International Journal of Communication Networks and Information Security
ISSN: 2073-607X, 2076-0930
Volume 15 Issue 04 Year 2023

IoT-Based Biometric Attendance System Using Arduino and ThingsBoard

Jarudin*
Doctor, Department of Information Engineering, Institut Teknologi dan Bisnis Bina Sarana Global, Indonesia
jarudin@global.ac.id
Prajka Ahmad Raihan
S.T, Department of Mechanical Engineering, Universitas Negeri Jakarta, Indonesia
prajkaahmad@gmail.com
Rahmat Tullah
Master, Department of Information Engineering, Institut Teknologi dan Bisnis Bina Sarana Global, Indonesia
rahmatullah@global.ac.id
M. Ramaddan Julianti
Ph.D. Candidate, Department of Business Digital, Institut Teknologi dan Bisnis Bina Sarana Global, Indonesia
mramaddanjulianti@global.ac.id
Syaipul Ramdhan
Master, Department of Business Digital, Institut Teknologi dan Bisnis Bina Sarana Global, Indonesia
syaipulramdhan@global.ac.id
M. Fitriansyah AK
S. Kom(Degree of Computer), Department of Information Engineering, Institut Teknologi dan Bisnis Bina Sarana Global, Indonesia
mfitriansyahak@global.ac.id
Sarah Fazilla
Ph.D. Candidate, Postgraduate University of Medan, Indonesia, Indonesia
sarahfazila@iainlhokseumawe.ac.id

Article History

Received: 2 October 2023
Revised: 17 November 2023
Accepted: 19 December 2023

Abstract

The fingerprint sensor is a sensor that detects fingerprints using an optical system, where detection is done by reading the contours of the fingerprints and the static electricity of the body. However, data generated by fingerprint sensors, in general, can only be accessed if connected directly to the fingerprint module. From these conditions, operational managers or business managers can't monitor the absence of discipline of their employees because attendance data cannot be accessed directly. They must go through the download process from the machine's default software. The purpose of this research is to build a fingerprint recognition system in the context of an abscess machine on an IoT basis so that fingerprint data processing is centralized so that it can be easily accessed without having to connect directly with the fingerprint sensor module that is available by implementing the client-server method. The test results of this study indicate that collaboration between the fingerprint sensor module integrated with the Arduino Uno module and the ThingsBoard IoT platform can be done with a fingerprint reading accuracy of 96.25%, and data can be accessed

CC License CC-BY-NC-SA 4.0	in real time through the ThingsBoard server. ***Keywords: Attendance, Fingerprint, IOT, Biometric, Arduino Uno, ThingsBoard***

1. Introduction

The rapid development of science and technology is very influential in human life. This can be seen with all the conveniences offered and provided. In terms of checking identification, for example, in addition to the many methods that can be used to distinguish a person's identity from others, with the development of computer technology, it is hoped that there is a design of a system that allows time spent on someone in terms of checking identification can be more efficient. The identification check itself is needed to increase [1] the level of security [2], for example, during absences. However, there are several obstacles to data processing from the identification [3] of fingerprint sensors. Data generated by fingerprint sensors, in general, can only be accessed if connected directly to the fingerprint module. This also applies to fingerprint attendance machines in general that uses a fingerprint sensor. For fingerprint attendance machines currently used at the PT Financial Multi Finance Tangerang Office, the process of withdrawing attendance data is done through default software from the attendance machine vendor installed on a PC connected directly to the attendance machine. If the operational manager or business manager requires employee attendance data, they must request the staff of the personal admin or admin's head. Requests can be made quickly if the relevant staff is in the office. If the related staff is not in the office, it takes time for the related staff to return to the office. From these conditions, operational managers or business managers can't monitor the absence of discipline of their employees because attendance data cannot be accessed directly and must go through the download process from the machine's default software. From these constraints, it can be identified that the fingerprint identification data for attendance needs cannot be directly accessed if it is not directly connected to the attendance machine and if it does not use the default software from the attendance machine vendor. Therefore, the time attendance data cannot be accessed in real-time because no client-server feature on the attendance machine is used.

Based on this background, it can be formulated that there is a need to develop the existing fingerprint biometric attendance machine model using the Arduino Uno module as a microcontroller integrated with the ethernet shield to be able to connect with the IoT ThingsBoard platform as a time attendance data storage by implementing [4] the client-server method, in where the client sends requests to the server and with server resources provides computing for many client components, fulfilling client requests [5], [6]. The protocol used is the MQTT protocol for communication with the publish/subscribe system [7]. This protocol has low packet overhead data size and small power consumption, which makes it ideal for use in Machine-to-Machine (M2M) communication and the Internet-of-Things (IoT) context [8], [9]. IoT is a concept of utilizing internet connectivity that is always connected at all times to connect one device to another over the internet in the hope that the system can help the user in carrying out a task [10], [11], [12].

The development of the attendance machine model designed in this study is only in the form of design (prototype) and the presentation of attendance data derived from the identification of integrated fingerprint sensor Arduino Uno modules only utilizing the features available on the IoT ThingsBoard platform. ThingsBoard is a popular open-source IoT software for device management, data collection, processing, and visualization [13],[14],[15],[16]. Biometrics, in general, is a study of measurable biological characteristics [17]. In information technology, biometrics is relevant to the technology used to conduct physical analysis and human behavior in the authentication process [18], [19]. Biometric identifiers are unique, and measurable characteristics are used to identify individuals. Fingerprint biometrics is an identification technology with physical characters in fingerprints. This type of biometrics is popularly used because it has good performance, a unique uniqueness, and is permanent, lifelong inherent in someone.

