Received April 12, 2022, accepted May 16, 2022, date of publication May 23, 2022, date of current version June 3, 2022.
Digital Object Identifier 10.1109/ACCESS.2022.3177225

A Robust Internet of Things-Based Aquarium Control System Using Decision Tree Regression Algorithm

MAMAN ABDUROHMAN[1], AJI GAUTAMA PUTRADA[1], (Member, IEEE), AND MUSTAFA MAT DERIS[2]
[1]School of Computing, Telkom University, Bandung 40257, Indonesia
[2]Department of Informatics, University Tun Husein Onn Malaysia (UTHM), Batu Pahat 86400, Malaysia
Corresponding author: Maman Abdurohman (abdurohman@telkomuniversity.ac.id)

This work was supported by Telkom University.

ABSTRACT The development of the Internet of Things (IoT) has shown significant contributions to many application areas, such as smart cities, smart homes, and smart farming, including aquarium control systems. Important things in an aquarium system are the level of ammonia in the water and the temperature of the water. Other research proposes several systems to make the aquarium control system robust for the aquarium monitoring and control system. However, those systems have weaknesses; namely, the user must actively access information to the server. This paper proposes a robust aquarium control system using the decision tree regression (DTR) algorithm. The development of this system was to overcome the problem of aquarium control by remote users. An accurate and real-time system is needed to monitor the aquarium so that it does not reach dangerous and critical points, such as in the case of an increase in water temperature. We did tests by developing an aquarium system connected to a server and an application that acts as a controller. Our measurements check the delay of sending data from the sensor to the server, process delay, actuator delay, user delay, and delay in reaching the aquarium's critical point. The measurement of the system's robustness is by calculating the probability of the information arrival to the user in the period of the critical point compared to the time needed to reach the critical point. Furthermore, we also made an analytical model based on the probability density function of the delay covered in this system. Analytically and experimentally, we show that the system can meet the needs of aquarium monitoring and control in an IoT-based environment.

INDEX TERMS Internet of Things, analytical model, probability density function, robustness, aquaculture, decision tree regression.

I. INTRODUCTION

The development of the IoT in recent years has shown a significant increase. The system advanced with the support of machine-to-machine (M2M) communication platforms that have developed in the past decade [1]. These communication platforms' development was to connect multiple devices, sensors, and actuators. At the above level, there are studies regarding middleware that communicates hardware with IoT applications [2].

One application developed in the study is a fish maintenance system that requires remote monitoring and control. Maintenance of fish requires a good system so that the owner can know the condition of fish even in a mobile position. Not every time a user can monitor the condition of ornamental fish that one has. Monitoring is needed to ensure that the fish's environmental and food conditions meet adequacy standards. Users need certainty that the monitoring and control are going well. Some considerations are the adequacy of fish feed and the water conditions where the fish reside. Water temperature is an important factor to control so that fish can live in a good water environment. Drastic changes in water temperature can cause fish to enter crucial and unstable conditions. Under certain conditions, fish can no longer survive in an environment with temperatures outside the limits. A system that can automatically detect and regulate to keep water conditions

The associate editor coordinating the review of this manuscript and approving it for publication was Aneel Rahim.

within good limits can actively anticipate the mentioned conditions.

A fish pet monitoring and control system has been widely implemented, manually and using the IoT system. The use of manual systems for aquarium management is very time-consuming, and vulnerable to human error [3], [4]. For example, people neglect maintenance when they forget or are busy. Human sensitivity to changes in water temperature cannot be relied upon to ensure water conditions are in good condition. A system is needed to automatically monitor the aquarium's condition precisely to ensure that the condition of pet fish remains good and cherished.

Besides the manual system, several studies have used the IoT system for aquarium monitoring and control. In some previous papers, aquarium control with IoT has been proposed but has a passive system [5], [6]. Users must actively monitor the condition of the aquarium. This passive system is not practical, while the user may be unaware of the usual hectic conditions, so controlling the aquarium's condition becomes missed. Users must monitor the aquarium's condition without knowing the aquarium's precondition. When conditions are good, access to aquarium information is not very useful except for just knowing. Conversely, delays in accessing information from the aquarium will cause fatal events in crucial conditions. An IoT system is needed that provides information and actively informs the aquarium's condition, especially in crucial conditions.

This research builds a smart aquarium management system, a push and mobile system, and an aquarium monitoring system. The intended push system is a system that actively sends information to the server, which forwards it to the user in critical conditions or normal conditions. With this push system, the user will obtain information automatically without actively accessing the web. Another advantage is that this system can be accessed on mobile so that users can easily access information on the aquarium's condition.

Our main contributions are summarized as follows:

1) We present a robust IoT-based aquarium control system using a decision tree regression (DTR) algorithm that processes the sensor data in a server, which is important to predict the condition of the aquarium system. The system automatically performs the actions needed when the fish environment meets critical conditions such as high temperature. The system actively sends sensor data to the server to be forwarded to the user in critical and normal conditions. Users will obtain information automatically through this push system without actively accessing the web.
2) We introduce the critical analysis on the possibility of information delay from the sensor system to the server, server processes, and information delay from the server to the actuator by observing the processes on the server.
3) We suggest a review of the delay in arriving information from the system to the user, ensuring that the aquaculture condition does not meet the critical phase. Research [3]–[5], and [6] have not discussed this in detail. We place the details of this analytical model based on the delay probability density function in Appendix A.
4) Another advantage of this paper is the server process delay, which previous studies have not calculated, such as [6].

The organization of the remainder of this study is as follows. In Section 2, we discuss the related works. Then Section 3 explains the testing environment of aquaculture and reports the results of testing the aquaculture. Section 4 contains the evaluation of the system's performance. Finally, Section 5 concludes our research.

II. RELATED WORK

A. APPLICATIONS OF INTERNET OF THINGS

IoT is one of the most trending issues in the ICT field. With the presence of IoT, the Internet has become increasingly pervasive. Recent studies have discussed IoT in various aspects, including agrarian, air pollution, smart cities, sports, vehicles, shopping, disasters, and electricity. In the agricultural field, there is a study of the plant wall system [7] where IoT is used to automate the monitoring and controlling activities of the plant wall. In the field of air pollution is a study of air pollution monitoring systems in a smart city [8] where the IoT system in the study involved 2500 devices spread across 29 different countries. An example of IoT research in smart cities addresses issues regarding vertical IoT platforms [9] where the research offers a multi-platform concept. This communication protocol can connect different platforms. In sports is a study of cycling [10] where accelerometers and gyroscope sensors in an inertial measurement unit (IMU) sensor provide various information needed by the cyclist to measure the quality of a completed exercise. In the field of vehicles is a study of car parking [11] where the solutions offered by the research are classified as reinforcement learning because this paper offers the optimum parking search solution from the available parking choices. In the field of shopping is a study of shopping cart [12] where IoT overcomes long queue problems in supermarkets by doing wireless billing when shopping carts go to the cashier. In the field of disaster is a study of flood prediction using IoT [13] where climate sensors from weather agencies around the world work together and 11 features for machine learning were collected. In the electric field, one of the studies on Smart Grid is the detection of damaged Smart Meters with edge computing and anomaly detection [14].

With various IoT products hitting the market, a new threat emerges, the security risk. Unlike other cyber fields, because IoT is still in its infancy, cyber risk assessment of IoT seems to be still not ready. Several studies have analyzed to calculate the performance of existing risk analysis and how to determine the optimum risk analysis.

B. SMART AQUARIUM

Several papers have researched IoT-based aquaculture using several sensors and actuators. Chen *et al.* [3] focused on the

TABLE 1. **IoT solutions for smart aquariums.**

Item	Haiyunnisa [3]	Tseng [4]	Raju [5]	Angani [27]	FishTalk [6]	Proposed Research
1. Sensors	pH, DO, Water Level, Temp	pH, DO, Temp, Water Level	DO, Salt, NH3, Nitrite, Temp, pH, Alkalinity	Temp, DO, Ultrasonic, pH	pH, EC, DO, TDS, Water Level, Temp	Temp, DO, EC
2. Actuators	Heater, Light, Feeder, Air Pump	No	Light	Solenoid Valve, Pump	Feeder, Fan, Heater, Light, Air Pump, RO Filter	Heater, Fan, Relay, RO Filter
3. Actuators Controlled by Sensors	Yes (simple threshold)	No	No (light is always on at night)	Yes	Yes	Yes
4. Smart Feeder	No (manual)	No	No	No	Yes	Yes
5. Control Board	MSP430	Arduino UNO	Raspberry Pi 3	Raspberry Pi	ESP8266, ESP12F	ESP8266
6. MQTT Communication	No	No	No	No	No	Yes
7. Prediction Server	No	No	No	No	No	Yes
8. Sensor-server-actuator delay measurement	No	No	No	No	Yes	Yes
9. Server process delay and user delay measurement	No	No	No	No	No	Yes
10. Android-based mobile application	No	No	No	No	No	Yes
11. iOS-based mobile application	No	No	No	No	No	Yes
12. MQTT	No	No	No	No	No	Yes

Dissolved Oxygen (DO) quality control system in aquariums. For input to the aquarium system, it uses a DO sensor. For actuators, the system uses a microbubble device. The sensors used by Tseng *et al.* [4] were water level sensors, pH sensors, and DO sensors.

Other studies on IoT-based smart aquariums utilize more specific sensors, resulting in expanded findings. Raju and Varma [5] also used sensors to monitor aquarium environments. In addition to the pH, DO, and temperature sensors, the system built is also equipped with nitrate, alkalinity, salt, and ammonia sensors. However, the system built does not have actuators that work automatically. Alerts arrive with instructions that the user must carry out.

Some studies add other useful functions to the aquaculture system. The theme of research by Angani *et al.* [15] was how to recycle water in an aquarium without wasting water. The sensors used are DO, pH, temperature, and water level sensors. The actuators used are solenoid valves to open waterways and pumps.

Some research conducted a comprehensive evaluation of the IoT system applied to aquaculture. FishTalk [6] used pH, EC, DO, total dissolved solids (TDS), water level, and temperature sensor. The actuators used are a fish feeder, fan, heater, light, air pump, and reserve osmosis (RO) filter. In addition to the completeness of sensors and actuators, what distinguishes this study from other smart aquarium studies is the measurement of delay. This research ensures probable and statistical delay measurement by conducting proper sensing and actuating. The Erlang and gamma distribution measurements prove that the system delay will not harm fish rearing. A research opportunity is to add temperature forecasting to enhance the delay safety of the system further. Table 1 shows a comparison of all the papers reviewed. The table highlights the contribution given by the proposed system.

C. WATER TEMPERATURE FORECASTING

Some researchers have implemented forecasting using machine learning techniques and applied them to environmental fields. East Asia Winter Monsoon (EAWM) forecasting used partial least square (PLS) regression [16]. Global solar irradiation forecasting used two persistent models with four types of regression tree [17]. Heatwave forecasting in Pakistan uses quantile regression forest [18].

