International Journal of Electrical and Computer Engineering (IJECE)
Vol. 14, No. 2, April 2024, pp. 2293~2312
ISSN: 2088-8708, DOI: 10.11591/ijece.v14i2.pp2293-2312

Developing a smart system for infant incubators using the internet of things and artificial intelligence

I Komang Agus Ady Aryanto[1], Dechrit Maneetham[1], Evi Triandini[2]
[1]Department of Mechatronics Engineering, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi, Pathum Thani, Thailand
[2]Department of Information Systems, Faculty of Informatics and Computers, Institute of Technology and Business STIKOM Bali, Bali, Indonesia

Article Info

Article history:

Received Sep 20, 2023
Revised Oct 11, 2023
Accepted Dec 5, 2023

Keywords:

Artificial intelligence
Infant incubator
Internet of things
Long short-term memory network
Message queue telemetry transport

ABSTRACT

This research is developing an incubator system that integrates the internet of things and artificial intelligence to improve care for premature babies. The system workflow starts with sensors that collect data from the incubator. Then, the data is sent in real-time to the internet of things (IoT) broker eclipse mosquito using the message queue telemetry transport (MQTT) protocol version 5.0. After that, the data is stored in a database for analysis using the long short-term memory network (LSTM) method and displayed in a web application using an application programming interface (API) service. Furthermore, the experimental results produce as many as 2,880 rows of data stored in the database. The correlation coefficient between the target attribute and other attributes ranges from 0.23 to 0.48. Next, several experiments were conducted to evaluate the model-predicted value on the test data. The best results are obtained using a two-layer LSTM configuration model, each with 60 neurons and a lookback setting 6. This model produces an R^2 value of 0.934, with a root mean square error (RMSE) value of 0.015 and a mean absolute error (MAE) of 0.008. In addition, the R^2 value was also evaluated for each attribute used as input, with a result of values between 0.590 and 0.845.

Corresponding Author:

Dechrit Maneetham
Department of Mechatronics Engineering, Faculty of Technical Education, Rajamangala University of Technology Thanyaburi
39 Village No. 1 Rangsit - Nakhon Nayok Road, Tambon Khlong Hok, Amphoe Thanyaburi Pathum Thani 12110, Thailand
Email: dechrit_m@rmutt.ac.th

1. INTRODUCTION

Infant incubators are important devices used to provide a controlled and stable environment for premature babies or babies with certain health conditions [1]. The incubator can maintain environmental conditions from outside conditions so that the baby becomes safe in the incubator [2], [3]. In the incubator, several important parameters must be closely monitored so that it requires more attention to care for babies in the incubator. These parameters include the baby's body temperature, incubator temperature, oxygen level, and heart rate [4]. All of these parameters are monitored to ensure that the baby receives appropriate care and that the environmental conditions in the incubator remain optimal.

However, the problem is monitoring the infant's condition and manually adjusting the incubator's environmental parameters, which can be a challenge. This requires medical personnel to constantly monitor and adapt to environmental conditions for each baby, which can be very tiring and increases the risk of human error.

Another problem often encountered is temperature instability inside the incubator because the incubator is often opened and closed for baby care and examination. As a result, the incubator's temperature conditions may differ now and, in the future, placing the baby at risk of hypothermia or hyperthermia. This incubator temperature condition that is not optimal can have a negative impact on health and affect the baby's healing process.

Based on these problems, we offer updates in conducting novel research on using automation technology in infant incubators enhanced by integrating internet of things (IoT) and artificial intelligence (AI) technologies. In particular, artificial intelligence technology uses the long short-term memory (LSTM) method to predict future incubator temperature conditions. This research is the first step for us in developing an incubator equipped with a combination of the internet of things and artificial intelligence models. The AI model is created using sensor data obtained at the incubator with the IoT concept. Thus, it is hoped that this research will improve the quality of care, especially for babies, and provide new insights regarding health technology development using IoT and AI artificial intelligence.

The internet of things is a concept that describes a physical device connected to the internet that can communicate with each other to collect and exchange data without involving human interaction [5]. So, in this study, internet of things technology is used to collect data in real-time from various sensors installed in the incubator. The system uses several types of sensors, including body temperature, incubator temperature, heart rate, and oxygen sensors that collect real-time data about the infant's physical condition and environment inside the incubator.

Then, for network communication devices, use the Wi-Fi module, which functions as a liaison between the incubator and the IoT broker so that sensor data measured in the incubator can be transmitted to the IoT broker with the message queue telemetry transport (MQTT) protocol. MQTT is a communication protocol used on the internet of things to send messages between devices with a publish (sender) and subscribe (recipient) mechanism. In addition, MQTT also supports quality of service (QoS) settings that allow users to determine the reliability level of message delivery [6].

Meanwhile, the microcontroller chip controls all sensor measurement processes and data transmission, which is the core brain of all hardware. The microcontroller used ATMega256 on the Arduino Mega board, with a large number of input/output pins and a large memory capacity compared to other Arduino-type boards, which allows reading data from various sensors and controlling other devices, such as relay actuators and communication modules. The Arduino Mega operates on 5 V, making it suitable for multiple sensor modules. The programming language used is wiring, which makes it easy to write program code for the various functions needed [7].

Furthermore, the data transmitted to the IoT broker is collected and processed by the server service to be stored in the database. The process of storing data received from the incubator in the database is adjusted to the database structure. The design of this database is based on data patterns obtained from sensor data in the incubator so that the stored data becomes structured and efficient to access and analyze. After the data is stored in the database, through the web application, the data is accessed and displayed on web pages in real-time so that medical staff can monitor the infant's condition directly if special attention is needed. The standard temperature setting in the incubator is 32 ˚C to 35 ˚C, depending on the baby's age [8].

Meanwhile, the LSTM-based deep learning model is used for the data analysis. LSTM are a particular type of recurrent neural networks (RNN) designed to learn and understand long-term dependencies in data. Through the analysis performed by the LSTM model, certain patterns in the data can be identified and predicted [9], [10]. Therefore, this study uses LSTM to analyze data from various sensors in the incubator collected in a database.

2. RELATED WORKS

In this section, we provide an overview of necessary research conducted in this field. As a basic framework, these studies have provided essential and in-depth insights into relevant methods, techniques, and approaches, all of which have become references in the planning and implementation of research conducted. The following are some of the studies that formed the basis for this research.

The first study we discussed concerned a real-time wireless temperature measurement system for infant incubators. This research in 2023 explains the implementation of the MQTT protocol in transmitting temperature data from the incubator to the server. In addition, perform an analysis of the QoS provided by MQTT and evaluate the overall performance of this system Sukma *et al.* [11]. Subsequent research focuses on developing incubators designed to read fingerprints for infant identification. In addition, this incubator is equipped with a monitoring system to monitor the infant's temperature and heart rate via the global system for mobile (GSM) communications network, which is integrated with IoT based applications Kapen *et al.* [12]. Subsequent research discusses the development of an incubator that can monitor the baby's temperature and heart rate on the liquid crystal display (LCD) module and web applications via the hypertext transfer protocol (HTTP) protocol with an ethernet shield device Irmansyah *et al.* [13]. Next is research that discusses the control system in the incubator by combining fuzzy-proportional integral derivative (fuzzy-PID) to regulate temperature and humidity in the incubator Alimuddin *et al.* [14]. Subsequent research regarding developing an incubator system to monitor temperature and humidity through a web application. The system uses a Wi-Fi network and MQTT protocol to send data to the broker service at node-RED Parra *et al*. [15]. Then, research on developing a system to monitor the respiratory rate and detect the incidence of apnea in premature babies with the internet of things concept. The system developed uses the ESP32 microcontroller, sensors, edge computing device (ECD) devices, and the MQTT protocol. It can be described in general that this system uses the internet of things architecture with the MQTT protocol to connect a wireless embedded system (WES) system or sensor with an ECD Cay *et al*. [16]. Subsequent research discusses the development of a model to detect system errors in reading sensor values in incubators. The model developed uses a classification method, namely support vector machine (SVM), decision tree (DT), and artificial neural network (ANN). The results of these three methods are compared to get the model with the best method. The dataset has four features: temperature, humidity, fan electric current, and heating electric current. The value of this dataset feature is obtained from the sensors attached to the incubator. In addition, the system also uses services from the Blynk application Mahdi *et al.* [17]. In Table 1, a summary of the state-of-the-art references related to the latest research conducted is presented for comparison.

Table 1. State-of-the-art of the existing works

Reference	Microcontroller	Microcomputer	Sensor	Communication protocol	Network	Broker IoT	Data process	Application interface	Programming	Data analytics
[11]	ESP32	Raspberry Pi	Temperature	MQTT	Wi-Fi	Node-Red	Node-Red	Web	Node-Red	-
[12]	ATMega 328	-	Phototherapy, Temperature, Humidity, Fingerprint, Heart Rate, Camera	HTTP	GSM	-	MySQL	Mobile Android	C++, PHP, Java	-
[13]	ATMega 2560	-	Heart rate, Weight	HTTP	Ethernet	-	-	Web	C++	-
[14]	ATMega 16	-	Temperature, Humidity	-	-	-	-	LCD, Desktop	C++, VB.Net	Fuzzy-PID
[15]	ESP32	-	Temperature, Humidity, Sound	MQTT	Wi-Fi	Node-Red	MySQL	Web	Node.js, PHP, JavaScript, C++	-
[16]	ESP32	Raspberry Pi	Respiration	MQTT	Wi-Fi	Mosquitto	Data filter, Peak Detection, Feature extraction	LCD, Desktop	Python	-
[17]	ESP8266	Raspberry Pi	Temperature, Humidity, Fan Current, Heater Current	-	Wi-Fi	-	-	Blynk Platform	C++	Decision Tree, SVM, Neural Network
Our work	ATMega 2560	Raspberry Pi	Temperature Incubator, Humidity Incubator, Body Temperature, Heart Rate, Saturation	MQTT and HTTP	Wi-Fi	Mosquitto	Data aggregator, Data filter, MySQL	Web, Mobile, Notification, LCD	C++, Python, PHP, JavaScript, CSS	LSTM, RMSE, MAE, MAPE, R^2

Based on previous research, we developed an incubator system with the internet of things concept to combine artificial intelligence, hardware, network, database management, and software. In terms of hardware, the system is designed using various sensors and actuators and is supported by a microcontroller and microcomputer. Furthermore, the communication network developed in this system uses the MQTT and HTTP protocols, which enable an efficient data transmission process according to its QoS. In addition, data is directly stored in the database according to the schema to facilitate analysis. The analysis process in this study utilizes the LSTM method to create a learning model. This study also developed a web application for monitoring, controlling, and providing real-time notifications regarding the condition of the incubator.

