

# Integrated Aquaculture Monitoring System Using Combined Wireless Sensor Networks and Deep Reinforcement Learning

Wen-Tsai Sung,<sup>1</sup> Indra Griha Tofik Isa,<sup>1</sup> and Sung-Jung Hsiao<sup>2\*</sup>

<sup>1</sup>Department of Electrical Engineering, National Chin-Yi University of Technology,  
Zhongshan Rd, Section 2, No. 57, Taichung City 411030, Taiwan

<sup>2</sup>Department of Information Technology, Takming University of Science and Technology,  
Taipei City 11451, Taiwan

(Received September 15, 2023; accepted March 1, 2024)

**Keywords:** aquaculture monitoring system, deep reinforcement learning, deep learning, IoT, WSN

Freshwater fish is one of the commodities experiencing an increasing growth rate from 1990 to 2018. Many efforts have been made to meet market needs, through both fisheries technology and applied technology, one of which is an integrated monitoring system. In this study, an aquaculture monitoring system was developed that integrates wireless sensor networks (WSNs) based on temperature, pH, and turbidity with deep reinforcement learning. The purpose of this study is to produce a convenient, precise, and low-cost aquaculture monitoring system. The stages of the study are (1) the integration of all the WSN components, (2) the validation of the WSNs, (3) the implementation of the analysis model in the system, (4) the implementation of the recommended model into the DRL system, and (5) practical experimentation using the aquaculture monitoring system. The WSN validation results indicate that the average percentage error is 3.23%, whereas at the system modeling stage, the optimal accuracy is 98.80%. In the experiment to monitor real aquaculture environmental conditions, an accuracy of 97% is obtained.

## 1. Introduction

The amount of freshwater fish consumed globally has continued to increase over the last few decades, with a total increase of 122% in fish consumption from 1990 to 2018.<sup>(1)</sup> Some of the challenges faced by several countries in freshwater fish production include (1) policy and governance, (2) climate change, (3) the quality and availability of water, (4) habitat degradation, and (5) the lack of data and information.<sup>(2)</sup> The challenges, namely, the quality and availability of water and the lack of data and information, can be solved by implementing machine-learning-based technology combined with wireless sensor networks (WSNs) that provide accurate real-time data to users.<sup>(3)</sup> Technically, this technology can be implemented, for example, to determine whether the condition of fish pond water is normal or abnormal. Thus, the user can immediately determine the condition of the pond water in real time and the recommendations for this condition, which can increase the productivity of freshwater fish production in the long term.

---

\*Corresponding author: e-mail: [sungjung@gs.takming.edu.tw](mailto:sungjung@gs.takming.edu.tw)  
<https://doi.org/10.18494/SAM4660>

WSNs are a subset of IoT and combine several nodes having sensors and microcontrollers that are then integrated into a single unit in a certain network pattern.<sup>(4)</sup> WSNs have several advantages, including speeding up the data processing, because nodes containing sensors directly transmit data to servers, and networks within WSNs can back up each other's data.<sup>(5)</sup> In several previous studies, WSN technology was implemented in fish management. For example, Wang *et al.* implemented WSNs combined with unmanned aerial vehicle systems in inland aquaculture water quality evaluation.<sup>(6)</sup> The monitoring device integrates the dissolved oxygen and turbidity sensors to monitor real-time environmental conditions. System quality analysis was performed by calculating the correlation between real-time and predicted values, or  $R^2$ , where the dissolved oxygen and turbidity sensors yielded 0.8042 and 0.8346, respectively. Tayo *et al.* developed a water quality monitoring system for shrimp aquaculture using WSNs combined with conductivity, temperature, and water level sensors, as well as an Arduino nano-microcontroller.<sup>(7)</sup> Their test results showed that the implemented sensor nodes can detect the real-time conditions of shrimp pond water in terms of both temperature and salinity. Moreover, Huy *et al.* developed WSNs by integrating Raspberry Pi 3 as an embedded computer and XBee S2C as wireless communication modules where the system can control the water pump, light condition, and actuator by using a web system and several sensors, i.e., a light sensor and a temperature sensor.<sup>(8)</sup> The implementation of the device can increase production efficiency through the provided real-time data and remote actuator control, and the systems can automatically respond to changes in environmental conditions. In this study, the system built is based on WSNs by integrating temperature, pH, and turbidity sensor components and determines the condition of fish pond water from these three parameters. The WSN system is integrated with deep reinforcement learning (DRL), which generates recommendations through agents based on predetermined parameters. Recommendations resulting from the system depend on whether the fish pond water conditions are “optimal” or “not optimal”.

DRL is a combination of deep learning (DL) and reinforcement learning (RL) and performs complex computational processes.<sup>(9)</sup> In data processing, DRL employs an agent that is integrated with DL, which consists of several layers and neurons.<sup>(10)</sup> The agent processes data on the basis of the state given by the environment, then processes it through the learning method mechanism, and finally produces output in the form of an action in response to the given initial state condition.<sup>(11)</sup> In this study, the initial state consisted of the temperature, pH, and turbidity conditions of the fish pond. Then, the agent receives the initial state in real time and processes data through deep neural network modeling. The results of data processing are in the form of binary actions, which are recommendations depending on whether the fish pond water conditions are optimal or not optimal. Rewards in the DRL system provide feedback to agents based on actions that have been taken; in this case, the aim is to enhance new knowledge on complex conditions in fish ponds, to be used by agents. Technically, Fig. 1 shows the mechanism of data processing on DRL.

