International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 5
ISSN: 2088-5334

# Smart Machine Learning-based IoT Health and Cough Monitoring System

Wai Leong Pang [a], Gwo Chin Chung [b,*], Kah Yoong Chan [b], Lee It Ee [b], Mardeni Roslee [b], Edzham Fitrey [c], Yee Wai Sim [a], Murman Dwi Prasetio [d]

[a] *School of Engineering, Taylor's University, Subang Jaya, 47500, Malaysia*
[b] *Faculty of Engineering, Multimedia University, Cyberjaya, 63100, Malaysia*
[c] *NXP Semiconductors, Petaling Jaya, 47300, Malaysia*
[d] *School of Industrial and System Engineering, Telkom University, Jawa Barat, 40257 Indonesia*
*Corresponding author: *gcchung@mmu.edu.my*

***Abstract*— Coronavirus 2019, more commonly known as COVID-19, was declared a global pandemic by the World Health Organization (WHO) on March 11, 2020. The β coronavirus culpable for the disease, SARS CoV-2, is known to be highly contagious with a relatively long incubation period of up to 14 days and is transmittable through small droplets, especially among people who are in close face-to-face contact. The Ministry of Health of Malaysia has recommended five days of quarantine for people who are positive for COVID-19 to avoid further disease transmission. Many resources are used to monitor patients throughout the quarantine period. Therefore, this project would like to present an IoT-enabled wearable device capable of monitoring COVID-19 quarantine patients by utilizing sensors to monitor the necessary health parameters and facilitate home quarantine. The low-cost ESP32 and Arduino Nano 33 BLE Sense microcontrollers are used in this device. They are connected to various IoT sensors to collect temperature, humidity, and sound data. The data obtained will then be uploaded to an IoT platform for doctors to analyze and monitor remotely via the health log throughout the 5-day quarantine period. An alert system is also devised to inform the medical staff if the patient is experiencing abnormal symptoms. The medical staff can then bring their attention to the patient and take the necessary actions to combat COVID-19.**

***Keywords*— COVID-19; wearable health monitoring system; cough detection system.**

*Manuscript received 15 Nov. 2022; revised 29 Mar. 2023; accepted 12 Jun. 2023. Date of publication 31 Oct. 2023.*


## I. Introduction

The year 2020 has witnessed COVID-19, or Coronavirus Disease 2019, being declared by the World Health Organization (WHO) as a global pandemic. The novel coronavirus attacks the victim's respiratory system and has further implications. This disease has affected the world economically just as much as it has affected the world's population regarding health and wellness. The virus culpable had been identified to be SARS CoV-2, which stands for "Severe Acute Respiratory Syndrome Coronavirus 2". The virus is proven to be transmittable through small droplets when people come into close contact [1]. The virus is highly contagious and has a long incubation period of up to 14 days. In this incubation period, the patient may or may not exhibit any symptoms of contracting the disease, thus rendering them possible silent carriers of the virus that could be spread to other people [1]. WHO has recommended a quarantine period of 14 days for people who have come into contact with any COVID-19 patient. In Malaysia, people considered suspect, probable, or confirmed to be suffering from low-risk COVID-19 undergo quarantine at Low-Risk Quarantine and Treatment Centers, where the local authority will monitor their symptoms.

The advancements of the Internet of Things (IoT) have created a smart ecosystem that utilizes processors, sensors, and communication devices to collect, analyze, and act based on data acquired from the environment. Multiple IoT devices may share the data collected under a unified cloud server where data analysis is performed without human intervention [2]–[6]. For healthcare, in particular, IoT has branched into a new specialized field known as the Internet of Medical Things (IoMT). IoMT platforms will usually revolve around smart biomedical sensors to acquire health-related parameters, including temperature, heart rate, respiration rate, blood pressure, and oxygen saturation. Health monitoring is considered a big part of IoMT implementation [7]. Having taken into account the quarantine measures recommended by WHO and enforced in many countries, including Malaysia, this project aims to help facilitate any home quarantine by providing health monitoring information for patients by utilizing an IoT wearable device to aid doctors and researchers in their quest to identify possible COVID-19 patients.

The World Health Organization (WHO) has recommended a quarantine period of 14 days for people who have traveled abroad and have come into close contact with a suspected COVID-19 carrier, and monitoring needs to be performed by medical staff during this period. Healthcare services have increased over the past few years and placed a high burden on either having doctors always be close to their patients during the diagnosis process or the patients having to remain in hospitals during the health monitoring process [8]. The increase in demand for health monitoring has encouraged the development of the healthcare field by making the most of IoT and wireless technology and integrating it with the internet to allow a continuous monitoring system for the patient [9]–[12]. This is important, especially since the high demand for hospitalization has made it difficult for patient screening to be done in a 24-hour manner [13]. Machine learning and deep learning are used to provide faster diagnosis in healthcare [14]. Various research and approaches have utilized deep learning to identify COVID-19 patients directly by analyzing the X-rays; these types of analysis will require complex devices that may not be available at home for quarantined patients [15]. In addition, a proper cough detection mechanism needs to be devised, as it is among the most common symptoms of COVID-19 at 67.7%, behind only fever [1]. Hence, this project aims to introduce a wearable device capable of obtaining data to monitor the health of COVID-19 patients and detect coughs and other possible complications. The objectives of this project can be summarized as follows:

- To monitor possible COVID-19 symptoms for patients by designing an IoT wearable device.
- To have an alert system that notifies medical staff of COVID-19 symptoms suffered by the patient.
- To reduce operation costs by advocating home quarantine using the health monitoring device rather than at a dedicated quarantine center.

The project scope can be summarized as follows.

- Designing a wearable health monitoring system to collect and obtain health parameters for quarantined people in relation to COVID-19.
- The compilation of data on an IoT platform is to be monitored by doctors or quarantined patients themselves.
- Alerting patients or doctors of possible COVID-19 symptoms based on data collected by the system.

The related works are reported as follows.

### *A. COVID-19*

Coronavirus Disease 2019, widely known as COVID-19, is a respiratory disease that has been characterized as a global pandemic by the WHO in February 2020. The disease attacks the respiratory system of victims, resulting in coughing, fatigue, fever, and breathlessness. The symptoms of COVID-19 range from showing no symptoms to acute respiratory distress syndrome (ARDS) [1]. Table 1 shows a list of symptoms associated with COVID-19 according to WHO.

