


Article

Enhancing Neonatal Incubator Energy Management and Monitoring through IoT-Enabled CNN-LSTM Combination Predictive Model

I Komang Agus Ady Aryanto , Dechrit Maneetham * and Padma Nyoman Crisnapati 

Department of Mechatronics Engineering, Rajamangala University of Technology Thanyaburi, Bangkok 12110, Thailand; i_komang@mail.rmutt.ac.th

* Correspondence: dechrit_m@rmutt.ac.th (D.M.); padma_c@mail.rmutt.ac.th (P.N.C.)

Abstract: This research focuses on enhancing neonatal care by developing a comprehensive monitoring and control system and an efficient model for predicting electrical energy consumption in incubators, aiming to mitigate potential adverse effects caused by excessive energy usage. Employing a combination of 1-dimensional convolutional neural network (1D-CNN) and long short-term memory (LSTM) methods within the framework of the Internet of Things (IoT), the study encompasses multiple components, including hardware, network, database, data analysis, and software. The research outcomes encompass a real-time web application for monitoring and control, temperature distribution visualizations within the incubator, a prototype incubator, and a predictive energy consumption model. Testing the LSTM method resulted in an RMSE of 42.650 and an MAE of 33.575, while the CNN method exhibited an RMSE of 37.675 and an MAE of 30.082. Combining CNN and LSTM yielded an RMSE of 32.436 and an MAE of 25.382, demonstrating the potential for significantly improving neonatal care.

Keywords: neonatal incubator; energy consumption prediction; CNN-LSTM; Internet of Things; artificial intelligence



Citation: Aryanto, I.K.A.A.; Maneetham, D.; Crisnapati, P.N. Enhancing Neonatal Incubator Energy Management and Monitoring through IoT-Enabled CNN-LSTM Combination Predictive Model. *Appl. Sci.* **2023**, *13*, 12953. <https://doi.org/10.3390/app132312953>

Academic Editor: Muhammad Awais Javed

Received: 26 October 2023
Revised: 28 November 2023
Accepted: 30 November 2023
Published: 4 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Newborn babies are referred to as neonates, and babies born before reaching 37 weeks of gestation are considered premature. Premature babies face higher health risks than those born at full term, requiring intensive care. Intensive care for these infants necessitates using a medical device known as a neonatal incubator. The incubator maintains a stable environmental temperature and humidity using an internal heater, assisted by a fan for air circulation. Operating the incubator requires significant knowledge, and medical personnel must be able to monitor the conditions inside the incubator at all times.

Additionally, efficient energy management within the incubator is crucial, as excessive energy usage could lead to the heater inside increasing the temperature conditions, potentially negatively impacting infant care and raising energy consumption. Therefore, research was conducted to develop an efficient energy-management system and enable real-time monitoring and control of the incubator's conditions to enhance neonatal care. In this study, a combination of LSTM-CNN methods and the Internet of Things (IoT) concept was utilized to create an energy prediction model and a real-time operational system for monitoring and control.

The Internet of Things (IoT) is a technological concept that enables hardware or objects, regardless of location, to communicate over the Internet without human intervention [1]. The IoT has been widely adopted in various aspects of life to enhance daily living, even within the industrial sector [2]. The IoT comprises hardware, networking, and software components [3,4]. In this research, IoT technology was applied in healthcare, specifically in neonatal incubators. Multiple sensors, actuators, and network modules are installed in the

incubator. All hardware sensors and actuator modules are connected to a microcontroller, which processes the data and sends them to an IoT broker using the message queuing telemetry transport (MQTT) communication protocol [5].

Several previous studies have explored using the Internet of Things (IoT) in baby incubators, incorporating various sensors. Furthermore, previous research has employed a hybrid fuzzy and PID approach in incubator control systems. Researchers have also researched incubator monitoring and apnea detection systems for premature babies using IoT and wearable devices [6–9]. However, no research has effectively combined artificial intelligence (AI) with IoT within the context of baby incubators. Our study addresses this gap by developing a system capable of understanding energy consumption, monitoring temperature conditions through a web application, and comprehending temperature distribution within the incubator.

Generally, healthcare professionals use neonatal incubators to maintain temperature and humidity conditions. Therefore, this research uses DHT11 sensors to monitor temperature and humidity and ACS712 devices to measure energy consumption [10,11]. Additionally, this research conducted experiments to understand the temperature distribution inside the incubator. Multiple DHT11 sensors were installed at various corners within the incubator to collect temperature data from each point in more detail and visualize. Through visualization, temperature distribution patterns were observed, which can provide further insights into improving temperature management efficiency and energy usage. This, in turn, contributes to the optimization of neonatal care.

The main hardware components utilized included a microcontroller, a single-board computer, a 1200-watt AC heater, and a 12 V DC motor with a propeller for the fan. The microcontroller was the ATmega256, integrated into the Arduino Mega 2560 board [12]. Arduino Mega 2560 has digital input/output (I/O) and analog I/O to connect with the DHT11 and ACS721 sensors. It also features three serial ports (Tx/Rx), with one of these serial ports used in the research to connect the ESP8266 WiFi module [13,14]. Furthermore, this study used the Raspberry Pi, a single-board computer capable of running Linux or Windows-based operating systems. The Raspberry Pi has a CPU, RAM, and I/O ports suitable for IoT-based projects [15]. Hence, in this research, Raspberry Pi was the server running the Mosquitto IoT Broker application, an Apache Server, and the MySQL database [16–18].

Subsequently, sensor measurements from the hardware components are sent to the IoT broker using a WiFi module with an MQTT communication protocol. MQTT is a protocol that offers three different quality of service (QoS) levels and employs publish/subscribe (pub/sub) terminology for communication [19,20]. The transmitted data enter the IoT broker and are further processed by the system, being stored in the database. The data stored in the database are then displayed in real-time on the web application. Furthermore, this web application is responsive and can be accessed on various screen sizes, including mobile devices. This feature allows for flexible control and monitoring of the incubator from anywhere and anytime over the Internet using the MQTT protocol.

The data stored in the database are then used as a dataset for creating a predictive model for electricity consumption. The process of creating this model utilizes a combination of methods from deep learning, namely long short-term memory (LSTM) and a 1-dimensional convolutional neural network (1D-CNN) [21]. Combining these two methods aims to increase the accuracy of predicting electrical energy consumption [22]. The choice of these two methods is based on the temporal or sequential data type. Therefore, the decision was made to employ the LSTM method to understand the data sequence that will occur next. The output from the 1D-CNN will be used as input for LSTM. Generally, the analysis process begins by inputting data into the 1D-CNN. The output results from the 1D-CNN are then used as input for LSTM. Following this, the final result of this model will be evaluated to measure the level of prediction it produces.

In the context of the incubator, numerous studies have been conducted previously, but most of them utilized control methods such as PID and Fuzzy [6–9], as shown in Table 1. Our research aims to extend the development of the incubator with artificial intelligence to monitor energy usage based on temperature conditions inside the incubator. In this work, our main contributions are:

- web application for real-time monitoring and control;
- visualization of the temperature data distribution inside the incubator;
- prototype of an electronic hardware incubator; and
- hybrid model utilizing 1D-CNN and LSTM methods for predicting energy consumption models.

Table 1. Summary of the works related to neonatal incubator systems.

Reference	Hardware		Network		Data	Software	Method
[6]	Microcontroller	ATMega 328	Protocol	HTTP	MySQL	Mobile	-
	Microcomputer	-	Network	GSM			
	Sensor	Phototherapy, Temperature, humidity, Fingerprint, Heart Rate, Camera	Broker IoT	-			
[7]	Microcontroller	ATMega16	Protocol	-	-	LCD, Desktop	Fuzzy-PID
	Microcomputer	-	Network	-			
	Sensor	Temperature, Humidity	Broker IoT	-			
[8]	Microcontroller	ESP32	Protocol	MQTT	Filter, Peak Detection, Feature Extraction	LCD, Desktop	-
	Microcomputer	Raspberry Pi	Network	WiFi			
	Sensor	Respiration	Broker IoT	Mosquitto			
[9]	Microcontroller	ESP32	Protocol	MQTT	MySQL	Web	-
	Microcomputer	-	Network	WiFi			
	Sensor	Temperature, humidity, Sound	Broker IoT	Node-Red			
Proposed work	Microcontroller	ATMega 2560	Protocol	MQTT, HTTP	Filter, MySQL	Web, Mobile	CNN, LSTM, RMSE, MAE, MAPE, MSE
	Microcomputer	Raspberry Pi	Network	WiFi			
	Sensor	Temperature, humidity, Energy	Broker IoT	Mosquitto			

2. Related Works

Numerous prior investigations have been conducted about neonatal incubators. Kapen et al. in 2019 [6] aimed to create an automatic neonatal incubator with various modules such as phototherapy, fingerprint reader, and remote monitoring. The remote monitoring module supervises the baby's temperature, heart rate, and video surveillance. Alimuddin et al., in 2021 [7], applied a hybrid Fuzzy-PID system to control temperature and humidity in the incubator. Next, Cay et al., 2022 [8] developed a respiratory monitoring and Apnea detection system for premature babies using the Internet of Things concept. This system utilizes the ESP32 microcontroller, sensors, edge computing devices (ECD), and the MQTT protocol. This system implements an Internet of Things architecture with the MQTT

protocol to connect the wireless embedded system (WES) or sensors to the edge computing device (ECD). Aya-Parra et al., 2023 [9] worked on developing an incubator system for temperature and humidity monitoring through a web application. This system uses a WiFi network and the MQTT communication protocol to send data to a broker service integrated with Node-RED. In Table 1, there is a summary of research related to neonatal incubator systems. This table summarizes research findings based on various aspects, including hardware, network, data management, software, analysis methods, and control systems. The last row of this table highlights the contributions of these research findings.

In Table 1, the hardware section features the adoption of the ATmega2560 microcontroller, chosen for its larger processing capacity. On the network side, the implementation of MQTT and HTTP protocols stands out compared to previous research that exclusively used MQTT. MQTT connects sensor devices within the neonatal incubator to the IoT broker, while HTTP establishes connections between web and mobile applications and the database. Notably, in the IoT broker section, Mosquitto is utilized, being directly installed on the Raspberry Pi, a departure from previous research that relied on third-party broker providers. Moving to the software section designed for monitoring the incubator’s condition through web and mobile devices, it follows a web-based approach, contrasting with prior research that predominantly focused on desktop applications and a 16 × 2 LCD module. Finally, the data analysis and monitoring of the neonatal incubator’s condition involve utilizing LSTM and CNN methods, a unique combination not previously explored in research.

3. Materials and Methods

3.1. System Overview

An overview of the system in this research is depicted in Figure 1. The entire developed system is divided into five components called modules. Each module encompasses several processes. These components include hardware, network, database, data analysis, and outputs. The initial stages of the research start from the hardware component and continue through the process to obtain the final results.

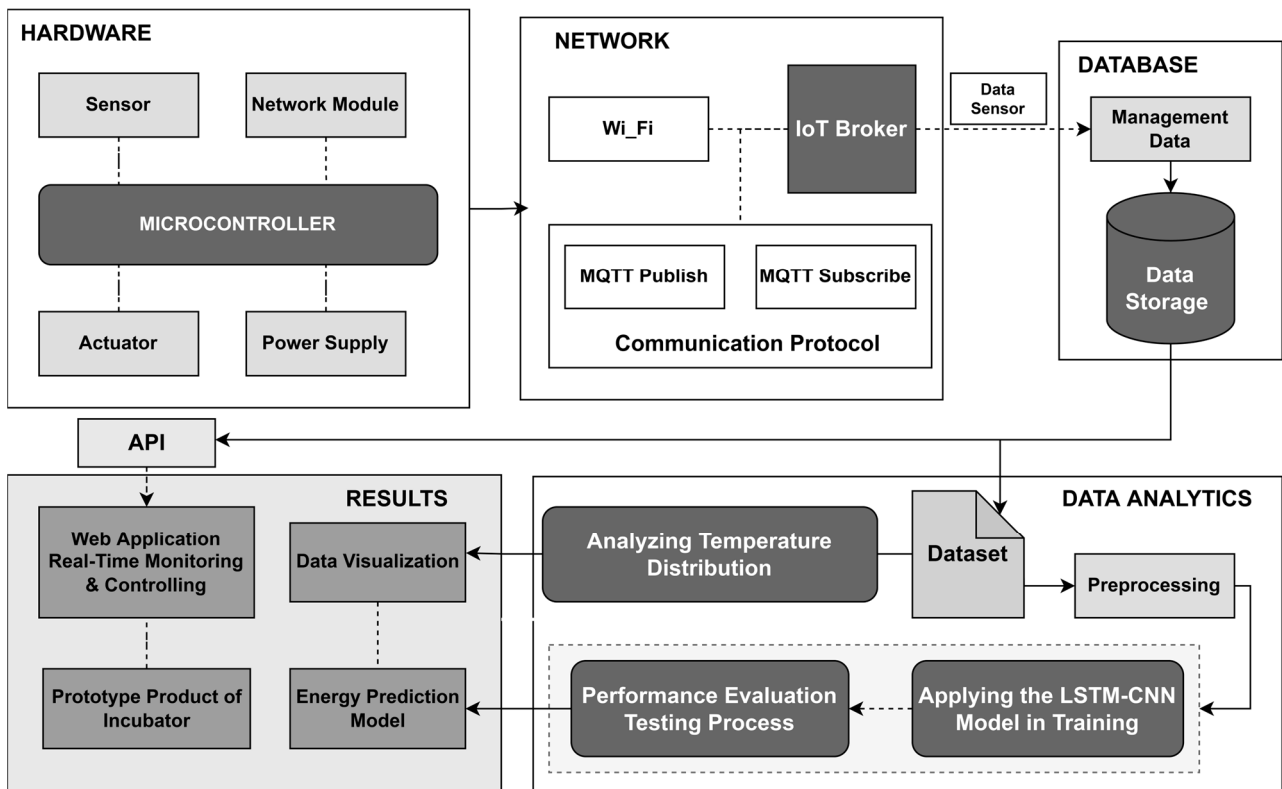


Figure 1. System overview.