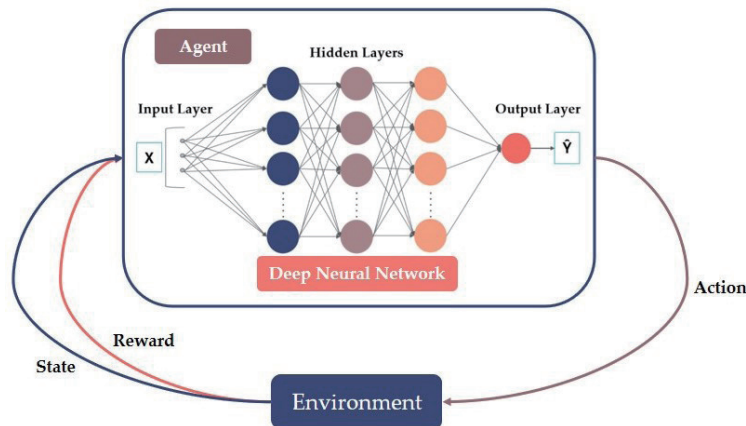


Fig. 1. (Color online) Deep reinforcement learning implemented in the study.

## 2. Materials and Methods

This section consists of several subsections on (1) the proposed system, where we describe how the elements are linked in the system, (2) the configuration of WSNs, (3) DRL setups, (4) hardware and software setups, and (5) the experimental setup.

### 2.1 Proposed system

The proposed system is shown in Fig. 2. There are four main components, namely, (1) WSNs, (2) Firebase Data Cloud, (3) the DRL system, and (4) mobile applications. In WSNs, there are several nodes, which are a combination of sensors, Arduino microcontrollers, and WiFi modules. WSNs combine pH, temperature, and turbidity sensors; they can also allow more than three sensors to be integrated, as indicated by the presence of an “*n*th-sensor”. Environmental conditions consisting of temperature, pH, and turbidity are read by nodes in real time and integrated by WSNs. Then, the data are transmitted to the Firebase data cloud real-time data storage medium. Firebase receives data from WSNs via the message queuing telemetry transport (MQTT) protocol, which is a machine-to-machine communication protocol commonly used in IoT devices. The data transmitted to Firebase is then processed through the DRL system with an agent as a data processor that determines the action based on the state received. The agent is integrated with a deep neural network to produce optimal accuracy in state processing, which consists of real-time conditions of temperature, pH, and fish pond turbidity. The processing results are translated into actions in the form of “optimal” or “non-optimal” binary conditions. The representation of the system is shown by a mobile application where the user can finally see how the real-time conditions of the fish pond are based on three parameters along with recommendations for fish pond conditions, whether optimal or not optimal.

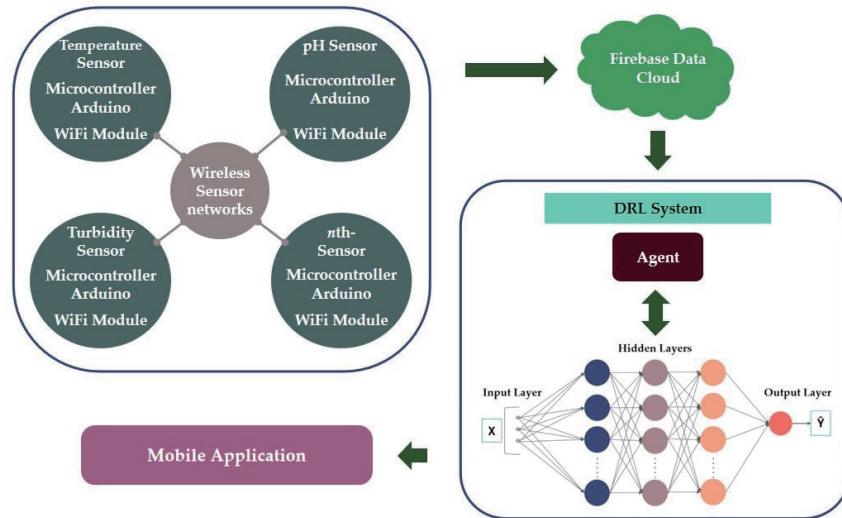


Fig. 2. (Color online) Proposed system.

## 2.2 Configuration of WSNs

As described in the previous section, WSNs are an integration of several nodes consisting of sensors, microcontrollers, and WiFi modules. Several topology configurations can be implemented, including the mesh topology used in this study. The mesh topology configuration represents the connectivity between nodes.<sup>(12)</sup> For example, if there are nodes A, B, C, and D, the connections built on a mesh topology are node A connected to nodes B, C, and D. Node B is connected to nodes A, C, and D, node C is connected to nodes A, B, and D, and node D is connected to nodes A, B, and C. All nodes contained in WSNs are interconnected to form a mesh topology structure, providing advantages in connection speed, increasing system coverage, increasing reliability in data transmission, and allowing nodes to back up each other in case of network constraints. Figure 3 shows the mesh topology structure of WSNs in this study.