TABLE I
COVID-19 SYMPTOMS

| **COVID-19 Symptom** | **Percentage** |
|---|---|
| Fever | 87.9% |
| Dry Cough | 67.7% |
| Tiredness or Fatigue | 38.1% |
| Phlegm Production | 33.4% |
| Breath Shortness | 18.6% |
| Arthralgia / Myalgia | 14.8% |
| Sore Throat | 13.9% |
| Headache | 13.6% |
| Chills | 11.4% |
| Nausea | 5.0% |
| Congestion of Nasal | 4.8% |
| Diarrhea | 3.7% |
| Hemoptysis | 0.9% |
| Conjunctival Congestion | 0.8% |
| Fever | 87.9% |

Since fever and dry cough have accounted for a significant percentage of COVID-19 symptoms at 87.9% and 67.7%, respectively, using these parameters as the primary symptoms to detect among patients using this project's health monitoring system makes sense. According to the National Health Service (NHS) Inform, COVID-19 symptoms include fever with a temperature of at least 37.8˚C. However, a high temperature does not necessarily indicate that the patient is positive for COVID-19. The only way to confirm a patient is positive for COVID-19 is through confirmatory tests known as Polymerase Chain Reaction (PCR) and Antigen Rapid Tests [16].

In humans, body temperature can vary from one person to another and is affected by many factors, including age, gender, activity levels, menstrual cycle (for females), and time of day. As such, there is no definite body temperature, and it can only be identified in the range [17]. Based on the comprehensive compilation of body temperature studies performed, the average temperature is 36.59˚C, considering all possible measuring sites for a patient. The study noted that rectal measurement, on average, shows the highest temperature reading compared to other mediums, including oral, tympanic, urine, and axillary or armpit. Infrared (IR) temperature measurement is another method of measuring body temperature and is popularly used due to its non-invasive method of use, unlike the previously mentioned measurement tools. It is low-cost and safe to use, although the reading may be inconsistent compared to other measurement methods compared to a reference mercury thermometer.

### *B. Health Monitoring System*

A few literature review papers were studied for health monitoring systems, although most do not cater to COVID-19 symptoms. A wireless patient monitoring system is proposed that can give electrocardiography (ECG), blood pressure, temperature, and pulse oximeter readings that are connected using Arduino and transmitted to the cloud [18]. Health monitoring devices with mobile applications, IoT sensors, Arduino, and IoT cloud platforms allow automatic health monitoring [19], [20]. Power BI was proposed to analyze the data collected [21]. However, Power BI has limited capability to analyze real-time data.

Various sensors were used to measure health data. The MAX30100 sensor measured the heartbeat rate and blood oxygen level. MLX90614 was used to measure the body temperature [22], [23]. An alternative approach to preventing the spread of the virus is to prioritize safety and social distancing measures. Researchers have developed an innovative solution called the IoT Safety Distance Monitoring Device [24]. This device aims to ensure that individuals maintain an adequate physical distance from others, thus effectively curbing the transmission of the virus. Another valuable tool for virus prevention is a face mask detector integrated with a health status monitoring system, as proposed in [25]. This system identifies the presence or absence of face masks and monitors individuals' health status. By combining these two functionalities, the system offers a comprehensive approach to prevention, emphasizing the importance of mask-wearing and health monitoring in mitigating the spread of the virus.

Tiny Machine Learning (TinyML), also referred to as embedded machine learning, is a growing field of machine learning running on TensorFlow Lite that is capable of being executed on low-power microcontroller units (MCUs) such as the Arduino, Microbit, and Raspberry Pi. It has allowed the collaboration of low-power embedded systems with machine learning to open up various possibilities for low-cost projects. One of the advantages of TinyML's implementation is energy efficiency. Due to MCUs requiring less power compared to their central processing unit (CPU) and graphical processing unit (GPU) counterparts, implementation of these systems is easier as they only require a battery [26].

### *C. Cough Detection*

On the other hand, this project focused more on utilizing the cough detection algorithm in a working prototype. An automated system for screening respiratory disease is designed by analyzing raw cough data using two convolution neural networks (CNN) [27]. An experiment was carried out to collect and label the patients' cough recordings for cough detection [28]. A cough classification device was proposed to identify COVID-19 coughs from non-COVID-19 ones using the Mel Frequency Cepstral Coefficient (MFCC) feature extraction and CNN classification methods [29].

## II. Material and Method

The architecture of the COVID-19 IoT Health Monitoring System is shown in Fig. 1. It consists of the ESP32 as the Health Monitoring Centre (HMC), sensors, Arduino Nano 33 BLE, ThingSpeak, and an alert system. The following sections will illustrate the function of each module in the COVID-19 IoT Health Monitoring System.

### *A. ESP32 as the IoT Health Monitoring Device*

The TTGO T-Display, which runs on an ESP32 microcontroller, acts as the primary IoT Health Monitoring Device (Fig. 2). The device is compact, weighing 7.81g (excluding headers), and can be powered through a 5V/1A USB type-C port or 3.7V lithium battery. Running on the FreeRTOS via the ESP32 microcontroller, it is programmed using C language on the Arduino platform, which is easy to understand and familiar. Arduino's wide compatibility with many libraries allows easy integration of several third-party sensors. The primary appeal of using TTGO T-Display ESP32 is the built-in 1.14-inch LCD Display. While the number of ports is limited compared to the conventional ESP32 single board computer (SBC), the built-in LCD makes the device compact and, with a dimension of 51.52mm × 25.04mm, makes it easier to implement as a wearable device. Furthermore, running on the ESP32 microcontroller, the TTGO T-Display is equipped with Wi-Fi 802.11 b/g/n and supports a speed of up to 150Mbps. The ESP32 TTGO T-Display (from here on abbreviated as 'ESP32' for simplicity purposes) acts as the Health Monitoring Centre (HMC) that collects data from all sensors, including MLX90614, LM35, DHT22, and the Arduino Nano 33 BLE.