In addition, other researchers have also applied forecasting in energy, economy, and electricity. Electricity price and load forecasting use an enhanced convolutional neural network (ECNN) [19]. District heating and cooling (DHC) forecasting used an online ensemble decision tree-neural network (DCNN) learning [20]. Copper price forecasting used the decision tree classification, a method also useful for economics [21]. Electric power load forecasting used a decision tree type reduced error pruning tree (REPTree) [22].

Some researchers have implemented forecasting at water temperature using machine learning. Research can be found regarding forecasting water temperature using a method called the non-linear regression model (NRM), which is superior to RBF and SVM [23]. A hybrid empirical mode-decomposition-back-propagation neural network (EMD-BPNN) method is applied to a prawn engineering culture pond [24]. A genetic algorithm-optimized long short-term memory (GA-LSTM) is used for forecasting urban water quality management [25]. A hybrid support vector regression-fruit fly optimization algorithm (SVR-FOA) method is used for forecasting river flow prediction [26]. The research gap found is to implement a forecasting method for aquaculture systems.

III. TESTING ENVIRONMENT

A. AQUARIUM SENSORS

The sensors used in this research are a Waterproof Temperature Sensor DS18B20, a TDS Sensor, and A DFRobot: DO sensor. Waterproof Temperature Sensor DS18B20 is a type of sensor to detect the ambient temperature in the water, as seen in Figure 1 section (a). A TDS Sensor measures the level of cleanliness of the water in an aquarium, as seen in Figure 1 section (b). DFRobot: Dissolved Oxygen Sensor is a low-power sensor that is compatible with microcontrollers, as seen in Figure 1 section (c).

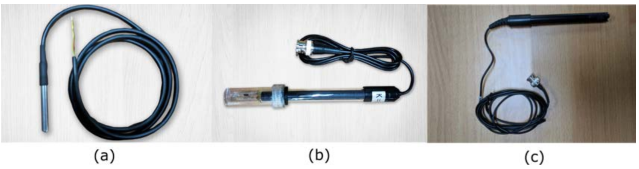

FIGURE 1. The aquaculture sensors: (a) Waterproof temperature sensor (b) TDS sensor (c) Dissolved oxygen sensor.

B. AQUARIUM ACTUATORS

Actuators used in this research are an HB-100 Waterheater, an FS-120 Fan, a 5 V Relay, an RO Filter, and a Feeder. The HB-100 Waterheater is a water heater that can help heat aquarium water with a volume range of 50-100 Liters, as seen in Figure 2 section (a). FS-120 Fan is a special fan for aquariums that has dimensions of 172 × 120 × 120 mm, a frequency of 50/60 Hz, and a power of 15 W, as seen in Figure 2 section (b). A 5 V Relay controls high-power devices with low-power microcontrollers. An RO filter filters water molecules in contaminated aquariums to yield clearer and cleaner water. Automatic fish feeders (or briefly, feeders) are controlled and triggered directly by the system based on need and is, as seen in Figure 2 section (c).

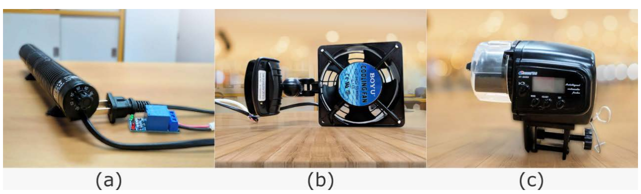

FIGURE 2. The aquaculture actuators: (a) Waterheater (b) Fan (c) Automatic fish feader.

C. IoT ENVIRONMENT

The proposed system uses an IoT Architecture in which the system consists of three layers, namely, the end device, server, and application [28]. The end device, in this case, is the aquarium. All sensors, actuators, and the aquarium are part of the end device. A microcontroller regulates the control of the aquarium, and in this study, the microcontroller used is an ESP8266. The end device communicates with the server and application via a message queue telemetry transport (MQTT) Broker wirelessly [29]. Wireless communication is the responsibility of the Wi-Fi module, which is part of the ESP8266.

The server used is a Python Server. Communication between the Server and End Device uses MQTT. The task of the Python server is to receive temperature data, make predictions, deduce predictions into decisions, and send the results of decisions. A DTR model carries out the prediction, where the DTR model goes through DTR training according to machine learning rules. Data received from and sent to the server come from the MQTT broker. Application is software that runs on the user's smartphone. An overview of the whole system is viewable in Figure 3.

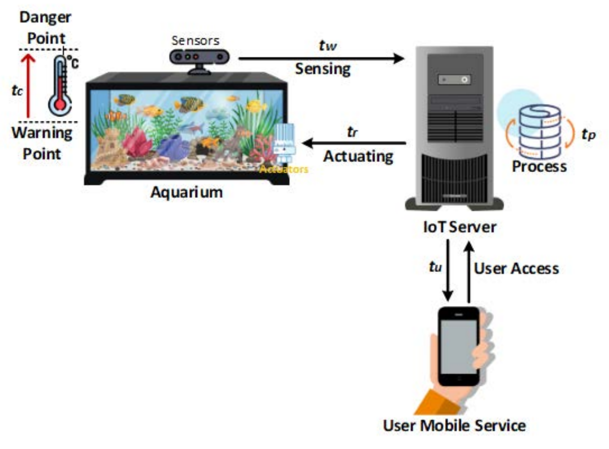

FIGURE 3. The proposed system block diagram.

The system's workflow is as follows: First, the sensor on the end device will send temperature data to the MQTT broker. The server and the application will receive this data. The application will display temperature information in the form of monitoring data. The server will receive data and then make predictions. The prediction results will then determine a decision. The decision is about which actuators should activate and which should not. The decision transmits to the MQTT broker. Both the end device and the application will receive the decision data. The end device will set the actuator according to the decision, while the application will show each actuator's status according to the decision.

D. DECISION TREE REGRESSION FORECASTING

A Decision Tree is one of the Machine Learning techniques used for classification and regression. In operating, the Decision Tree works like a flowchart. The feature values go through a tree-like model then the branches in the model will direct these values towards a final decision. Each branch contains conditions that resemble the concept of if-else.

The Decision Tree model is a result of training data processing. One of the most famous types of Decision Trees, namely, ID3, conducts training in several stages. The first is calculating the Entropy of each feature. Entropy is the value of diversity. The higher the value of Entropy is, the more diverse the data. The maximum value of Entropy is 1. After Entropy, the next stage is to calculate the information gain value. The information Gain value indicates which feature most influences the output. The higher the value of Information Gain, the greater the influence of features. The final step is to arrange the tree based on each feature's Information Gain and Entropy value.

The resulting Decision Tree model goes through a testing process using the testing data. Testing data is data that has been collected and has an output. Testing data cannot be the same as the training data. The Decision Tree model's performance measurement is with the model's accuracy in producing output from testing data. Accuracy is comparing the output of the testing data with the actual output.

There are two types of Decision Trees, classification and regression. If used for classification, the input of the decision tree is usually the attributes of an event. An example of classification is determining a person's sex from the person's height, shoe size, weight, and chest circumference. When used for regression, input data is usually a time series event or continuous value.

DTR in the proposed system is used to predict the next temperature data so that the system becomes more responsive than conventional systems. DTR is a type of supervised learning, a machine learning trained based on labeled data. The data used for training and testing are temperature data taken in real-time for approximately 24 hours. These data separate into training and testing data by using 80% of the data for training and 20% for testing. The model is exportable for further use. An explanation of the process of Decision Training is in Algorithm 1. SSE in the algorithm is the sum of squared error or all differences in the values of array members with the average value of the squared array members.

The result of DTR training is a model. This model is then exported and embedded in the Prediction Server, where the model's input is data entering the Prediction Server. The model then processes the input to yield the output. The model's output is the predicted temperature, which will then be classified. There are three classification results classes: LOW, HIGH, and NEUTRAL. LOW will turn on the heater, while HIGH will turn on the fan. NEUTRAL will deactivate both. The results of this classification transmit back to the End Device to control the temperature. Communication between the End Device and Server uses MQTT. To receive data from the MQTT Broker, Subscribe is used. The MQTT Publish protocol sends data to the MQTT broker. The explanation of the complete process of the prediction server is in Algorithm 2. Based on the tests that have been carried out, it is obtained a temperature dataset taken for 1 week which shows fluctuations in temperature changes as shown in Figure 4. The data obtained in this test is used as training data to predict the next temperature value.

Algorithm 1: Decision Tree Regression Training

```
Data: Training Data
Result: Decision Tree Regression Model
Procedure Split (Training Data)
    n = sizeof(Training Data);
    for i in (n) do
        for j in (0,i) do
            Low[i].append(Training Data[j])
        for j in (i,n) do
            High[i].append(Training Data[j])
        LowLabel = average(Low[i]);
        HighLabel = average(High[i]);
        SquaredLoss[i] = SSE(Low[i]) + SSE(High[i]);
    choice = min(SquaredLoss);
    /* recurse split with chosen Low
       and chosen High */
    Split (Low[choice]);
    Split (High[choice]);
```

Algorithm 2: Decision Tree Regression Prediction Server

```
Data: Temperature
Result: Decision
Procedure PredictionServer (Temperature)
    Temperature = mqtt.Subscribe();
    Prediction = DTRModel.Predict(Temperature);
    if Prediction < Min Temperature then
        Decision = "LOW";
    else if Prediction > Max Temperature then
        Decision = "HIGH";
    else
        Decision = "NEUTRAL";
    mqtt.Publish(Decision);
```

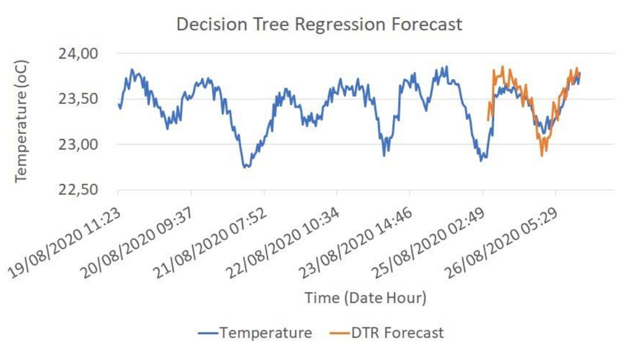

FIGURE 4. DTR forecast.

The performance test compares a generated temperature prediction data set with real data. Figure 4 shows predictive data and real temperature data. The model's accuracy is measured based on the match between the predicted temperature data and the real temperature data.

Prediction accuracy is measured using three methods, namely the Root Mean Squared Error (*RMSE*) [30], Mean Absolute Percentage Error (*MAPE*) [31], and R Squared (R^2) methods [32]. These three methods are well-known methods to assess the accuracy of predictions. The following are the formulas for the three methods.

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \tilde{y}_i)^2} \tag{1}$$

where n is number of data, y_i is observed values, and $\tilde{y}_i$ is predicted values, and

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}|PE_i| \tag{2}$$

where n is number of data, PE_i is the percentage of error of two variables, and

$$R^2 = \left(\frac{n(\sum xy) - (\sum x)(\sum y)}{\sqrt{(n\sum x^2 - (\sum x)^2)(n\sum y^2 - (\sum y)^2)}}\right)^2 \tag{3}$$

where n is number of dataset, x is first variable, and y is second variable in the context.