In the hardware component, various electronic components are used to collect data. These electronic components include sensors, actuators, power supply, communication or network devices, and microcontrollers. These electronic components are connected to the microcontroller according to their respective I/O signals. The sensors carry out the process of reading environmental data values inside the incubator, and the microcontroller processes the acquired values. The microcontroller can use these data to send commands to the actuators or transmit the data to the database via the network.

An IoT Broker is used and installed on a server in the network component. The data sent by the hardware will enter this IoT broker. Data communication in this system uses the publish/subscribe method, where each piece of data sent has a corresponding topic [23]. Additionally, each device has a unique ID or identity for identification purposes. WiFi is the communication medium during the data transmission from hardware to the IoT Broker.

The data collected in the IoT broker are then processed for storage in the database. The data storage process is conducted in real-time from the broker to the database using a socket service created with Python. However, before the data are stored in the database, they are filtered to ensure no data loss or inappropriate data occurs during storage. Subsequently, the data stored in the database are used as a dataset in the analysis stage to create a model to predict energy consumption and temperature distribution patterns within the incubator.

The dataset contains features or attributes such as temperature, humidity, and electric current values in the analysis stage. Data patterns in this dataset are recorded every second. Therefore, energy consumption is calculated for each second in the analysis process. Suppose you want to calculate the total energy consumption for a specific period. In that case, you can multiply the electric current data by the electrical voltage (220 V) with the desired duration in seconds. For example, if you want to find energy consumption per minute, the result can be multiplied by 60 s.

The next step in the process is modeling, using deep learning methods to analyze the sequential data from the incubator. This analysis process combines two techniques, 1D-CNN and LSTM, to develop a predictive model for electric energy consumption [24]. Before that, a data preprocessing process is performed, which includes steps such as calculating the average values based on time and scaling. The purpose is to optimize and prepare the data to create an accurate and reliable predictive model.

Furthermore, in the next phase, an analysis of the temperature distribution patterns within the incubator is conducted, aiding in the optimization of infant care. The outcome of this analysis will manifest as visualizations of the temperature distribution conditions within the incubator. Additionally, a web application has been developed to be accessed from anywhere via the Internet. This web application allows users to monitor and control the incubator's conditions in real-time.

3.2. Electronic Design

The hardware design in this research is illustrated in Figure 2. Figure 2a represents the design process, while Figure 2b displays the wiring schematic. The hardware design utilizes several DHT11 sensors [25] and a one-channel relay device as actuators. All of these components are connected to the microcontroller through digital I/O pins. An ACS712 sensor is also employed to measure the electrical current linked to the microcontroller via an analog I/O pin (A0) [26]. To control the speed of the DC motor, a BTS7960 device is used, capable of controlling motors up to 48 V. The DC motor circulates the heated air produced by the heater into the incubator. This heater device is connected to a relay module, facilitating easy on/off control. A comprehensive list of the electronic components used can be found in Table 2.

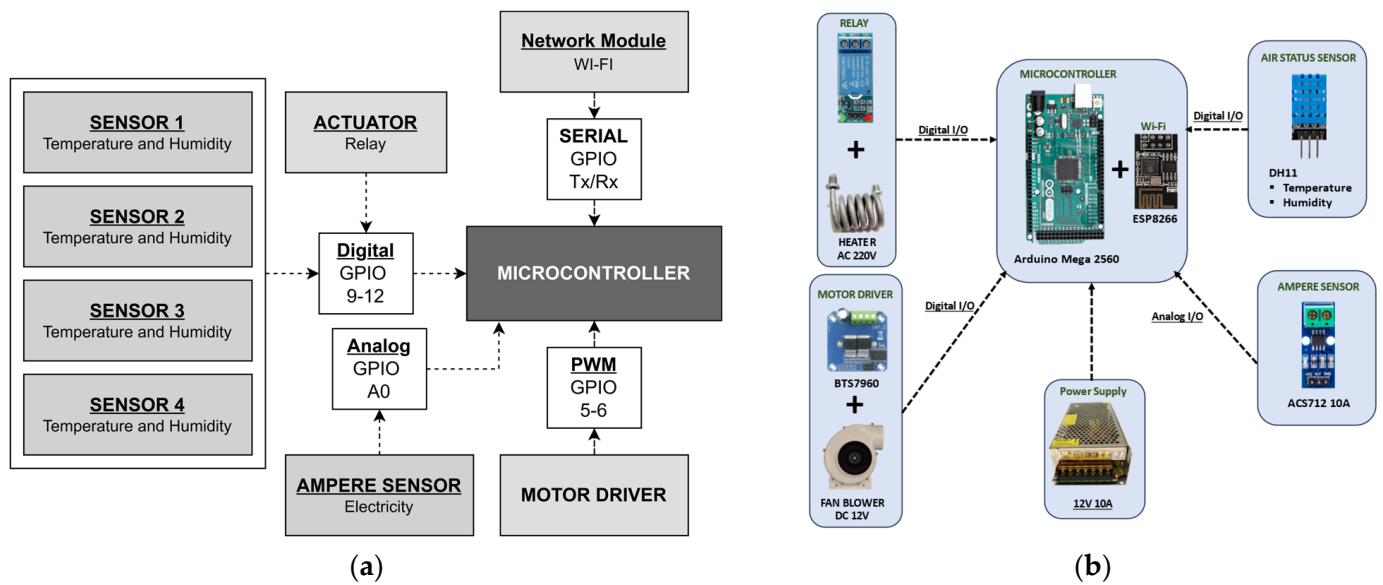


Figure 2. Architecture of hardware: (a) Electronic design; (b) Electronic schematic.

Table 2. Electronic component.

Component	Description	Qty.	Unit Price (\$)	Total (\$)
Board Arduino Mega2560	Microcontroller	1	8.65	8.65
ESP8266	WiFi	1	0.86	0.86
DHT11	Temperature measurement range: 0 °C to 50 °C, and Humidity measurement range: 20% to 90% [27]	4	0.60	2.4
ACS712 10 A	Ampere sensor	1	0.73	0.73
Relay 1 channel	Relay	1	0.30	0.3
Motor DC 12 V	Motor	1	15.38	15.38
BTS7960	Driver motor	1	3.89	3.89
Heater Incubator 1200 Watt	Heater	1	115.34	115.34
Power supply 12 V	Power supply	1	8.78	8.78
Total				156.33

Furthermore, the placement procedure of these sensors is designed to optimize monitoring and measurements inside the incubator, as depicted in Figure 3. Four temperature and humidity sensors are positioned on each side of the incubator. The distribution of these sensors across the incubator’s sides aims to assist in monitoring even temperature and humidity changes throughout the entire space. Meanwhile, the current sensor placed on the AC power cable of the primary power source enables precise monitoring of the electrical current entering the incubator.

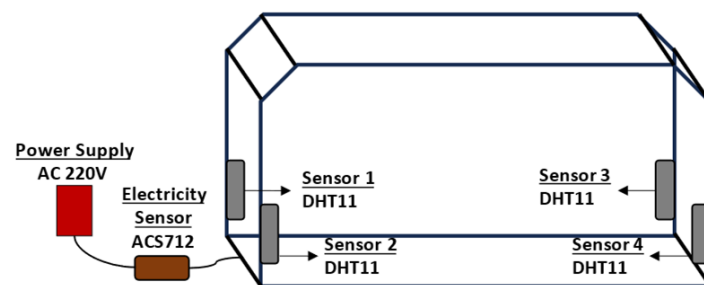


Figure 3. Sensor placement in the incubator.

The next step after connecting the sensor and actuator devices to the microcontroller involves developing the program code. The program development process employs the Arduino IDE as the code editor [28]. The program is divided into several functions, which include (a) connecting to the internet network, (b) communicating with the IoT broker, (c) reading sensor values, and (d) the control system. The program begins with the device connecting to the Internet using the SSID and password. Subsequently, the device establishes a connection with the IoT broker by configuring several parameters such as the host, port, username, and password. Once connected, the device starts reading sensor data and repeatedly sends them to the IoT broker. The following complete details regarding the program on the hardware can be seen in Algorithm 1.

Algorithm 1: Hardware system

```

1.  INPUT: Hardware sensor
2.  OUTPUT: Temperature sensor value, humidity, and electric current
3.  DESCRIPTION:
4.  IF network not conected:
5.      | network = reconnect and set parameter (ssid, password)
6.  END
7.  IF network connected:
8.      | client = connect to an IoT broker with parameters (host, port, id, username,
9.      | password)
9.  END
10. WHILE client connected:
11.     | temperature, humidity = read sensor temperature and humidity
12.     | IF(AVERAGE(temperature < 33):
13.         | setHeater = ON
14.     | END
15.     | FOR i in range 150:
16.         | current += read sensor current
17.         | set delay = 3 milliseconds
18.     | END
19.     | current = current/150
20.     | current = (2.5 - (current × (4.9/1024.0)))/0.066;
21.     | response = send data (temperature, humidity, current)
22.     | IF response:
23.         | PRINT response
24.     | END
25. END

```

3.3. Network Design

The network design is explained in Figure 4. The communication process between objects or incubators uses the MQTT protocol and a WiFi network to transmit sensor data. Communication using the MQTT protocol involves the terms subscribe and publish. An IoT broker functions to receive messages from publishers and then distribute them to subscribers who have subscribed to relevant topics [29].

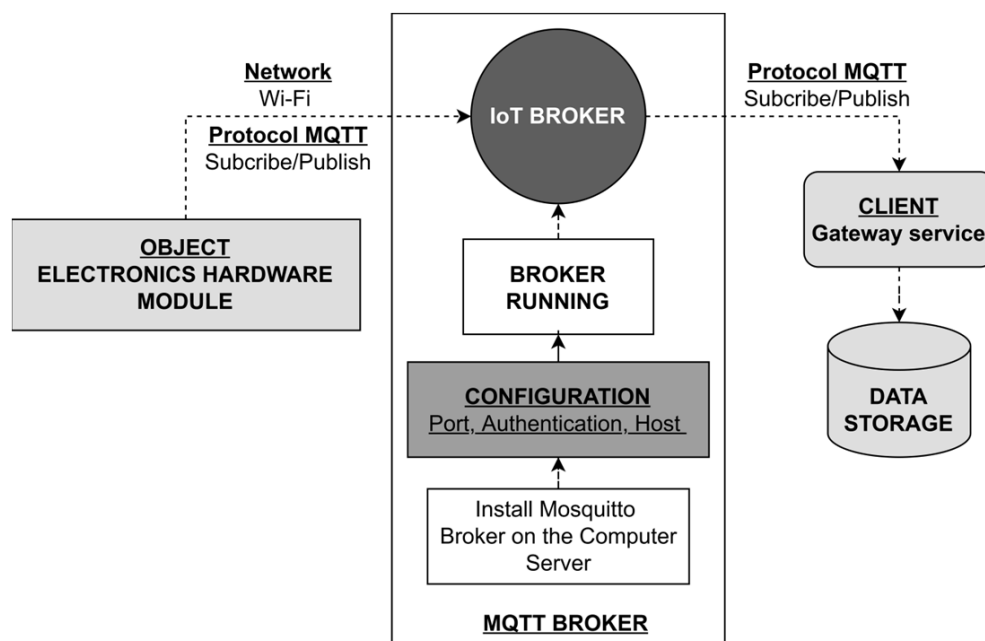


Figure 4. Architecture of network.

Subsequently, the data on the broker are subscribed to by clients and then displayed in the web application or stored in the database. In this research, the IoT broker is installed on a Raspberry Pi mini-computer, which also functions as an Apache server for the web application [17]. Mosquitto is used as the MQTT broker service for the IoT broker [30]. Several configurations on the IoT broker, such as port settings, authentication, and host settings, must be configured in advance.