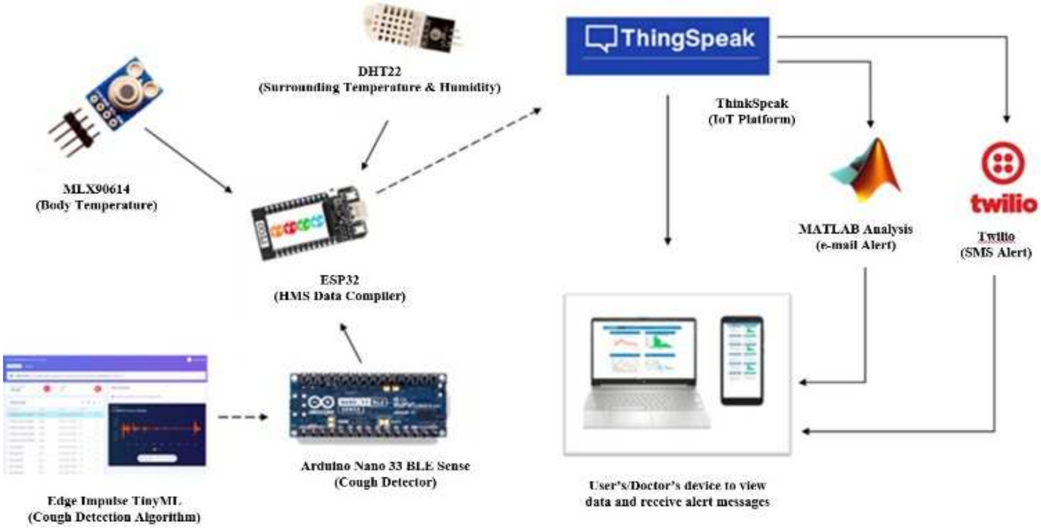


Fig. 1 Full architecture of the system proposed

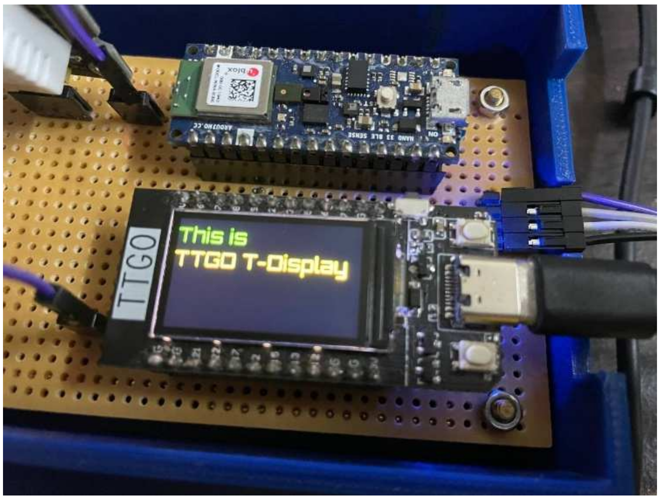


Fig. 2 TTGO T-Display ESP32 (abbreviated as ESP32)

### B. Sensors

For body temperature, the sensor considered for measuring body temperature is MLX90614 (Fig. 3). At 7.41 USD, it is almost double the price of the LM35 sensor used in many research papers. It is a non-contact temperature sensor that relies on infrared (IR) to measure the temperature of an object or ambient environment. It operates based on Stefan Boltzmann's theory of radiation, which states that every object and living organism emits radiation energy directly proportional to its area, emissivity, and absolute temperature to the power of four, as shown in Equation 1 [30].

$$E = A\sigma T^4 \tag{1}$$

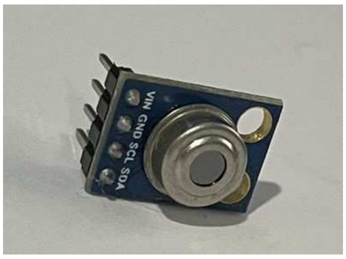


Fig. 3 MLX90614 uses infrared to measure temperature

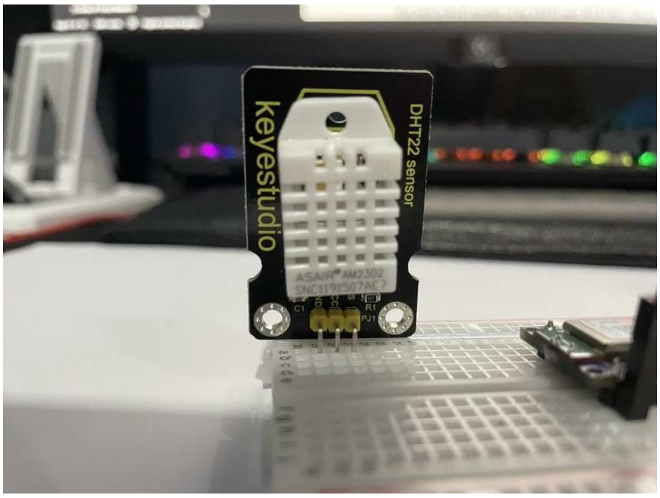


Fig. 4 DHT22 measures surrounding temperature and humidity

MLX90614 measures the temperature of the emitted IR rays from objects or surfaces. It is operated through a 3.3V or 5V voltage source and communicates to the ESP32 using the I2C protocol. It can operate from the -70 to 382.2 ˚C range when measuring body temperature. According to the datasheet, it can measure temperature for objects ranging in emissivity from 0.1 to 1, where clean skin has an emissivity value of around 0.98 [31]. The company that developed MLX90614, Melexis, used this particular IR sensor for healthcare temperature sensors, signifying its reliability in measuring at a medical-grade level. According to the datasheet, one major downside of the MLX90614 is its lifespan, as it can only undergo a minimum of 100,000 erase/write cycles. In this project's implementation, 14 days of continuous temperature monitoring will use many erase/write cycles, and thus, the reading frequency needs to be reduced to ensure the sensor can continue to operate within the quarantine duration. Since body temperature hardly changes within a few minutes, the MLX90614 is configured to obtain a reading once every 30 minutes. This allows a total erase/write cycle of around 672 cycles per patient throughout his/her 14-day quarantine period and allows up to 148 patients to use the system throughout their quarantine periods before requiring the replacement of the sensor. Data reading using the MLX90614 is direct and simple. VIn, GND, SDA (Data), and SCL (Clock) pins are connected to the corresponding 5V/3.3V, GND, SDA, and SCL pins of the ESP32, respectively. It is programmed simpler than the LM35 as it only requires one command line due to its more accurate and consistent reading.

The DHT22 is a digital sensor used for measuring surrounding temperature and humidity (Fig. 4). It is relatively cheap at around USD3.50. It can be powered at 3V to 5V power from any digital I/O port and consumes 2.5mA current at max usage. The DHT11 can measure 20% to 80% humidity levels at 5% accuracy compared to its predecessor. The DHT22 can read a wider humidity range from 0 to 100% at 2-5% accuracy. Regarding temperature, the DHT22 has a wider range of reading at -40 to 80 °C compared to the DHT11's range at 0 to 50 °C. The DHT22 is used to obtain environmental data that is used as a reference for medical staff to consider should there be any abnormal reading in the patient's body temperature and for calibration of the MLX90614 sensor.