The calculation results obtained show the *RMSE* value of 0.15, *MAPE* value of 0.52% and R^2 of 84.52% which is shown in Figure 5.

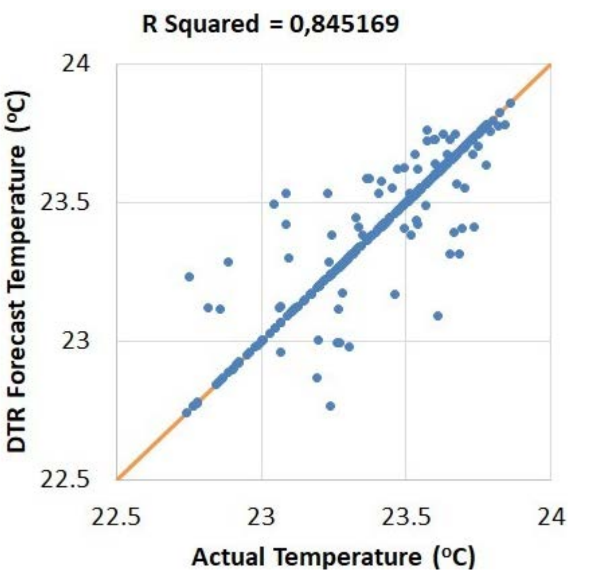

FIGURE 5. R squared value.

The *RMSE* and *MAPE* measurement results indicate that the difference between predictive data and real data is low—the closer to 0, the *RMSE* and *MAPE* value, the better. Meanwhile, the R^2 value is 84.52%, which shows the closeness of the prediction variable with the real value variable. It shows that the predicted value strongly correlates with the real value.

E. APPLICATION

Applications in the three layers of the IoT architecture are the user's domain. The application development in the proposed system was with Android Studio. Users can monitor the aquarium through the application. Users can also monitor the status of actuators. Figure 6 shows the User Interface (UI) of the created application. The Application UI consists of two UIs: the main UI and the Setting UI. The main UI is called the Monitoring UI.

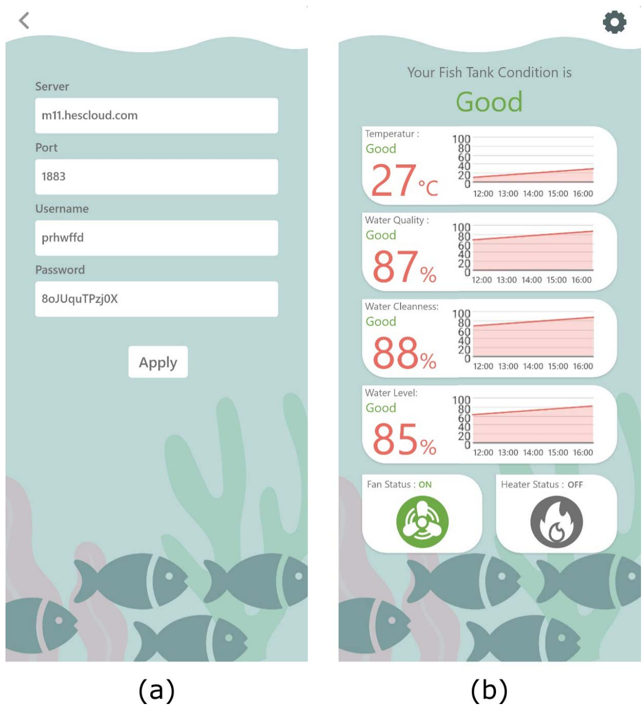

FIGURE 6. Mobile app user interface. (a) Setting. (b) Monitoring.

The main function of the Monitoring UI is to display the monitoring of aquarium values and the status of the actuators. The application monitors four aquarium values: temperature, water quality, water cleanness, and water level. There are two statuses of the actuator displayed, namely, Fan and Heater.

In addition, there is the status of each aquarium value. A ''Good'' text will appear for each value if all values are within the threshold range. If all values are good, there will be large ''Good'' text at the top of the Monitoring UI. If the aquarium value goes out of the threshold range, the text will change to ''Bad'' on monitoring the concerned value. If there is at least one monitoring value in the Bad status, then the large status text above will change to bad, then there will be an alert to the user.

Display Settings are for connecting to the MQTT broker. The settings view consists of each server, port, username, and password field that the user can fill in. Server and port are the IP address and port number of the MQTT broker used.

Username and password comprise an authentication system set up to access the MQTT broker securely.

All data obtained from the application are the contents of the MQTT broker published by the end device or server. The end device sends monitoring data, and the server sends actuator states.

F. TESTING

A performed testing is to measure the regression tree forecasting performance. Betta fish are in the test environment during testing, where the water condition of the betta fish becomes the test parameter. Betta fish live in water with a minimum temperature of 25 ° C and a maximum of 27 ° C. So the rules will be arranged to follow these restrictions. The methodology is as follows. First, the IoT system is set up and activated. Sensing data begins to collect, the computer network system will also be active, communication will establish, and sensing data will enter the forecasting system—the testing process applies testing data to the model. The output data from the model is named the prediction data. The prediction data is then entered into inference to determine the actuator action at the intended forecast time. The output of the inference is named decision data. Performance measurement is the next step. The measurement parameter is accuracy. The way to measure accuracy is to compare the forecast decision data with the actual decision data. The accuracy is 99.82 %.

IV. SYSTEM PERFORMANCE EVALUATION

A. IoT SYSTEM DELAYS AND MEASUREMENT

The system has gone through the previous test, which is a field testing by an empirical approach using an aquarium system connected to the IoT system. The protocol used in this test is MQTT using a Wi-Fi network. The test results show that the system can monitor and control the aquarium following the user requirements. The server response time to changes in the aquarium environment is good and can send information to the actuator following the processes carried out on the server. Likewise, information that reaches the user is still within the system's tolerance.

In addition to empirical testing, this section also discusses the development of an analytical model to measure the worst possible system failure in monitoring and controlling the aquarium from the user's perspective. In this case, there are several parameters used concerning time. The parameters are the time required for sending data from the sensor to the server (t_w) as a warning from the system to the user and the processing time on the server (t_p) (i.e., the time needed by the server to define the information received and determine the next action). The time the data are sent from the server to the actuator (t_r) in response to the information the server receives. Another delay analyzed is the time of sending data from the server to the user (t_u) and the deadline between the warning time and the critical point (t_C). The mentioned time is required to raise the aquarium water temperature from the initial warning point until it is critical. Figure 7 shows a timing diagram from the trigger sensor to the actuator and the user. When the $\tau_{t,0}$ sensor first detects that the system is entering a critical condition. For example, a damaged equipment event (such as a heater that turns on suddenly, which increases the water temperature). The sensor will notify the server by sending warning information. The server receives a warning when $\tau_{t,1}$ and then processes the information up to $\tau_{t,2}$. The server makes a decision and sends information to two destinations: the actuator and the user system. The actuator works when $\tau_{t,3}$, for example, the system turns off the heater. The deadline required from the start of the system to detect the possibility of a critical login system ($\tau_{t,0}$) to the time the active actuator ($\tau_{t,3}$) is shown as t_i time. At approximately the same time, the server sends the information to users, which reaches the user at $\tau_{t,u}$. The deadline required from the system to detect the possibility of a critical login system ($\tau_{t,0}$) until the information reaches the user application ($\tau_{t,u}$) is t_j time. While the system sends information to servers and users, the aquarium's condition will continue to lead to critical conditions. Critical conditions, which are undesirable, will be completely exceeded when it reaches the point $\tau_{t,4}$. Resulting in a controlled condition so that the system can restore normal conditions before a critical condition occurs.

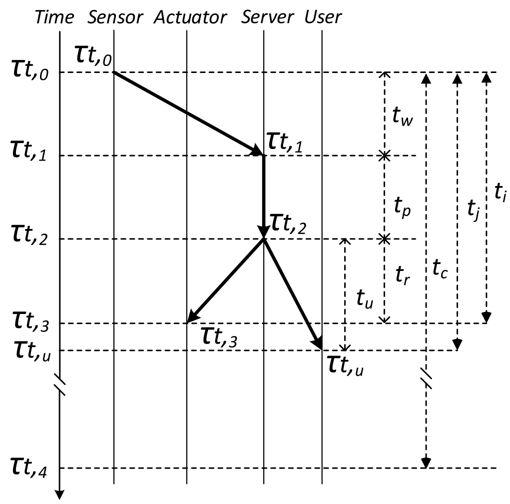

FIGURE 7. The proposed system time diagram.

Analytical models explore the worst possible conditions in IoT network connections and the effect on aquarium monitoring systems. The model creation is by looking at empirical delay data generated in the communication process between entities in the IoT system. The approximation of delay data is by a Probability Density Function, proven by using the Kolmogorov-Smirnov normality test. The developed analytical model references and compares empirical data on testing results. This model is important to ensure the system runs well by considering various possible system failures due to message delays and packet loss sent from one entity to another IoT entity. The theoretical proof is more reliable than just testing empirical data on the results of testing in the lab. By calculating these worst conditions, the system provides a certain level of confidence to overcome problems in the IoT communication network. The worst condition in question is the uncontrolled delay when a congested network occurs, or some functions are not running well, causing packet loss so that information does not arrive properly from one entity to another IoT entity.

Two situations become important concerns in this evaluation: the time needed to activate the actuator and the arrival of information to the user compared to the time needed to achieve critical conditions. Assuming good network conditions and worst network conditions, certain important variables that become a reference for this goal are the time for sending data from the sensor to the server (t_w), server processing time (t_p), time from the server to the actuator (t_r), and time from the server to the user (t_u). Unfortunately, there has been no testing for server time and information delivery to the user in previous studies, even though both are very important. Fortunately, as value added from our research, we have considered all delay variables, including server process time and sending message delay from the server to the user. The delay formula is as follows.

$$\tau_{t,3} = t_w + t_p + t_r \tag{4}$$

and

$$\tau_{t,u} = t_w + t_p + t_u \tag{5}$$

The testing process assumes that the server used is active with several running tasks. Server busyness is important to show that the system is running in a natural environment with many other tasks running on the server. This processing time is not negligible because it can affect the overall processing time, unlike the previous system that ignores it. The next evaluation explores the worst conditions in systems where the network is not running smoothly and there is a long delay in sending data. In this study, an analysis ensures that the worst process time can still guarantee a good control process.

The second situation is the arrival of information to users in different locations. Under certain conditions, the user can be in a complicated situation and not have time to check the aquarium system; therefore, that information must transmit from the system to the user. A push information system to users is an advantage of the system we have developed that previous papers have not discussed. There are several time components to consider: delay in the transfer of data from the sensor to server (t_w), server processing (t_p), and delay from the server to user (t_u). There is a comparison of this overall time to the probability of an aquarium environment critical condition. The evaluation of this second situation is under normal conditions and poor network conditions.