3.4. Data Management

3.4.1. Database

In this study, the sensor measurement data from the incubator are sent to the broker and then processed by the system for storage in the database. The database application used in this research is MySQL, which is one of the relational database management systems (DBMS) that makes it easy to manage structured data as tables which can have relationships with each other [18,31]. In storing sensor data, the data must go through a filtering process, which is carried out by setting upper and lower threshold values to improve the quality of the data stored in the database.

This process is carried out with the aim of filtering out non-outlier data. The process is controlled by the filter low and filter high parameters in Algorithm 2, at Lines 10 and 11. The values of these parameters have been determined based on the characteristics of the DHT11 sensor device, which has a temperature measurement range of 0 °C to 50 °C and humidity range of 20% to 90%, as presented in Table 2. The methodology used to determine these threshold values is based on the range of sensor characteristic values [27] and standard temperature values for the incubator range from 32 °C to 35 °C [32]. If the measurement point of sensor data is below the filter low or filter high threshold, it will be identified as an outlier, as in Lines 14–19 of Algorithm 2.

After going through the filtering process, the data are separated based on the data topic and then stored in variables for storage. A description of the variable structure and sensor data types can be seen in Table 3, and the data-management program is shown in Algorithm 2. After that, the data stored in the database can be accessed for analysis and monitoring purposes. The data in the database are used as a dataset in the analysis process.

Algorithm 2: Data management

```

1.  INPUT: Sensor data
2.  OUTPUT: Data stored in the database
3.  DESCRIPTION:
4.  db = connect to a database and set parameters (host, username, password,
    db_name)
5.  IF db error:
6.      | PRINT database error
7.  END
8.  IF db connected:
9.      | object = subscribe topic
10.     | filter low = set value low
11.     | filter high = set value high
12.     WHILE object:
13.         | payload = read value(object)
14.         IF payload < filter low:
15.             | outliers = payload
16.         END
17.         ELSEIF payload > filter high:
18.             | outliers = payload
19.         END
20.         ELSE
21.             | query = insert data to database and set query(payload, tabel
22.             | name)
23.             IF query check:
24.                 | PRINT response
25.             END
26.         END
27.     END

```

Table 3. Database structure for electronic sensors.

Variable	Electronic	Data Type	Unit	Variable	Electronic	Data Type	Unit
temperature_1	DHT11	Float	°C	humidity_1	DHT11	Float	%
temperature_2	DHT11	Float	°C	humidity_2	DHT11	Float	%
temperature_3	DHT11	Float	°C	humidity_3	DHT11	Float	%
temperature_4	DHT11	Float	°C	humidity_4	DHT11	Float	%
electricity	ACS712	Float	A	date	DS1307	DateTime	yy:mm:dd hh:mm:ss

3.4.2. Dataset Description

This research uses the sensor data stored in the database as the dataset. Based on the variable data in the database, this dataset consists of 10 attributes corresponding to the sensor data. The attributes of this dataset are T_1, T_2, T_3, T_4, Electricity, RH_1,

RH_2, RH_3, RH_4, and date. The attributes T_1, T_2, T_3, T_4 are obtained from the temperature variables (temperature_1, temperature_2, temperature_3, temperature_4), and RH_1, RH_2, RH_3, RH_4 are obtained from the humidity variables (humidity_1, humidity_2, humidity_3, humidity_4). Additionally, the electricity attribute is obtained from the electricity variable. Detailed information about the dataset attributes is shown in Table 4.

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

Table 4. Dataset attribute.

Variable	Attribute	Unit	Descriptions	Variable	Attribute	Unit	Descriptions
temperature_1	T_1	°C	Temperature on the right rear side	humidity_1	RH_1	%	Humidity on the right rear side
temperature_2	T_2	°C	Temperature on the right front side	humidity_2	RH_2	%	Humidity on the right front side
temperature_3	T_3	°C	Temperature on the left front side	humidity_3	RH_3	%	Humidity on the left front side
temperature_4	T_4	°C	Temperature on the left rear side	humidity_4	RH_4	%	Humidity on the left rear side
electricity	Electricity	A	Electrical energy usage	date	Date	yy:mm:dd hh:mm:ss	Date and time

Next, after obtaining the dataset, an analysis is conducted to determine the correlation coefficient (r) between attributes using Equation (1) [33]. Data correlation analysis is essential to understand the relationship between the target attribute, in this case, the electricity attribute, and other attributes [34]. Knowing the correlation coefficient values between the target and other attributes makes model development or training easier. Attributes positively correlate if the correlation value approaches 1 or $0 < r < 1$. Conversely, attributes have a negative correlation if the correlation value falls within the range $-1 < r < 0$, and if the correlation value approaches 0, it indicates no correlation between the attributes.

3.5. Software Design

The software design is illustrated in Figure 5. The software architecture design process involves several components: the database, gateway service, an application programming interface (API), and web application. MySQL version 8.0.30 is utilized as the database to store sensor data, falling under the category of relational databases [35]. Clients requesting data from the database must go through the gateway service system. Therefore, the gateway service is responsible for interacting with clients and managing all client data requests via the API [2]. The generated data will be in JavaScript Object Notation (JSON) format to facilitate communication with the web. The JSON format is favored for its lightweight and structured nature [36].

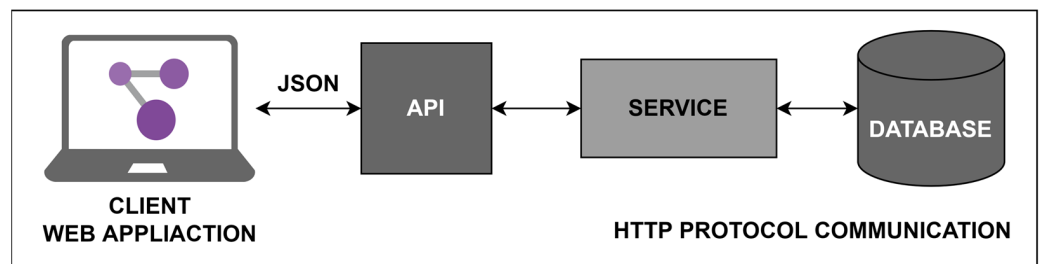


Figure 5. Architecture of software.

Furthermore, the web application design is implemented with a responsive interface, meaning it can be accessed on various screen types. The web interface also includes charts

and tables to visually represent data from the database. The development of this web application involves several programming languages, including CSS version 3, JavaScript version ES13, PHP version 7.2.27, and Python version 3.10.6 [37]. The software programming process is illustrated in Algorithm 3.

Algorithm 3: Web application

```

1.  INPUT: Sensor data in the database
2.  OUTPUT: Web application
3.  DESCRIPTION:
4.  FUNCTION database (host, username, password, db_name)
5.      db = connect_db(host, username, password, db_name)
6.      IF !db:
7.          db = db.error()
8.      END
9.      RETURN db
10. END
11. FUNCTION gateway(request, db_con)
12.     getrequest = API(request)
13.     result = db.select(getrequest)
14.     WHILE row IN RANGE result.length():
15.         json = {T_1: row[temperature_1], T_2: row[temperature_2],
16.               T_3: row[temperature_3], T_4: row[temperature_4];
17.               H_1: row[humidity_1], H_2: row[humidity_2],
18.               H_3: row[humidity_3], H_4: row[humidity_4],
19.               Electricity: row[electricity], date: row[date]}
20.     END
21.     RETURN json
22. END
23. MAIN:
24.     db_con = database(parameter[host, user, pass, db_name])
25.     data json = gateway(data_request, db_con)
26.     data = convert_json(data json)
27.     plot graph-table(data[T_1], data[T_2], data[T_3], data[T_4], data[H_1],
28.                     data[H_2], data[H_3], data[H_4], data[Electricity], data[date])
29. END

```

3.6. Model Design

3.6.1. Preprocessing

Figure 6 depicts the dataset preprocessing process performed in this study. Dataset preprocessing comprises a series of steps on raw data to ensure their quality and suitability before they can be used for analysis or modeling. In this research, the dataset includes temperature, humidity, and electric current data, with attributes as shown in Table 4. Subsequently, this dataset is processed by calculating the average values of temperature and humidity attributes over time (T). Meanwhile, the electric current attribute (I) is calculated to obtain its energy value (W) by multiplying voltage (V) with electric current

and time, as indicated in Equation (2) [38]. As a result, a new attribute called energy is added to the dataset, rendering the electric current attribute obsolete. Unlike temperature and humidity attributes with averaged values, the energy attribute is summed according to periods (T).

$$W = V \times I \times T \tag{2}$$

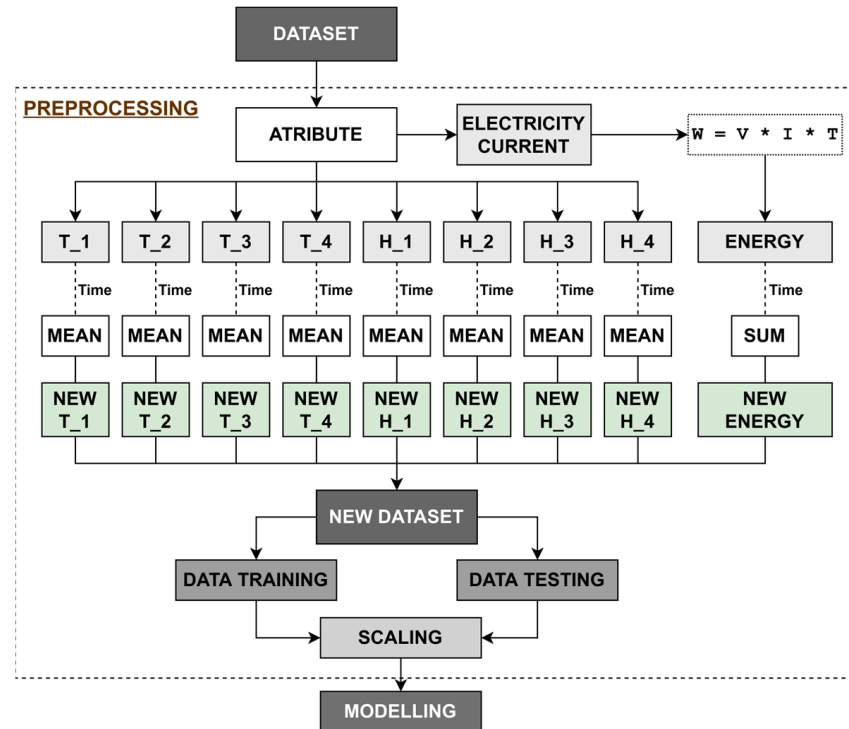


Figure 6. Preprocessing flow diagram.

After obtaining the new dataset, the next step is to scale the training and testing data. The fitting process is carried out in the training data by applying the MinMaxScaler method [39]. This method helps to transform the data range to distribute it on a scale between 0 and 1 [40]. The use of MinMaxScaler on the training data is intended to balance the influence of each variable on the model. Additionally, the testing data are scaled to adapt the testing data to the same transformation as the training data, and this is performed to maintain a balance between the training and testing data.

3.6.2. Long Short-Term Memory (LSTM)

LSTM [41] is used in this research to address short-term and long-term dependency issues in sequential data. As a recurrent neural network (RNN) type, the LSTM workflow uses gates and memory cells to retain information over extended periods. There are three types of gates in LSTM: the input gate, the forget gate, and the output gate. In the context of this research, the LSTM method is employed to create a model based on the dataset obtained from sensor data.

The modeling process to recognize data patterns begins by determining the appropriate number of layers, neurons, and activation functions. The model’s performance is evaluated using test data to measure its effectiveness. The mathematical calculations depict the operations within LSTM that enable it to recognize patterns in sequential data and maintain and manipulate information over extended periods through Equations (3)–(10). The architectural design of the LSTM is illustrated in Figure 7 [33].

$$F_t = \sigma\left(\left([H_{t-1}, X_t] \cdot W_f\right) + b_f\right) \tag{3}$$

$$I_t = \sigma((H_{t-1}, X_t) \cdot W_i) + b_i) \tag{4}$$

$$\tilde{C}_t = \tanh((H_{t-1}, X_t) \cdot W_c) + b_c) \tag{5}$$

$$C_t = (F_t \cdot C_{t-1}) + (I_t \cdot \tilde{C}_t) \tag{6}$$

$$O_t = \sigma((W_o \cdot [H_{t-1}, X_t]) + b_o) \tag{7}$$

$$H_t = O_t \cdot \tanh(C_t) \tag{8}$$

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{9}$$

$$\tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}} \tag{10}$$

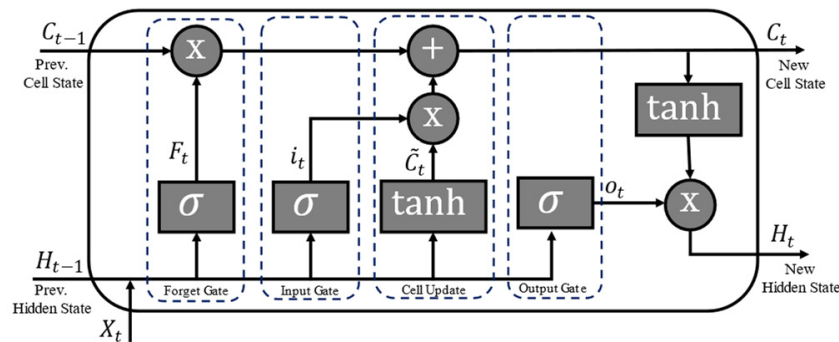


Figure 7. The architecture of LSTM.