### C. Cough Detection System

In this project, cough detection is primarily achieved through machine learning. Tiny Machine Learning (TinyML) allows deploying machine learning algorithms on various devices, including development boards such as Arduino. Edge Impulse is a free online machine-learning platform that allows data collection, training, and deployment of the TinyML model. The features are extracted for audio datasets through the Mel-Frequency Cepstral Coefficient (MFCC), which is suitable for the human voice. Keras neural network is then chosen as the learning block.

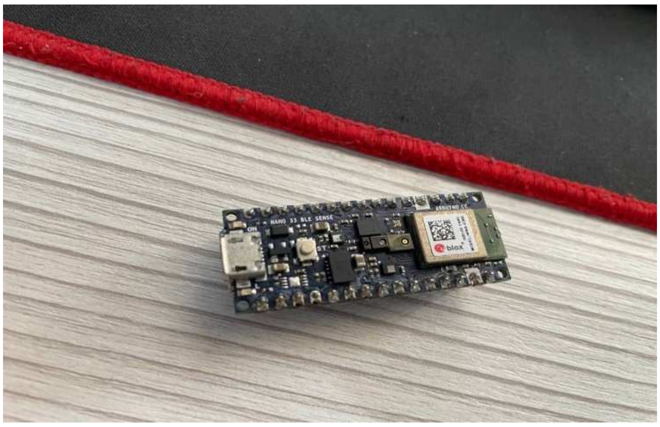

Fig. 5 Arduino Nano 33 BLE Sense

Developing the cough detection algorithm involves data acquisition for both cough and noise sounds, extraction of the features via MFCC methods, and training the datasets using the Keras neural network. The Arduino Nano 33 BLE (Fig. 5) is chosen as the device for deployment. It is compatible with running TinyML models and can be directly used within Edge Impulse API for data acquisition and testing. With a built-in MP34DT05-A microphone equipped, the Arduino Nano 33 BLE can be deployed as the cough detection device capable of running the machine learning algorithm and recording audio without requiring a separate microphone.

### D. *ThingSpeak*

The ThingSpeak platform allows users to create multiple channels that can store data in their respective fields. For free users, ThingSpeak allows up to 4 channels per account, but it can be extended to 250 with a paid membership. In the real-life implementation, the doctors and/or nurses can have a full list of patients under their care, with each patient allocated a single channel to monitor their health parameters. These channels can also be configured publicly, privately, or selectively shareable, thus allowing patients to view their data logs restrictively.

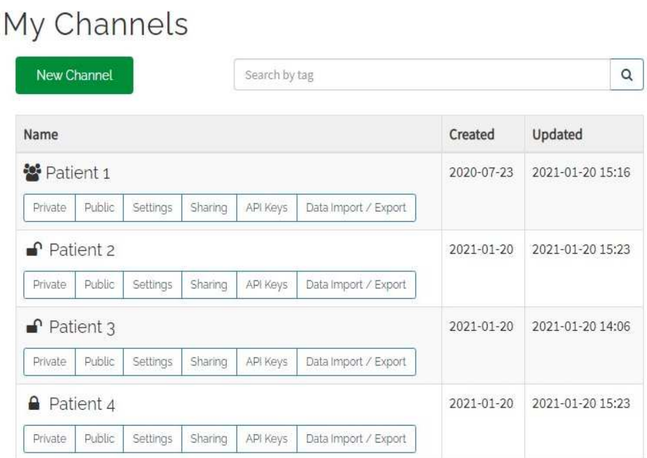


Fig. 6 ThingSpeak Patients' Channels

Each channel is given a channel ID, read and write Application Programming Interface (API) key. Data uploading is done in the data uploading phase of the ESP32 by using the channel number, and the write API key obtained on the channel is assigned to the patient. The read API key, on the other hand, is mostly used in the alert system of the project to notify medical staff via e-mail or SMS. The example of the patients' channels created is shown in Fig. 6. ThingSpeak allows medical staff to monitor multiple patients. The privacy setting can be adjusted so that only the patient and the corresponding medical staff can view the health data log.

### E. *Alert System*

The COVID-19 IoT Health Monitoring System is equipped with an alert system that utilizes the Read API Key provided by ThingSpeak to access data that is stored in the IoT platform. The alert, which is targeted for medical staff when a patient's temperature exceeds normal level, is done through e-mail and SMS, albeit with a slight difference in method. An advantage of the alert system implanted is that no additional modules are required for execution; thus, the lower cost can be maintained.

Fig. 7 shows the proposed alert system that revolves around a few components, including ThingSpeak, React, ThingHTTP, Twilio, and MATLAB Analysis. Essentially, both mediums require the use of React, which is an app also developed by ThingSpeak. It functions as the automation service that can be configured to launch an alert event, either through ThingHTTP or MATLAB Analysis, periodically or when a certain threshold value of a particular field in ThingSpeak is exceeded.

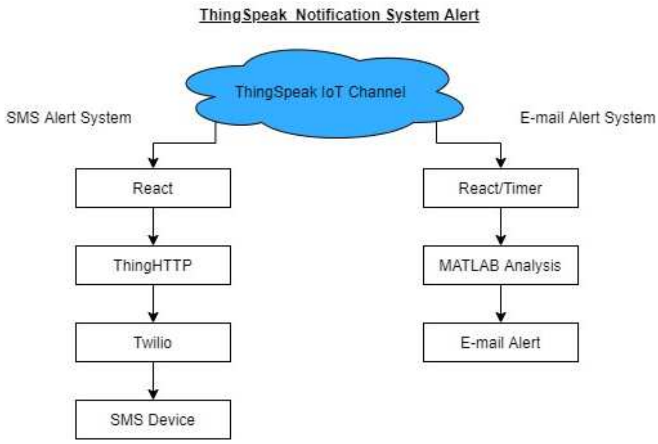


Fig. 7 The alert system proposed

In this project, React is set to activate the alert event when the patient's body temperature exceeds the threshold value of 37.5˚C, which is beyond a regular healthy person's temperature range. In the case of the e-mail alert system, React is set to automate the execution of the MATLAB Analysis program "COVID-19 Email system v2.0 (BETA)" when data in the Field 1 (Body Temperature) of Channel 1104440 (Patient 1) exceeds 37.5 ˚C. It is also set to automate each time the condition is met. For SMS alerts, React is configured to allow automation in almost a similar manner to e-mail alerts, but the difference is in the action, where it executes ThingHTTP. ThingHTTP is an app developed by ThingSpeak that acts as the bridge for communication between websites, web services, and devices. In this system, ThingHTTP acts as the bridge to Twilio SMS, which is a free-to-use API-based SMS service that sends SMS alert to medical staff.