B. ANALYSIS OF THE WORST EFFECTS OF MESSAGE DELAY ON THE IoT NETWORK

The following is an analysis of the results of the test for the delay caused in sending data from one entity to another on the IoT network. For example, t_w is the delay of sending data from the sensor to the server, t_p is the processing delay

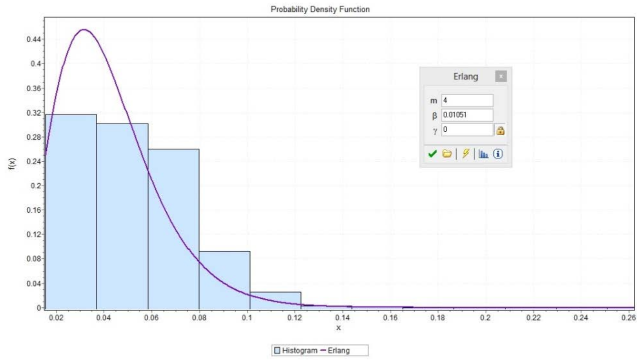

FIGURE 8. The delay t_p histogram.

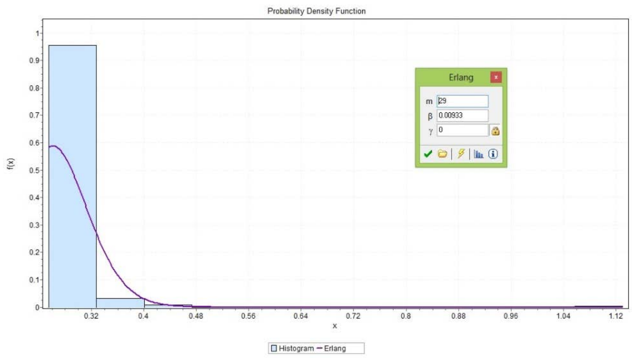

FIGURE 9. The delay t_r histogram.

of the server, t_r is the delay when the server sends data to the actuator, and t_u is the delay when sending data from the server to the user. We assume that the four t_w, t_p, t_r, and t_u are random variables with density functions $f_w(t_w)$, $f_p(t_p)$, $f_r(t_r)$, and $f_u(t_u)$. We have obtained histograms for t_w, t_p, t_r, and t_u with the results of 3000, 3041, 3000, and 3022 measurement data for Wi-Fi transmission delays, respectively. Based on the data obtained from the sample measurement, the results $t^*_{N,w}$, $t^*_{N,p}$, $t^*_{N,r}$, and $t^*_{N,u}$ obtained by Erlang distributions are the expected value $E[t^*_{N,w}] = 0.27704$ ms and the variance $V[t^*_{N,w}] = 0.00135$, expected value $E[t^*_{N,p}] = 0.05223$ ms, and the variance $V[t^*_{N,p}] = 0.0005491$, expected value $E[t^*_{N,r}] = 0.28036$ ms and the variance $V[t^*_{N,r}] = 0.00131$, expected value $E[t^*_{N,u}] = 0.28036$ ms and the variance $V[t^*_{N,u}] = 0.00131$. The histogram for each PDF with an Erlang distribution for each function $f_w(t_w)$, $f_p(t_p)$, $f_r(t_r)$, and $f_u(t_u)$ are shown in Figure 11, Figure 8, Figure 9, and Figure 10, respectively.

The Erlang density function with the shape parameter m and the scale parameter β is formulated as follows.

$$f_E(t, m, \beta) = \frac{\beta^m t^{m-1} e^{-\beta t}}{(m-1)!} \tag{6}$$

and

$$\int_{\tau=0}^{t} f_E(\tau, m, \beta) \mathrm{d}_\tau = 1 - \sum_{k=0}^{m-1} \frac{\beta^k t^k e^{-\beta t}}{k!} \tag{7}$$

where $E[t] = \frac{m}{\beta}$ and $V[t] = \frac{m}{\beta^2}$

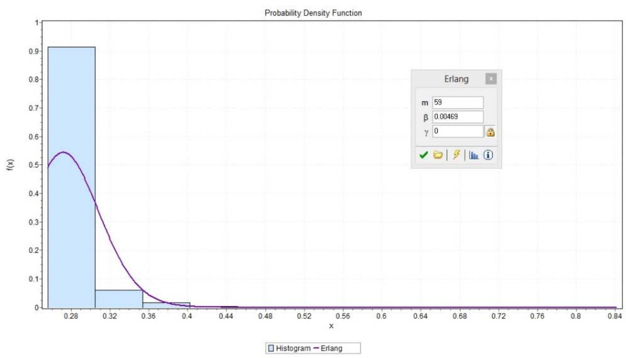

FIGURE 10. The delay t_u histogram.

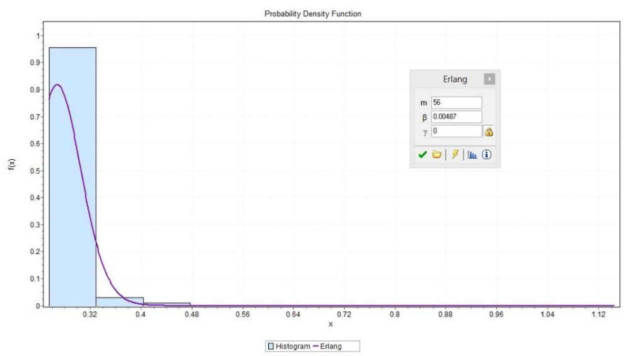

FIGURE 11. The delay t_w histogram.

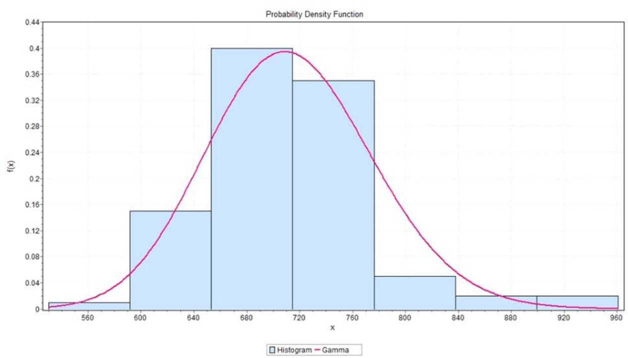

FIGURE 12. The delay t_C histogram.

From equation (6) an estimate $f_w(t^*_{N,w})$ can be obtained as $f_E(t^*_{N,w}, m^*_{N,w}, \beta^*_{N,w})$, where the shape parameter is $m^*_{N,w} = 56$ and the scale parameter is $\beta^*_{N,w} = 0.00487$. We approximate also some data as follows: $f_p(t^*_{N,p})$ as $f_E(t^*_{N,p}, m^*_{N,p}, \beta^*_{N,p})$, where the shape parameter is $m^*_{N,p} = 4$ and the scale parameter is $\beta^*_{N,p} = 0.01051$, $f_r(t^*_{N,r})$ as $f_E(t^*_{N,r}, m^*_{N,r}, \beta^*_{N,r})$, where the shape parameter is $m^*_{N,r} = 29$ and the scale parameter is $\beta^*_{N,r} = 0.00933$, and $f_u(t^*_{N,u})$ as $f_E(t^*_{N,u}, m^*_{N,u}, \beta^*_{N,u})$, where the shape parameter is $m^*_{N,u} = 59$ and the scale parameter is $\beta^*_{N,u} = 0.00469$. The PDF curve estimate is validated by the Kolmogorov-Smirnov suitability test.

As explained in the previous discussion, four units of time are taken into account in monitoring and controlling the aquarium system, namely, t_w, t_p, t_r, and t_u. For example, in certain situations, the water temperature shows a gradual increase resulting from faults, such as activating the heater or hot weather conditions from outside. This condition makes the water warmer, and the temperature continues to rise to a certain extent that is no longer good for the fish, called a crucial point; in this case, the water temperature reaches 25^o C. The system must be able to send information before it exceeds this crucial point. For example, at a temperature of 24^o C, the first time, the sensor detects the conditions that allow the necessary conditions in the aquarium ($\tau_{t,0}$). Then, the system sends a warning to the server with a t_w travel time. The server receives data from the aquarium when $\tau_{t,1}$. Upon receiving this information, the server then processes the information and determines the next action in a time of t_p, and the server sends a response on time $\tau_{t,2}$. Then, the response data sent reaches the aquarium system within t_r and arrives at the actuator when $\tau_{t,3}$. Simultaneously, the server also sends information to the user through a mobile phone application. The time it takes to send a message to the user is t_u. The message arrives at the user's mobile phone application when $\tau_{t,u}$. Meanwhile, the time needed by the system to reach a critical point is, for example, t_C, which is at $\tau_{t,4}$ time point. The time needed to achieve this critical condition is $t_C = \tau_{t,4} - \tau_{t,0}$. This control system is important to guarantee and ensure that the value of $t_C > t_w + t_p + t_r$ or $\tau_{t,4} > \tau_{t,3}$ and that the value of $t_C > t_w + t_p + t_u$ or $\tau_{t,4} > \tau_{t,u}$. The system can control the aquarium's condition by activating the actuator, as important as the arrival of the information to the user. Therefore, there are two controls by the system that is controlling the actuator directly and giving awareness to the user. On the other hand, the time needed to reach the crucial point we call t_C is the time when the system starts going towards the critical point ($\tau_{t,0}$) and to the critical point ($\tau_{t,4}$). Let $f_C(t_C)$ be the density function for t_C. We have empirically tested the temperature rise time to a critical point and obtained 100 data points with a histogram shown in Figure 12. The mark of the measured value is t^*_C. From these measurements, we can approach t^*_C with a Gamma distribution with an expected value $E[t^*_C] = 712.14s$ and the variance $V[t^*_C] = 3.902$. Therefore, $f_C(t^*_C)$ has the shape parameter $\alpha = 130.69$ and the scale parameter $\hat{A}\mu = 5.464$. The Kolmogorov-Smirnov suitability test validates the PDF curve estimate.