Based on the structure and equations, LSTM explains that the input values of weights are represented by W_f , W_i , W_c , and W_o . The variable t indicates the current time, while $t - 1$ represents the previous time. Furthermore, b_f , b_i , b_c , and b_o are biases. The symbol X refers to the input, H to the output, and C to the cell state. The notation σ describes the sigmoid function, producing inputs between 0 and 1. A value of 0 indicates that no values are allowed to pass to the next stage, while a value of 1 signifies that the output can fully enter the next stage. Additionally, the hyperbolic tangent function (\tanh) is used to address gradient loss during the training process.

3.6.3. Convolutional Neural Network (CNN)

The one-dimensional convolutional neural network (1D-CNN) [42,43] is an artificial neural network architecture used for processing one-dimensional data, such as time-series data. This research applied a 1D-CNN architecture consisting of several key layers, as shown in Figure 8. These layers consist of an input layer, which functions to receive input data consisting of nine fields, namely T_1, H_1, T_2, T_3, H_3, T_4, H_4, and electricity. Five timesteps (lookback) are used in the input to predict future energy usage. Next, the convolutional layer is the main layer used to extract features from the data. In this study, we set several values for neurons in each of Convolutional Layer 1 and 2 and use ReLU activation. This is followed by the flattening layer process, which is used to transform the data into a one-dimensional vector. Finally, the dense layer process with 10 neurons is responsible for processing data representations and producing the output of the modeling process. In Table 5, information about the parameters or properties of each layer is displayed. Mathematically, the 1D-CNN method can be expressed in Equation (11).

In Equation (11), the input vector x with length n is convolved with the filter vector ω with length l , resulting in the one-dimensional output layer c [44,45].

$$c(n) = f\left(\sum_{i=-l}^l \omega(i) \sum x(n-i) + b\right) \tag{11}$$

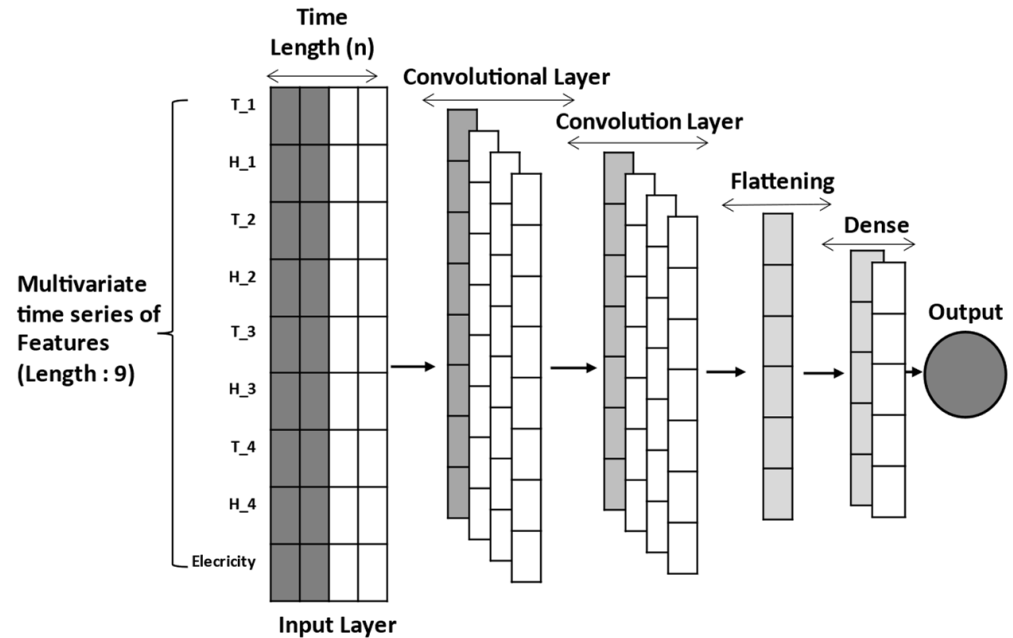


Figure 8. Architecture of CNN.

Table 5. CNN properties of the proposed model.

Layer	Properties
1st Convolutional	filter = 55, kernel size = 3, activation = ReLU
2nd Convolutional	filter = 55, kernel size = 2, activation = ReLU
Flattening	-
Dense	unit = 10

3.6.4. CNN-LSTM

Figure 9 shows the architecture of the method used in creating a predictive model for energy consumption by combining two techniques: 1D-CNN and LSTM [46–48]. The modeling process consists of three main stages: input, processing, and output. In the input stage, the process begins by using the CNN method, which consists of several conventional layers with 50 neurons each. The input data consist of the fields: T_1, H_1, T_2, T_3, H_3, T_4, H_4, and electricity. The CNN is used to extract features from the input data. Next, the results from the CNN are used as input for the LSTM method, which consists of two layers with 50 neurons each. LSTM is used to understand the sequential patterns of the data in the future period. Each layer uses the ReLU activation function. Furthermore, before generating the output, the flattening process is performed, followed by the dense layer stage with 10 neurons. This dense layer is responsible for processing the previously extracted data representations and generating energy usage predictions. Furthermore, during the model training process, optimization is performed using the Adam algorithm. A summary of the properties of each layer can be seen in Table 6.

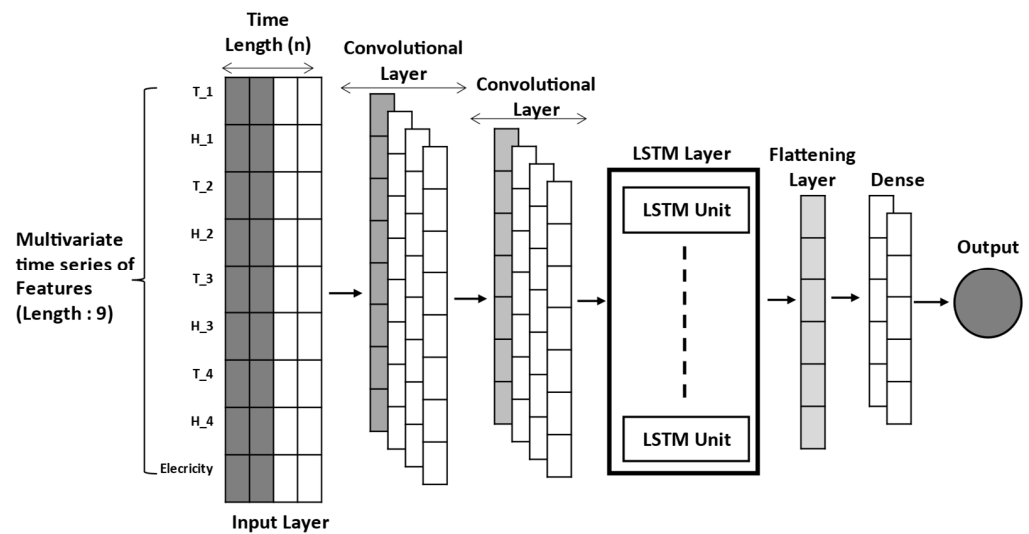


Figure 9. The architecture of CNN-LSTM.

Table 6. CNN-LSTM properties of the proposed model.

Layer	Properties
1st Convolutional	filter = 50, kernel size = 3, activation = ReLU
2nd Convolutional	filter = 50, kernel size = 2, activation = ReLU
1st LSTM	unit = 50, activation = ReLU
2nd LSTM	unit = 50, activation = ReLU
Flattening	-
Dense	unit = 10

3.6.5. Activation Function

Activation functions transform inputs into outputs that are then passed to the next layer of neurons. One of the activation functions in deep learning is the rectified linear unit (ReLU) [49]. The working process of the ReLU activation is as follows: if the input value is positive ($\alpha > 0$), then the output value is equal to the input value, which is expressed as $f(\alpha) = \alpha$. Conversely, if the input value is negative ($\alpha \leq 0$), the output value is always 0, expressed as $f(\alpha) = 0$. ReLU activation is highly computationally efficient and does not suffer from the vanishing gradient problem. In general, the equation for ReLU activation is as shown in Equation (12) [50]:

$$f(\alpha) = \max(0, \alpha) \tag{12}$$

3.6.6. Evaluation Performance Model

The evaluation process is carried out to determine the extent to which the model matches the data used and how well the model produces predictions. This research used four evaluation techniques for the model evaluation process, each with its function. The methods used are the root mean squared error (RMSE), mean absolute error (MAE), mean absolute percentage error (MAPE), and mean squared error (MSE) [42].

The RMSE technique, as shown in Equation (13), measures the error between predicted and actual values. Meanwhile, MAPE, as shown in Equation (14), is used to calculate prediction errors as a percentage. As Equation (15) indicates, MAE measures the error between predicted and actual values, considering tolerance for outliers. Additionally, MSE is used to measure the difference between predicted values and actual values. The MSE equation is presented in Equation (16) [51,52].

In this research, the symbols in the evaluation equation can be explained as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^n (y_i - \tilde{y}_i)^2}{n}} \quad (13)$$

$$MAPE = \frac{1}{n} \sum_{i=1}^n \frac{|y_i - \tilde{y}_i|}{y_i} \times 100\% \quad (14)$$

$$MAE = \frac{\sum_{i=1}^n |y_i - \tilde{y}_i|}{n} \quad (15)$$

$$MSE = \frac{\sum_{i=1}^n |y(i) - \tilde{y}_i|^2}{n} \quad (16)$$

n = Number of data points;

y_i = Actual values;

\tilde{y}_i = Predicted values;

Σ = Summation of all data.

3.6.7. Data Visualization

Data visualization is transforming data into visual information or other graphical forms. This research transforms the temperature data inside the incubator into visualizations to understand temperature patterns and distributions, making the information more easily understandable [53]. During this process, data from four temperature sensors on each side of the incubator are visualized in three dimensions (3D) with coordinates (x, y, z). The creation of 3D graphs is aided by the Matplotlib library, one of the libraries in the Python programming language. The type of visualization used in this research is surface visualization, which helps to create a three-dimensional representation corresponding to the temperature data. Furthermore, adjustments are made to set the color scale, axis labels, and titles in the visualization graph.

3.7. Output Design

In the research output design shown in Figure 10, there are several types of outputs, including:

- A web application for real-time monitoring and control: the first output is a web application designed for monitoring real-time data from sensors inside the incubator, such as temperature, humidity, electrical current, and heater status. This application provides real-time information to medical staff or the patient's family.
- Visualization of the temperature data distribution inside the incubator: the second output is the visualization of the temperature data distribution within the incubator. This visualization provides insights into temperature fluctuations in different areas within the incubator, enabling the identification of areas that may require further attention.
- A prototype of an electronic hardware incubator: the third output is a prototype product of an incubator equipped with several hardware modules such as sensors, actuators, and microcontrollers. This prototype is a tangible representation of the designed incubator by integrating various hardware components and technologies.
- Energy consumption prediction models: the final output is an energy consumption prediction model. This model can assist in optimizing resource usage by making predictions for more efficient energy consumption. The research's outcomes are expected to improve neonatal management and care.