## III. Result and Discussion

### A. Sensor Testing

An experiment was done by wearing both LM35 and MLX90614 temperature sensors on each wrist and measuring the body temperature. The experiment was conducted in two situations, one in an air-conditioned room and another at room temperature. The results of the LM35 and MLX90614 are shown in Table 2 and Table 3, respectively.

TABLE II
LM35 TESTING RESULTS

| LM35 | Air-Condition | Room Temperature |
|---|---|---|
| Standard Deviation | 0.23 | 0.54 |
| Error vs Reference IR Temperature (˚C) | -3.08 | -0.61 |
| Error vs Reference Oral Temperature (˚C) | -3.48 | -1.01 |

TABLE III
MLX90614 TESTING RESULTS

| MLX90614 | Air-Condition | Room Temperature |
|---|---|---|
| Standard Deviation | 0.15 | 0.26 |
| Error vs Reference IR Temperature (˚C) | -0.87 | -0.78 |
| Error vs. Reference Oral Temperature (˚C) | -1.27 | -1.18 |

The MLX90614 can be used in both environments in a relatively convincing manner. It features a low level of standard deviation in both air condition and room temperature. Moreover, in both conditions' reference IR temperature, it can obtain a reading with lower than -0.9˚C error in both environments, which allows easy implementation of correction factor if needed to recalibrate the sensor, especially since the low standard deviation indicates a high consistency level in the reading. The MLX90614 is therefore chosen as the sensor of choice for monitoring body temperature due to its consistency in the reading and acceptable level of errors that can be corrected via calibration in the final product. Moreover, as it is known to be used for medical purposes, the MLX90614 fits the need for a reliable, non-penetrable temperature reading sensor.

An experiment was done using DHT11 and DHT22 sensors to compare the accuracy of both sensors in measuring surrounding temperature and humidity. The DHT11 is the predecessor of the DHT22 with a lower cost but also at a smaller temperature range reading. The experiment was conducted in many different environments with two hygrometers used as reference. The average of the hygrometer reading across 15 minutes is computed and compared to the DHT11 and DHT22 sensors. The results are tabulated in Table 4, and the experimental testbed configuration is shown in Fig. 8. The DHT11 vs. DHT22 measured the surrounding temperature and humidity, and the results are compared with two reference hygrometers.

TABLE IV
DHT11 VS DHT22 TESTING RESULTS

| Location | P | H1 | H2 | Avg H | D11 | D22 | ΔDHT11 | ΔDHT22 |
|---|---|---|---|---|---|---|---|---|
| Room without air-conditioning or fan | Temp (˚C) | 31.3 | 31.7 | 31.5 | 33.0 | 32.0 | 1.5 | 0.5 |
| | Hum (%) | 65.0 | 66.0 | 65.5 | 66.0 | 64.5 | 0.5 | -1.0 |
| Room with fan | Temp (˚C) | 31.7 | 31.8 | 31.75 | 33.2 | 32.1 | 1.45 | 0.35 |
| | Hum (%) | 73.0 | 75.0 | 74.0 | 76.0 | 75.2 | 2.00 | 1.20 |
| Room with air-conditioning | Temp (˚C) | 28.8 | 29.1 | 28.95 | 30.1 | 29.1 | 1.15 | 0.15 |
| | Hum (%) | 51.0 | 50.0 | 50.50 | 51.0 | 51.7 | 0.50 | 1.20 |
| The car without air condition | Temp (˚C) | 37.9 | 38.9 | 38.4 | 46.4 | 43.8 | 8.0 | 5.4 |
| | Hum (%) | 38.0 | 35.0 | 36.5 | 32.0 | 37.6 | -4.5 | 1.1 |
| Car with air-conditioning | Temp (˚C) | 30.9 | 30.2 | 30.55 | 29.3 | 27.5 | -1.25 | 11.9 |
| | Hum (%) | 32.0 | 32.0 | 32.00 | 41.0 | 43.9 | 9.0 | -0.55 |
| Garden (morning) | Temp (˚C) | 26.5 | 26.0 | 26.25 | 26.4 | 25.7 | 0.15 | -0.55 |
| | Hum (%) | 83.0 | 77.0 | 80.00 | 85.0 | 82.3 | 5.00 | 2.30 |

Notes: Abbreviations used; P = Parameter, H1 = Hygrometer 1, H2 = Hygrometer 22, Avg H = Average Hygrometer, D11 = DHT11, D22 = DHT22, ΔDHT11 = Error of DHT11, ΔDHT22 = Error of DHT22

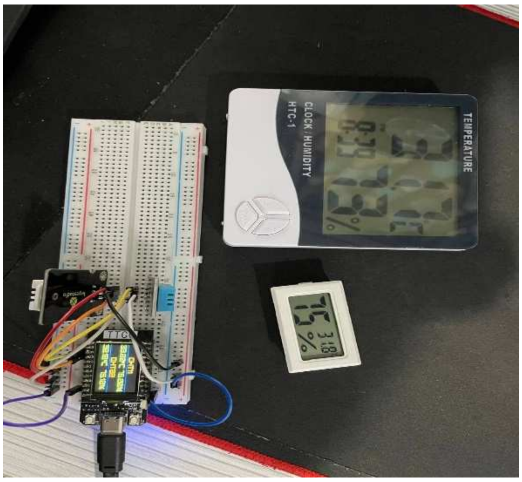


Fig. 8 The experimental testbed for DHT11 and DHT22

In general, the readings obtained by DHT11 and DHT22 sensors are both acceptable, although both sensors have had significant errors in certain circumstances. For DHT11, there was a significant error in the temperature (8˚C) in the car without an air-conditioned environment and humidity (9%) in the car with air-cond. In the latter environment, the DHT22 had a more significant error at 11.9% but this sensor had a lower maximum error for the temperature at only 5.4˚C in the car without an air-conditioned environment. On average, for temperature, the DHT22 had a lower error (0.47 ˚C) than DHT11 (1.83˚C) while in terms of humidity, the DHT11 average error is lower at 2.08% than the DHT22 sensor's 2.78% average error. In overall terms, the DHT22 has a lower error at 1.63 compared to DHT11's 1.96 reading. Due to the small price difference between the two sensors at only 2.46 USD, and the subtle temperature reading and humidity error, the

DHT22 is therefore chosen ahead of its predecessor as the environment sensor to be utilized in the health monitoring system.