3. METHOD

3.1. System overview

Figure 1 shows an overview of the system design in this study. This innovation applies the latest technology to improve baby care, especially for premature babies and babies with certain health conditions requiring intensive monitoring. This system is centered on a baby incubator with various sensors, including sensors for the infant's body temperature, temperature inside the incubator, heart rate, and oxygen level. These sensors continuously collect data in real-time, providing a real-time picture of the infant's condition and the environment inside the incubator.

The microcontroller then handles the data measured by these sensors [18]. The microcontroller functions as a data collector from various sensors and as an actuator controller whose job is to regulate the temperature in the incubator. These data and control instructions are sent to the IoT broker via the network, making this system part of the internet of things scope. Sending data to the IoT broker uses the MQTT protocol and Wi-Fi network [19].

The data collected in the IoT broker is then processed and stored in the database through data processing stages such as screening and data filtering. The data storage process is adjusted to the database schema, such as attributes and data types. This approach also enables large-scale and systematic collection of medical data. These data are invaluable for further research and development in neonatal care.

After that, the data is analyzed using a learning method based on deep learning LSTM to produce a model that can predict future incubator temperature values. LSTM is an artificial neural network method for pattern processing in sequential datasets [20]. This model is designed to learn from the incubator dataset collected in the database, understand the patterns that emerge from the data, and then make accurate predictions about future incubator temperature conditions.

Furthermore, in the data output process, a web based IoT application plays an essential role in making it easier for users to monitor and control the condition of the incubator. The web application is developed using modern web technology with a responsive design and is equipped with various features, making it easier for users to understand and operate this system. In addition, the web application also has a notification feature that provides a warning if an event occurs.

Then, the data displayed on the web application is obtained in real-time from the database through the application programming interface (API) and the POST method. API allows programmatic access to data stored in the database. In addition, this application also uses asynchronous JavaScript and XML (AJAX) and JavaScript object notation (JSON) technology, and AJAX is used to send asynchronous requests to the server to retrieve data from the database [21]. Meanwhile, JSON is used as a data format that is lightweight and easy for software to understand, allowing data to be transmitted quickly and efficiently between servers and applications [22]. After the data is received in JSON format, the application can parse and process it to display it in a graphical and tabular interface.

3.2. Software design

In Figure 2, the software design is shown. This design describes the process of data from sensors in the incubator being sent to the IoT broker. Next, on the client side, a service is created that runs continuously through a looping process. In this process, sensor data on the broker is taken in real-time and displayed on a web page. The process of showing this data is carried out automatically without the need to refresh the web page. Thus, sensor data from the incubator is displayed in real-time in a web interface without requiring manual intervention for refreshing.

3.3. Hardware design

The hardware schematic design is shown as shown in Figure 3. After completing the schematic design, we continued with making a printed circuit board (PCB) to connect the microcontroller, sensor, and actuator hardware on one board [23]. The PCB design is shown in Figure 4, and information about the hardware used is shown in Table 2.

The microcontroller used is the Arduino Mega board; this device was chosen because of its ability to interact with various sensors and actuators simultaneously. This board has 54 digital input/output pins, of which 15 provide pulse width modulation (PWM) outputs and 16 analog input pins. Each input/output pin can supply 20 mA of current. Then, the Arduino Mega is equipped with 256 KB of flash memory, of which the bootloader uses 8 KB. Arduino Mega operates at a clock speed of 16 MHz, which allows fast processing of instructions [24], [25]. In this research, Arduino Mega is used to read data from DHT22, MLX90614, and MAX30102 sensors, each of which measures parameters such as temperature, heart rate, and blood oxygen level [26]–[28]. In the system configuration process, the DHT22 sensor is connected to the Arduino Mega via a digital pin, enabling data communication between the two. Meanwhile, the MLX90614 and MAX30102 sensors are connected to the Arduino Mega using the inter-integrated circuit (I2C) communication bus, an efficient and reliable two-way communication system for exchanging data between devices. Arduino Mega is also used to control actuator devices based on data from these sensors.

In addition, we also use the DS3231 real-time clock (RTC) module to provide a time marker for data taken from the sensor. This tracks parameter changes over time and aids in data analysis. Next, we use the ESP8266 Wi-Fi module for wireless communication, which is connected to the Arduino Mega using serial (TX/RX). This module provides IoT capabilities for Arduino Mega to send data to the Raspberry Pi via a Wi-Fi network.

Raspberry Pi is a microcomputer that has more than 2GB of random-access memory (RAM) with a quad-core cortex-A72 64-bit processor with a speed of 1.5GHz, equipped with IEEE 802.11ac wireless connectivity at a frequency of 2.4 and 5.0 GHz, Bluetooth 5.0, and also ethernet. Raspberry Pi can also be referred to as a low-cost computer [29]. Then, in this study, the Raspberry Pi functions as an IoT broker in the system, processing and forwarding data from Arduino Mega to the server. The Raspberry Pi is designed to communicate with the Arduino Mega using the MQTT protocol. In addition, the Raspberry Pi also functions as a server, storing sensor data in a database and serving data requests from other devices or users. This design structure combines sensor devices, actuators, network modules, microcontrollers, and microcomputers to create an incubator IoT system. The process of how integration programs from various hardware devices can form an IoT system is shown in Algorithm 1.

