.my/man_hp1/LBT2020.pdf

[2] N. Afzan and M. Hasni, “THE PENINSULAR MALAYSIA FLOODING-A SPATIO-TEMPORAL ANALYSIS OF PRECIPITATION RECORDS,” 2014.

[3] Mohammed, Nurashikin, Rodger Edwards, and Andrew Gale. “Optimisation of Flooding Recovery for Malaysian Universities.” Procedia Engineering 212 (2018): 356–62. https://doi.org/10.1016/j.proeng.2018.01.046.

[4] ASEANUP. Malaysia's economic plan 2016-2020 - ASEAN UP. 2015 2015-05-28; Accessed: Feb. 26, 2024. [Online] Available: https://aseanup.com/malaysia-economic-plan-2016-2020/

[5] MoHE. MoHE - Pelan Pembangunan Pendidikan Malaysia 2015-2025 (Pendidikan Tinggi). Accessed: Feb. 26, 2024. [Online] Available: https://www.mohe.gov.my/en/pppm-pt.

[6] Department of Statistics Malaysia. Department of Statistics Malaysia Official Portal. 2017 Accessed: Feb. 26, 2024. [Online] Available: https://www.dosm.gov.my/v1/

[7] Jabatan Pengairan dan Saliran Malaysia. Pengurusan Banjir - Program dan Aktiviti - Pengurusan Banjir di Malaysia. 2017 Accessed: Feb. 26, 2024. [Online] Available: http://www.water.gov.my/our-services-mainmenu-252/flood-mitigation-mainmenu-323/programme-aamp-activities mainmenu-199

[8] Brookings, "Protecting Persons Affected by Natural Disasters". 2006.

[9] Department of Primary Industries, NSW Government, Natural disaster reporting and declaration, in POLICY O-103 NATURAL DISASTER REPORTING AND DECLARATION. 2011.

[10] California Department of Public Health. Know and Understand Natural Disasters. 2016 Accessed: Feb. 26, 2024. [Online] Available: http://www.bepreparedcalifornia.ca.gov/BeInformed/NaturalDisasters/Pages/KnowandUnderstandNaturalDisasters.aspx.

[11] Renaud, F.G., K. Sudmeier-Rieux, and M. Estrella, "The role of ecosystems in disaster risk reduction". 2013.

[12] Adnan, M.S.G., Siam, Z.S., Kabir, I., Kabir, Z., Ahmed, M.R., Hassan, Q.K., Rahman, R.M., Dewan, A.," A novel framework for addressing uncertainties in machine learning-based geospatial approaches for flood prediction" J. Env. Manag (2023). 326, https://doi.org/10.1016/j.jenvman.2022.116813.

[13] Ahmed, N., Hoque, M.A.A., Arabameri, A., Pal, S.C., Chakrabortty, R., Jui, J. "Flood susceptibility mapping in Brahmaputra floodplain of Bangladesh using deep boost, deep learning neural network, and artificial neural network". Geocarto Int. 37(2021), 8770–8791. https://doi.org/10.1080/10106049.2021.2005698

[14] Al-Abadi, A.M."Mapping flood susceptibility in an arid region of southern Iraq using ensemble machine learning classifiers: a comparative study". Arab. J. Geosci. 11(2018), 1–19. https://doi.org/10.1007/s12517-018-3584-5

[15] Hinks, J L, and C A Goff. "Fuse-Plug Spillways." Smart Dams and Reservoirs, January 2018. https://doi.org/10.1680/sdar.64119.249.

[16] Jiang, Lili. "A Visual Explanation of Gradient Descent Methods (Momentum, AdaGrad, RMSProp, Adam)." Medium, September 21, 2020. https://towardsdatascience.com/a-visual-explanation-of-gradient-descent-methods-momentum-adagrad-rmsprop-adam-f898b102325c.

[17] Geoffrey Hinton Nish Srivastava and Kevin Swersky. Overview of mini-batch gradient descent Accessed: Feb. 26, 2024. [Online] Available: https://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[18] Emilien Dupont. Optimization Algorithms Visualization. Accessed: Feb. 26, 2024. [Online] Available: https://gist.github.com/EmilienDupont/aaf429be5705b219aaaf8d691e27ca87

[19] "Precision and Recall in Machine Learning - Javatpoint." www.javatpoint.com. Accessed February 26, 2024. https://www.javatpoint.com/precision-and-recall-in-machine-learning

[20] Dang, Tommy. "Guide to Accuracy, Precision, and Recall." Mage.ai. Accessed February 26, 2024. https://www.mage.ai/blog/definitive-guide-to-accuracy-precision-recall-for-product-developers.