## 2.3 DRL setup

DRL is implemented to perform complex computational processing, which is carried out by agents integrated with deep learning, or technically through deep neural networks. Fundamentally, the DRL formulation refers to the Bellman equation that represents the Q-learning modeling that updates the rule in the DRL sequence, as described in Eq. (1).<sup>(13)</sup>

$$Q(s, a) = Q(s, a) + \alpha (r + \gamma \times \max_{a'} Q(s', a') - Q(s, a)) \quad (1)$$

$Q(s, a)$  is described as a Q-value, which consists of a series of state  $s$  consisting of temperature, pH, and turbidity, as well as action  $a$ , which is the result of a decision from the

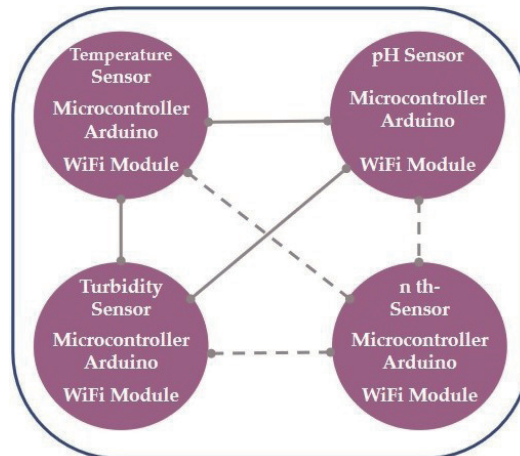


Fig. 3. (Color online) WSN mesh topology.

agent, which, in this study, is in the form of a binary condition, whether the fish pond is in an optimal or non-optimal condition. In this formulation, the Q-value is updated with  $Q(s, a)$  plus  $\alpha(r + \gamma \times \max Q(s^n, a^n) - Q(s, a))$ .  $\alpha$  shows the learning rate, which represents how big the step size is in each update and helps to control the information received by the Q-value. Reward  $r$  as feedback for the agent every time an action is taken, with the addition of new information in each subsequent iteration.  $\gamma$  is a discount value that is in the range of 0 and 1 and serves as a multiplier factor in determining  $r$ .  $Q(s', a')$  is the next state and action after the agent receives feedback from  $r$ .

As described in the equation above, since the model implemented in DRL is Q-Learning, a model scenario is needed as an initial state. This model scenario then becomes basic knowledge for agents in determining actions. The model scenario, which then becomes the dataset, consists of state temperature, pH, and turbidity, as well as action recommendations for pond water conditions in the form of “optimal”, represented by the number 1, and “non-optimal”, represented by the number 0. The optimal conditions for fish ponds are shown in Table 1, which refers to the Food and Agriculture Organization’s warm freshwater fishpond conditions.<sup>(14)</sup>

The next stage is the development of the dataset, where the total dataset is 2000 data records with “optimal” conditions, or class label 1 of 1000 data, and “non-optimal” or class label 0 of 1000 data. Non-optimal conditions are conditions outside the optimal range shown in Table 1. If at least one of the parameters does not meet the optimal range, a non-optimal condition will arise. Table 2 shows the dataset developed in this study.

After determining the dataset, the next step is to create a training model scenario using a deep neural network, which in this study uses five layers consisting of one input layer, three hidden layers, and one output layer. The input layer consists of three nodes that represent input data consisting of temperature, pH, and turbidity. The output layer is a binary class consisting of only one node, which represents binary class label 1 for optimal conditions and class label 0 for non-optimal conditions. The formulation used in deep neural networks is shown in Eq. (2).

Table 1  
Warm freshwater fishpond optimal ranges.

No.	Parameter	Optimal range
1	Temperature	25–30 °C
2	pH	6–9
3	Turbidity	0–50 NTU

Table 2  
The developed dataset.

No.	Temperature (°C)	pH	Turbidity (NTU)	Class
1	27	8	10	1
2	30	8	47	1
3	25	4	73	0
4	29	6	3	1
5	14	5	66	0
6	2	2	71	0
7	29	7	50	1
8	29	7	39	1
9	71	8	97	0
10	25	12	51	0
...	...	...	...	...
2000	45	14	59	0

$$z = f(w_1x_1 + w_2x_2 + w_3x_3 + \dots + w_nx_n + b) \quad (2)$$

$z$  in the deep neural network formulation shows the final result of the modeling, in which there is a combination of input  $x$  and weight  $w$ . Bias  $b$  is used as an additional parameter in optimizing the model to generalize new data, which is usually summed after all  $x$  and  $w$  are summed.<sup>(15)</sup> The activation function  $f$  can activate the addition of  $x$  and  $w$  in the context of the nonlinear transformation.<sup>(16)</sup> In this study, the activation function implemented is the sigmoid, which has a range between 0 and 1. The result of the sigmoid is in the form of a trend value, where if the resulting output is less than 0.5, then it shows a tendency close to 0, or if it is relevant to the dataset, it will refer to class label 0, which is translated as non-optimal conditions. Likewise, if the resulting output is equal to or greater than 0.5, then the tendency falls into class label 1 or optimal conditions. The sigmoid formulation is represented by Eq. (3).

$$f(x) = \frac{1}{1 + e^{(-x)}} \quad (3)$$

$f(x)$  denotes the activation function in the input data and is determined as 1 divided by the sum of 1 and  $e$ , where  $e$  is a constant with a value of 2.71838.<sup>(17)</sup> Sigmoid has the characteristic that the resulting values form an S-shaped curve in the range between 0 and 1 and provide a smooth result.