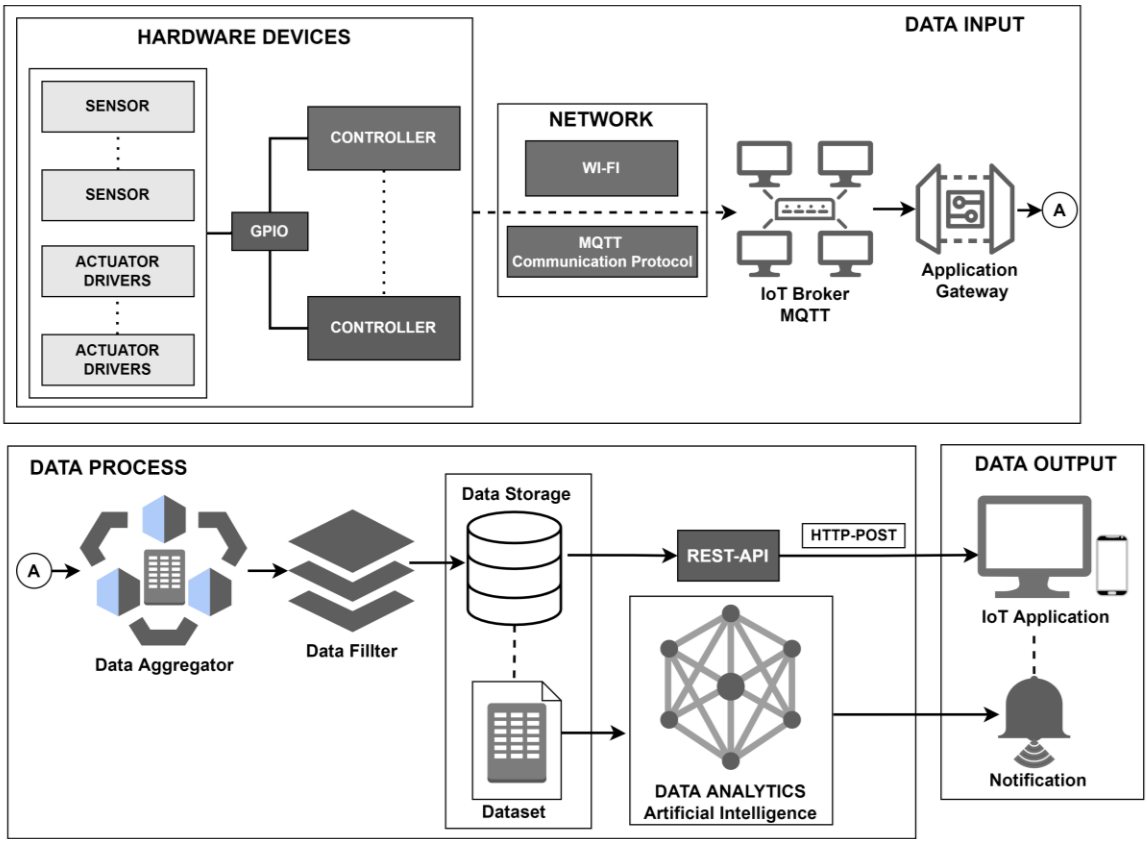


Figure 1. System design overview

Figure 2. Software design

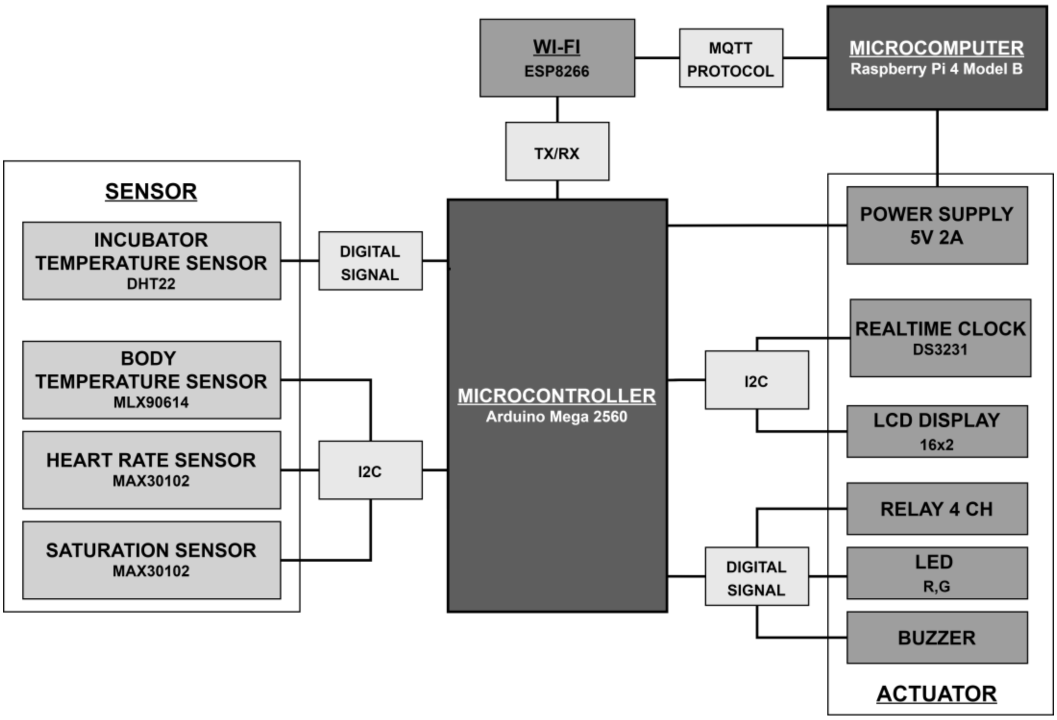


Figure 3. Hardware design

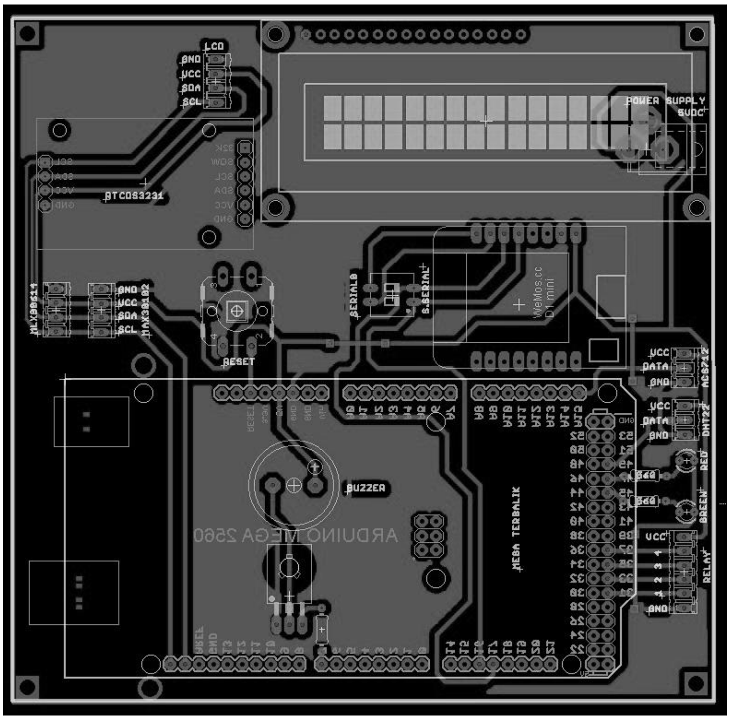


Figure 4. Schematic design of the PCB

Table 2. Module bill of materials

Product/Module	Qty.	Unit price ($)	Total ($)
Board Arduino Mega	1	37.75	37.75
Board Raspberry Pi	1	80.95	80.95
DHT22	1	4.49	4.49
MLX90614	1	22.4	22.4
MAX30102	1	11.16	11.16
ESP8266	1	1.05	1.05
RTC DS3231	1	2.05	2.05
Relay	1	2.88	2.88
LCD	1	1.08	1.08
Light emitting diode (LED)	2	0.13	0.26
Buzzer	1	0.40	0.40
Power Supply 5V 2A	1	9.59	9.59
Total			174.06

Algorithm 1. Integrating hardware into the system

```
Input: hardware components
Output: sensor data is transmitted to the IoT Broker using the MQTT protocol.
Description:
 connect A to sensor dht, mxl, max, esp, rtc
 connect esp, rasp to network
  while the system microcontroller is running:
     read data[ ] = dht, mxl, max,
     read data[time] = rtc
     send data[ ] to microcomputer via network using MQTT protocol
     if response from microcomputer == true:
          process response
     end
  end
  while the system microcomputer is running:
     if data[ ] from microcontroller == true:
          analysis data[ ]
          process store data[ ] into the database
     end
     if data[ ] request from the server or other devices == true:
          send data[ ] to the requesting device
     end
  end
```

3.4. Design network and server

This section discusses how the network design is used in communication in the baby incubator system. Wi-Fi networks are used as a means of wireless connection between IoT devices. Wi-Fi works by using radio waves to send and receive data between devices. This is an advantage of using Wi-Fi as devices can easily connect, allowing for greater mobility compared to wired networks. The IEEE developed the Wi-Fi standards, starting with the number 802.11 [30]. Through Wi-Fi, IoT devices such as sensors or actuators in incubators can connect to the Internet, allowing them to send and receive data with the MQTT communication protocol.

MQTT is a lightweight communication protocol for exchanging data in IoT devices on incubators. MQTT is designed for efficient two-way communication and reliable message delivery, even in unstable network conditions. In addition, MQTT also supports QoS mechanisms up to level three, which enables reliable message delivery. The MQTT architecture has two types of entities, namely, publisher and subscriber. The publisher sends a message to the broker IoT, which then distributes the message to subscribers who are subscribed to the relevant topic of the message. The Broker acts as a message handler and ensures the messages arrive at the right destination [31], [32].

This study uses the Eclipse Mosquitto application as an IoT broker, which is open-source software that facilitates the implementation of the MQTT protocol [33]. For this broker installation, configuration involves address and port settings, authentication and authorization, and QoS. Configuration is done on the client side using the Paho-MQTT library [34]. The IoT broker configuration is illustrated in Figure 5, which illustrates the configuration process flowchart.

Meanwhile, the gateway application functions as an access point between IoT devices and servers via the network, also carrying out processes such as light data processing, filtering, and security. Then, the server is used for data processing and control in IoT devices. This server is implemented on the Raspberry Pi microcomputer module. Application services running on this server are responsible for collecting, storing, and analyzing data from IoT devices and sending instructions back to the device based on data processing results.

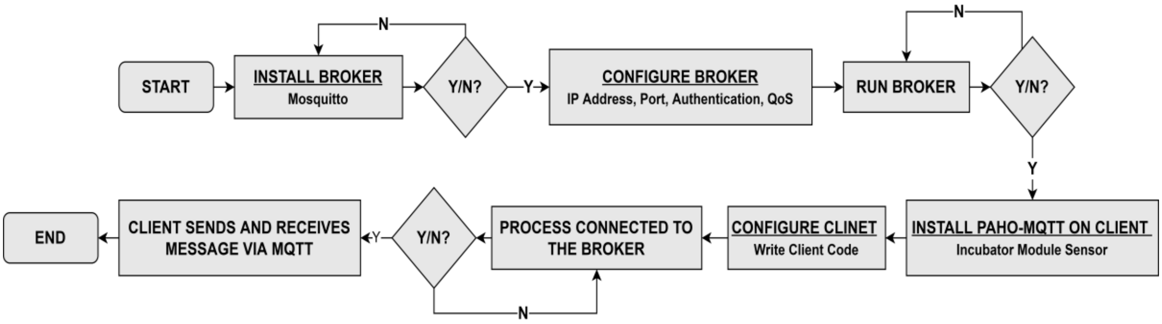


Figure 5. The flow of IoT broker configuration

3.5. Design data process
3.5.1. Data aggregator

Collecting data from the internet of things devices in the incubator uses a data aggregator service. This service is needed because of the number and variety of IoT devices that can connect to the network at one time, so it is essential to manage the flow of information and ensure data originating from various IoT devices can be analyzed and utilized effectively. The processes carried out on the data aggregator are: i) collecting data from various sensor devices installed in the incubator; ii) normalizing data by converting sensor data into a consistent format so that it can be processed; iii) simple data processing to identify patterns or detecting data anomalies obtained on sensor and iv) presenting the final data. The following Algorithm 2 shows the procedure for the data aggregator.

Algorithm 2. data aggregator service
```
INPUT: Sensor data from IoT devices
OUTPUT: Sensor data is shaped according to the format of the system
 Description:
  data_store = {
     JSON_object: []
  }
  handle_sensor_data(data):
     normalized_data = normalize(data)
     data_store[sensor_id].append(normalized_data)
  end
  receive_sensor_data(device_type, data):
     for each_item_sensor in id_sensor do
        if device_type == sensor_id:
            handle_sensor_data(data)
        end
    end
  end
  analyze_sensor_data():
    result = analyze_process (data_store[sensor_id])
    return:
        result
    end
  end
```

3.5.2. Data filtering

In this research, the data filtering process needs to be done because the data generated by internet of things devices is extensive and varied. By implementing data filtering, the system can enhance the quality of sensor device data and bolster cybersecurity by mitigating potential cyberattacks [35]. There are two types of applications of the data filtering process carried out on the system, namely: i) Noise reduction because the data generated by IoT devices can contain noise or interference that can affect the accuracy of the data; the process is carried out by calculating the average number of data points in a row in a data; and ii) Outlier detection because the data obtained from the sensor may have a value that is much different from other values and can be caused by various factors such as measurement errors from the sensor or abnormal conditions. The process is carried out by determining the upper and lower limits of the data and identifying any values outside these limits as outliers. Algorithm 3 shows the procedure of the data filter module.

Algorithm 3. Data filter
```
Input: Sensor data
Output: Filtered sensor data of the system
 Description:
 data = data_sensor
 noise (data, size):
    results []
    for i in range(len(data) - size + 1):
         x = sum(data[i:i+size])/ size
         results[] = x
    end
    return:
         results[]
    end
 end
 detect_outliers(data):
    low = set_lower_bound
    high = set_upper_bound
    for x in data:
         if data < low:
             outliers = data
         end
         elseif data > high:
             outliers = data
         end
    end
    return:
         outliers
    end
 end
```

3.5.3. Data storage

In this study, sensor devices are used to collect data from the incubator environment through connected sensors, then send this data to a database by passing through a filtering process. Data storage in the database requires several tables related to each other to enable efficient access and further data analysis. The design of the database schema on the system defines the characteristics of the data stored in the database, including the type of data as shown in Table 3.

Table 3 describes the sensor data, namely, the incubator temperature, measured in degrees Celsius, as a parameter stored with the float data type in the database so that it has a high precision value in representing temperature. Furthermore, body temperature is monitored and recorded. Like the incubator temperature, the database stores body temperature as a float data type. Meanwhile, heart rate is another vital parameter monitored, stored as an Integer data type in the database. Next, the oxygen saturation level in the blood is expressed as a percentage. Oxygen saturation is stored as a decimal data type in the database. In addition to these parameter data, records of data retrieval times are stored as the Timestamp data type in the database. Thus, these various types of data can be stored efficiently in the database, facilitating the analysis process.