### *B. Cough Detection System*

A raw dataset labeled cough and non-cough sounds is required for generating a machine-learning algorithm in Edge Impulse. The dataset includes various types of coughs and noises, recorded using Arduino Nano 33 BLE and smartphones, with an emphasis on distinguishing communication voice from coughs. The dataset is preprocessed, and a Keras neural network is trained using the generated features to classify cough and noise sounds. The total memory usage of the cough detection system developed on Edge Impulse is shown in Fig. 9. At 10.4 kB and 29.6 kB of peak RAM and ROM usage, respectively, the resource consumption is well within the 256 kB of RAM and 1 MB of RAM found on the nRF52840 microcontroller of the ESP32.

The cough detection algorithm is then deployed as an Arduino library header file that is imported into Arduino Nano 33 BLE cough detection file. The result is monitored through the Serial Monitor of the Platform IO interface. For every 2s, the Arduino will record a 1s period audio through its microphone to classify it as either cough or noise. If noise is detected, a '0' is printed on the serial monitor, whereas when a cough is detected, the line '1' is printed on the serial monitor. The integer displayed on the serial monitor indicates the integer cough flag that is sent to ESP32 when connected. The cough detection system deployed on the Arduino Nano 33 BLE is tested against other environmental noises - specifically, ones that usually occur within a quarantine room of an average person in solitary confinement - to test its robustness to noise within a quiet environment and the corresponding integer flag that it produces.


Fig. 9 The total memory usage of the cough detection system developed on Edge Impulse

### *C. ThingSpeak IoT Platform*

The ThingSpeak IoT platform features a dashboard that displays all sensors' data reading. The dashboard displays the reading in widget form, which shows the latest reading uploaded to ThingSpeak. In addition, medical staff and patients can view the data log of the reading throughout the 14-day quarantine period. The body temperature reading from MLX90614 is displayed in both gauge and line graphs. Surrounding temperature and humidity readings from DHT22 are displayed in the numerical widget and line graphs.

For cough events, a red icon widget lights up with each cough event, and a step chart is plotted with a '1' value to display the cough event and a '0' value to show that no coughs were heard. An additional widget displays the number of coughs over the past 24 hours. The last widget is for channel location, where the patient's location is plotted according to the longitude and latitude information entered. However, this position is not dynamic and does not automatically update according to the user's location.

The easy-to-use user interface makes ThingSpeak ideal for uploading and monitoring data over a long period. The ThingSpeak dashboard for Channel 1104440 (Patient 1) shown in Fig. 10 allows viewing of data log history and has multiple customizable widgets that offer an intuitive experience for the user and medical staff.

### *D. Alert System*

The incorporation of the free Twilio SMS feature with ThingHTTP and React by ThingSpeak allows a seamless SMS alert system to be implemented. The SMS alert is sent to the phone number registered under Twilio (the medical staff for actual implementation), and an alert is sent if ThingSpeak detects the patient's body temperature exceeding the threshold temperature of 37.5˚C. An alert is then sent mentioning the ThingSpeak Channel ID of the patient, the temperature reading, and the number of coughs over the past 24 hours. For e-mail alerts, it undergoes an almost similar process, except for using MATLAB Analysis (an application also under ThingSpeak) in place of the Twilio SMS service. The resulting e-mail alert then similarly displays the message to the SMS alert. These alert features are useful for medical staff to keep track of any abnormal health readings experienced by the patients. An example of the SMS and e-mail alerts sent to the medical staff is shown in Fig. 11. SMS and e-mail alerts are sent to medical staff when a patient's temperature exceeds the threshold of 37.5˚C. The alert message also specifies the temperature reading and the number of coughs experienced by the patient over the past 24 hours.

### *E. Final Prototype*

The final prototype circuit of the health monitoring system is soldered on a doughnut board and mounted onto a smartphone armband which can also fit a power bank. The MLX90614 is mounted on the wrist for the user to measure their body temperature through the wrist comfortably.

TABLE V
COUGH DETECTION RESULTS

| | **Environment Sound** | **Cough Integer Flag** | | | |
|---|---|---|---|---|---|
| | | Expected | 1st Attempt | 2nd Attempt | 3rd Attempt |
| 1 | Single Cough | 1 | 0 | 1 | 1 |
| 2 | Double Cough | 1 | 1 | 0 | 0 |
| 3 | Triple Cough or more | 1 | 1 | 1 | 1 |
| 4 | Keyboard typing and mouse clicking | 0 | 0 | 0 | 0 |
| 5 | Table hitting | 0 | 0 | 0 | 0 |
| 6 | Chair Pulling | 0 | 0 | 0 | 0 |
| 7 | Door Closing | 0 | 0 | 1 | 0 |
| 8 | Monologuing while working in slow volume | 0 | 0 | 0 | 0 |
| 9 | Guitar Strumming | 0 | 1 | 0 | 0 |
| 10 | Phone Conversation | 0 | 0 | 1 | 0 |
| 11 | Phone ringtone | 0 | 0 | 0 | 0 |
| 12 | Computers Speakers | 0 | 0 | 0 | 0 |

Fig. 10 ThingSpeak dashboard for Channel 1104440 (Patient 1)

This prototype design provides mobility to the users when using the device as it does not confine the users to stay at a particular spot throughout their quarantine period. The total cost of the system components and prototype equates to around 67.99 USD. As the system is reusable for the most part at up to 148 patients, the average cost is 0.46 USD per user, achieving a very low cost in the bigger picture. Regarding total power usage, 24-hour usage testing was conducted with the whole system powered by a 10,000 mAh power bank equipped with a battery-level indicator. The 24-hour testing yields 23% battery usage, equal to 2300mAh. When divided by 24 hours, it indicates that the whole system uses 95.83mA in total. The low power usage allows the system to be used continuously for slightly over four days with a 10,000mAh power bank before requiring charging.