One important thing is to ensure that the system stays under control before it reaches a critical point that is dangerous for the fish. The deadline to reach the tipping point is t_C, while the time needed to get the information and activate the actuator is t_i, and the deadline for information to the user is t_j. A model is needed to determine the probability value to reach that critical point. The smaller the probability is, the better the system. So that the system does not reach a critical point that is dangerous for fish, it is necessary to ensure that $\tau_{t,4} > \tau_{t,3}$ or $t_C > t_i$ and that $\tau_{t,4} > \tau_{t,u}$ or $t_c > t_j$. In Appendix B,

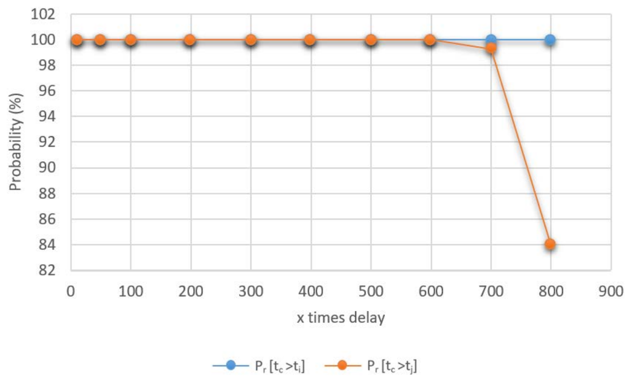

FIGURE 13. Effect of network delay to probability of $P_r[t_C > t_i]$ and $P_r[t_C > t_j]$.

we have derived the probability $Pr[\tau_{t,4} > \tau_{t,3}]$ or $Pr[t_C > t_i]$ as follows.

$$P_r[t_C > t_i] = 1 - \sum_{k=0}^{m_w+m_p+m_r-1} \binom{\alpha+k-1}{k} \left[\frac{\beta^k \mu^\alpha}{(\beta+\mu)^{\alpha+k}}\right]$$

Moreover, for the probability of $Pr[\tau_{t,4} > \tau_{t,u}]$ or $Pr[t_C > t_j]$ is as follows.

$$P_r[t_C > t_j] = 1 - \sum_{k=0}^{m_w+m_p+m_u-1} \binom{\alpha+k-1}{k} \left[\frac{\beta^k \mu^\alpha}{(\beta+\mu)^{\alpha+k}}\right]$$

As part of the proof of the analytical model developed, testing has been performed by sampling empirical data to prove the two equations. After several calculations with the time parameters t_C, t_i, and t_j, the conclusion is that the probability is extremely small and under 0.01%. It is the empirical and appropriate proof of the developed analytical model of the test environment. This test shows that the system is quite good at monitoring and controlling the aquarium system in an IoT environment. The next test is to anticipate poor network conditions that are assumed to be with a high increase in delay with expected values of $E[t_i] \gg E[t_i^*]$ and variance $V[t_i] \gg V[t_i^*]$, as well as the expected value of $E[t_j] \gg E[t_j^*]$ and variance $V[t_j] \gg V[t_j^*]$. We have tested by increasing the value of the delay significantly, as shown in Figure 13. The test results show that the new system is affected after the delay is increased 800 times with the values $Pr[t_C > t_i] = 99.99\%$ and $Pr[t_C > t_j] = 84\%$. This shows that the opportunity for the system to remain under control is still very high, even though the network condition is not good as indicated by a significant increase in delay. The system will continue to run well even if the IoT network condition is having problems increasing the message delivery delay.

In the section so far, the system testing has been presented empirically and mathematically. Both show consistently that the system can run stably to monitor and control the aquarium by using the IoT network, even with poor network conditions. In addition to the achievements we have found, there are several system limitations. This system is designed specifically for a certain aquarium size and for the temperature of a certain environment. External air conditions also affect the system.

Our tests have proven that the proposed system runs with an understandable delay, but there is still room for improvement in the future. The cutting-edge concepts offered are edge computing, where in the cloud-powered IoT architecture, intelligent computing shifts from the cloud to end devices. This concept demands a compact intelligent model solution. Several studies on the quantization of machine learning models already exist and can be adopted.

V. CONCLUSION

Our research proposes a novel IoT system with predictive functionalities to enhance the environment control of an aquarium and ensure delay tolerance, where the system adopts a decision tree regression (DTR) algorithm for prediction. The equipment involved, among others, are water temperature sensors, TDS sensors, DO sensors, a water heater, a fan, a relay, and an RO filter. The system also equips a Python server and an android-based application, where MQTT is the communication protocol. The test results show that the system can effectively send messages from sensors to the server, perform server processes to respond to information from sensors, and actively provide the necessary actions by controlling actuators. The advantage of this system is that the system can actively push information to the user when conditions are near critical. In addition, we have developed an analysis model to ensure data transmission between sensors, the server, actuators, and the user continues running well in high network delay conditions. It is a model of the sensor to the server, the server processing, the sensor to the actuator, and the user delay measurement. Abundant data is collected, namely, 3000, 3041, 3000, and 3022 delay data for t_w, t_p, t_a, and t_u, respectively. This analysis model shows that the possibility of the system entering a critical condition without control is quite little. It means that, most likely, the system can control properly even though the network delay conditions are quite high. For a delay of 800 times normal conditions, the chances of a controlled system remain high, namely, $P_r[t_C > t_i] = 99.99\%$ and $Pr[t_C > t_j] = 84\%$. Under high delay conditions, the system can still activate the actuator and convey information to the user's application. The system works well in a static environment at a certain temperature with a certain aquarium water volume. Finally, the contributions of our research are a robust IoT-based aquarium control system using a DTR algorithm and a critical analysis of server process delay that previous studies have not calculated.

APPENDIX A

In this appendix, the probability formula is derived $P_r[\tau_{t,4} > \tau_{t,3}]$ or $P_r[t_C > t_i]$. Here, it is clear that $P_r[t_C > t_i] = P_r[t_C > t_w + t_p + t_r]$, as follows.

$$P_r[t_C > t_w + t_p + t_r]$$

$$= \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C}\int_{t_p=0}^{t_C-t_r}\int_{t_w=0}^{t_C-t_p} f_C(t_C) \times f(E)(t_r, m_r, \beta_r) \times f(E)(t_p, m_p, \beta_p) \times f(E)(t_w, m_w, \beta_w)\mathrm{d}t_w\mathrm{d}t_p\mathrm{d}t_r\mathrm{d}t_C \quad (8)$$

In equation (8), if $\beta_w \neq \beta_p \neq \beta_r$ it is assumed that $\beta_r > \beta_p > \beta_w$, we have

$$\int_{t_w=0}^{t_C-t_p} f(E)(t_w, m_w, \beta_w)\mathrm{d}t_w$$
$$= 1 - \sum_{k=0}^{m_w-1}\left[\frac{\beta_w^k(t_C-t_p)^k e^{-\beta_w(t_C-t_p)}}{k!}\right]$$
$$= 1 - \sum_{k=0}^{m_w-1}\left[\frac{\beta_w^k(t_C-t_p)^k e^{-\beta_w(t_C-t_p)}}{k!}\right]\sum_{i=0}^{k}\binom{k}{i} t_C^{k-i}(-t_p)^i$$
$$= 1 - \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i} t_C^{k-i}e^{-\beta_w t_C}(-t_p)^i e^{\beta_w t_p} \quad (9)$$

Substitute (8) and (9)

$$P_r[t_C > t_w + t_p + t_r] = \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C}\int_{t_p=0}^{t_C-t_r} f_C(t_C)f(E)(t_r, m_r, \beta_r) \times f(E)(t_p, m_p, \beta_p)\mathrm{d}t_p\mathrm{d}t_r\mathrm{d}t_C - \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C}\int_{t_p=0}^{t_C-t_r} f_C(t_C)f(E)(t_r, m_r, \beta_r) \times f(E)(t_p, m_p, \beta_p) \times \left[\sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i} t_C^{k-i}e^{-\beta_w t_C}(-t_p)^i e^{\beta_w t_p}\right. \quad (10)$$

This equation is divided into two parts to make it easier, namely, the parts M and N, so

$$P_r[t_C > t_w + t_p + t_r] = M - N$$

Equation formula M

$$M = \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C}\int_{t_p=0}^{t_C-t_r} f_C(t_C)f(E)(t_r, m_r, \beta_r) \times f(E)(t_p, m_p, \beta_p)\mathrm{d}t_p\mathrm{d}t_r\mathrm{d}t_C \quad (11)$$

From equation (11), if $\beta_p \neq \beta_w$ and it is assumed that $\beta_p > \beta_w$, then

$$\int_{t_p=0}^{t_C-t_r} f(E)(t_p, m_p, \beta_p)\mathrm{d}t_p$$
$$= 1 - \sum_{k=0}^{m_p-1}\left[\frac{\beta_p^k(t_C-t_r)^j e^{-\beta_p(t_C-t_r)}}{j!}\right]$$
$$= 1 - \sum_{k=0}^{m_p-1}\left[\frac{\beta_p^k e^{-\beta_p(t_C-t_r)}}{j!}\right]\sum_{i=0}^{k}\binom{k}{i} t_C^{k-i}(-t_p)^i$$
$$= 1 - \sum_{k=1}^{m_p-1}\left(\frac{\beta_p^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i} t_C^{k-i}e^{-\beta_p t_C}(-t_p)^i e^{\beta_p t_p} \quad (12)$$

Substitute (12) and (11)

$$M = \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C} f_C(t_C)f(E)(t_r, m_r, \beta_r)\mathrm{d}t_r\mathrm{d}t_C - \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C} f_C(t_C)f(E)(t_r, m_r, \beta_r) \times \sum_{k=0}^{m_p-1}\left(\frac{\beta_p^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i} t_C^{k-i}e^{-\beta_p t_C}(-t_p)^i e^{\beta_p t_p}\mathrm{d}t_r\mathrm{d}t_C \quad (13)$$

The equation M is divided into two parts, namely, P and Q, so that the following equation is obtained

$$M = P - Q$$

with

$$P = \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C} f_C(t_C)f(E)(t_r, m_r, \beta_r)\mathrm{d}t_r\mathrm{d}t_C \quad (14)$$

and Q is the remainder.

Look for the value of P based on equations (14) and (6)

$$P = \int_{t_C=0}^{\infty} f_C(t_C)\left[1 - \textstyle\sum_{k=0}^{m_r-1}\frac{\beta_r^k t_C^k e^{-\beta_r t_C}}{k!}\right]\mathrm{d}t_C$$
$$= 1 - \int_{t_C=0}^{\infty} f_C(t_C)\left[\textstyle\sum_{k=0}^{m_r-1}\frac{\beta_r^k t_C^k e^{-\beta_r t_C}}{k!}\right]\mathrm{d}t_C$$
$$= 1 - \sum_{k=0}^{m_r-1}\left(\frac{\beta_r^k}{k!}\right)\int_{t_C=0}^{\infty} t_C^k f_C(t_C)e^{-\beta_r t_C}\mathrm{d}t_C \quad (15)$$

Based on the Laplace transformation formula for the $f(x)$ function, the Laplace transformation $f^*(s)$ is obtained as follows

$$\int_{t=0}^{\infty} t^k f(t)e^{xt}\mathrm{d}t = (-1)^k\left[\frac{d^{(k)}t_k(x)}{d_x^k}\right] \quad (16)$$

Based on equations (15) and (16), we obtain

$$P = 1 - \sum_{k=0}^{m_r-1}\left(\frac{\beta_r^k}{k!}\right)(-1)^k\left[\frac{d^{(k)}f_C^{\alpha}(x)}{d_x^k}{}^{\alpha}\right]\Bigg|_{x=\beta_r}$$
$$= 1 - \sum_{k=0}^{m_r-1}\left[\frac{-\beta_r^k}{k!}\right]\left[\frac{d^{(k)}f_C^{\alpha}(x)}{d_x^k}{}^{\alpha}\right]\Bigg|_{x=\beta_r} \quad (17)$$