Table 3. Data description for sensor devices

Device	Data variable	Example value	Unit	Data type
DHT22	Incubator temperature	33.00	℃	Float
MLX90614	Body temperature	37.00	℃	Float
MAX30102	Heart rate	150	Beats per minute (BPM)	Integer
MAX30102	Saturation	99	SpO_2	Decimal
RTC Real-time	Date-time	2023-07-21 10:00:00	yyyy-mm-dd hh:mm:ss	Timestamp

3.5.4. Dataset

In this study, the data set in the database was organized to become the dataset used for modeling analysis. This dataset consists of features retrieved from database columns. The features of the dataset are incubator temperature (*T_i*), body temperature (*T_i*), heart rate (HR), and saturation peripheral oxygen (SPO), as shown in Table 4. Then, each row of data will include values for each feature, which are used to determine the infant's condition from time to time based on model training to produce predictive values or patterns of incubator temperature conditions. In addition, given the complexity and sensitivity of health data, we are also aware of handling this dataset with care, ensuring data privacy and security, and only using data for legal and ethical purposes.

Table 4. Dataset attribute

Attribute	Unit	Descriptions
T_b	℃	Body temperature
T_i	℃	Temperature around the incubator environment
HR	BPM	Heart rate in infants
SPO	SpO_2	Saturation (oxygen level)

Next, we use this dataset to calculate the correlation value between the attributes. The target or label chosen in this dataset is the incubator temperature or *T_i*. Selecting attributes that correlate with the target can increase the effectiveness of the model training process. The correlation equation r uses the formula shown in (1). This equation can be used if the dataset has a sequential time series with vector $X=(x_1,x_2,x_3,\ldots,x_n)$, and there must also be vector $Y=(y_1,y_2,y_3,\ldots,y_n)$. Then, the results of the value of r are considered to have a positive correlation if the attribute value is in the range $0 < r < 1$, while it is considered to have a negative correlation if the attribute value is in the range $-1 < r < 0$ [36].

$$r = \frac{n\sum_{i=1}^{n} x_i\, y_i - \sum_{i=1}^{n} x_i \sum_{i=1}^{n} y_i}{\sqrt{n \sum_{i=1}^{n} x_i^2 - (\sum_{i=1}^{n} x_i)^2}\ \sqrt{n \sum_{i=1}^{n} y_i^2 - (\sum_{i=1}^{n} y_i)^2}} \quad (1)$$

3.5.5. Long short-term memory

The LSTM method is a particular recurrent neural network (RNN) designed to solve the problem of long-term and short-term dependencies in data sequences. LSTM overcomes this problem with gate operations and memory cells that retain information longer. Three types of gates are used, namely, the forget gate to determine how far information from the previous step must be maintained or forgotten, the input gate to determine how far new data from the current input must be stored in the memory cell, the output gate to determine how far the data from the cell memory must be used to calculate the output current [37], [38].

The application of the LSTM method in the IoT system in the incubator in this study begins with the initialization step and data preparation to be analyzed based on the attribute dataset, as shown in Table 3. Then, the LSTM model is built and trained using training data. This process involves determining parameters such as the number of layers and neurons and selecting activation and optimization operations. Once the model is trained, it can make predictions based on test data. This makes evaluating the model performance possible by measuring the difference between the prediction and the actual value. Algorithm 4 represents the basic steps of the LSTM.

Algorithm 4. LSTM application in IoT systems in incubators

```
Input: Dataset sensor
Output: Predictive value
 DESCRIPTION:
 initialize LSTM parameters
 prepare training_data
 prepare test_data
 training (training_data):
     for each sample in training_data do:
         compute all gate outputs and states (w_f, w_i, w_o, cell state)
         compute final output
     end
 end
 testing (test_data):
     for each sample in test_data do:
         compute all gate outputs and states
         record final output as prediction
     end
     for each prediction in actual value do:
         compute difference between prediction and actual value
         record differences
     end
 end
```

Then mathematically, the operations within the LSTM unit, which enable it to recognize patterns in sequential data and retain and manipulate information over long periods, can be described by (2)-(9), and the structure of the LSTM is shown in Figure 6 [39]–[41]. All stages in model building, from data training to testing, were carried out by using the Python programming language. In addition, we also rely on several additional libraries to support the modeling process.

$$F_t = sigmoid([H_{t-1}, X_t] \,.\, W_f + b_f) \tag{2}$$

$$I_t = sigmoid([H_{t-1}, X_t] \,.\, W_i + b_i) \tag{3}$$

$$\widetilde{C_t} = \tanh([H_{t-1}, X_t] \,.\, W_c + b_c) \tag{4}$$

$$C_t = (F_t \,.\, C_{t-1}) + (I_t \,.\, \widetilde{C_t}) \tag{5}$$

$$O_t = sigmoid(W_o \,.\, [H_{t-1}, X_t] + b_o) \tag{6}$$

$$H_t = O_t \,.\, \tanh(C_t) \tag{7}$$

$$sigmoid(x) = \frac{1}{1+e^{-x}} \tag{8}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{9}$$

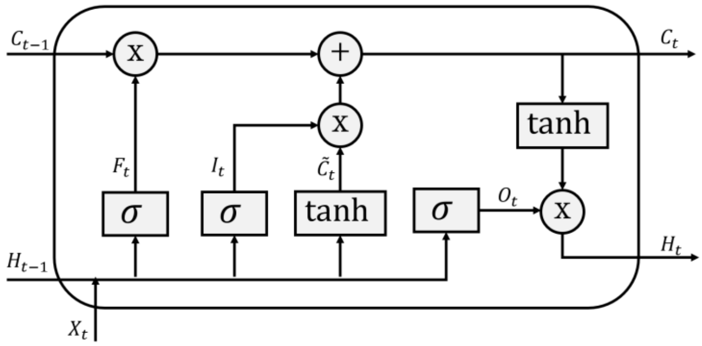


Figure 6. LSTM design structure

3.6. Evaluation of predictive performance

The model obtained from training with the LSTM was evaluated using several techniques. Evaluation techniques used in this study include root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and coefficient of determination (R^2). RMSE measures the difference between the predicted value generated by the model and the actual value, as shown in (10) [42], [43]. Meanwhile, MAE is similar to RMSE, but the difference lies in its tolerance for outliers, which is stated in (11) [42], [43]. In addition, MAPE is used to determine the scaling of the prediction error in percentage terms, the calculation of which is described in (12) [42], [43].

On the other hand, R^2 is used to measure the variation of the variables in the model. If the R^2 value is high, the model can be considered good, whereas if the value is low, then the model may not be effective. The equation for R^2 is shown in (13)-(15) [42], [43]. In this study, the symbols in the evaluation formula can be explained as follows: *n* represents the amount of data, y_i is the actual value, $\widetilde{y_i}$ is the predicted value, and Σ represents the sum of all data.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(y_i - \widetilde{y_i})^2}{n}} \tag{10}$$

$$MAE = \frac{\sum_{i=1}^{n}|y_i - \widetilde{y_i}|}{n} \tag{11}$$

$$MAPE = \frac{1}{n}\sum_{i=1}^{n}\frac{|y_i - \widetilde{y_i}|}{y_i} \times 100\% \tag{12}$$

$$SS_{res} = \sum_{i=1}^{n}(y_i - \widetilde{y_i})^2 \tag{13}$$

$$SS_{tot} = \sum_{i=1}^{n}(y_i - \bar{y})^2 \tag{14}$$

$$R^2 = 1 - \frac{SS_{res}}{SS_{tot}} \quad (15)$$

3.7. Design data output

Figure 7 shows the design for the output data generated by the incubator. This includes how data is collected, processed, and presented to users. This design includes essential elements that ensure critical information can be easily understood and applied to make informed decisions about baby care. The components in this data output include data visualization and monitoring, notification, control system, data logging, predictive analytics, and system integration.

In this study, the process of visualizing and monitoring data is carried out after the sensor data on the incubator has been successfully sent to the database, where the data is then visualized and monitored through a web dashboard that has been designed. This dashboard allows users to monitor real-time status indicators and incubator condition trends through a responsive web interface with intuitive graphs and tables. Then, the designed notification system can automatically warn users when abnormal or emergency conditions occur. Apart from providing notifications, this system can also be controlled remotely via a web dashboard by adjusting the operational parameters of the incubator. Furthermore, data collected and stored in a database can be downloaded in various file formats to facilitate data access and analysis. For example, data can be downloaded as XLSX, CSV, JSON, PDF, or XML files, allowing users to perform further analysis.

The analysis process uses the LSTM method to develop a model that can predict the temperature value in the incubator based on previous data patterns stored in the database. The results of this prediction can then be used to optimize incubator operations, predict changes in conditions, and make adjustments before conditions change significantly. This system is based on an IoT application so that it can be integrated with other systems such as electronic medical record systems through the API. Thus, the data generated by this IoT incubator system can be used to create various applications to improve baby care.