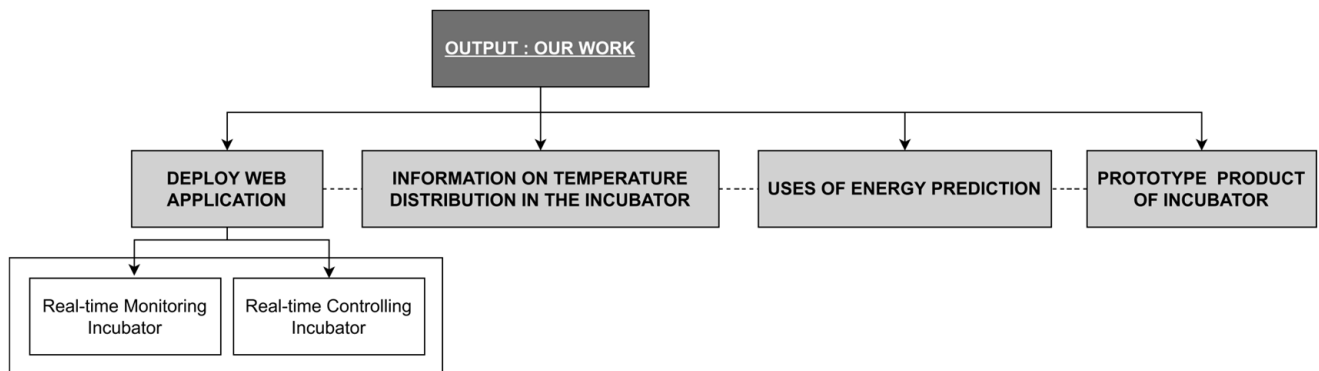
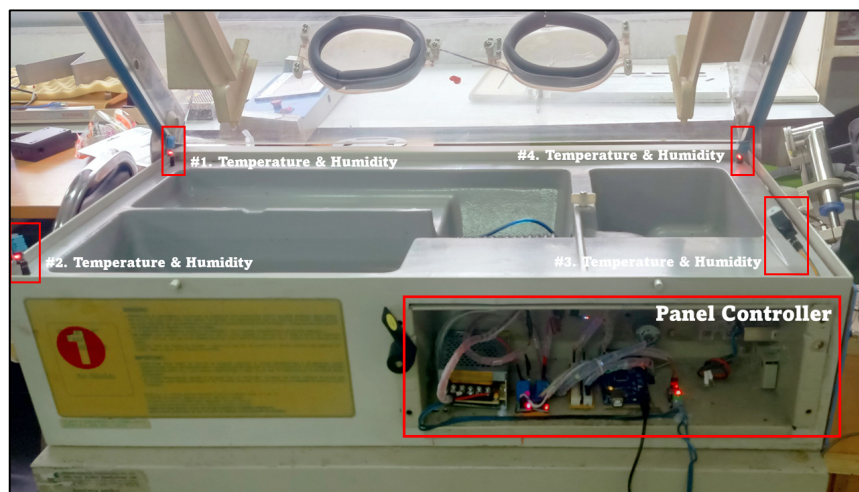


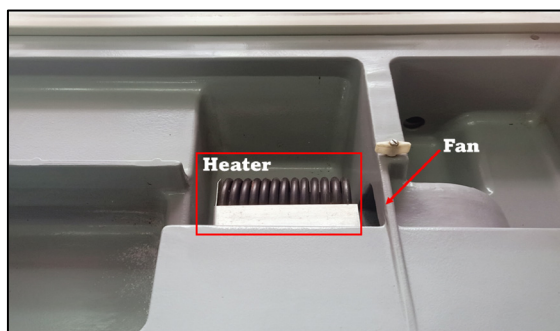
Figure 10. Output design.

4. Results and Discussion

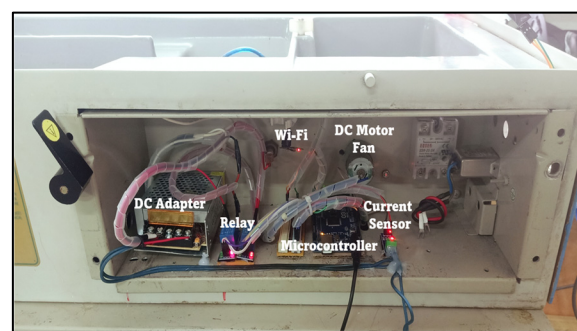
Figure 11 presents an overview of the hardware implementation used in the incubator system. Figure 11a displays the main incubator box, which includes temperature and humidity sensors on each side. Subsequently, in Figure 11b, the implementation of the heater device is shown. This device generates heat and distributes it throughout the incubator using a blower fan. Finally, Figure 11c illustrates the hardware implementation for the electronic components that control the incubator. These electronic components include a microcontroller, current sensor, WiFi module, and power supply.



(a)



(b)



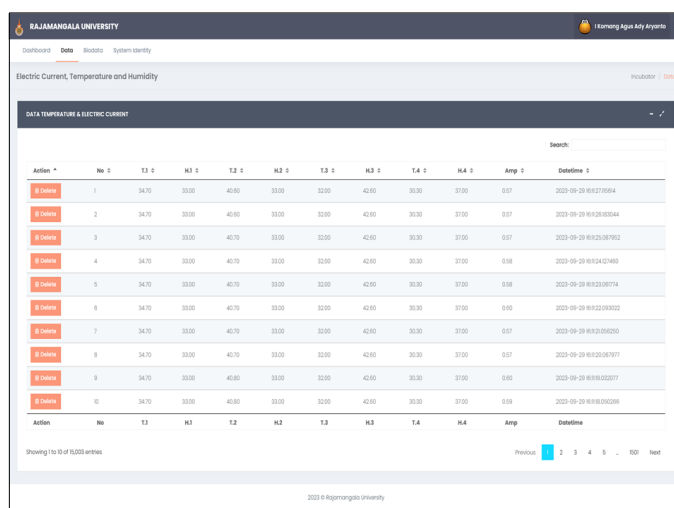
(c)

Figure 11. Hardware prototype: (a) neonatal incubator box; (b) actuator heater and blower; (c) electronic components.

Figure 12 shows the results of the responsive web application implementation, making it accessible on various screen sizes, including mobile phones. In the application, the main page is a dashboard, as seen in Figure 12a. This dashboard page displays data from the incubator in real-time as graphs. Figure 12b also indicates the stored data from the database in a tabular format. This page provides various features for users, including the ability to sort data, perform searches, and download data in Excel, CSV, PDF, and JSON formats.



(a)



(b)

Figure 12. Web application: (a) dashboard page; (b) data record page.

Based on the data stored in the database, which have been converted into a dataset and displayed in Figure 13, there are several primary attributes. These attributes include T_1 (as shown in Figure 13a), T_2 (Figure 13b), T_3 (Figure 13c), T_4 (Figure 13d), H_1 (Figure 13e), H_2 (Figure 13f), H_3 (Figure 13g), H_4 (Figure 13h), and Ampere (Figure 13i). New data based on the Ampere data were also generated to measure energy consumption. The energy (W) calculation process follows Equation (2). The results of these energy data are displayed in Figure 13j. There is a significant drop in temperature in the data T_1 , T_2 , T_3 , and T_4 (Figure 13a–d). This was caused by the incubator’s box being opened, allowing hot air to escape. The box was intentionally opened to observe the data pattern, and once the incubator box was closed again, the temperature returned to normal. Meanwhile, there is an increase in humidity in the data H_1 , H_2 , H_3 , and H_4 (Figure 13e–h).

Table 7 presents statistical information from the dataset. The statistical results are categorized into variables: count, mean, std, min, and max. The total number of data entries for each attribute is 65,535. The average temperature value in the dataset ranges from 31 °C to 39 °C. The average energy consumption is 161.47 Joule, with a minimum value of 99 Joule and a maximum of 202.40 Joule.

Furthermore, Table 8 displays the correlation coefficient values from the dataset. Noticeably, the correlation coefficient values for the Joule and Ampere attributes are identical. This is because the Joule value is calculated based on the Ampere attribute. Some attributes show negative values in the correlation coefficient to the Joule attribute. The correlation between the T_1 and T_3 data attributes is 0.69, indicating a strong relationship. Similarly, the correlation between T_2 and T_3 is 0.35, suggesting that the temperature flows from T_1 and T_2 to T_3 , as observed in Figure 14. Furthermore, the humidity data attributes H_1 and H_2 and H_3 and H_4 have correlations exceeding 0.79. Additionally, the H_4 attribute exhibits a significant correlation with the target attribute Joule, with a value of 0.91.

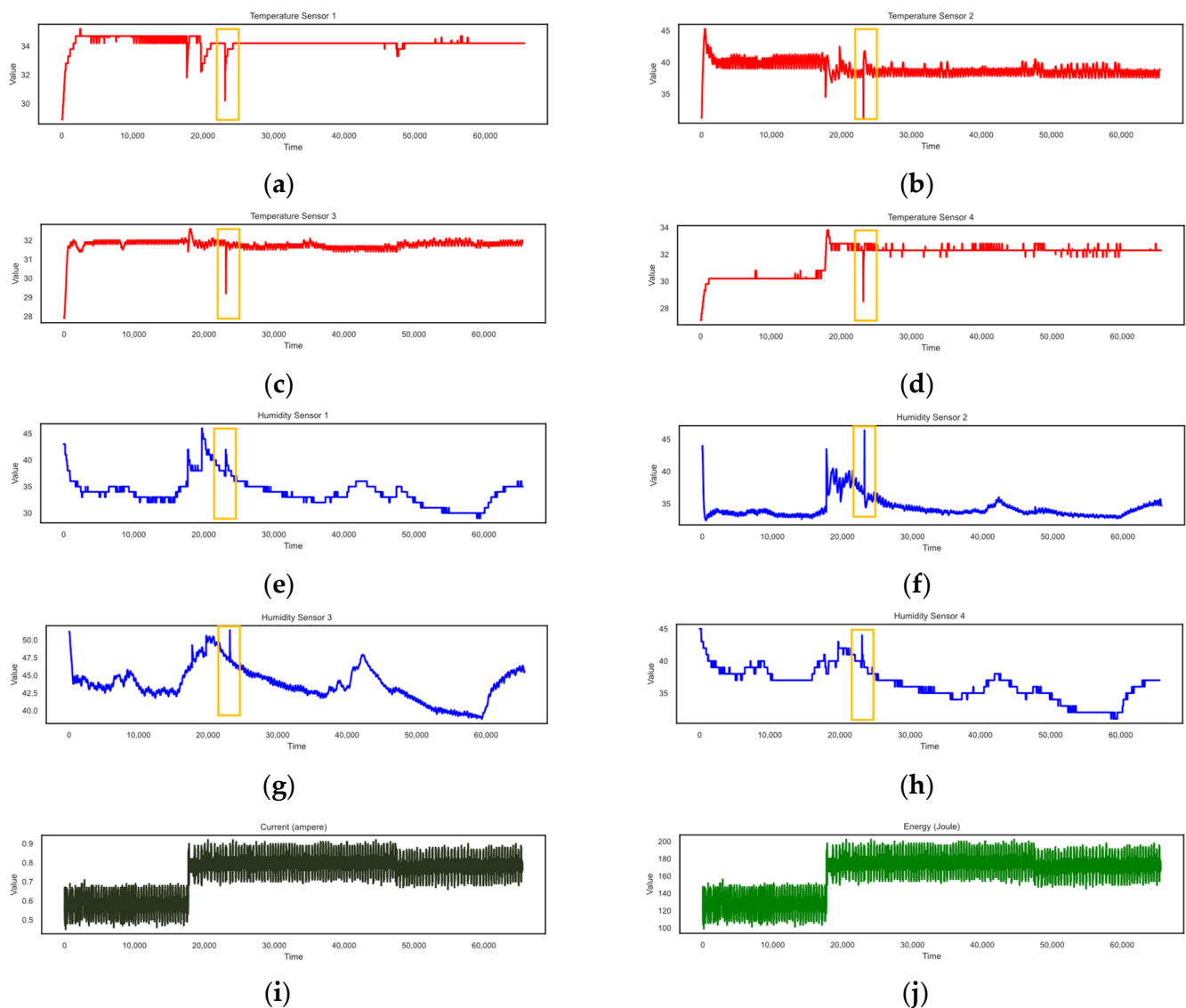


Figure 13. Data sensor: (a) Temperature 1 (T₁); (b) Temperature 2 (T₂); (c) Temperature 3 (T₃); (d) Temperature 4 (T₄); (e) Humidity 1 (H₁); (f) Humidity 2 (H₂); (g) Humidity 3 (H₃); (h) Humidity 4 (H₄); (i) electric current (Ampere); and (j) energy (Joule).

Table 7. Statistical information of the dataset.

Variable	T ₁	T ₂	T ₃	T ₄	H ₁	H ₂	H ₃	H ₄	Ampere	Joule
count	65,535	65,535	65,535	65,535	65,535	65,535	65,535	65,535	65,535	65,535
mean	34.22	39.02	31.76	31.77	34.03	34.22	43.76	36.45	0.73	161.47
std	0.48	1.14	0.27	1.02	2.62	1.51	2.47	2.60	0.09	21.63
min	28.90	31.10	27.90	27.10	29.00	32.40	38.80	31.00	0.45	99.00
max	35.20	45.30	32.60	33.80	46.00	46.40	51.40	45.00	0.92	202.40

Figure 14 depicts the temperature distribution pattern in the incubator. From the image, it is evident that the section of the incubator occupied by Sensor 1 and Sensor 2 has a higher temperature than the section occupied by Sensor 3 and Sensor 4. This temperature difference can be attributed to two primary factors. First, the proximity of Sensor 1 and Sensor 2 to the heat source results in higher temperatures in their vicinity due to the direct influence of the heat source. The second factor is the blower fan’s lack of even hot air distribution, so there are hot areas in the incubator.

Table 8. Correlation coefficient values of attributes in the dataset.