## 2.4 Hardware and software setup

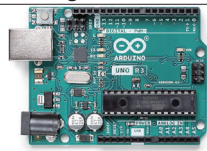

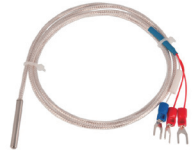
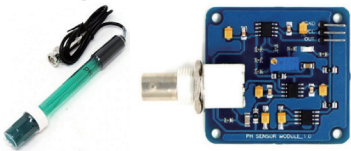

The hardware components used consist of an Arduino Uno R3 microcontroller, a WiFi module (ESP 8266), a temperature sensor (RTD PT 100), a pH sensor (SEN 0161), and a turbidity sensor (SEN 0189). Table 3 shows the details of the hardware components used in the system.

The software used in the system is Firebase Data Cloud, which is a Google product that provides a database system that can be accessed by users in real time. Firebase supports the MQTT protocol so that it can be implemented in the WSN configuration.<sup>(18)</sup> On the user side, the software interface is developed on a mobile basis with the Android Studio Integrated Development Environment (IDE). The Android Studio IDE is a software development program for Android-based applications using the programming languages Java and Kotlin when designing interfaces using extensible markup language.

## 2.5 Experimental setup

Figure 4 shows the experimental setup used in this study. It consists of several parts. The initial part is the integration of all the WSN components, which consist of nodes that include sensors, microcontroller units, and WiFi modules. Node integration uses a mesh topology where nodes are interconnected. In the next stage, WSNs of the system were validated to determine the percentage error level of the system. The WSNs were evaluated by comparing the actual

Table 3  
(Color online) Hardware components.

No.	Name of component	Specification	Figure detail
1	Arduino Uno R3	Microcontroller uses ATmega328P, which has 14 pinout digital input/output, 6 analog input pins of 5 and 3.3 V pin.	
2	WiFi module (ESP 8266)	Pinout: 3V3, Ground (GND), Reset (RST), Enable (EN), Serial Transmit (TX), Serial Receive (RX), General Purpose I/O pin (GPIO1 and GPIO2)	
3	Temperature sensor (RTD PT 100)	Material: PVC, PTFE, silicone rubber, glass fiber, and stainless steel Coverage range: 0–100 °C Pinout: VCC, Out, GND	
4	pH sensor (SEN 0161)	Coverage range: 0–14 level of pH Time of response: ≤1 min Voltage required: 5 V Pinout: VCC, Out, GND	
5	Turbidity sensor (SEN 0189)	Optimum temperature: 5–90 °C Time of response: ≤0.5 min Voltage required: 5 V Pinout: 8 pins	

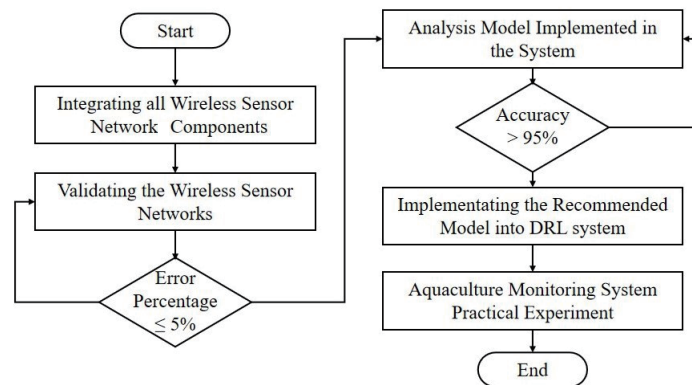


Fig. 4. Experimental setup.

environment's WSN readings with the results of measurements made with the measurement tool. The gap between WSNs and measurement tools becomes a parameter in determining the percentage error, and in this study, the percentage error tolerance of the system is a maximum of 5%. Thus, if there is an error percentage greater than 5%, the stage will be repeated for the previous process. The next stage is the analysis of the model implemented in the system by applying hyperparameter tuning scenarios to produce the highest accuracy for the model being developed. The model with the highest accuracy will be implemented in the DRL system. The threshold of the accuracy is greater than 95%. The final stage is a practical experiment using the system in a real-world environment to determine the effectiveness and accuracy of the aquaculture monitoring system.