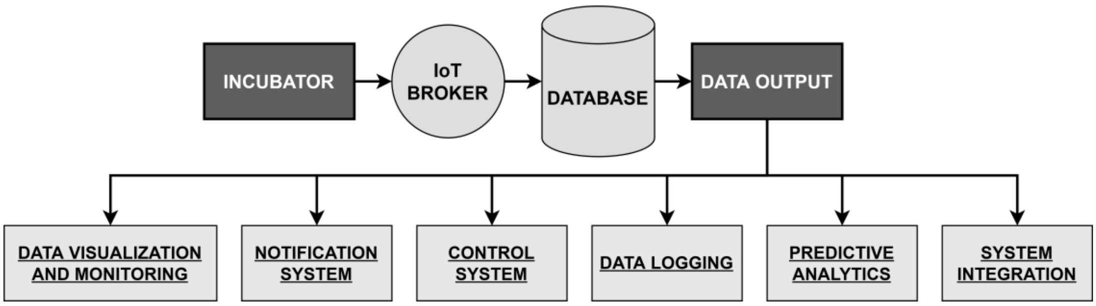


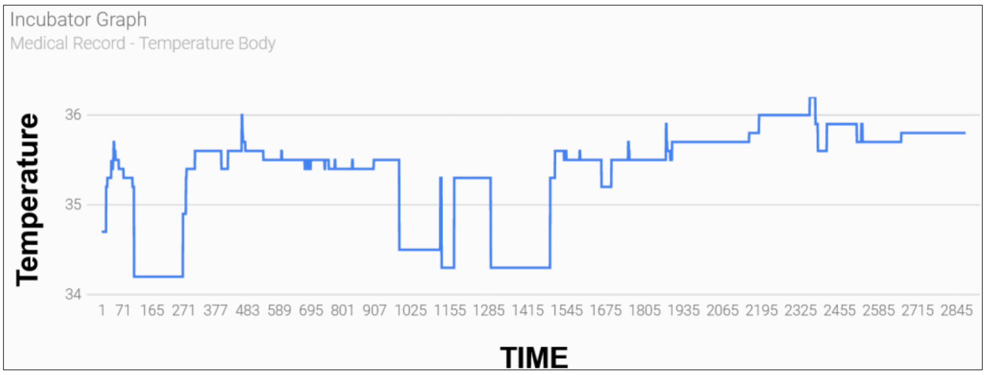
Figure 7. Design of output data

4. RESULTS AND DISCUSSION

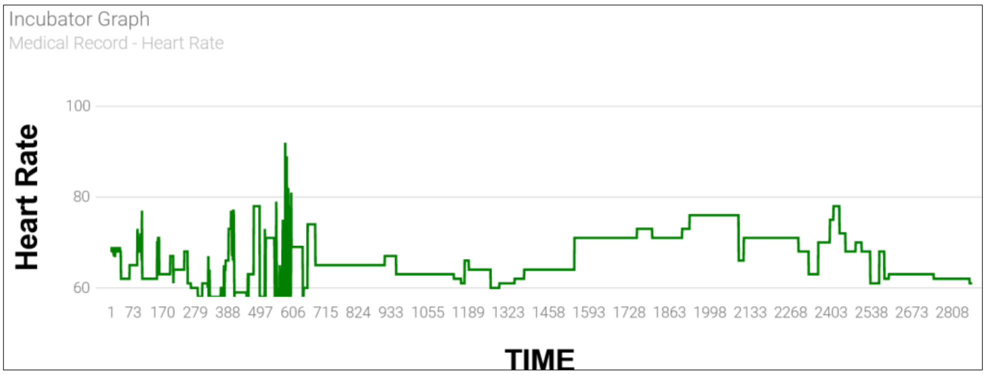
This section will present the system analysis and design results implemented previously. This research aims to develop a baby incubator system by adopting the IoT concept, which is combined with the LSTM method to make predictions about incubator temperature values in the future. In addition, sensor data from the hardware installed on the incubator will be visualized using web technology and stored in a database for easy access and further analysis.

The first process involves collecting data from measurements made by sensors attached to the incubator. Data from this sensor is divided into four categories: baby body temperature sensors, incubator environmental sensors, heart rate sensors, and saturation sensors. Each of these sensors has an essential role in monitoring the infant's condition and the environment in the incubator. Visualization of these various sensors produces patterns of information, as shown in Figure 8(a) shows body temperature data visualization, Figure 8(b) shows heart rate data, Figure 8(c) shows incubator temperature data, and Figure 8(d) shows saturation data. We then use these patterns to predict future temperature values in developing learning models.

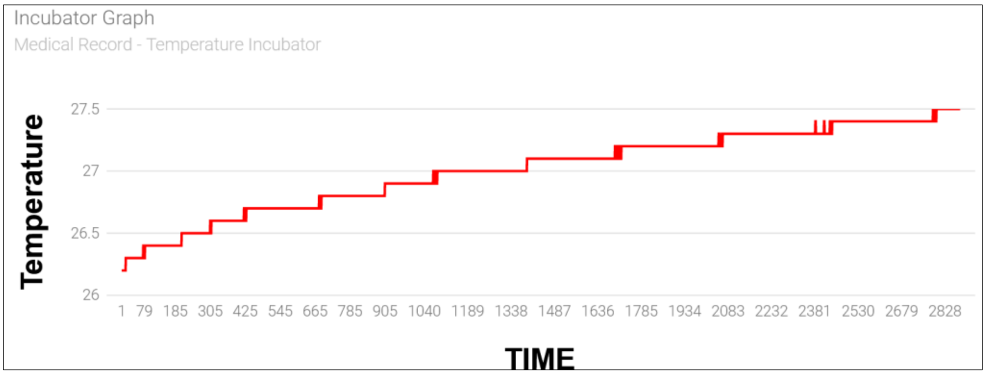
Measurement results data from sensors in the incubator are stored directly in the database, and the characteristics of the data are shown in Table 5. This data is then used as a dataset in making learning models. This dataset is multivariate data, where each row includes various features such as time, body temperature, incubator temperature, heart rate, and saturation. This dataset is also a time series data, meaning that the data is collected sequentially over time, and each data point is related to the data point before and after. In this dataset, the time intervals are five seconds apart, and the total amount of data is 2,880 taken over four hours, as shown in Table 6.



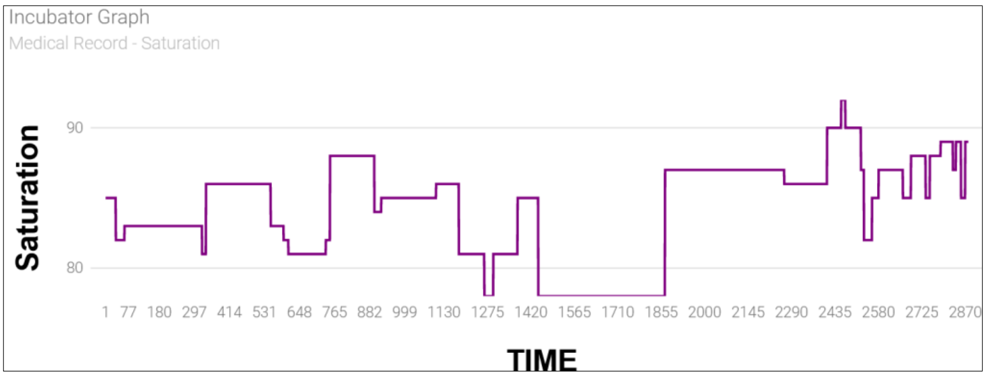
(a)



(b)



(c)



(d)

Figure 8. Pattern of data for each sensor measurement (a) body temperature, (b) heart rate, (c) incubator temperature, and (d) saturation

Table 5. Statistical summary of the sensor data characteristics

Variable	T_b	T_i	HR	SPO
count	2,880	2,880	2,880	2,880
mean	35.36	27.01	66.43	84.21
std	0.54	0.32	5.04	3.54
min	34.2	26.2	58	78
max	36.2	27.5	92	98

Table 6. Dataset information in the database record table

Item	Hour	T_b	T_i	HR	SPO
1	15:00:05	34.7	26.2	68	85
2	15:00:10	34.7	26.2	68	85
3	15:00:15	34.7	26.2	68	85
.	.	.	.	.	.
.	.	.	.	.	.
2878	18:59:50	35.8	27.5	61	89
2879	18:59:55	35.8	27.5	61	89
2880	19:00:00	35.8	27.5	61	89

The dataset is divided into two parts, namely 80% training data, or 2,304 data lines, and 20% test data, or 576 data lines. The training data trains the model and sets parameters to make predictions. At the same time, the test data is used to evaluate the extent to which the model can make correct predictions on previously unknown data.

Based on Table 7, it can be seen that the incubator temperature attribute has a positive correlation with all other attributes. The correlation value between incubator temperature and body temperature is 0.48, with a heart rate of 0.29 and a saturation of 0.23. The correlation results show that the dataset used in this study tends to have a positive correlation with incubator temperature. Therefore, we can identify the factors that influence the incubator temperature for the analysis process and make decisions based on the correlation between the attributes in the dataset.

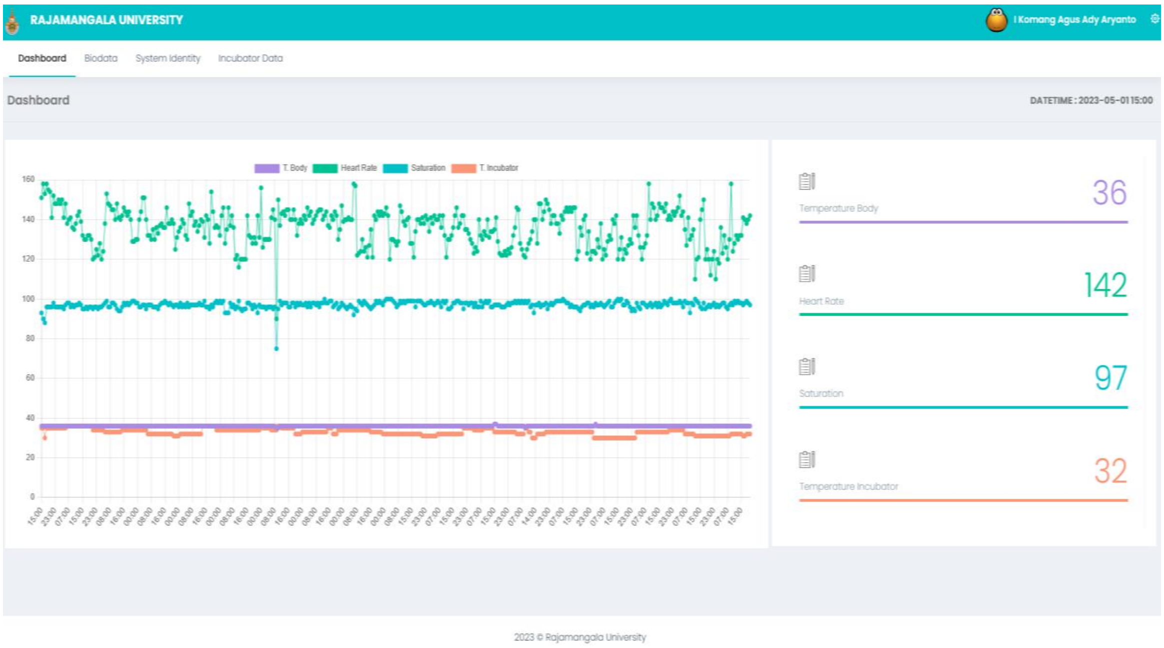
Table 7. Correlation coefficients of attribute in the dataset

	T_b	T_i	HR	SPO
T_b	1.0	0.48	0.35	0.31
T_i	0.48	1.0	0.29	0.23
HR	0.35	0.29	1.0	0.022
SPO	0.31	0.23	0.022	1.0