Moreover, with Q from equation 13

$$Q = \int_{tC=0}^{\infty}\int_{tr=0}^{tC=0} f(C)t_C f(E)(t_r, m_r, \beta_r) \times \left[\textstyle\sum_{k=0}^{m_p-1}\left(\frac{\beta_p^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i} t_C^{k-i}e^{-\beta_p t_C}(-t_p)^i e^{\beta_p t_p}\right] \times \mathrm{d}t_r\mathrm{d}t_C$$

$$Q = \sum_{k=0}^{m_p-1}\left(\frac{\beta_p^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}\int_{t_C=0}^{\infty} f_C(t_C) \times \int_{t_r=0}^{t_C} f(E)(t_r, m_r, \beta_r)\, t_C^{k-i}e^{-\beta_p t_C}(-t_p)^i e^{\beta_p t_p}\mathrm{d}t_r\mathrm{d}t_C$$

$$= \sum_{k=0}^{m_p-1}\left(\frac{\beta_p^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\int_{t_C=0}^{\infty} f_C(t_C)t_C^{k-i}e^{\beta_p t_C} \times \int_{t_r=0}^{t_C} f(E)(t_r, m_r, \beta_r)t_r^i e^{\beta_p t_r}\,\mathrm{d}t_r\mathrm{d}t_C \tag{18}$$

Based on (18) it is obtained

$$\begin{aligned}
&\int_{t_r=0}^{t_C} f(E)(t_r, m_r, \beta_r)t_r^i e^{\beta_p t_r}\,\mathrm{d}t_r\\
&= \int_{t_r=0}^{t_C}\left[\frac{\beta_r^{m_r}t_r^{m_r+i-1}t_r^{m_r-1}e^{-\beta_r t_r}}{(m_r-1)!}\right]t_r^i e^{\beta_p t_r}\,\mathrm{d}t_r\\
&= \int_{t_r=0}^{t_C}\left[\frac{\beta_r^{m_r}t_r^{m_r+i+1}e^{-(\beta_r-\beta_p)t_r}}{(m_r-1)!}\right]\mathrm{d}t_r\\
&= \left[\frac{\beta_r^{m_r}(m_r+i+1)!}{(\beta_r-\beta_p)^{m_r+i}(m_r-1)!}\right]\\
&\quad\times\int_{t_r=0}^{t_C}\left[\frac{(\beta_r-\beta_p)^{m_r+i}t_r^{m_r+i-1}e^{-(\beta_r\beta_p)t_r}}{(m_r+i-1)!}\right]\mathrm{d}t_r\\
&= \left[\frac{\beta_r^{m_r}(m_r+i+1)!}{(\beta_r-\beta_p)^{m_r+i}(m_r-1)!}\right]\\
&\quad\times\left[1-\sum_{l=0}^{m_r+i-1}\frac{(\beta_r-\beta_p)^l t_C^l e^{-(\beta_r\beta_p)t_C}}{l!}\right]
\end{aligned}\tag{19}$$

Substitute (19) and (18)

$$\begin{aligned}
Q &= \sum_{k=0}^{m_p-1}\left(\frac{\beta_p^k}{k!}\right)\sum_{i=0}^{k}(-1)^i\int_{t_C=0}^{\infty} f_C(t_C)t_C^{k-i}e^{\beta_p t_C}\\
&\quad\times\left[\frac{\beta_r^{m_r}(m_r+i-1)!}{(\beta_r-\beta_p)^{m_r+i}(m_r-1)!}\right]\\
&\quad\times\left[1-\sum_{l=0}^{m_r+i-1}\frac{(\beta_r-\beta_p)^l t_C^l e^{-(\beta_r-\beta_p)t_r}}{l!}\right]\mathrm{d}t_c\\
Q &= \sum_{k=0}^{m_p-1}\left(\frac{\beta_p^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\left[\frac{\beta_r^{m_r}(m_r+i-1)!}{(\beta_r-\beta_p)^{m_r+i}(m_r-1)!}\right]\\
&\quad\times(R-S)
\end{aligned}\tag{20}$$

with

$$R = \int_{t_C=0}^{\infty} f_C(t_C)t_C^{k-i}e^{-\beta_p t_C}\,\mathrm{d}t_C \tag{21}$$

and

$$\begin{aligned}
S &= \int_{t_C=0}^{\infty} f_C(t_C)t_C^{k-i}e^{-\beta_p t_C}\\
&\quad\times\sum_{l=0}^{m_r+i-1}\frac{(\beta_r\beta_p)^l t_C^l e^{-(\beta_r-\beta_p)t_C}}{l!}\mathrm{d}t_C\\
S &= \sum_{l=0}^{m_r+i-1}\int_{t_C=0}^{\infty} f_C(t_C)t_C^{k-i}e^{-\beta_p t_C}\\
&\quad\times\left[\frac{(\beta_r-\beta_p)^l t_C^l e^{-(\beta_r-\beta_p)t_C}}{l!}\right]\mathrm{d}t_C
\end{aligned}\tag{22}$$

Equation (21) in a different form

$$\begin{aligned}
R &= \int_{t_C=0}^{\infty} f_C(t_C)t_C^{k-i}e^{-\beta_p t_C}\,\mathrm{d}t_C\\
&= (-1)^{k-i}\left[\frac{d^{k-i}f_C^{\alpha(\mu)}}{d_x^{k-i}}\right]\Bigg|_{x=\beta_p}
\end{aligned}\tag{23}$$

Equation (22)

$$S = \sum_{l=0}^{m_r+i-1}\left[\frac{(\beta_r-\beta_p)^l}{l!}\right](-1)^{k+l-i}\left[\frac{(d^{k+l-i}f_C^{*(x)}}{d_x^{k+l-i}}\right]\Bigg|_{x=\beta_r} \tag{24}$$

Based on equations (20), (23) and (24),

$$\begin{aligned}
Q &= \sum_{k=0}^{m_p-1}\left[\frac{(-\beta_p)^k}{k!}\right]\sum_{i=0}^{k}\binom{k}{i}\left[\frac{\beta_r^{m_r}(m_r+i-1)!}{(\beta_r-\beta_p)^{m_r+i}(m_r-1)!}\right]\\
&\quad\times\left\{\left[\frac{d^{(k-i)}f_C^{\alpha}(x)}{d_x^{k-i}}\right]\Bigg|_{x=\beta_p}\right.\\
&\quad\left.-\sum_{l=0}^{m_r+i-1}\left[\frac{(\beta_p-\beta_r)^l)}{l!}\right]\left[\frac{(d^{k+l-i}f_C^{\alpha(x)}}{d_x^{(k+l-i)}}\right]\Bigg|_{x=\beta_r}\right.\\
&= \sum_{k=0}^{m_p-1}\sum_{i=0}^{k}\binom{m_r+i-1}{i}\left[\frac{(-\beta_p)^k\beta_r^{m_r}}{(\beta_r-\beta_p)^{m_r+i}(k-i)!}\right]\\
&\quad\times\left\{\left[\frac{d^{k-i}f_C^{\alpha}(x)}{d_x^{(k-i)}}\right]\Bigg|_{x=\beta_p}\right.\\
&\quad\left.-\sum_{l=0}^{m_r+i-1}\left[\frac{(\beta_p-\beta_r)^l)}{l!}\right]\left[\frac{(d^{k+l-i}f_C^{\alpha(x)}}{d_x^{(k+l-i)}}\right]\Bigg|_{x=\beta_r}\right.
\end{aligned}\tag{25}$$

Based on equations (13), (17) and (25), we obtain

$$\begin{aligned}
M &= P-Q\\
&= P_r[t_C > t_r + t_p]\\
&= 1-\sum_{k=0}^{m_r-1}\left[\frac{(-\beta_r)^k)}{k!}\right]\left[\frac{(d^{(k)}f_C^{\alpha(x)}}{d_x^{\alpha}}\right]\Bigg|_{x=\beta_r}\\
&\quad-\sum_{k=0}^{m_p-1}\sum_{i=0}^{k}\binom{m_r+i-1}{i}\left[\frac{(-\beta_p)^k\beta_r^{m_r}}{(\beta_r-\beta_p)^{m_r+i}(k-i)!}\right]\\
&\quad\times\left\{\left[\frac{d^{(k-i)}f_C^{\alpha}(x)}{d_x^{k-i}}\right]\Bigg|_{x=\beta_p}\right.\\
&\quad\left.-\sum_{l=0}^{m_r+i-1}\left[\frac{(\beta_p-\beta_r)^l)}{l!}\right]\left[\frac{(d^{k+l-i}f_C^{\alpha(x)}}{d_x^{(k+l-i)}}\right]\Bigg|_{x=\beta_r}\right.
\end{aligned}\tag{26}$$

From equation (10), we obtain

$$\begin{aligned}
&P_r[t_C > t_w + t_p + t_r]\\
&\quad= M-N\\
N &= \int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C}\int_{t_p=0}^{t_C-t_r} f_C(t_C)f(E)(t_r, m_r, \beta_r)\\
&\quad\times f(E)(t_p, m_p, \beta_p)\\
&\quad\times\left[\sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}t_C^{k-i}e^{-\beta_w t_C}(-t_p)^i e^{\beta_w t_p}\right.\\
&\quad\mathrm{d}t_p\mathrm{d}t_r\mathrm{d}t_C\\
&= \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}\int_{t_C=0}^{\infty} f_C(t_C)\int_{t_r=0}^{t_C}\\
&\quad\times f(E)(t_r, m_r, \beta_r)\\
&\quad\times\int_{t_p=0}^{t_C-t_r} f(E)(t_p, m_p, \beta_p)t_C^{k-i}e^{-\beta_w t_C}(-t_p)^i e^{\beta_w t_p}
\end{aligned}$$

$$\begin{aligned}
&\mathrm{d}t_p\mathrm{d}t_r\mathrm{d}t_C\\
&= \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C} f_C(t_C)\\
&\quad\times f(E)(t_r,m_r,\beta_r)\\
&\quad\times\int_{t_p=0}^{t_C-t_r} f(E)(t_p,m_p,\beta_p)t_p^i e^{\beta_w t_p} t_C^{k-i}e^{-\beta_w t_C}\\
&\quad\times \mathrm{d}t_p\mathrm{d}t_r\mathrm{d}t_C\\
N &= \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C} f_C(t_C)\\
&\quad\times f(E)(t_r,m_r,\beta_r)t_C^{k-i}e^{-\beta_w t_C}\\
&\quad\times\int_{t_p=0}^{t_C-t_r} f(E)(t_p,m_p,\beta_p)t_p^i e^{\beta_w t_p}\mathrm{d}t_p\mathrm{d}t_r\mathrm{d}t_C
\end{aligned}\tag{27}$$

$$\begin{aligned}
&\int_{t_p=0}^{t_C-t_r} f(E)(t_p,m_p,\beta_p)t_p^i e^{\beta_w t_p}\mathrm{d}t_p\\
&= \int_{t_p=0}^{t_C-t_r}\left[\frac{\beta_p^{m_p}t_p^{m_p-1}e^{-\beta_p t_p}}{(m_p-1)!}\right]t_p^i e^{\beta_w t_p}\mathrm{d}t_p\\
&= \int_{t_p=0}^{t_C-t_r}\left[\frac{\beta_p^{m_p}t_p^{m_p+i-1}e^{-(\beta_p-\beta_w)t_p}}{(m_p-1)!}\right]\mathrm{d}t_p\\
&= \left[\frac{\beta_p^{m_p}(m_p+i-1)!}{(\beta_p-\beta_w)^{m_p+i}(m_p-i)!}\right]\\
&\quad\times\int_{t_p=0}^{t_C-t_r}\left[\frac{(\beta_p-\beta_w)^{m_p+i}t_p^{m_p+i-1}e^{-(\beta_p-\beta_w)t_p}}{(m_p+i-1)!}\right]\mathrm{d}t_p
\end{aligned}$$