### 3. Results and Discussion

#### 3.1 Validating the WSNs

The WSNs are validated by testing the reading results of WSNs compared with the measurement results from a measurement tool, which consists of (1) a digital thermometer for measuring the temperature, (2) a pH meter for measuring the pH, and (3) a turbidity meter for measuring the turbidity of water. The results of subsequent measurements produce gaps used for analysis in determining the percentage error of the WSN set. The data were collected from 6:00 a.m. to 8:00 p.m. every 2 h, so there were eight data collection times, namely, 6:00 a.m., 8:00 a.m., 10:00 a.m., noon, 2:00 p.m., 4:00 p.m., 6:00 p.m., and 8:00 p.m. The measurement results of the WSN device and measurement tools are shown in Table 4.

Table 4 shows differences between the measurement results of the WSN device and measurement tools for each parameter. For example, when the temperature was measured at 6:00 a.m., the result obtained using the WSN device was 24.2 °C, whereas the temperature measured with a thermometer was 23.1 °C. The absolute difference between the two values, hereinafter referred to as the gap, is 1.1 °C. The largest gap in temperature measurement occurred at 6:00



Table 4  
Results obtained using WSN device and measurement tools.

Time	WSN device			Measurement tools		
	Temperature (°C)	pH	Turbidity (NTU)	Temperature (°C)	pH	Turbidity (NTU)
6:00 a.m.	24.2	6.9	18.8	23.1	6.7	20.2
8:00 a.m.	24.2	7.0	18.7	24.7	7.0	20.3
10:00 a.m.	25.1	7.0	19.2	24.8	7.0	21.5
Noon	26.8	6.9	20.0	26.9	7.1	21.3
2:00 p.m.	26.6	6.9	20.0	26.5	7.0	21.1
4:00 p.m.	24.9	6.9	20.1	25.2	7.0	20.7
6:00 p.m.	24.4	6.7	19.4	23.8	7.0	20.3
8:00 p.m.	24.1	6.9	19.4	23.6	7.0	20.3

a.m. The largest gap in pH was 0.3 at 6:00 p.m.; i.e., the pH measurement results for the WSN device and measurement tools were 6.7 and 7.0, respectively. In the case of the turbidity measurement results, the largest gap of 2.3 NTU was obtained at 10:00 a.m.; i.e., the WSN device and measurement tool results were 19.2 and 21.5, respectively. The next step is to calculate the gap with the measurement result as an expected value, then multiply it by 100% to produce an error percentage, as in Eq. (4).

$$\text{Error Percentage} = \frac{\text{gap result}}{\text{measurement result}} \times 100\% \quad (4)$$

The measurement results of error percentage are shown in Table 5, where the average error percentages of temperature, pH, and turbidity are 1.82, 1.80, and 6.07%, respectively. Overall, the average of the three parameters is 3.23%. The value of 3.23% is less than 5%, which is the maximum threshold of error percentage in this study shown in Fig. 4; thus, the WSN device can be validated and we progress to the next stage.

### 3.2 Implementing the recommended model into DRL system

In the context of DRL, modeling is performed by Q-Learning, where scenarios are needed in the form of datasets, which become basic knowledge used by agents to make decisions. The dataset used in the modeling in this study is the scenario of optimal and non-optimal conditions for fish ponds and consists of 2000 data records that are divided into optimal conditions, which are represented by 1, and non-optimal conditions, which are represented by 0. The dataset consists of (1) state, which is the basic parameter of fish pond conditions (temperature, pH, and turbidity), and (2) action in the form of a binary class of optimal and non-optimal conditions. Table 6 shows the dataset used in the study.

Furthermore, 80% of the dataset is used as training data and 20% as testing data. The data training was carried out on a Windows 10 operating system with an Intel® Core™ i7-10510U CPU @ 1.80 GHz and software training data using Jupyter Notebook. Initial parameters in the

Table 5  
Error percentage (EP) of each parameter.

Time	EP of temperature (%)	EP of pH (%)	EP of turbidity (%)
6:00 a.m.	4.76	2.99	6.93
8:00 a.m.	2.02	0.00	7.88
10:00 a.m.	1.21	0.00	10.70
Noon	0.37	2.82	6.10
2:00 p.m.	0.38	1.43	5.21
4:00 p.m.	1.19	1.43	2.90
6:00 p.m.	2.52	4.29	4.43
8:00 p.m.	2.12	1.43	4.43
Average per parameter	1.82	1.80	6.07
Average EP		3.23	

Table 6  
Aquaculture monitoring dataset.

ID	Temperature	pH	Turbidity	Class
1	27	8	10	1
2	30	8	47	1
3	29	6	3	1
4	29	7	50	1
5	9	9	14	0
6	27	7	22	1
7	25	7	13	1
8	27	8	4	1
9	25	9	23	0
10	3	8	3	0
11	25	8	32	1
12	28	9	7	1
13	15	10	23	0
14	13	11	4	0
15	15	11	8	0
16	27	6	14	1
17	27	7	39	1
18	63	12	92	0
...	...	...	...	...
2000	45	14	59	0

modeling process are a learning rate of 0.0005, hidden layer 1 consisting of 32 neurons, hidden layer 2 consisting of 64 neurons, hidden layer 3 consisting of 64 neurons, the input layer consisting of 3 features, and the output layer consisting of 2 classes. The experiment was carried out by conducting data training, also called a variation of 20 epochs every 10 epochs from epoch 10 to epoch 200. From the results of the epoch process, the loss was also obtained as an indicator of how much error the model produced. The accuracy of the epoch is shown in Fig. 5, and the loss for each epoch is shown in Fig. 6.