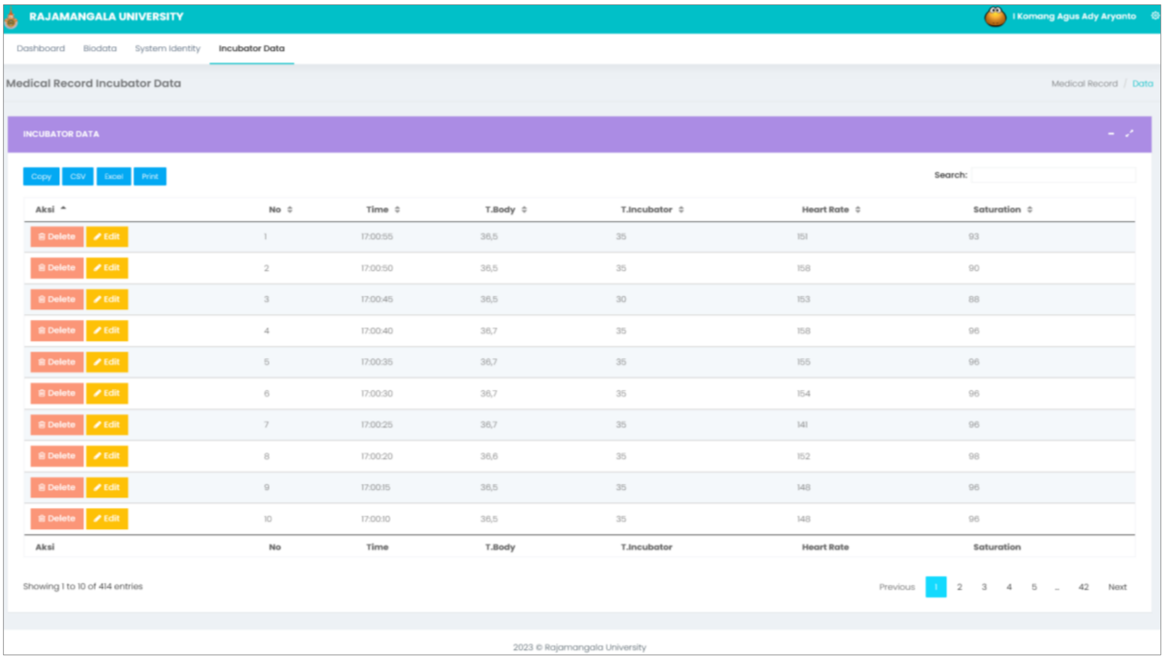
The results of web application development are shown in Figure 9(a) for the dashboard and control system pages, Figure 9(b) for sensor data pages, and Figure 9(c) for notification pages. The web application includes a dashboard page that displays real-time data from various sensors, including the baby's body temperature, incubator temperature, heart rate, and oxygen saturation sensors. This data is presented in graphs and numbers so users can easily understand and analyze patterns in the data based on specific time records. In addition, an essential feature of this application is to provide notifications or warnings if the data from the sensor shows abnormal conditions. Through the web application, users can also manually control the conditions in the incubator. This gives the user the flexibility to set and monitor temperature conditions in real-time, thus ensuring an optimal environment for the baby being treated in the incubator.

Furthermore, this application has a page to display data stored in database data. The data displayed on this page can be printed in various formats such as CSV, Excel, JSON, or PDF. Through this page, users can also sort and search data. This sorting feature can be done by determining the data parameters such as time, sensor value, or sensor type. Meanwhile, the data search feature can be used by entering certain keywords, helping users quickly find specific data in the system.

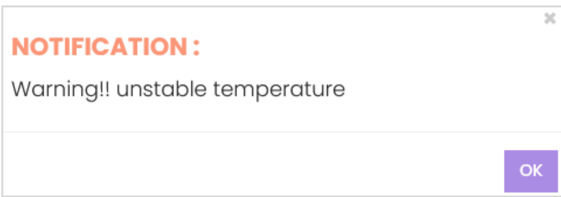
Figure 10 shows the results of implementing the electronic incubator designed in this study. Figure 10(a) shows the hardware installation results on the incubator, Figure 10(b) depicts the implementation of the PCB along with sockets for connecting sensors and microcontrollers, while Figure 10(c) displays the sensor devices installed on the incubator. Each component, from sensors to actuators, is structured in a PCB layout. After that, the program is uploaded to the microcontroller for testing. During the test, all components are operational and active, and sensor data such as heart rate, incubator temperature, body temperature, and oxygen saturation can also be collected. In addition, testing is also carried out to ensure all actuator devices function correctly.



(a)



(b)



(c)

Figure 9. Web application (a) dashboard page and control system, (b) data record page, and (c) notification

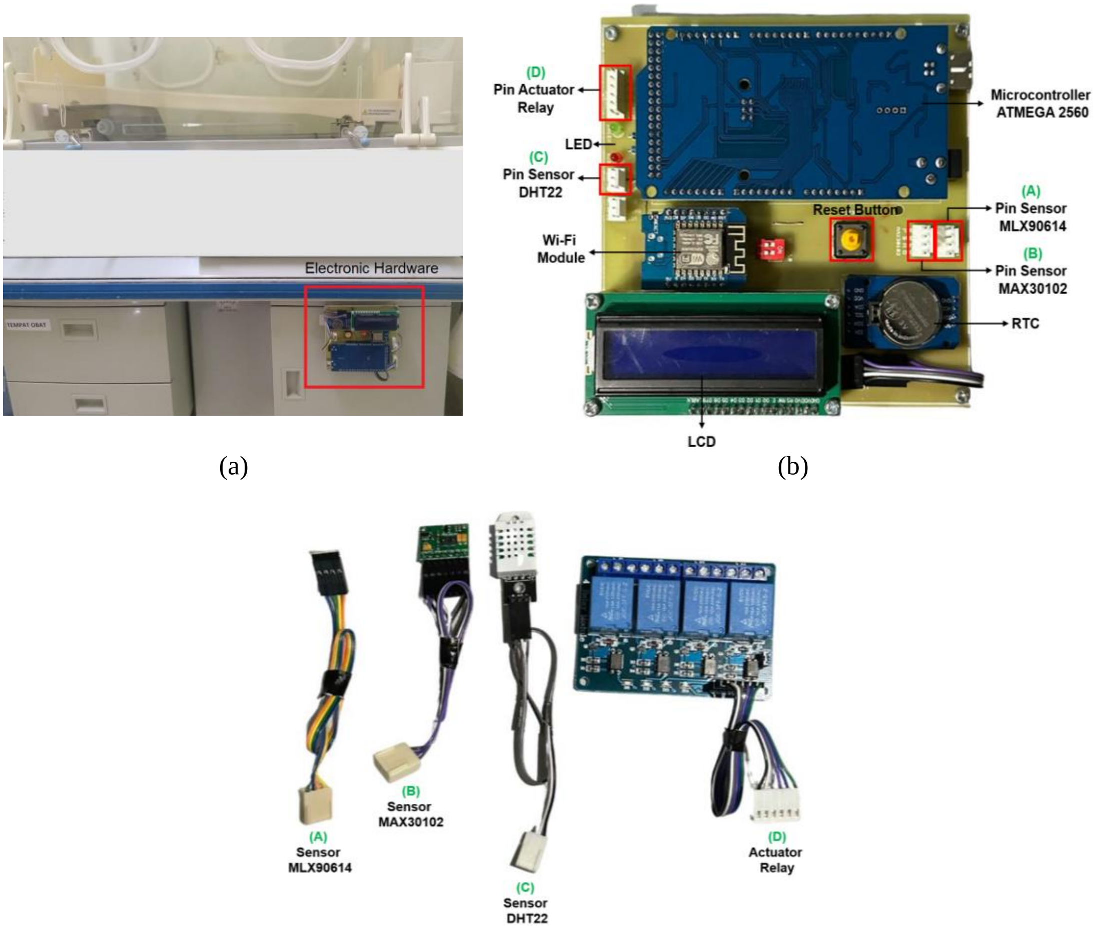


Figure 10. The hardware results including (a) device installation, (b) PCB and microcontroller components, and (c) sensors and actuators

Furthermore, testing was also carried out to test the integration process between electronic devices and the IoT platform. The Wi-Fi module is used as a connecting network so that sensor data that has been collected can be sent to an IoT Broker, where the data is then stored in a database and visualized via web pages, as shown in Table 4, Table 5, and Figure 8. In addition, this sensor data is also used for the dataset for making a model with the LSTM method, with results as shown in Table 7.

Based on the results of experiments carried out on the learning model, the results are shown in Table 8. We evaluated this model using RMSE, MAE, MAPE, and R^2 metrics. This model uses two layers of LSTM, and we conducted experiments with several numbers of neurons in each layer. The results show that as many as 60 neurons produce the highest predictive value or R^2, equal to 0.934. However, it is essential to note that model performance does not always increase as the number of neurons increases.

Furthermore, we also conducted experiments on the model by adjusting the lookback value. Lookback refers to the model ability to look at data at previous time intervals before making predictions about future data. For example, in this dataset, with a time interval of 5 seconds, if lookback is set as 6, then the model will predict 1 data forward based on 6 previous data. We conducted an experimental scenario by setting the lookback value from 1 to 10, with 60 neurons for each layer. The experimental results show that lookback 6 produces the best predictive value, which is 0.934, as shown in Table 9.

Next, we also experimented with inputting each attribute from the dataset into the model. From Table 10, it can be seen that the predicted value for the input attribute *T_b* has the highest value of 0.845. As for the *HR* input, a predictive value of 0.805 is obtained, and the *SPO* input produces a predictive value of 0.590. The results of the predicted value of each attribute show that all attributes positively impact model performance because each attribute has a different contribution to the final prediction. This analysis reveals that the attributes *T_b*, HR, and SPO each make important contributions to predictions, with *T_b* having the most significant impact.