for example

$$\begin{aligned}
g = (m_p+i-1) &= \left[\frac{\beta_p^{m_p}(m_p+i-1)!}{(\beta_p-\beta_w)^{m_p+i}(m_p-i)!}\right]\\
&\quad\times\left[1-\sum_{g=0}^{m_p+i-1}\frac{(\beta_p-\beta_w)^g(t_C-t_r)^g e^{-(\beta_p-\beta_w)(t_C-t_r)}}{g!}\right]
\end{aligned}$$

combined with (27)

$$\begin{aligned}
N &= \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C} f_C(t_C)\\
&\quad\times f(E)(t_r,m_r,\beta_r)t_C^{k-i}e^{-\beta_w t_C}\\
&\quad\times\left[\frac{\beta_p^{m_p}(m_p+i-1)!}{(\beta_p-\beta_w)^{m_p+i}(m_p-i)!}\right]\\
&\quad\times\left[1-\sum_{g=0}^{m_p+i-1}\frac{(\beta_p-\beta_w)^g(t_C-t_r)^g e^{-(\beta_p-\beta_w)(t_C-t_r)}}{g!}\right]\\
&\quad\times\mathrm{d}t_r\mathrm{d}t_C\\
N &= \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\int_{t_C=0}^{\infty}\int_{t_r=0}^{t_C}\\
&\quad\times f_C(t_C)t_C^{k-i}e^{-\beta_w t_C}\\
&\quad\times\int_{t_r=0}^{t_C} f(E)(t_r,m_r,\beta_r)\times\left[\frac{\beta_p^{m_p}(m_p+i-1)!}{(\beta_p-\beta_w)^{m_p+i}(m_p-i)!}\right]\\
&\quad\times\left[1-\sum_{g=0}^{m_p+i-1}\frac{(\beta_p-\beta_w)^g(t_C-t_r)^g e^{-(\beta_p-\beta_w)(t_C-t_r)}}{g!}\right]\\
&\quad\times\mathrm{d}t_r\mathrm{d}t_C\\
&= \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\times T\times O
\end{aligned}$$

$$\begin{aligned}
&\quad\times\left[\frac{\beta_p^{m_p}(m_p+i-1)!}{(\beta_p-\beta_w)^{m_p+i}(m_p-i)!}\right]\\
&\quad\times\left[1-\sum_{g=0}^{m_p+i-1}\frac{(\beta_p-\beta_w)^g(t_C-t_r)^g e^{-(\beta_p-\beta_w)(t_C-t_r)}}{g!}\right]\\
&\quad\times\mathrm{d}t_r\mathrm{d}t_C\\
T &= \int_{t_C}^{\infty} f_C(t_C)t_C^{(k-i)}e^{-\beta_w t_C}\mathrm{d}t_C\\
O &= \int_{t_o}^{t_C} f(E)(t_r,m_r.\beta_r)\mathrm{d}t_r
\end{aligned}$$

The equation T can be written as follows

$$T = (-1)^{k-i}\left[\frac{\mathrm{d}^{(k-i)}f_C^{\alpha}(x)}{\mathrm{d}x^k}\right]\Bigg|_{x=\beta_r}\tag{28}$$

Moreover, the O equation can be written as follows (the same as equation (19))

$$\begin{aligned}
O &= \int_{t_r=0}^{t_C} f(E)(t_r,m_r,\beta_r)\mathrm{d}t_r\\
&= \left[\frac{\beta_r^{m_r}(m_r+i-1)!}{(\beta_r-\beta_p)^{m_r+i}(m_r-1)!}\right]\\
&\quad\times\left[1-\sum_{l=0}^{m_r+i-1}\frac{(\beta_r-\beta_p)^l t_C^l e^{-(\beta_r\beta_p)t_C}}{l!}\right]
\end{aligned}\tag{29}$$

Obtaining equation N

$$\begin{aligned}
N &= \sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{i=0}^{k}\binom{k}{i}(-1)^i\\
&\quad\times\left[(-1)^{k-i}\left[\frac{\mathrm{d}^{(k-i)}f_C^{\alpha}(x)}{\mathrm{d}x^k}\right]\Bigg|_{x=\beta_r}\right]\\
&\quad\times\left[\left[\frac{\beta_r^{m_r}(m_r+i-1)!}{(\beta_r-\beta_p)^{m_r+i}(m_r-1)!}\right]\right.\\
&\quad\times\left.\left[1-\sum_{l=0}^{m_r+i-1}\frac{(\beta_r-\beta_p)^l t_C^l e^{-(\beta_r-\beta_p)t_C}}{l!}\right]\right]\\
&\quad\times\left[\left[\frac{\beta_p^{m_p}(m_p+i-1)!}{(\beta_p-\beta_w)^{m_p+i}(m_p-1)!}\right]\right.\\
&\quad\times\left.\left[1-\sum_{g=0}^{m_p+i-1}\frac{(\beta_p-\beta_w)^l(t_C-t_r)^g e^{-(\beta_p-\beta_w)(t_C-t_r)}}{g!}\right]\right]
\end{aligned}\tag{30}$$

If combined with equations (10), (26) and (30), we obtain

$$\begin{aligned}
&Pr[t_C > t_w+t_p+t_r] = M-N\\
&= \left[1-\sum_{k=0}^{m_r-1}\left[\frac{(-\beta_r)^k}{k!}\right]\left[\frac{\mathrm{d}^{(k-i)}f_C^{\alpha}(x)}{\mathrm{d}x^k}\right]\Bigg|_{x=\beta_r}\right.\\
&\quad-\left[1-\sum_{k=0}^{m_r-1}1-\sum_{i=0}^{k}\binom{m_r+i-1}{i}\left[\frac{(-\beta_p)^k\beta_r^{m_r}}{(\beta_r-\beta_p)^{m_r+i}(k-i)}\right]\right.\\
&\quad\times\left\{\left[\frac{\mathrm{d}^{(k-i)}f_C^{\alpha}(x)}{\mathrm{d}x^{k-i}}\right]\Bigg|_{x=\beta_r}\right.\\
&\quad-\left.\left.\sum_{l=0}^{m_r+i-1}\left[\frac{(\beta_p-\beta_r)^l}{l!}\right]\mathrm{I}\left[\frac{\mathrm{d}^{(k+l-i)}f_C^{\alpha}(x)}{\mathrm{d}x^{k+l-i}}\right]\Bigg|_{x=\beta_r}\right]\right]\\
&\quad-\left[\sum_{k=0}^{m_w-1}\left(\frac{\beta_w^k}{k!}\right)\sum_{k!}^{k}\sum_{i=0}^{k}\binom{k}{i}(-1)^i\right.
\end{aligned}$$

$$\times \left[(-1)^{k-i}\left[\frac{\mathrm{d}^{(k-i)} f_C^{\alpha}(x)}{\mathrm{d}x^k}\right]\Bigg|_{x=\beta_r}\right]$$
$$\times \left[\left[\frac{\beta_r^{m_r}(m_r+i-1)!}{\beta_r-\beta_p)^{m_r+i}(m_r-1)}\right]\Bigg|_{x=\beta_r}\right.$$
$$\times \left[1-\sum_{l=0}^{m_r+i-1}\frac{(\beta_r-\beta_p)^l t_C e^{-(\beta_r-\beta_p)t_C}}{l!}\right]\Bigg]$$
$$\times \left[\left[\frac{\beta_p^{m_p}(m_p+i-1)!}{(\beta_p-\beta_w)^{m_p+i}(m_p-i)!}\right.\right.$$
$$\times \left[1-\sum_{g=0}^{m_p+1-i}(\beta_p-\beta_w)^g(t_C-t_r)^g\right.$$
$$\times e^{-(\beta_p-\beta_w)(t_C-t_r)}\Bigg]\Bigg]\Bigg] \tag{31}$$

APPENDIX B

For example, $t_i = t_w + t_p + t_r$. In equation (8), when $\beta_w = \beta_p = \beta_r = \beta$, with the Erlang convolution distribution approach, then t_i also has an Erlang distribution with a density function as $f_E(t_i, m_w + m_p + m_r, \beta)$ so that $P_r[\tau_{t,4} > \tau_{t,3}] = P_r[t_C > t_w + t_p + t_r] = P_r[t_C > t_i]$, where

$$P_r[t_C > t_i] = \int_{t_C=0}^{\infty}\int_{t_i=0}^{t_C} f_C(t_C) f_E(t_i, m_w + m_p + m_r, \beta)\mathrm{d}t_i\mathrm{d}t_C$$

By looking at the equations (14) and (17), the following formula can be derived.

$$P_r[t_C > t_i] = 1 - \sum_{k=0}^{m_w+m_p+m_r-1}\left[\frac{(-\beta)^k}{k!}\right]\left[\frac{(d^{(k)} f_C^*(x)}{dx^k}\right]\Bigg|_{x=\beta} \tag{32}$$

Since $f_C(t_C)$ is a function approximated by the Gamma distribution with the shape parameter α and the scale parameter μ, the results of the Laplace transform are shown as follows.

$$f_C^*(x) = \int_{x=0}^{\infty} f_C(t_C) = e^{-xt_C}\mathrm{d}t_C = \frac{\mu^\alpha}{(x+\mu)^\alpha}$$

Therefore, the equation (32) can be rewritten as follows

$$P_r[t_C > t_i] = 1 - \sum_{k=0}^{m_w+m_p+m_r-1}\binom{\alpha+k-1}{k}\left[\frac{\beta^k\mu^\alpha}{(\beta+\mu)^{\alpha+k}}\right] \tag{33}$$

With the same pattern, we can look for opportunities to convey a message to the user, $P_r[\tau_{t,4} > \tau_{t,u}] = P_r[t_C > t_w + t_p + t_u] = P_r[t_C > t_j]$ as follows

$$P_r[t_C > t_j] = 1 - \sum_{k=0}^{m_w+m_p+m_u-1}\binom{\alpha+k-1}{k}\left[\frac{\beta^k\mu^\alpha}{(\beta+\mu)^{\alpha+k}}\right] \tag{34}$$

ACKNOWLEDGMENT

The authors would like to thank colleagues who have supported this research, namely, Dr. Khoirul Anwar at the AICOMS Laboratory, Dr. Deni Saepudin, Jonri, Indwiarti, Aniq Antiki, Dr. Edy Wibowo, Dr. Mamat Rohmat, Dr. Andri Alamsyah, Endriawan Tria Sukmara, Akbar Taufiqurrahman, Abdullah Hadi, Pertiwang Sismananda, Dede Suryadi, and Maya Ameliasari. Hopefully, this research will have a good impact on the advancement of science.