In Fig. 5, it can be seen that epoch 10 has the lowest accuracy of 90.10%, whereas epochs 120, 140, 170, 180, 190, and 200 have the highest accuracy of 98.80%. At epochs 150 and 160, there was an anomaly in the form of a decrease in accuracy to 96.90 and 96.00%, respectively. As

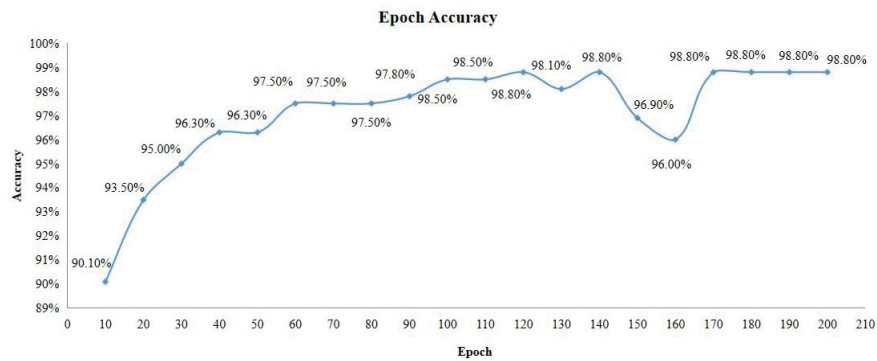


Fig. 5. (Color online) Epoch accuracy.

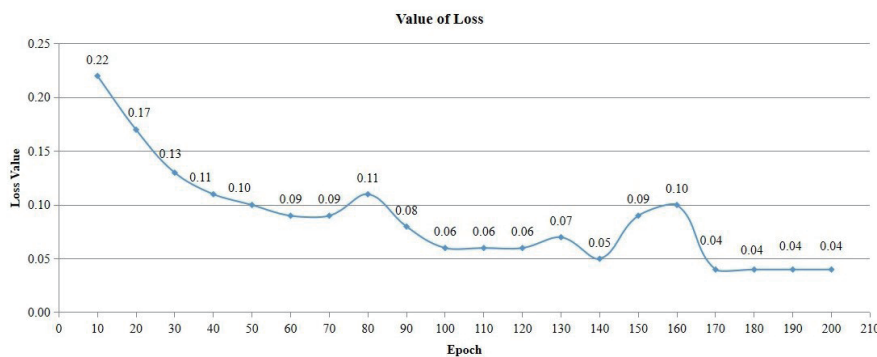


Fig. 6. (Color online) Loss of each epoch.

shown in Fig. 6, the highest loss of 0.22 was obtained at epoch 10 and the lowest loss of 0.04 was obtained at epochs 170, 180, 190, and 200. On the basis of the results of epoch analysis and the value of loss, the optimal point is obtained by looking at the highest accuracy along with the lowest value of loss and seeing how effective the memory is used in the epoch process. In this case, the optimal epoch is epoch 170, which has the highest accuracy of 98.80% and the lowest value of loss of 0.04. The visualization of data distribution based on epoch 170 training can be seen in Fig. 7, which shows that the classification is distributed between the optimal class or 1 (green) and the non-optimal class or 0 (yellow). When observed in more detail, there are only a few incorrect data values (red and orange), which in this case represents the value of loss.

### 3.3 Practical experiment using aquaculture monitoring system

Models that have undergone the training and evaluation process are then integrated into the DRL system. To facilitate communication between the user and the system, a user interface that includes parameter indicators of temperature, turbidity, and pH for fish pond conditions is designed. The results of data analysis are in the form of actions performed by agents on the DRL system through the binary classification of conditions indicating whether the pond conditions are optimal or not optimal. Figure 8 shows the user interface, wherein the application development uses an Android platform that can be easily installed and used by users.

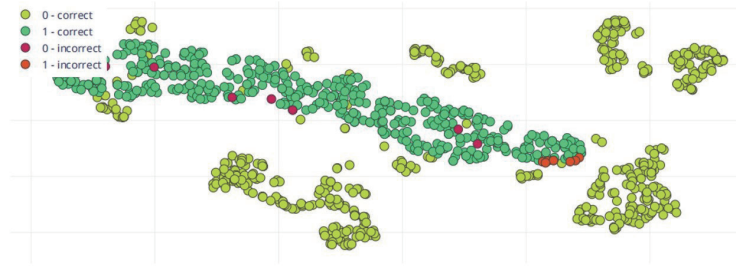


Fig. 7. (Color online) Visualization of data distribution.