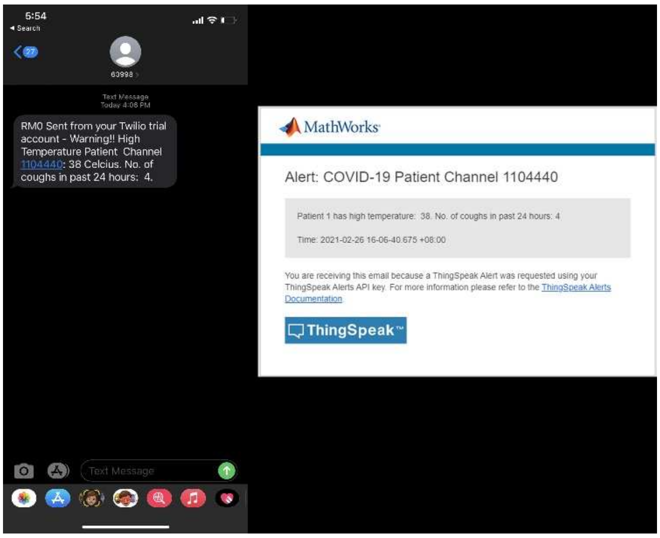

Fig. 11 SMS and e-mail alerts are sent to medical staff when a patient's temperature exceeds the threshold of 37.5˚C

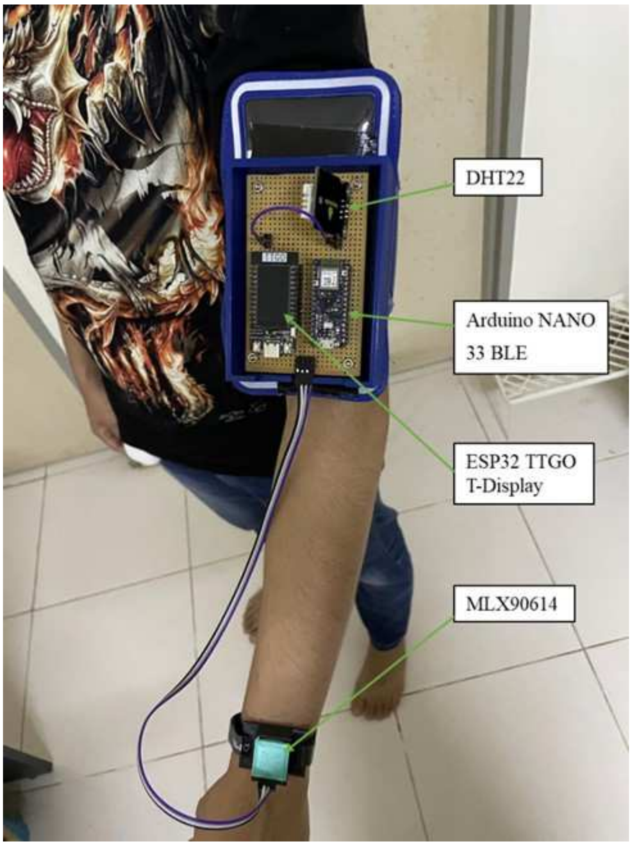

Fig. 12 The final prototype for COVID-19 IoT Health Monitoring System is enclosed in 3D-printed housing and attached to a smartphone armband

## IV. Conclusion

The project has managed to fulfill its three main objectives, which are to present a wearable device capable of monitoring COVID-19 symptoms, have an alert system for medical staff to be aware of any possible COVID-19 symptoms shown by the patient and achieve a low total cost to present home quarantine as a viable option compared to undergoing health monitoring at quarantine centers. In future work, it is recommended to further enhance the cough detection feature to allow more robust detection. A geofence system with alerts can be introduced to monitor and prevent patients from wandering beyond their quarantine area. Lastly, a group analysis feature that will allow all project users to be unified under a single system can be presented to allow large-scale analysis by medical staff and allow them to study more on the disease for the long-term healthcare of society.


## ACKNOWLEDGMENT

This research project is sponsored by Research Grant, MMUE/220061, Multimedia University.


## REFERENCES


[1] V. Chamola, V. Hassija, V. Gupta, and M. Guizani, "A comprehensive review of the COVID-19 pandemic and the role of IoT, drones, AI, blockchain, and 5G in managing its impact," IEEE Access, vol. 8, no. April, pp. 90225–90265, 2020, doi: 10.1109/ACCESS.2020.2992341.

[2] B. Farahani, F. Firouzi, and M. Luecking, "The convergence of IoT and distributed ledger technologies (DLT): Opportunities, challenges, and solutions," J. Netw. Comput. Appl., vol. 177, no. September 2020, p. 102936, 2021, doi: 10.1016/j.jnca.2020.102936.

[3] Z. Chang, S. Liu, X. Xiong, Z. Cai, and G. Tu, "A survey of recent advances in edge-computing-powered artificial intelligence of things," IEEE Internet Things J., vol. 8, no. 18, pp. 13849–13875, 2021, doi: 10.1109/JIOT.2021.3088875.

[4] A. Barnawi, P. Chhikara, R. Tekchandani, N. Kumar, and B. Alzahrani, "Artificial intelligence-enabled Internet of things-based system for COVID-19 screening using aerial thermal imaging," Futur. Gener. Comput. Syst., vol. 124, pp. 119–132, 2021, doi: 10.1016/j.future.2021.05.019.

[5] S. Shriram, B. Nagaraj, J. Jaya, S. Shankar, and P. Ajay, "Deep learning-based real-time AI virtual mouse system using computer vision to avoid COVID-19 spread," J. Healthc. Eng., vol. 2021, 2021, doi: 10.1155/2021/8133076.

[6] Defni, - Johan, A. E. Putra, F. Nova, and W. Andriani, "Utilizing requirement testing methods on web-based swab data information system," Int. J. Adv. Sci. Comput. Eng., vol. 4, no. 1, pp. 22–31, Mar. 2022, doi: 10.30630/ijasce.4.1.75.

[7] S. Jain et al., "Internet of medical things (IoMT)-integrated biosensors for point-of-care testing of infectious diseases," Biosens. Bioelectron., vol. 179, no. September 2020, p. 113074, 2021, doi: 10.1016/j.bios.2021.113074.

[8] N. Y. Philip, J. J. P. C. Rodrigues, H. Wang, S. J. Fong, and J. Chen, "Internet of things for in-home health monitoring systems: Current Advances, Challenges and Future Directions," IEEE J. Sel. Areas Commun., vol. 39, no. 2, pp. 300–310, 2021, doi: 10.1109/JSAC.2020.3042421.

[9] X. Li, Y. Lu, X. Fu, and Y. Qi, "Building the Internet of things platform for smart maternal healthcare services with wearable devices and cloud computing," Futur. Gener. Comput. Syst., vol. 118, pp. 282–296, 2021, doi: 10.1016/j.future.2021.01.016.