REFERENCES

[1] M. Abdurohman, A. G. Putrada, S. Prabowo, C. Wijiutomo, and A. Elmangoush, "M2M device connectivity framework," *Int. J. Electr. Eng. Informat.*, vol. 9, no. 3, pp. 441–454, Sep. 2017.
[2] M. Abdurohman, A. Sasongko, and A. Herutomo, "M2M middleware based on OpenMTC platform for enabling smart cities solution," in *Smart City 360°* (Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering), vol. 166, A. Leon-Garcia, Ed. Berlin, Germany: Springer, 2016.
[3] J.-H. Chen, W.-T. Sung, and G.-Y. Lin, "Automated monitoring system for the fish farm aquaculture environment," in *Proc. IEEE Int. Conf. Syst., Man, Cybern.*, Oct. 2015, pp. 1161–1166.
[4] S.-P. Tseng, Y.-R. Li, and M.-C. Wang, "An application of Internet of Things on sustainable aquaculture system," in *Proc. Int. Conf. Orange Technol. (ICOT)*, Dec. 2016, pp. 17–19.
[5] K. R. S. R. Raju and G. H. K. Varma, "Knowledge based real time monitoring system for aquaculture using IoT," in *Proc. IEEE 7th Int. Adv. Comput. Conf.*, Jan. 2017, pp. 318–321.
[6] Y.-B. Lin and H.-C. Tseng, "FishTalk: An IoT-based mini aquarium system," *IEEE Access*, vol. 7, pp. 35457–35469, 2019.
[7] Y. Liu, K. A. Hassan, M. Karlsson, O. Weister, and S. Gong, "Active plant wall for green indoor climate based on cloud and Internet of Things," *IEEE Access*, vol. 6, pp. 33631–33644, 2018.
[8] L.-J. Chen, Y.-H. Ho, H.-C. Lee, H.-C. Wu, H.-M. Liu, H.-H. Hsieh, Y.-T. Huang, and S.-C. C. Lung, "An open framework for participatory $PM_{2.5}$ monitoring in smart cities," *IEEE Access*, vol. 5, pp. 14441–14454, 2017.
[9] S. Kubler, J. Robert, A. Hefnawy, K. Främling, C. Cherifi, and A. Bouras, "Open IoT ecosystem for sporting event management," *IEEE Access*, vol. 5, pp. 7064–7079, 2017.
[10] Y.-X. Zhao, Y.-S. Su, and Y.-C. Chang, "A real-time bicycle record system of ground conditions based on Internet of Things," *IEEE Access*, vol. 5, pp. 17525–17533, 2017.
[11] T. N. Pham, M.-F. Tsai, D. B. Nguyen, C.-R. Dow, and D.-J. Deng, "A cloud-based smart-parking system based on Internet-of-Things technologies," *IEEE Access*, vol. 3, pp. 1581–1591, 2015.
[12] M. Shahroz, M. F. Mushtaq, M. Ahmad, S. Ullah, A. Mehmood, and G. S. Choi, "IoT-based smart shopping cart using radio frequency identification," *IEEE Access*, vol. 8, pp. 68426–68438, 2020.
[13] M. Khalaf, H. Alaskar, A. J. Hussain, T. Baker, Z. Maamar, R. Buyya, P. Liatsis, W. Khan, H. Tawfik, and D. Al-Jumeily, "IoT-enabled flood severity prediction via ensemble machine learning models," *IEEE Access*, vol. 8, pp. 70375–70386, 2020.
[14] F. Liu, C. Liang, and Q. He, "Remote malfunctional smart meter detection in edge computing environment," *IEEE Access*, vol. 8, pp. 67436–67443, 2020.
[15] A. Angani, J. C. Lee, and K. J. Shin, "Vertical recycling aquatic system for Internet-of-Things-based smart fish farm," *Sensors Mater.*, vol. 31, no. 12, pp. 3987–3998, 2019.
[16] L. Yu, Z. Wu, R. Zhang, and X. Yang, "Partial least regression approach to forecast the east Asian winter monsoon using Eurasian snow cover and sea surface temperature," *Climate Dyn.*, vol. 51, nos. 11–12, pp. 4573–4584, Dec. 2018.
[17] C. Voyant, F. Motte, G. Notton, A. Fouilloy, M.-L. Nivet, and J.-L. Duchaud, "Prediction intervals for global solar irradiation forecasting using regression trees methods," *Renew. Energy*, vol. 126, pp. 332–340, Oct. 2018.
[18] N. Khan, S. Shahid, L. Juneng, K. Ahmed, T. Ismail, and N. Nawaz, "Prediction of heat waves in Pakistan using quantile regression forests," *Atmos. Res.*, vol. 221, pp. 1–11, Jun. 2019.

[19] M. Zahid, F. Ahmed, N. Javaid, R. Abbasi, H. Z. Kazmi, A. Javaid, M. Bilal, M. Akbar, and M. Ilahi, ''Electricity price and load forecasting using enhanced convolutional neural network and enhanced support vector regression in smart grids,'' *Electronics*, vol. 8, no. 2, p. 122, Jan. 2019.

[20] C. Johansson, M. Bergkvist, D. Geysen, O. De Somer, N. Lavesson, and D. Vanhoudt, ''Operational demand forecasting in district heating systems using ensembles of online machine learning algorithms,'' in *Proc. 15th Int. Symp. District Heating Cooling (DHC)*. Seoul, South Korea: Elsevier, 2017, pp. 208–216.

[21] C. Liu, Z. Hu, Y. Li, and S. Liu, ''Forecasting copper prices by decision tree learning,'' *Resour. Policy*, vol. 52, no. 1, pp. 427–434, Jun. 2017.

[22] A. O. J. Hambali, M. Akinyemi, and N. JYusuf, ''Electric power load forecast using decision tree algorithms,'' *Comput. Inf. Syst. Device Inf. Allied Res. J.*, vol. 7, no. 4, pp. 29–42, 2016.

[23] H. Yuan, M. Tan, and Y. Chen, ''A model for Fishery forecast based on cluster analysis and nonlinear regression,'' in *Proc. Decis. Making Soft Comput.* Singapore: World Scientific, Sep. 2014, pp. 174–181.

[24] S. Liu, L. Xu, and D. Li, ''Multi-scale prediction of water temperature using empirical mode decomposition with back-propagation neural networks,'' *Comput. Electr. Eng.*, vol. 49, pp. 1–8, Jan. 2016.

[25] S. Stajkowski, D. Kumar, P. Samui, H. Bonakdari, and B. Gharabaghi, ''Genetic-algorithm-optimized sequential model for water temperature prediction,'' *Sustainability*, vol. 12, no. 13, p. 5374, Jul. 2020.

[26] S. Samadianfard, S. Jarhan, E. Salwana, A. Mosavi, S. Shamshirband, and S. Akib, ''Support vector regression integrated with fruit fly optimization algorithm for river flow forecasting in lake Urmia basin,'' *Water*, vol. 11, no. 9, p. 1934, Sep. 2019.

[27] A. K. Erlang, ''The theory of probabilities and telephone conversations,'' *Nyt. Tidsskr. Mater. Ser. B*, vol. 20, pp. 131–137, 1909. [Online]. Available: https://www.jstor.org/stable/i24528012

[28] M. Abdurohman, A. G. Putrada, S. Prabowo, C. W. Wijiutomo, and A. Elmangoush, ''Integrated lighting enabler system using M2M platforms for enhancing energy efficiency,'' *J. Inf. Process. Syst.*, vol. 14, no. 4, pp. 1033–1048, 2018.

[29] F. A. Rachman, A. G. Putrada, and M. Abdurohman, ''Distributed campus bike sharing system based-on Internet of Things (IoT),'' in *Proc. 6th Int. Conf. Inf. Commun. Technol. (ICoICT)*, May 2018, pp. 333–336.

[30] I. Hussain, T. Shah, M. A. Gondal, and H. Mahmood, ''Analysis of S-box in image encryption using root mean square error method,'' *Zeitschrift Naturforschung A*, vol. 67, nos. 6–7, pp. 327–332, Jul. 2012.

[31] U. Khair, H. Fahmi, S. A. Hakim, and R. Rahim, ''Forecasting error calculation with mean absolute deviation and mean absolute percentage error,'' *J. Phys., Conf. Ser.*, vol. 930, Dec. 2017, Art. no. 012002.

[32] K. H. Zou, K. Tuncali, and S. G. Silverman, ''Correlation and simple linear regression,'' *Radiology*, vol. 227, no. 3, pp. 617–628, 2003.

MAMAN ABDUROHMAN received the master's and Ph.D. degrees from ITB, in 2004 and 2010, respectively. Since 2000, he has been working as a full-time Lecturer and a Researcher with Telkom University (formerly STT Telkom). He is currently a Professor in information technology with Telkom University. Previously, he served as the Director of Bandung Techno Park (BTP)—Telkom University, which focuses on research commercialization and business incubation. He has authored over 60 papers published in international and national journals and conferences, such as *International Journal of Electrical Engineering and Informatics*, *International Journal of Informational and Education Technology*, International Conference of ICT, and International Conference on Computer Engineering and Technology. He has received institutional or national research grants from Telkom University and the Ministry of Higher Education Board in the last three years. His current research interests include the IoT, notably smart card technology, and smart lighting.

AJI GAUTAMA PUTRADA (Member, IEEE) was born in Jakarta, in November 1985. He received the bachelor's degree in electrical engineering and the master's degree in microelectronics from ITB, in 2003 and 2009, respectively. Currently, he is pursuing the Ph.D. degree in computer science doctoral program with Telkom University. His dissertation topic concerns user comfort in smart lighting with machine learning. He was the Research and Development Director of PT. Cetta Nusantara Technology, where he developed low level broadband wireless access (BWA) solutions, such as WiMAX and LTE. He became a Lecturer at Telkom University, Bandung, where he is currently an Assistant Professor. He taught various courses, such as computer architecture, microcontroller systems, and operating systems. Since 2015, he has been involved in various research grants from the Government about smart lighting. From 2020 to 2022, he was entrusted to become as the Vice Director of the Advanced and Creative Networks Research Center (AdCNet RC)—Telkom University.

MUSTAFA MAT DERIS received the Ph.D. degree from University Putra Malaysia, in 2002. He is currently a Professor of computer science with the Faculty of Computer Science and Information Technology, UTHM, Malaysia. He published more than 220 papers in journals and conference proceedings with Scopus H-index of 19. He was appointed as a keynote speaker for several conferences and served as a program committee member and a co-organizer for numerous international conferences/workshops. His research interests include distributed databases, rough set theory, soft set theory, and data mining. He was a recipient of ICT Excellent Teacher Award from Malaysian National Computing Confederation (MNCC), in 2006. He has appointed as an Editorial Board Member of *Journal of Next Generation Information Technology* (JNIT, South Korea), *International Journal of Rough Sets and Data Analysis* (IGI-Global, USA), and *Encyclopedia on Mobile Computing and Commerce* (Idea Group, USA); a Guest Editor of *International Journal of Biomedical Soft Computing and Human Science* for a Special Issue on ''Soft Computing Methodologies and Its Applications;'' and a Reviewer of several outstanding international journals, such as IEEE TRANSACTION ON PARALLEL AND DISTRIBUTED COMPUTING, *Journal of Parallel and Distributed Databases*, *Journal of Future Generation on Computer Systems*, *Journal of Information Sciences* (Elsevier), and *Knowledge-Based Systems* (Elsevier).