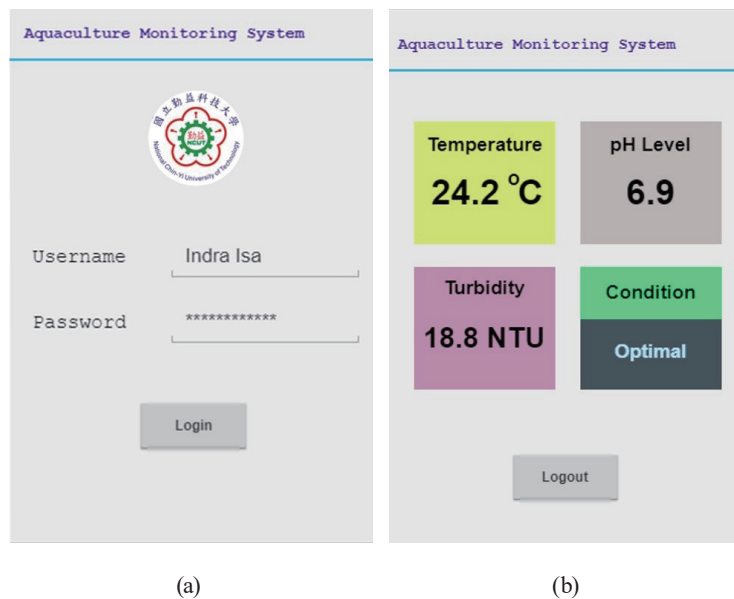


Fig. 8. (Color online) User interface of system. (a) Main menu and (b) monitoring menu.

The DRL system was tested by creating test scenarios. There were 30 test scenarios with different variations of temperature, turbidity, and pH, as shown in Table 7. The test scenarios contain an expected value and a real value where the expected value is the ideal temperature, turbidity, and pH. Moreover, the real value is the real-time measurement result of the aquaculture monitoring system. Of the 30 test results, there is one condition that does not match the expected value at ID 18. It can be seen that the expected value is “not optimal”, whereas the real value indicates “optimal”. This happens because the temperature is close to the specified threshold, which is between 25 and 30 °C, resulting in a bias or misinterpretation between the sensor readings and the DRL system. In the experiment, the accuracy percentage is 97%, which results from 29 of the correct test results of the aquaculture monitoring system (as shown in the “real value” column in Table 7) divided by 30 scenarios of expected values.

Table 7  
Experimental results of aquaculture monitoring system.

ID	Temperature	pH	Turbidity	Expected value	Real value
1	28	6	32	optimal	optimal
2	26	9	25	optimal	optimal
3	27	6	3	optimal	optimal
4	27	8	23	optimal	optimal
5	28	7	7	optimal	optimal
6	25	7	9	optimal	optimal
7	25	6	22	optimal	optimal
8	27	7	14	optimal	optimal
9	29	8	25	optimal	optimal
10	29	9	37	optimal	optimal
11	25	6	47	optimal	optimal
12	30	8	32	optimal	optimal
13	25	6	49	optimal	optimal
14	29	7	49	optimal	optimal
15	28	9	20	optimal	optimal
16	27	2	33	not optimal	not optimal
17	25	3	47	not optimal	not optimal
18	31	7	40	not optimal	optimal
19	16	5	3	not optimal	not optimal
20	7	3	45	not optimal	not optimal
21	14	2	11	not optimal	not optimal
22	19	8	52	not optimal	not optimal
23	12	9	79	not optimal	not optimal
24	2	8	72	not optimal	not optimal
25	12	8	91	not optimal	not optimal
26	44	8	17	not optimal	not optimal
27	33	7	26	not optimal	not optimal
28	25	11	42	not optimal	not optimal
29	42	8	92	not optimal	not optimal
30	31	7	66	not optimal	not optimal

#### 4. Conclusions

In this research, we developed an aquaculture monitoring system that is convenient to users, especially fish cultivators. The system was developed by integrating WSNs and DRL systems, which produce optimal accuracy. From the validation results of the accuracy of WSNs in detecting fish pond conditions using the parameters of temperature, pH, and turbidity, an average error percentage of 3.23% was obtained. In contrast, in selecting the model to be implemented on the DRL system, the optimal accuracy was obtained at epoch 170 with an accuracy of 98.80% and a loss function of 0.04. The experimental results of 30 tests on the aquaculture monitoring system device implemented with WSNs and DRL show that an accuracy of 97% was obtained. As the next development, we propose that the aquaculture monitoring system be integrated with an actuator that can provide a physical response to the fish area, such as an actuator to automatically spray feed, or the addition of physical sensors to increase the complexity of the aquaculture monitoring system.

## Acknowledgments

The authors would like to thank the Department of Electronic Engineering of National Chin-Yi University of Technology for supporting this research.