	T_1	H_1	T_2	H_2	T_3	H_3	T_4	H_4	Ampere	Joule
T_1	1	-0.43	0.1	-0.31	0.69	-0.27	-0.034	-0.22	-0.2	-0.2
H_1	-0.43	1	0.11	0.79	-0.22	0.94	-0.008	0.87	-0.003	-0.003
T_2	0.1	0.11	1	-0.37	0.35	0.015	-0.58	0.36	-0.63	-0.63
H_2	-0.31	0.79	-0.73	1	-0.22	0.8	0.3	0.57	0.29	0.29
T_3	0.69	-0.22	0.35	-0.22	1	-0.21	0.13	-0.16	-0.11	-0.11
H_3	-0.27	0.94	0.015	0.8	-0.21	1	0.018	0.86	0.022	0.022
T_4	-0.034	-0.008	-0.58	0.3	0.13	0.018	1	-0.43	0.91	0.91
H_4	-0.22	0.87	0.36	0.57	-0.16	0.86	-0.43	1	-0.42	-0.42
Ampere	-0.2	-0.003	-0.63	0.29	-0.11	0.022	0.91	-0.42	1	1
Joule	-0.2	-0.003	-0.63	0.29	-0.11	0.022	0.91	-0.42	1	1

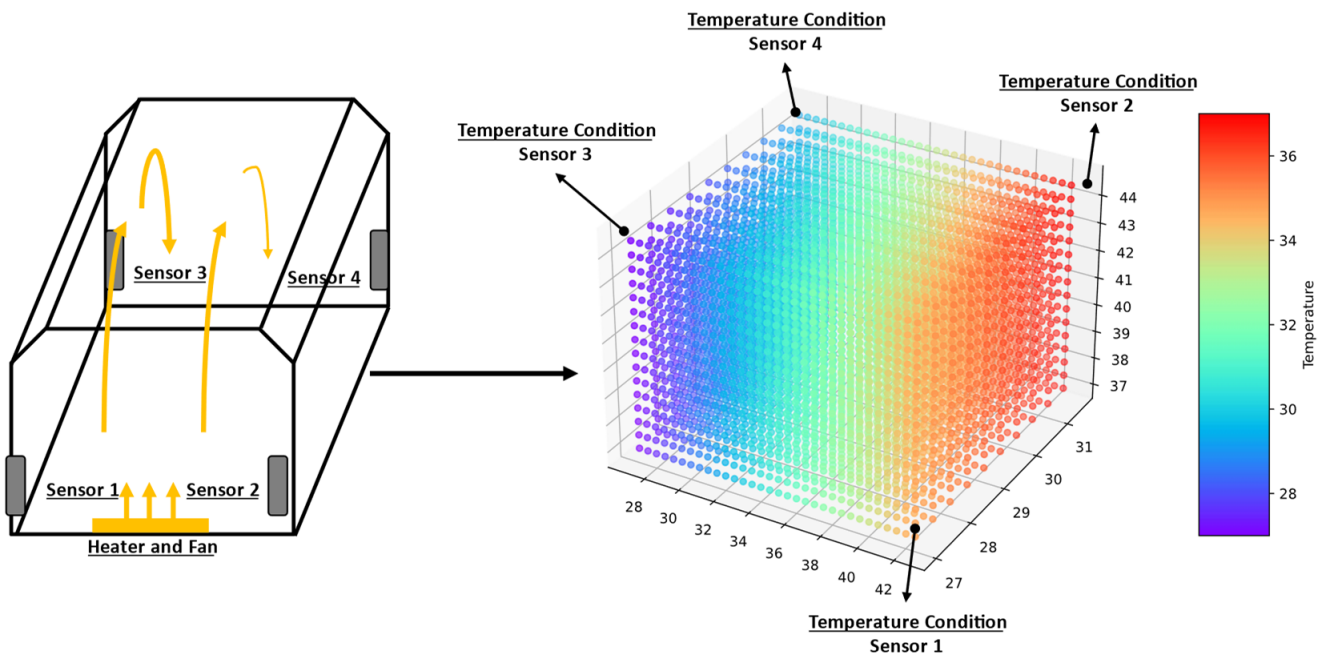


Figure 14. Temperature distribution.

Next, the dataset undergoes preprocessing in training and testing the model for predicting electricity usage. This involves calculating the total energy consumption (Joules) and computing the average values for the temperature attributes (T_1, T_2, T_3, T_4) and humidity (H_1, H_2, H_3, H_4) for each minute (60 s). As a result, a new dataset is obtained with energy values ranging from 7.000 to 11.000 Joules, as depicted in Figure 15. Subsequently, the data are preprocessed to scale for the training dataset.

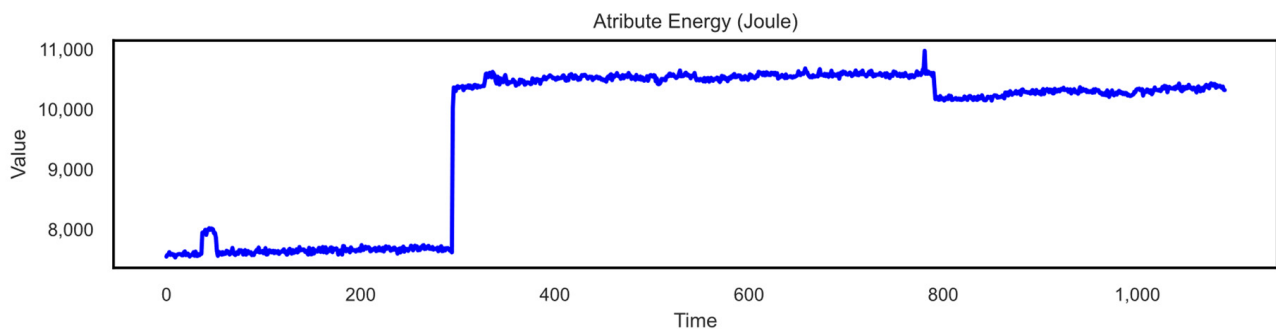
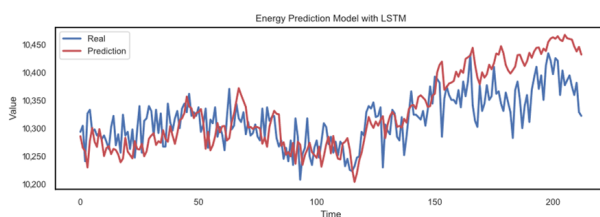


Figure 15. Target energy attribute data (Joule).

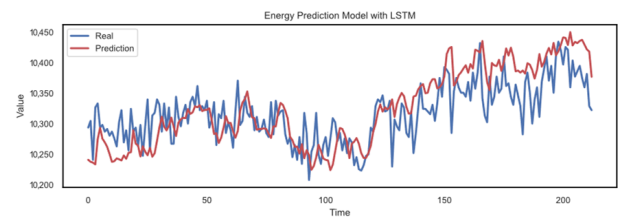
In this research, the testing of the model is divided into three parts. First, the initial model is created using the LSTM method. Second, the model is created using the CNN method. Third, a model is developed by combining the CNN and LSTM methods. The best RMSE and MAE results for the LSTM method are 42.650 and 33.575, respectively. The LSTM layer consists of 35 neurons, the dense layer contains 10 neurons, and the lookback value is set to 5. The complete testing results are presented in Table 9. The comparison between the predicted and actual values is shown in Figure 16a (with five neurons-1 and five neurons-2) and Figure 16b (with thirty-five neurons-1 and thirty-five neurons-2).

Table 9. The result of LSTM testing.

Neurons LSTM-1	Neurons LSTM-2	Dense	Lookbacks	RMSE (Joule)	MAE (Joule)	MSE (Joule)	MAPE (%)
5	5	10	5	47.824	37.945	0.004	0.4
10	10	10	5	53.587	42.816	0.004	0.4
15	15	10	5	60.238	51.231	0.005	0.5
20	20	10	5	95.778	87.149	0.008	0.8
25	25	10	5	83.127	75.284	0.007	0.7
30	30	10	5	64.154	52.420	0.005	0.5
35	35	10	5	42.650	33.574	0.003	0.3
40	40	10	5	72.566	57.842	0.006	0.6
45	45	10	5	64.616	54.205	0.005	0.5
50	50	10	5	60.420	49.234	0.005	0.5
55	55	10	5	67.376	55.351	0.005	0.5
60	60	10	5	53.054	42.452	0.004	0.4
65	65	10	5	47.793	38.947	0.004	0.4
70	70	10	5	68.445	54.603	0.005	0.5



(a)



(b)

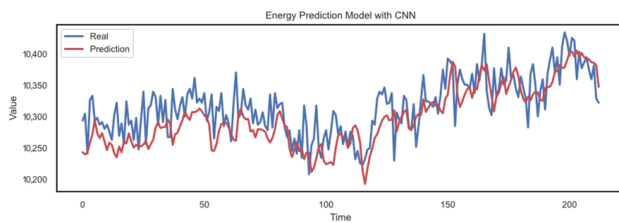
Figure 16. Results of an energy prediction model with LSTM: (a) 5 neurons-1 and 5 neurons-2; (b) 35 neurons-1 and 35 neurons-2.

In the second test using the CNN method, the best results for RMSE and MAE were 37.675 and 30.082, respectively. The optimal model for this test was achieved using 55 neurons-1 and 55 neurons-2. The comprehensive results can be found in Table 10, and the comparison between the predicted and actual values is illustrated in Figure 17a (with 30 neurons-1 and 30 neurons-2) and Figure 17b (with 55 neurons-1 and 55 neurons-2).

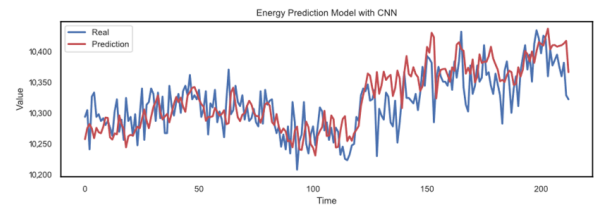
In the third test using the combination of CNN and LSTM, the best results were achieved with RMSE values of 32.436 and MAE values of 25.382. Furthermore, compared to the previous methods' results, the combination of CNN and LSTM outperformed them, indicating an improvement in the model's performance using this combination. Detailed results can be found in Table 11, and the comparison between the predicted and actual values using the CNN-LSTM method is illustrated in Figure 18a,b.

Table 10. The result of CNN testing.

Neurons CNN-1	Neurons CNN-2	Dense	Lookbacks	RMSE (Joule)	MAE (Joule)	MSE (Joule)	MAPE (%)
5	5	10	5	85.404	67.058	0.007	0.7
10	10	10	5	41.459	33.377	0.003	0.3
15	15	10	5	38.380	30.588	0.003	0.3
20	20	10	5	45.492	36.244	0.004	0.4
25	25	10	5	41.660	33.074	0.003	0.3
30	30	10	5	37.873	30.209	0.003	0.3
35	35	10	5	47.039	37.867	0.004	0.4
40	40	10	5	59.837	48.592	0.005	0.5
45	45	10	5	61.525	51.413	0.005	0.5
50	50	10	5	45.261	35.964	0.003	0.3
55	55	10	5	37.675	30.082	0.003	0.3
60	60	10	5	70.768	62.246	0.006	0.6
65	65	10	5	46.593	37.493	0.004	0.4
70	70	10	5	50.010	40.676	0.004	0.4



(a)



(b)

Figure 17. Results of an energy prediction model with CNN: (a) 30 neurons-1 and 30 neurons-2; (b) 55 neurons-1 and 55 neurons-2.

Table 11. The result of CNN–LSTM testing.

Neurons CNN-1	Neurons CNN-2	Neurons LSTM-1	Neurons LSTM-2	Lookbacks	RMSE (Joule)	MAE (Joule)	MSE (Joule)	MAPE (%)
5	5	5	5	5	53.678	43.037	0.004	0.4
10	10	10	10	5	38.052	30.266	0.003	0.3
15	15	15	15	5	53.819	43.775	0.004	0.4
20	20	20	20	5	56.922	46.486	0.005	0.5
25	25	25	25	5	55.123	43.133	0.004	0.4
30	30	30	30	5	51.156	40.531	0.004	0.4
35	35	35	35	5	52.600	44.532	0.004	0.4
40	40	40	40	5	43.773	35.523	0.003	0.3
45	45	45	45	5	64.274	51.416	0.005	0.5
50	50	50	50	5	32.436	25.382	0.002	0.2
55	55	55	55	5	44.562	36.165	0.004	0.4
60	60	60	60	5	40.758	32.583	0.003	0.3
65	65	65	65	5	46.439	37.802	0.004	0.4
70	70	70	70	5	115.992	96.323	0.009	0.9

In Figure 19, a graph or visualization of the loss values computed during the model training process is presented. These results measure how much the predicted values differ from the actual values. In Figure 19a, the loss results for Model 1 are shown, utilizing the CNN-LSTM method, where each CNN and LSTM layer has 10 neurons, resulting in an RMSE of 38.052. Meanwhile, in Figure 19b, the loss results for Model 2 using the CNN-LSTM method are displayed, with each CNN and LSTM layer having 50 neurons and an RMSE of 32.436, as indicated in Table 11. Each model in the training process utilized 240 epochs and the observed spikes in the early stages of the training loss. This

phenomenon can be attributed to the occurrence of vanishing gradients, particularly when the ReLU activation function is employed during the training process.