[10] B. Pradhan, S. Bhattacharyya, and K. Pal, "IoT-based applications in healthcare devices," J. Healthc. Eng., vol. 2021, 2021, doi: 10.1155/2021/6632599.

[11] I. D. M. B. Filho, G. Aquino, R. S. Malaquias, G. Girao, and S. R. M. Melo, "An IoT-based healthcare platform for patients in ICU beds during the COVID-19 outbreak," IEEE Access, vol. 9, pp. 27262–27277, 2021, doi: 10.1109/ACCESS.2021.3058448.

[12] W. R. Malatji, R. VanEck, and T. Zuva, "A systematic review of the adoption of eHealth cloud-based technology applications during COVID-19," Int. J. Adv. Sci. Comput. Eng., vol. 3, no. 2, pp. 53–64, Oct. 2021, doi: 10.30630/ijasce.3.2.44.

[13] M. Alkhodari et al., "Screening cardiovascular autonomic neuropathy in diabetic patients with microvascular complications using machine learning: a 24-hour heart rate variability study," IEEE Access, vol. 9, pp. 119171–119187, 2021, doi: 10.1109/ACCESS.2021.3107687.

[14] S. K. Mohamed, N. A. Sakr, and N. A. Hikal, "A review of breast cancer classification and detection techniques," Int. J. Adv. Sci. Comput. Eng., vol. 3, no. 3, pp. 128–139, Oct. 2021, doi: 10.30630/ijasce.3.3.55.

[15] L. Brunese, F. Mercaldo, A. Reginelli, and A. Santone, "Explainable deep learning for pulmonary disease and coronavirus COVID-19 detection from X-rays," Comput. Methods Programs Biomed., vol. 196, p. 105608, Nov. 2020, doi: 10.1016/j.cmpb.2020.105608.

[16] G. Guglielmi, "Fast coronavirus tests: what they can and can't do," Nature, vol. 585, no. 7826, pp. 496–498, Sep. 2020, doi: 10.1038/d41586-020-02661-2.

[17] J. Dunn et al., "Wearable sensors enable personalized predictions of clinical laboratory measurements," Nat. Med., vol. 27, no. 6, pp. 1105–1112, 2021, doi: 10.1038/s41591-021-01339-0.

[18] S. K., P. A. S., P. U., P. S., and S. P. V., "Patient health monitoring system using IoT," Mater. Today Proc., vol. 80, pp. 2228–2231, 2023, doi: 10.1016/j.matpr.2021.06.188.

[19] N. Chinnamadha, R. Z. Ahmed, and K. Kalegowda, "Development of health monitoring system using smart intelligent device," Indones. J. Electr. Eng. Comput. Sci., vol. 28, no. 3, pp. 1381–1387, 2022, doi: 10.11591/ijeecs.v28.i3.pp1381-1387.

[20] N. A. Ahmad Yani and M. F. Zolkipli, "Computerized senior citizen health monitoring using mobile application," Int. J. Adv. Sci. Comput. Eng., vol. 3, no. 3, pp. 140–152, Oct. 2021, doi: 10.30630/ijasce.3.3.56.

[21] S. Karamehic, "Analysis of Covid-19 daily results and information about patients using SQL and PowerBi," Int. J. Data Sci., vol. 4, no. 1, pp. 1–9, May 2023, doi: 10.18517/ijods.4.1.1-9.2023.

[22] M. R. Islam, M. M. Kabir, M. F. Mridha, S. Alfarhood, M. Safran, and D. Che, "Deep learning-based IoT system for remote monitoring and early detection of health issues in real-time," Sensors, vol. 23, no. 11, pp. 1–18, 2023, doi: 10.3390/s23115204.

[23] A. N. Ejin, H. T. Yew, M. Mamat, F. Wong, A. Chekima, and S. K. Chung, "Internet of things based real-time coronavirus 2019 disease patient health monitoring system," Int. J. Electr. Comput. Eng., vol. 12, no. 6, pp. 6806–6819, 2022, doi: 10.11591/ijece.v12i6.pp6806-6819.

[24] C. Sen Chan, W. L. Pang, K. Y. Chan, and A. S. T. Ng, "Design of an IoT safety distance monitoring device for COVID-19," in Lecture Notes in Electrical Engineering, vol. 835, 2022, pp. 261–272. doi: 10.1007/978-981-16-8515-6_21.

[25] C. E. You, W. L. Pang, and K. Y. Chan, "AI-based low-cost real-time face mask detection and health status monitoring system for COVID-19 prevention," WSEAS Trans. Inf. Sci. Appl., vol. 19, pp. 256–263, Nov. 2022, doi: 10.37394/23209.2022.19.26.

[26] R. Sanchez-Iborra and A. F. Skarmeta, "TinyML-enabled frugal smart objects: challenges and opportunities," IEEE Circuits Syst. Mag., vol. 20, no. 3, pp. 4–18, 2020, doi: 10.1109/MCAS.2020.3005467.

[27] C. Bales et al., "Can machine learning be used to recognize and diagnose coughs?," in 2020 International Conference on e-Health and Bioengineering (EHB), Oct. 2020, pp. 1–4. doi: 10.1109/EHB50910.2020.9280115.

[28] M. You, W. Wang, Y. Li, J. Liu, X. Xu, and Z. Qiu, "Automatic cough detection from realistic audio recordings using C-BiLSTM with boundary regression," Biomed. Signal Process. Control, vol. 72, no. PA, p. 103304, 2022, doi: 10.1016/j.bspc.2021.103304.

[29] V. Bansal, G. Pahwa, and N. Kannan, "Cough classification for COVID-19 based on audio mfcc features using Convolutional Neural Networks," in 2020 IEEE International Conference on Computing, Power and Communication Technologies (GUCON), Oct. 2020, pp. 604–608. doi: 10.1109/GUCON48875.2020.9231094.

[30] A. Dasari, A. Revanur, L. A. Jeni, and C. S. Tucker, "Video-based elevated skin temperature detection," IEEE Trans. Biomed. Eng., vol. 70, no. 8, pp. 2430–2444, Aug. 2023, doi: 10.1109/TBME.2023.3247910.

[31] P. E. Aylwin, S. Racinais, S. Bermon, A. Lloyd, S. Hodder, and G. Havenith, "The use of infrared thermography for the dynamic measurement of skin temperature of moving athletes during competition; methodological issues," Physiol. Meas., vol. 42, no. 8, p. 084004, Aug. 2021, doi: 10.1088/1361-6579/ac1872.