Table 8. The result of selecting the number of neurons

Number of neurons LSTM - 1	Number of neurons LSTM - 2	RMSE (ºC)	MAE (ºC)	MAPE (%)	R^2
100	100	0.017	0.008	0	0.918
90	90	0.020	0.014	0.1	0.893
80	80	0.027	0.022	0.1	0.793
70	70	0.029	0.026	0.1	0.770
60	60	0.015	0.008	0	0.934
50	50	0.015	0.006	0	0.933
40	40	0.045	0.023	0.1	0.436
30	30	0.042	0.027	0.1	0.503
20	20	0.059	0.046	0.2	0.035
10	10	0.023	0.016	0.1	0.851
5	5	0.031	0.021	0.1	0.726

Table 9. The result of the selection is the number of lookbacks

Lookbacks	RMSE (ºC)	MAE (ºC)	MAPE (%)	R^2
1	0.023	0.015	0.1	0.859
2	0.047	0.035	0.1	0.392
3	0.024	0.017	0.1	0.843
4	0.030	0.027	0.1	0.751
5	0.034	0.018	0.1	0.677
6	0.015	0.008	0	0.934
7	0.028	0.022	0.1	0.787
8	0.032	0.030	0.1	0.719
9	0.026	0.021	0.1	0.814
10	0.041	0.028	0.1	0.522

Table 10. The results of the selection of input types for the LSTM

Input type	Correlation coefficient (r)	RMSE (ºC)	MAE (ºC)	MAPE (%)	R^2
T_b	0.48	0.024	0.020	0.1	0.845
HR	0.29	0.026	0.019	0.1	0.805
SPO	0.23	0.038	0.035	0.1	0.590

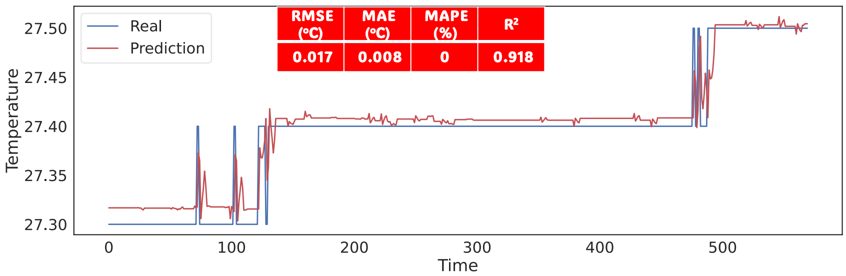
The plotting results between the predicted results and the actual values in the test data are shown in Figure 11 (in Appendix). The red line shows the predicted result, while the blue line shows the actual value. In general, the plotting results show that the predicted value follows the pattern of the actual value. Even so, the model does not follow the actual value pattern perfectly. Several differences or deviations between the predicted results and the actual values need further attention to improve the accuracy and quality of model predictions.

In Figure 11(a), the plot results are obtained with LSTM-1 100 neurons and LSTM-2 100 neurons with a lookback of 6. Figure 11(b) shows the plot results with LSTM-1 90 neurons and LSTM-2 90 neurons, also with a lookback of 6. Figure 11(c) displays the plot results with LSTM-1 60 neurons and LSTM-2 60 neurons, again with a lookback of 6. Lastly, in Figure 11(d), the plot results are achieved with LSTM-1 50 neurons and LSTM-2 50 neurons, maintaining a lookback of 6.

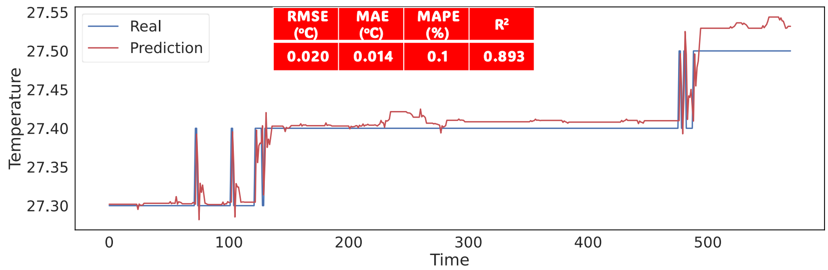
5. CONCLUSION

This research presents a new approach to incubator system development with the internet of things and deep learning concept, designed by combining hardware, network, software, and database management. Based on the research results, it was found that hardware such as sensors installed in incubators could measure environmental conditions and send their values to the database via the Internet using the MQTT protocol. The sensor device connected to the microcontroller performs measurements every five-second interval. In the experiment, the incubator was turned on for 4 hours, and the data was successfully stored in a database of 2,880 rows. In addition, through the web application, users can view real-time data with visual graphs and tables. This web application also provides notifications if conditions change in the incubator. In the web application, the user can also manually control the temperature conditions inside the incubator. This system is also equipped with API features to facilitate integration with other systems related to patient health data. The data stored in the database is also used as a dataset for model building using the LSTM deep learning method. This dataset has four attributes, namely *T_i*, *T_b*, *HR*, and *SPO*. The attribute selected as the target or label in the dataset is *T_b*. Furthermore, the value of the correlation coefficient (*r)* on the dataset for each attribute with a target (*T_b*) is obtained, namely *T_i* of 0.48, *HR* of 0.29, and *SPO* of 0.23. Then, based on this dataset, the model was developed using LSTM by dividing the dataset into two types, namely 80% training data and 20% test data. The best prediction results are obtained using two LSTM layers from several model experiments on test data. Each layer has 60 neurons and 6 lookback settings, producing an R^2 prediction value of 0.934. In addition, the error value of the model for the test data for each matrix is RMSE of 0.015 and MAE of 0.008. Furthermore, based on the results of input experiments with each attribute in the dataset, it appears that the best R^2 predictions are obtained for the *T_b* attribute with a value of 0.845, for the *HR* attribute with a value of 0.805, and the *SPO* attribute with a value of 0.590. These results show the variability of predictive performance for each input attribute.

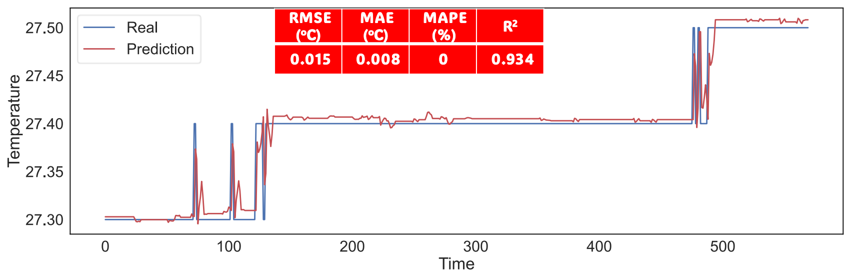
APPENDIX



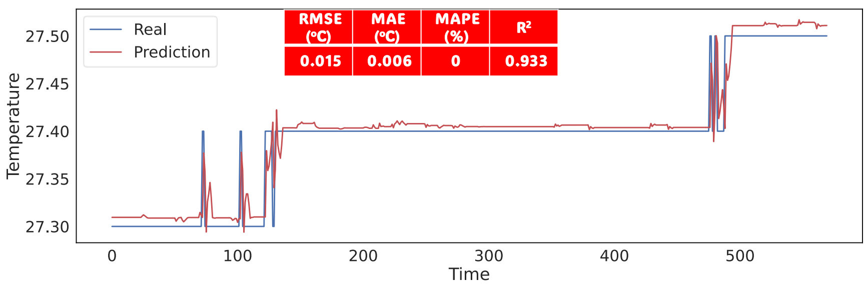
(a)



(b)



(c)



(d)

Figure 11. Results comparison of actual value and predicted value (a) LSTM 1 and 2: 100, (b) LSTM 1 and 2: 90, (c) LSTM 1 and 2: 60, and (d) LSTM 1 and 2: 50

REFERENCES

[1] A. Latif, H. A. Widodo, R. A. Atmoko, T. N. Phong, and E. T.Helmy, "Temperature and humidity controlling system for baby incubator," *Journal of Robotics and Control (JRC)*, vol. 2, no. 3, pp. 190–193, 2021, doi: 10.18196/jrc.2376.

[2] M. A. Mahdi, A. H. Issa, and S. A. Gitaffa, "Machine learning techniques based data-driven fault detection of multiple transducers," in *AIP Conference Proceedings*, 2023, doi: 10.1063/5.0154268.

[3] F. Ahmed, A. Sufyan, S. Hussain, A. Rehman, S. Amin, and Y. Salam, "Smart baby incubator," in *2021 International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, Oct. 2021, pp. 1–5, doi: 10.1109/ICECube53880.2021.9628245.

[4] L. Lamidi, A. Kholiq, and M. Ali, "A low cost baby incubator design equipped with vital sign parameters," *Indonesian Journal of electronics, electromedical engineering, and medical informatics*, vol. 3, no. 2, pp. 53–58, May 2021, doi: 10.35882/ijeeemi.v3i2.3.

[5] A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, and M. Ayyash, "Internet of things: a survey on enabling technologies, protocols, and applications," *IEEE Communications Surveys & Tutorials*, vol. 17, no. 4, pp. 2347–2376, 2015, doi: 10.1109/COMST.2015.2444095.

[6] M. A. Khan *et al.*, "A deep learning-based intrusion detection system for MQTT enabled IoT," *Sensors*, vol. 21, no. 21, pp. 1–25, Oct. 2021, doi: 10.3390/s21217016.

[7] W. Widhiada, I. N. G. Antara, I. N. Budiarsa, and I. M. G. Karohika, "The robust PID control system of temperature stability and humidity on infant incubator based on Arduino ATMega 2560," *IOP Conference Series: Earth and Environmental Science*, vol. 248, no. 1, pp. 1–12, Apr. 2019, doi: 10.1088/1755-1315/248/1/012046.

[8] W. H. Organization, R. Health, P. Cooper, and M. McCormick, "Managing newborn problems: a guide for doctors, nurses, and midwives," *World Health Organization*, 2003.

[9] I. N. K. Wardana, J. W. Gardner, and S. A. Fahmy, "Optimising deep learning at the edge for accurate hourly air quality prediction," *Sensors*, vol. 21, no. 4, pp. 1–28, Feb. 2021, doi: 10.3390/s21041064.

[10] J. Krohkaew, P. Nilaphruek, N. Witthayawiroj, S. Uapipatanakul, Y. Thwe, and P. N. Crisnapati, "Thailand raw water quality dataset analysis and evaluation," *Data*, vol. 8, no. 9, pp. 1–17, Sep. 2023, doi: 10.3390/data8090141.

[11] I. Sukma *et al.*, "Real-time wireless temperature measurement system of infant incubator," *International Journal of Electrical and Computer Engineering (IJECE)*, vol. 13, no. 1, pp. 1152–1160, Feb. 2023, doi: 10.11591/ijece.v13i1.pp1152-1160.

[12] P. T. Kapen, Y. Mohamadou, F. Momo, D. K. Jauspin, N. Kanmagne, and D. D. Jordan, "Development of a neonatal incubator with phototherapy, biometric fingerprint reader, remote monitoring, and heart rate control adapted for developing countries hospitals," *Journal of Neonatal Nursing*, vol. 25, no. 6, pp. 298–303, Dec. 2019, doi: 10.1016/j.jnn.2019.07.011.

[13] M. Irmansyah, "Design and application of portable heart rate and weight measuring tool for premature baby with microcontroller base," *International Journal of GEOMATE*, vol. 17, no. 61, pp. 195–201, Sep. 2019, doi: 10.21660/2019.61.ICEE12.

[14] A. Alimuddin, R. Arafiyah, I. Saraswati, R. Alfanz, P. Hasudungan, and T. Taufik, "Development and performance study of temperature and humidity regulator in baby incubator using fuzzy-PID hybrid controller," *Energies*, vol. 14, no. 20, pp. 1–21, Oct. 2021, doi: 10.3390/en14206505.