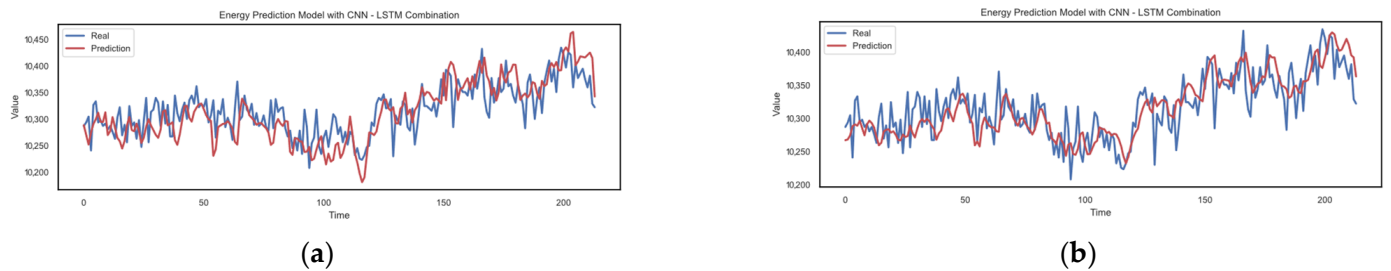


Figure 18. Results of an energy prediction model with the CNN–LSTM combination: (a) Neuron 10; (b) Neuron 50.

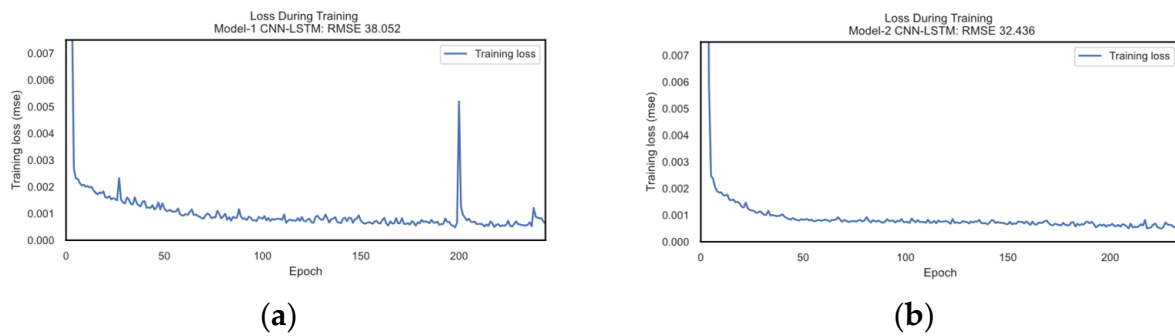


Figure 19. Losses during model training: (a) Model 1; (b) Model 2.

In this section, we also compare the results of the energy consumption prediction model with similar methods. We have compared the results of several methods, namely elastic net (ELNET) regression, support vector regression (SVR), gradient boosting regression (GBR), linear regression (LR), ridge regression (RR), and kernel ridge regression (KRR). Based on the test results, the model evaluations for each method are as follows: ELNET has an RMSE of 274.217 Joule, SVR has an RMSE of 82.572 Joule, GBR has an RMSE of 51.213 Joule, LR has an RMSE of 43.438 Joule, RR has an RMSE of 41.258 Joule, and KRR has an RMSE of 38.758 Joule. These comparison results are also presented in Table 12, with the last row mentioning the results of the model with the combination of CNN-LSTM. Additionally, for the comparison between the actual and predicted values for each method, refer to Figure 20a for ELNET, Figure 20b for SVR, Figure 20c for GBR, Figure 20d for LR, Figure 20e for RR, and Figure 20f for KRR.

Table 12. The result of comparison with other methods.

Method	RMSE (Joule)	MAE (Joule)	MSE (Joule)	MAPE (%)
Elastic Net Regression	274.217	272.205	0.026	2.6
Support Vector Regression	82.572	69.823	0.007	0.7
Gradient Boosting Regression	51.213	41.882	0.004	0.4
Linear Regression	43.438	34.700	0.003	0.3
Ridge Regression	41.258	32.404	0.003	0.3
Kernel Ridge Regression	38.758	29.841	0.003	0.3
CNN-LSTM	32.436	25.382	0.002	0.2

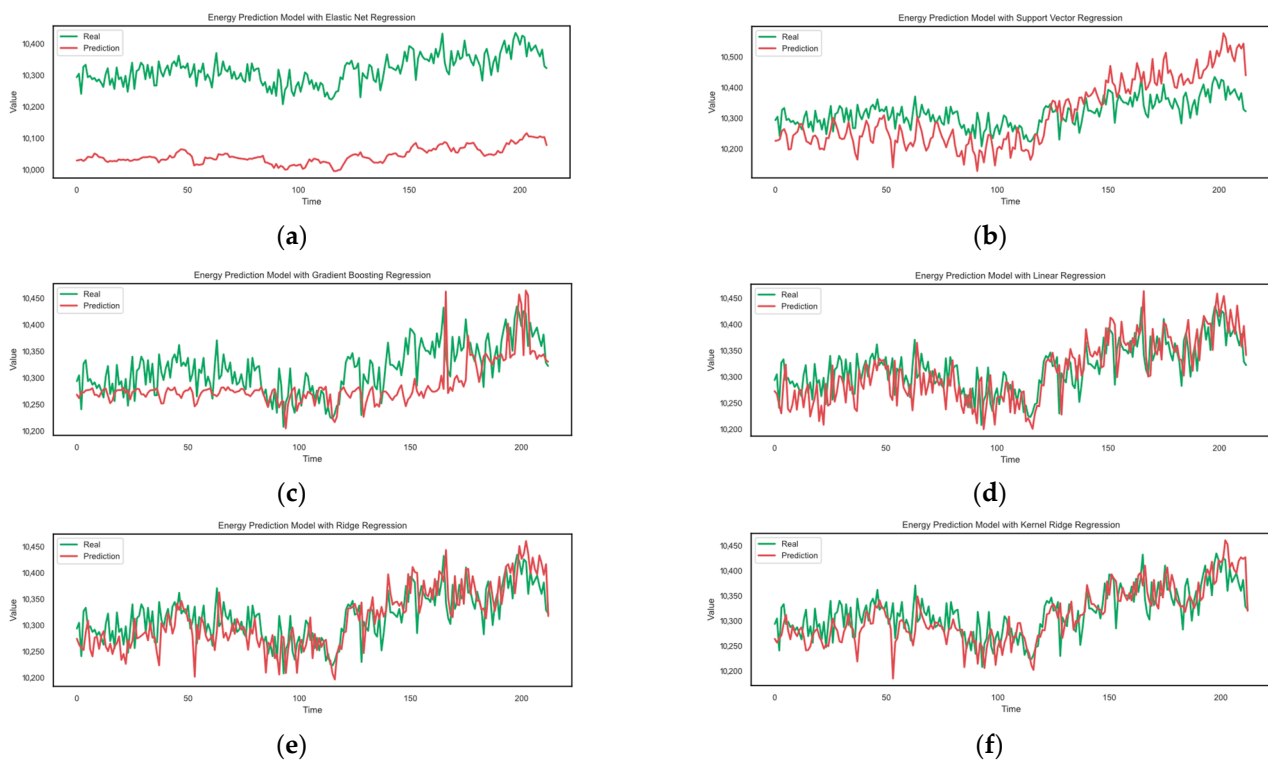


Figure 20. Comparison with other methods: (a) elastic net regression; (b) support vector regression; (c) gradient boosting regression; (d) linear regression; (e) ridge regression; (f) kernel ridge regression.

5. Conclusions

In summary, this study has effectively devised a comprehensive framework for the real-time control of energy and monitoring of incubator conditions in neonatal care facilities. The system effectively integrates various hardware components and a highly efficient database management system by utilizing the Internet of Things (IoT) architecture in combination with the MQTT communication protocol. A rigorous and validated predictive energy consumption model was developed, integrating convolutional neural networks (CNN) and long short-term memory (LSTM) techniques. The temperature distribution within the incubator exhibits variation throughout its sides, as indicated by the results of the tests. The measurements obtained from the incubator's sensors can be recorded within a database system. These stored data may be accessible in real-time using a web application, facilitating the monitoring and control functions. During the experimental phase, the long short-term memory (LSTM) approach produced root mean square error (RMSE) and mean absolute error (MAE) values of 42.650 and 33.575, respectively.

On the other hand, the convolutional neural network (CNN) technique gave RMSE and MAE values of 37.675 and 30.082, respectively. The integration of both convolutional neural network (CNN) and long short-term memory (LSTM) techniques resulted in notable enhancements in the root mean square error (RMSE) and mean absolute error (MAE) values, which reached 32.436 and 25.382, respectively. This exemplifies the integrated methodology's efficacy in augmenting the model's prediction precision. The results indicate that the model successfully enhanced predicted accuracy to a significant degree. Moreover, the technology facilitates the visualization of temperature distribution within the incubator, augmenting the standard of care provided to infants. This study significantly contributes to the progression of neonatal incubator technology, facilitating the development of more efficient and data-informed newborn care solutions in healthcare settings.

Author Contributions: Conceptualization, I.K.A.A.A. and D.M.; methodology, I.K.A.A.A., D.M. and P.N.C.; software, I.K.A.A.A.; data collection, I.K.A.A.A.; data analysis I.K.A.A.A.; writing—original draft preparation, I.K.A.A.A., D.M. and P.N.C.; writing—review and editing, I.K.A.A.A., D.M. and P.N.C.; and supervision, I.K.A.A.A., D.M. and P.N.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Acknowledgments: Sincere gratitude is extended to Rajamangala University of Technology Thanyaburi and Institute Technology and Business STIKOM Bali for their unwavering support and the invaluable resources that greatly facilitated the successful completion of this research.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Fortino, G.; Savaglio, C.; Spezzano, G.; Zhou, M. Internet of Things as System of Systems: A Review of Methodologies, Frameworks, Platforms, and Tools. *IEEE Trans. Syst. Man Cybern. Syst.* **2021**, *51*, 223–236. [\[CrossRef\]](#)
2. Wozniak, M.; Zielonka, A.; Sikora, A.; Piran, M.J.; Alamri, A. 6G-Enabled Iot Home Environment Control Using Fuzzy Rules. *IEEE Internet Things J.* **2021**, *8*, 5442–5452. [\[CrossRef\]](#)
3. Fortino, G.; Russo, W.; Savaglio, C.; Viroli, M.; Zhou, M. Modeling Opportunistic IoT Services in Open IoT Ecosystems. *CEUR Workshop Proc.* **2017**, *1867*, 90–95.
4. Krohkaew, J.; Nilaphruek, P.; Witthayawiroj, N.; Uapipatanakul, S.; Thwe, Y.; Crisnapati, P.N. Thailand Raw Water Quality Dataset Analysis and Evaluation. *Data* **2023**, *8*, 141. [\[CrossRef\]](#)
5. De Rango, F.; Potrino, G.; Tropea, M.; Fazio, P. Energy-Aware Dynamic Internet of Things Security System Based on Elliptic Curve Cryptography and Message Queue Telemetry Transport Protocol for Mitigating Replay Attacks. *Pervasive Mob. Comput.* **2020**, *61*, 101105. [\[CrossRef\]](#)
6. Kapen, P.T.; Mohamadou, Y.; Momo, F.; Jauspin, D.K.; Kanmagne, N.; Jordan, D.D. Development of a Neonatal Incubator with Phototherapy, Biometric Fingerprint Reader, Remote Monitoring, and Heart Rate Control Adapted for Developing Countries Hospitals. *J. Neonatal Nurs.* **2019**, *25*, 298–303. [\[CrossRef\]](#)
7. Alimuddin, A.; Arafiyah, R.; Saraswati, I.; Alfanz, R.; Hasudungan, P.; Taufik, T. Development and Performance Study of Temperature and Humidity Regulator in Baby Incubator Using Fuzzy-Pid Hybrid Controller. *Energies* **2021**, *14*, 14206505. [\[CrossRef\]](#)
8. Cay, G.; Solanki, D.; Rumon, M.A.A.; Ravichandran, V.; Hoffman, L.; Laptook, A.; Padbury, J.; Salisbury, A.L.; Mankodiya, K. NeoWear: An IoT-Connected e-Textile Wearable for Neonatal Medical Monitoring. *Pervasive Mob. Comput.* **2022**, *86*, 101679. [\[CrossRef\]](#)
9. Aya-Parra, P.A.; Rodriguez-Orjuela, A.J.; Rodriguez Torres, V.; Cordoba Hernandez, N.P.; Martinez Castellanos, N.; Sarmiento-Rojas, J. Monitoring System for Operating Variables in Incubators in the Neonatology Service of a Highly Complex Hospital through the Internet of Things (IoT). *Sensors* **2023**, *23*, 5719. [\[CrossRef\]](#)
10. Essa, M.E.-S.M.; El-shafeey, A.M.; Omar, A.H.; Fathi, A.E.; Maref, A.S.A.E.; Lotfy, J.V.W.; El-Sayed, M.S. Reliable Integration of Neural Network and Internet of Things for Forecasting, Controlling, and Monitoring of Experimental Building Management System. *Sustainability* **2023**, *15*, 2168. [\[CrossRef\]](#)
11. Hadi, W.N.H.W.A.; Rashid, R.A.; Sarijari, M.A.; Hamid, S.Z.A.; Muhammad, N.A. Machine Learning Bill Prediction for IoT-Based Utility Management System. In Proceedings of the 2022 IEEE 6th International Symposium on Telecommunication Technologies (ISTT), Johor Bahru, Malaysia, 14–16 November 2022; pp. 74–78.
12. Dinçer, K.; Semczuk, M. Open-Source Magnetometer for Characterizing Magnetic Fields in Ultracold Experiments. *Appl. Sci.* **2023**, *13*, 10620. [\[CrossRef\]](#)
13. Rouillard, J.; Vannobel, J.-M. Multimodal Interaction for Cobot Using MQTT. *Multimodal Technol. Interact.* **2023**, *7*, 78. [\[CrossRef\]](#)
14. Woźniak, M.; Szczotka, J.; Sikora, A.; Zielonka, A. Fuzzy Logic Type-2 Intelligent Moisture Control System. *Expert Syst. Appl.* **2024**, *238*, 121581. [\[CrossRef\]](#)
15. Guo, S.; Mao, X.; Dai, D.; Wang, Z.; Chen, D.; Wang, S. Embedded Yolo-Fastest V2-Based 3D Reconstruction and Size Prediction of Grain Silo-Bag. *Remote Sens.* **2023**, *15*, 4846. [\[CrossRef\]](#)
16. Siddharthan, H.; Deepa, T.; Chandhar, P. SENMQTT-SET: An Intelligent Intrusion Detection in IoT-MQTT Networks Using Ensemble Multi Cascade Features. *IEEE Access* **2022**, *10*, 33095–33110. [\[CrossRef\]](#)
17. Barreto, M.V.S.; Barra, W.; Rocha, E.M.; Nogueira, F.G.; Marcillo, K.L.; De Medeiros, R.L.P.; Abreu, T.W.M.; Alves, M.S. Apache Dynamic Update for Feedback Control of Computing Resources. *IEEE Access* **2019**, *7*, 55861–55872. [\[CrossRef\]](#)