## References

- 1 Food and Agriculture Organization of The United Nations: <https://www.fao.org/state-of-fisheries-aquaculture/2020/en> (accessed July 2023).
- 2 A. Avadí, S. M. Cole, F. Kruijssen, M. H. Dabat, and C. M. Mungule: *Aquaculture* **547** (2022) 2022 <https://www.doi.org/10.1016/j.aquaculture.2021.737494>
- 3 P. Shi, G. Li, Y. Yuan, and L. Kuang: *Sensors* **18** (2018) 11. <https://www.doi.org/10.3390/s18113851>
- 4 O. S. Egwuche, O. S. Adewale, S. A. Oluwadare, and O. A. Daramola: *Proc. 2020 Int. Conf. Mathematics, Computer Engineering and Computer Science (ICMCECS, 2020)* 1–7. <https://doi.org/10.1109/ICMCECS47690.2020.240887>
- 5 W. T. Sung, I. G. T. Isa, and S. J. Hsiao: *Electron* **12** (2023) 9. <https://doi.org/10.3390/electronics12092032>
- 6 L. Wang, X. Yue, H. Wang, K. Ling, Y. Liu, J. Wang, J. Hong, W. Pen, and H. Song: *Remote Sens.* **12** (2020) 1. <https://doi.org/10.3390/rs12030402>
- 7 C. Tayo, N. D. Perez, and J. Villaverde: *Proc. 2022 Int. Conf. Electrical, Computer, and Energy Technologies (ICECET, 2022)* 1–6.
- 8 T. N. Huy, K. N. Tuan, and T. T. Trung: *Proc. 2019 Int. Conf. Research in Intelligent and Computing in Engineering (RICE, 2019)* 975–979.
- 9 A. D. R. Torres, D. S. Andreiana, Á. O. Roldán, A. H. Bustos, and L. E. A. Galicia: *Appl. Sci.* **12** (2023) 1. <http://doi.org/10.3390/app122312377>
- 10 K. Sivamayil, E. Rajasekar, B. Aljafari, S. Nikolovski, S. Vairavasundaram, and I. Vairavasundaram: *Energies* **16** (2023) 1. <http://doi.org/10.3390/en16031512>
- 11 C. T. Lee and W. T. Sung: *Electronics* **11** (2022) 1. <http://doi.org/10.3390/electronics11060928>
- 12 Z. Q. M. Ali and S. T. Hasson: *Proc. 2022 4th Int. Conf. Advanced Science and Engineering (ICOASE, 2022)* 55–59.
- 13 V. Anantharam: *Syst. Control Lett.* **169** (2022) 1. <http://doi.org/10.1016/j.sysconle.2022.105382>
- 14 Food and Agriculture Organization of The United Nations (FAO Fisheries Training): [https://www.fao.org/fishery/docs/CDrom/FAO\\_Training/FAO\\_Training/General/x6709e/x6709e02.htm#top](https://www.fao.org/fishery/docs/CDrom/FAO_Training/FAO_Training/General/x6709e/x6709e02.htm#top) (accessed July 2023).
- 15 Y. Matsuo, Y. LeCun, M. Sahani, D. Precup, D. Silver, M. Sugiyama, E. Uchibe, and J. Morimoto: *Neural Networks* **152** (2022) 267. <http://doi.org/10.1016/j.neunet.2022.03.037>
- 16 K. Nantomah: *Asia Math.* **3** (2019) 1. <http://doi.org/10.2307/3602191>
- 17 S. Afaq and S. Rao: *Int. J. Sci. Technol. Res.* **9** (2020) 485. <https://www.ijstr.org/final-print/jun2020/Significance-Of-Epochs-On-Training-A-Neural-Network.pdf>
- 18 A. B. Semma, M. Ali, M. Saerozi, Mansur, and Kusriani: *J. Electr. Eng. Comput. Sci.* **29** (2023) 1719. <http://doi.org/10.11591/ijeecs.v29.i3>

## About the Authors



**Wen-Tsai Sung** (Member, IEEE) received his M.S. and Ph.D. degrees from the Department of Electrical Engineering, National Central University, Taoyuan City, Taiwan, in 2000 and 2007, respectively. He is a distinguished professor in the Department of Electrical Engineering, National Chin-Yi University of Technology, Taichung, Taiwan. Dr. Sung won the 2009 Journal of Medical and Biological Engineering (SCIE index) Best Annual Excellent Paper Award (Ph.D. Dissertation) and the Dragon Thesis Award (Master Thesis) sponsored by Acer Foundation. In October 2023, he was recognized as one of the world's top 2% scientists (Career Impact) (2001–2023) by Mendeley Data, a company owned by Elsevier. He has authored over 100 peer-reviewed journal papers and articles indexed by SCI/SCIE/EI. His research interests include artificial intelligence Internet of Things, intelligent computing and systems, and wireless sensor networks. ([songchen@ncut.edu.tw](mailto:songchen@ncut.edu.tw))



**Indra Griha Tofik Isa** received his B.S. degree in informatics engineering from Pelita Bangsa University in 2011 and his M.S. degree in computer science from STMIK LIKMI Indonesia in 2014. He is currently a lecturer at Sriwijaya State Polytechnic and is continuing his Ph.D. studies at National Chin-Yi University of Technology (NCUT) in the fields of electrical engineering and computer science. His research interests are in deep learning, neural network applications, data science, image processing, and AIOT.



**Sung-Jung Hsiao** is an associate professor in the Department of Information Technology, Takming University of Science and Technology. He received his B.S. degree in electrical engineering from National Taipei University of Technology, Taiwan, in 1996, his M.S. degree in computer science and information engineering from National Central University, Taiwan in 2003, and his Ph.D. degree in electrical engineering from National Taipei University of Technology, Taiwan in 2014. He has work experience in research and design at Acer Universal Computer Co., Mitsubishi, and First International Computer. ([sungjung@gs.takming.edu.tw](mailto:sungjung@gs.takming.edu.tw))