[15] P. A. Aya-Parra, A. J. Rodriguez-Orjuela, V. Rodriguez Torres, N. P. Cordoba Hernandez, N. Martinez Castellanos, and J. Sarmiento-Rojas, "Monitoring monitoring system for operating variables in incubators in the neonatology service of a highly complex hospital through the internet of things (IoT)," *Sensors*, vol. 23, no. 12, pp. 1–18, Jun. 2023, doi: 10.3390/s23125719.

[16] G. Cay *et al.*, "NeoWear: an IoT-connected e-textile wearable for neonatal medical monitoring," *Pervasive and Mobile Computing*, vol. 86, pp. 1–17, Oct. 2022, doi: 10.1016/j.pmcj.2022.101679.

[17] M. A. Mahdi, S. A. Gittaffa, and A. H. Issa, "Multiple fault detection and smart monitoring system based on machine learning classifiers for infant incubators using raspberry Pi 4," *Journal Européen des Systèmes Automatisés*, vol. 55, no. 6, pp. 771–778, Dec. 2022, doi: 10.18280/jesa.550609.

[18] P. N. Crisnapati and D. Maneetham, "RIFIS: a novel rice field sidewalk detection dataset for walk-behind hand tractor," *Data*, vol. 7, no. 10, pp. 1–16, Sep. 2022, doi: 10.3390/data7100135.

[19] H. Shi, L. Niu, and J. Sun, "Construction of industrial internet of things based on MQTT and OPC UA protocols," in *2020 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA)*, Jun. 2020, pp. 1263–1267, doi: 10.1109/ICAICA50127.2020.9182598.

[20] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural Computation*, vol. 31, no. 7, pp. 1235–1270, Jul. 2019, doi: 10.1162/neco_a_01199.

[21] A. Huiyao, S. Yang, Y. Tao, L. Hui, Z. Peng, and Z. Jun, "A new architecture of ajax web application security crawler with finite-state machine," in *2014 International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery*, Oct. 2014, pp. 112–117, doi: 10.1109/CyberC.2014.27.

[22] H. K. Dhalla, "A performance analysis of native JSON parsers in java, python, MS.NET core, JavaScript, and PHP," in *2020 16th International Conference on Network and Service Management (CNSM)*, 2020, pp. 1–5, doi: 10.23919/CNSM50824.2020.9269101.

[23] H. Shamkhalichenar, C. J. Bueche, and J.-W. Choi, "Printed circuit board (PCB) technology for electrochemical sensors and sensing platforms," *Biosensors*, vol. 10, no. 11, pp. 1–16, Oct. 2020, doi: 10.3390/bios10110159.

[24] A. Bouraiou *et al.*, "Display on web browser using ethernet shield and LM35 sensor based on Arduino board," in *2022 19th International Multi-Conference on Systems, Signals & Devices (SSD)*, 2022, pp. 421–424, doi: 10.1109/SSD54932.2022.9955956.

[25] K. Chaarmart, A. Vora-ud, and T. Seetawan, "Temperature controlling and monitoring system of Mao-wine aging tank for local enterprises," in *2021 Research, Invention, and Innovation Congress: Innovation Electricals and Electronics (RI2C)*, Sep. 2021, pp. 243–246, doi: 10.1109/RI2C51727.2021.9559768.

[26] Y. A. Ahmad, T. Surya Gunawan, H. Mansor, B. A. Hamida, A. Fikri Hishamudin, and F. Arifin, "On the evaluation of DHT22 temperature sensor for IoT application," in *2021 8th International Conference on Computer and Communication Engineering (ICCCE)*, Jun. 2021, pp. 131–134, doi: 10.1109/ICCCE50029.2021.9467147.

[27] C. C. Paglinawan, G. A. R. Bordonada, and A. V. D. Roseus, "Infrared temperature detection using MLX90614 sensor for wearable applications," in *2023 15th International Conference on Computer and Automation Engineering (ICCAE)*, Mar. 2023, pp. 385–388, doi: 10.1109/ICCAE56788.2023.10111139.

[28] A. C. Bento, "An experimental survey with NodeMCU12e+shield with Tft Nextion and MAX30102 sensor," in *2020 11th IEEE Annual Information Technology, Electronics and Mobile Communication Conference (IEMCON)*, Nov. 2020, pp. 82–86, doi: 10.1109/IEMCON51383.2020.9284870.

[29] C.-H. Lee, W.-H. Lee, and S.-M. Kim, "Development of IoT-based real-time fire detection system using raspberry pi and fisheye camera," *Applied Sciences*, vol. 13, no. 15, pp. 1–14, Jul. 2023, doi: 10.3390/app13158568.

[30] S. Avallone, P. Imputato, G. Redieteab, C. Ghosh, and S. Roy, "Will OFDMA improve the performance of 802.11 Wi-Fi networks," *IEEE Wireless Communications*, vol. 28, no. 3, pp. 100–107, Jun. 2021, doi: 10.1109/MWC.001.2000332.

[31] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," *Applied Sciences*, vol. 9, no. 5, pp. 1–10, Feb. 2019, doi: 10.3390/app9050848.
[32] H. Siddharthan, T. Deepa, and P. Chandhar, "SENMQTT-SET: an intelligent intrusion detection in IoT-MQTT networks using ensemble multi cascade features," *IEEE Access*, vol. 10, pp. 33095–33110, 2022, doi: 10.1109/ACCESS.2022.3161566.
[33] R. A. Light, "Mosquitto: server and client implementation of the MQTT protocol," *The Journal of Open Source Software*, vol. 2, no. 13, p. 265, 2017, doi: 10.21105/joss.00265.
[34] T. N. Ford, E. Gamess, and C. Ogden, "Performance evaluation of different raspberry Pi models as MQTT servers and clients," *International journal of Computer Networks & Communications*, vol. 14, no. 2, pp. 1–18, Mar. 2022, doi: 10.5121/ijcnc.2022.14201.
[35] K. Manandhar, X. Cao, F. Hu, and Y. Liu, "Detection of faults and attacks including false data injection attack in smart grid using kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 1, no. 4, pp. 370–379, Dec. 2014, doi: 10.1109/TCNS.2014.2357531.
[36] Q. Tao, F. Liu, Y. Li, and D. Sidorov, "Air pollution forecasting using a deep learning model based on 1D convnets and bidirectional GRU," *IEEE Access*, vol. 7, pp. 76690–76698, 2019, doi: 10.1109/ACCESS.2019.2921578.
[37] I. N. K. Wardana, J. W. Gardner, and S. A. Fahmy, "Estimation of missing air pollutant data using a spatiotemporal convolutional autoencoder," *Neural Computing and Applications*, vol. 34, no. 18, pp. 16129–16154, Sep. 2022, doi: 10.1007/s00521-022-07224-2.
[38] Y.-C. Liang, I. Wijaya, M.-T. Yang, J. R. C. Juarez, and H.-T. Chang, "Deep learning for infant cry recognition," *International Journal of Environmental Research and Public Health*, vol. 19, no. 10, pp. 1–10, May 2022, doi: 10.3390/ijerph19106311.
[39] T. Li, M. Hua, and X. Wu, "A hybrid CNN-LSTM model for forecasting particulate matter (PM2.5)," *IEEE Access*, vol. 8, pp. 26933–26940, 2020, doi: 10.1109/ACCESS.2020.2971348.
[40] Q. Wang, R.-Q. Peng, J.-Q. Wang, Z. Li, and H.-B. Qu, "NEWLSTM: an optimized long short-term memory language model for sequence prediction," *IEEE Access*, vol. 8, pp. 65395–65401, 2020, doi: 10.1109/ACCESS.2020.2985418.
[41] V. Gude and S. Corns, "Integrated deep learning and supervised machine learning model for predictive fetal monitoring," *Diagnostics*, vol. 12, no. 11, pp. 1–16, Nov. 2022, doi: 10.3390/diagnostics12112843.
[42] D. Chicco, M. J. Warrens, and G. Jurman, "The coefficient of determination R-squared is more informative than SMAPE, MAE, MAPE, MSE and RMSE in regression analysis evaluation," *PeerJ Computer Science*, vol. 7, pp. 1–24, Jul. 2021, doi: 10.7717/peerj-cs.623.
[43] T. O. Hodson, "Root-mean-square error (RMSE) or mean absolute error (MAE): when to use them or not," *Geoscientific Model Development*, vol. 15, no. 14, pp. 5481–5487, Jul. 2022, doi: 10.5194/gmd-15-5481-2022.

BIOGRAPHIES OF AUTHORS

I Komang Agus Ady Aryanto Completed a computer systems undergraduate program at the ITB STIKOM Bali in 2015. After that, continued his master's degree in computer systems at the Ganesha University of Education and finished his education in 2018. He lecturer on the ITB STIKOM Bali since 2018 until now and teaching web programming, internet of things, sensors and transducers, microcontroller programming, data structures, and databases. The research topics occupied are the internet of things, information systems, computer science, robotics, and programming. In addition, he also works as a software developer to develop web-based and mobile applications that have been implemented in the fields of government, transportation, industry, academics, hospitals, and others. He can be contacted at email: i_komang@mail.rmutt.ac.th and komang.aryanto@gmail.com.

Dechrit Maneetham is a lecturer in the Faculty of Engineering Education at Rajamangala University, Thailand for more than 15 years. Fields taught by lecturers are mechatronics, robotics, and internet of things. Education in Computer Engineering with M.S and B.Tech degrees at the Raja Mongkut University of Technology in Thailand. D.Eng doctoral education in the field of mechatronics at the Asian Institute of Technology. Then the second Ph.D. doctoral education at Mahasarakham University in the field of computer engineering. Author of books on PLC, microcontroller, pneumatics, and robotics topics. Research interest in the fields of mechatronics, automation, internet of things, and robotics and has published more than 30 international papers. He can be contacted at email: dechrit_m@rmutt.ac.th.

Evi Triandini as a lecturer at the ITB STIKOM Bali in the information systems study program. Teach courses related to software engineering, object-oriented modeling, and information systems design. Undergraduate education was taken by the University of Brawijaya in Indonesia in the field of Agricultural Cultivation. Master's degree at Asia Institute of Technology Thailand with M.Eng. Doctoral education at the Institute of Technology Sepuluh November Surabaya Indonesia in the department of computer science with the title Dr. Published research has topics that include information systems, software, and electronic commerce. She can be contacted at email: evi@stikom-bali.ac.id.