18. Hernández-Gutiérrez, C.A.; Delgado-del-Carpio, M.; Zebadúa-Chavarría, L.A.; Hernández-de-León, H.R.; Escobar-Gómez, E.N.; Quevedo-López, M. IoT-Enabled System for Detection, Monitoring, and Tracking of Nuclear Materials. *Electronics* **2023**, *12*, 3042. [[CrossRef](#)]
19. Jara Ochoa, H.J.; Peña, R.; Ledo Mezquita, Y.; Gonzalez, E.; Camacho-Leon, S. Comparative Analysis of Power Consumption between MQTT and HTTP Protocols in an IoT Platform Designed and Implemented for Remote Real-Time Monitoring of Long-Term Cold Chain Transport Operations. *Sensors* **2023**, *23*, 4896. [[CrossRef](#)]
20. Dinculeană, D.; Cheng, X. Vulnerabilities and Limitations of MQTT Protocol Used between IoT Devices. *Appl. Sci.* **2019**, *9*, 848. [[CrossRef](#)]
21. Alhussein, M.; Aurangzeb, K.; Haider, S.I. Hybrid CNN-LSTM Model for Short-Term Individual Household Load Forecasting. *IEEE Access* **2020**, *8*, 180544–180557. [[CrossRef](#)]
22. Peng, Y.; Shen, H.; Tang, X.; Zhang, S.; Zhao, J.; Liu, Y.; Nie, Y. Energy Consumption Optimization for Heating, Ventilation and Air Conditioning Systems Based on Deep Reinforcement Learning. *IEEE Access* **2023**, *11*, 88265–88277. [[CrossRef](#)]
23. El-Hassan, F.T.; Ionescu, D. Design and Implementation of a Hardware Versatile Publish-Subscribe Architecture for the Internet of Things. *IEEE Access* **2018**, *6*, 31872–31890. [[CrossRef](#)]
24. Singh, T.; Solanki, A.; Sharma, S.K.; Jhanjhi, N.Z.; Ghoniem, R.M. Grey Wolf Optimization Based CNN-LSTM Network for the Prediction of Energy Consumption in Smart Home Environment. *IEEE Access* **2023**, *11*, 114917–114935. [[CrossRef](#)]
25. Govarthan, R.; Hariharan, S.; Mary, T.B.; Paul, J.J.; Manimekhalai, M.A.P.; Thilagavathi, K. IoT Based Health Monitoring and Tracking in Combat. In Proceedings of the 2023 4th International Conference on Signal Processing and Communication (ICSPPC), Coimbatore, India, 6–8 September 2023; pp. 297–301.
26. Widura, A.; Hadiatna, F.; Anugerah, D. Fuzzy-Based Smart Farming and Consumed Energy Comparison Using the Internet of Things. *IEEE Access* **2023**, *11*, 69241–69251. [[CrossRef](#)]
27. Lalitha, K.; Ramya, G.; Shunmugathammal, M. AI-Based Safety Helmet for Mining Workers Using IoT Technology and ARM Cortex-M. *IEEE Sens. J.* **2023**, *23*, 21355–21362. [[CrossRef](#)]
28. El-Leathey, L.-A.; Anghelita, P.; Constantin, A.-I.; Circumaru, G.; Chihaia, R.-A. System for Indoor Comfort and Health Monitoring Tested in Office Building Environment. *Appl. Sci.* **2023**, *13*, 11360. [[CrossRef](#)]
29. Tran, M.Q.; Elsis, M.; Liu, M.K.; Vu, V.Q.; Mahmoud, K.; Darwish, M.M.F.; Abdelaziz, A.Y.; Lehtonen, M. Reliable Deep Learning and IoT-Based Monitoring System for Secure Computer Numerical Control Machines Against Cyber-Attacks With Experimental Verification. *IEEE Access* **2022**, *10*, 23186–23197. [[CrossRef](#)]
30. Ling, Z.; Gao, C.; Sano, C.; Toe, C.; Li, Z.; Fu, X. STIR: A Smart and Trustworthy IoT System Interconnecting Legacy IR Devices. *IEEE Internet Things J.* **2020**, *7*, 3958–3967. [[CrossRef](#)]
31. Györfödi, C.A.; Turtureanu, T.; Györfödi, R.Ş.; Zmaranda, D.R. Implementing a Synchronization Method between a Relational and a Non-Relational Database. *Big Data Cogn. Comput.* **2023**, *7*, 153. [[CrossRef](#)]
32. World Health Organization Research; Reproductive Health. *Managing Newborn Problems: A Guide for Doctors, Nurses, and Midwives*; Integrated Management of Pregnancy and Childbirth; World Health Organization: Geneva, Switzerland, 2003; ISBN 9789241546225.
33. Wardana, I.N.K.; Gardner, J.W.; Fahmy, S.A. Optimising Deep Learning at the Edge for Accurate Hourly Air Quality Prediction. *Sensors* **2021**, *21*, 1064. [[CrossRef](#)]
34. Wardana, I.N.K.; Gardner, J.W.; Fahmy, S.A. Estimation of Missing Air Pollutant Data Using a Spatiotemporal Convolutional Autoencoder. *Neural Comput. Appl.* **2022**, *34*, 16129–16154. [[CrossRef](#)]
35. Wen, S.; Jia, P.; Yang, P.; Hu, C. Squill: Testing DBMS with Correctness Feedback and Accurate Instantiation. *Appl. Sci.* **2023**, *13*, 2519. [[CrossRef](#)]
36. Babovic, Z.B.; Protic, J.; Milutinovic, V. Web Performance Evaluation for Internet of Things Applications. *IEEE Access* **2016**, *4*, 6974–6992. [[CrossRef](#)]
37. Brito, T.; Ferreira, M.; Monteiro, M.; Lopes, P.; Barros, M.; Santos, J.F.; Santos, N. Study of JavaScript Static Analysis Tools for Vulnerability Detection in Node.js Packages. *IEEE Trans. Reliab.* **2023**, *72*, 1324–1339. [[CrossRef](#)]
38. Elmannai, W.; Elleithy, K.; Benz, A.A.; DeAngelis, A.C.; Weaver, N. An Enhanced Piezoelectric-Generated Power Technique for Qi Wireless Charging. *Clean Technol.* **2023**, *5*, 94–115. [[CrossRef](#)]
39. Ahsan, M.M.; Mahmud, M.A.P.; Saha, P.K.; Gupta, K.D.; Siddique, Z. Effect of Data Scaling Methods on Machine Learning Algorithms and Model Performance. *Technologies* **2021**, *9*, 52. [[CrossRef](#)]
40. Aaryan, A.; Kanisha, B. Forecasting Stock Market Price Using LSTM-RNN. In Proceedings of the 2022 2nd International Conference on Advance Computing and Innovative Technologies in Engineering (ICACITE), Greater Noida, India, 28–29 April 2022; pp. 1557–1560.
41. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780. [[CrossRef](#)] [[PubMed](#)]
42. Lawal, A.; Rehman, S.; Alhems, L.M.; Alam, M.M. Wind Speed Prediction Using Hybrid 1D CNN and BLSTM Network. *IEEE Access* **2021**, *9*, 156672–156679. [[CrossRef](#)]
43. Wang, X.; Xia, M.; Deng, W. MSRN-Informer: Time Series Prediction Model Based on Multi-Scale Residual Network. *IEEE Access* **2023**, *11*, 65059–65065. [[CrossRef](#)]
44. Zhao, J.; Mao, X.; Chen, L. Speech Emotion Recognition Using Deep 1D & 2D CNN LSTM Networks. *Biomed. Signal Process Control* **2019**, *47*, 312–323. [[CrossRef](#)]

45. Mitiche, I.; Nesbitt, A.; Conner, S.; Boreham, P.; Morison, G. 1D-CNN Based Real-Time Fault Detection System for Power Asset Diagnostics. *IET Gener. Transm. Distrib.* **2020**, *14*, 5766–5773. [[CrossRef](#)]
46. Han, Z.; Cui, B.; Xu, L.; Wang, J.; Guo, Z. Coupling LSTM and CNN Neural Networks for Accurate Carbon Emission Prediction in 30 Chinese Provinces. *Sustainability* **2023**, *15*, 13934. [[CrossRef](#)]
47. Zhang, C.; Chen, P.; Jiang, F.; Xie, J.; Yu, T. Fault Diagnosis of Nuclear Power Plant Based on Sparrow Search Algorithm Optimized CNN-LSTM Neural Network. *Energies* **2023**, *16*, 2934. [[CrossRef](#)]
48. Rubasinghe, O.; Zhang, X.; Chau, T.K.; Chow, Y.; Fernando, T.; Iu, H.H.C. A Novel Sequence to Sequence Data Modelling Based CNN-LSTM Algorithm for Three Years Ahead Monthly Peak Load Forecasting. *IEEE Trans. Power Syst.* **2023**, 1–15. [[CrossRef](#)]
49. Jain, R.; Semwal, V.B. A Novel Feature Extraction Method for Preimpact Fall Detection System Using Deep Learning and Wearable Sensors. *IEEE Sens. J.* **2022**, *22*, 22943–22951. [[CrossRef](#)]
50. Lee, S.Y.; Byeon, S.; Kim, H.S.; Jin, H.; Lee, S. Deep Learning-Based Phase Prediction of High-Entropy Alloys: Optimization, Generation, and Explanation. *Mater. Des.* **2021**, *197*, 109260. [[CrossRef](#)]
51. Chicco, D.; Warrens, M.J.; Jurman, G. The Coefficient of Determination R-Squared Is More Informative than SMAPE, MAE, MAPE, MSE and RMSE in Regression Analysis Evaluation. *PeerJ Comput. Sci.* **2021**, *7*, e263. [[CrossRef](#)]
52. Hodson, T.O. Root-Mean-Square Error (RMSE) or Mean Absolute Error (MAE): When to Use Them or Not. *Geosci. Model Dev.* **2022**, *15*, 5481–5487. [[CrossRef](#)]
53. Embarak, O. *Data Analysis and Visualization Using Python: Analyze Data to Create Visualizations for BI Systems*; Apress: New York, NY, USA, 2018; ISBN 9781484241097.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.