Fingerprints are the reproduction of fingerprints, whether intentionally taken, stamped with ink, or marks left on an object because it was once touched on the skin of the palm or foot [20]. The fingerprint is one of the unique parts of the human body [21]. With fingerprints, humans have a

unique identity that distinguishes them from other people, and this identity can be accessed easily [22]. Fingerprints prove to be entirely accurate, easy, comfortable, and safe compared to other human identity recognition systems [23]. At present, the developing technology is fingerprint recognition using the fingerprint sensor. A fingerprint sensor is a sensor that detects fingerprints by utilizing an optical system, where the detection process is done by reading the contours of fingerprints and static electricity that exists in the human body [23], [24]. This results in a high level of security because it cannot be falsified with artificial fingerprints or fingerprint copies [25], [26].

Fingerprint sensors have been widely used in several previous studies. The fingerprint sensor was used for attendance systems using the MySQL database for attendance data storage [27]. The fingerprint sensor was collaborated with an ethernet module to record fingerprints and store them on a database server built with Raspberry PI [28]. The result is an attendance module consisting of the Arduino Uno module, RTC, fingerprint sensor, Raspberry PI 3, UI, and data storage. The concept is the same as the scanning and matching process, except that the log data is sent via ethernet protocol to the raspberry PI 3 module. The user can access the local web server on the Raspberry PI 3 for his attendance data dashboard.

Meanwhile, in Hoo and Ibrahim's [29] research, the fingerprint sensor created a fingerprint attendance system using SD Card storage media. The result is an attendance module consisting of an Arduino Uno module, fingerprint sensor, RTC, 16x2 LCD and SD Card Module [30]. When the user performs a scan, the flow is recognized if it is recognized, the LCD will display the name according to what is stored in the program and save the log to the SD Card. If it is not recognized, the LCD will not display anything. The weakness is that the storage is still local storage, where the data storage is attached to the attendance device. It has not implemented the IoT concept.

Arduino Uno and ThingsBoard are expected to provide alternative technologies that can be used in fingerprint attendance where the data can be centralized so that it can be easily accessed without being connected directly to a physical device so that attendance data can be seen in real time[31]. ThingsBoard provides device management, data collection, processing, and visualization, making it easier for end-users to get better data. Data visualization, real-time and remote device control, customizable settings, plugins, widgets, and transport system implementations, can monitor attributes from the client and server sides, supporting transport encryption for MQTT and HTTP protocols, and experienced nodes damage can be replaced without downtime. This certainly eases the duties of Personal Admins and Chief Admin Staff in managing attendance data, because the data can be downloaded anytime and anywhere.

2. Related Work

IoT-based biometric attendance systems using Arduino Uno and ThingsBoard offer a modern and secure approach to employee or student timekeeping. Functionality: Fingerprint scanning: Most systems utilize fingerprint sensors for user identification, providing a tamper-proof and unique biometric identifier [32], [33]. Cloud storage: Attendance data is securely stored on cloud platforms like ThingsBoard, enabling remote access and eliminating reliance on local storage [34], [35]. Real-time monitoring: Systems often display attendance data in real-time on dashboards, allowing for immediate oversight and management [36]. Enhanced security: Biometrics provide a more reliable and secure method of identification compared to traditional methods like cards or codes.

IoT Based Biometric Attendance System Benefits:

1. Improved accuracy: Fingerprint recognition minimizes errors and eliminates buddy punching, ensuring accurate attendance records.

2. Remote accessibility: Cloud storage allows for easy access to attendance data from anywhere, facilitating data analysis and reporting.

3. Scalability: These systems can be readily adapted to accommodate different user sizes and locations.

IoT-Based Biometric Attendance System Challenges and Considerations:

1. Cost: Implementing and maintaining an IoT-based system may involve higher upfront costs compared to traditional methods.

2. Complexity: Setting up and configuring the hardware and software components requires some technical expertise.

3. Privacy concerns: Storing biometric data raises privacy concerns, necessitating robust data security measures and user consent protocols.

Overall, IoT-based biometric attendance systems using Arduino Uno and ThingsBoard offer a promising solution for organizations seeking to improve the accuracy, security, and efficiency of their timekeeping practices [26], [37], [38]. However, careful consideration of the costs, complexity, and privacy implications is essential before implementation.

3. Methodology

IoT-based attendance system development methodology uses Arduino and Thingsboard using prototype methodology. The steps taken are as follows:

- Analyze the IoT-based fingerprint attendance system process flow in data transmission flow between 2 Arduino Uno modules and send data to the Thingsboard server.
- Analyze the fingerprint identification work scheme using a fingerprint sensor integrated with the Arduino module for the implementation of biometric mechanisms for the enrollment phase, where input will be scanned by the sensor and represent digital characteristics, phase matching where data is in the database will be matched with the identification data, and the identification phase (introduction) [39].
- Analyze the integration scheme of the Ethernet Shield Module with the Arduino Uno module as a network connection feature to connect to the ThingsBoard server.
- Determine supporting components for communication between sensors with Arduino Uno, communication between Arduino Uno I (master) and Arduino Uno II (slave), components for notifications, and components for display.
- Make wiring diagrams to illustrate the overall design scheme of the tool to be built.
- Designing a program code to operate the device to be built.
- Build a prototype following the wiring diagram; see Figure 1.

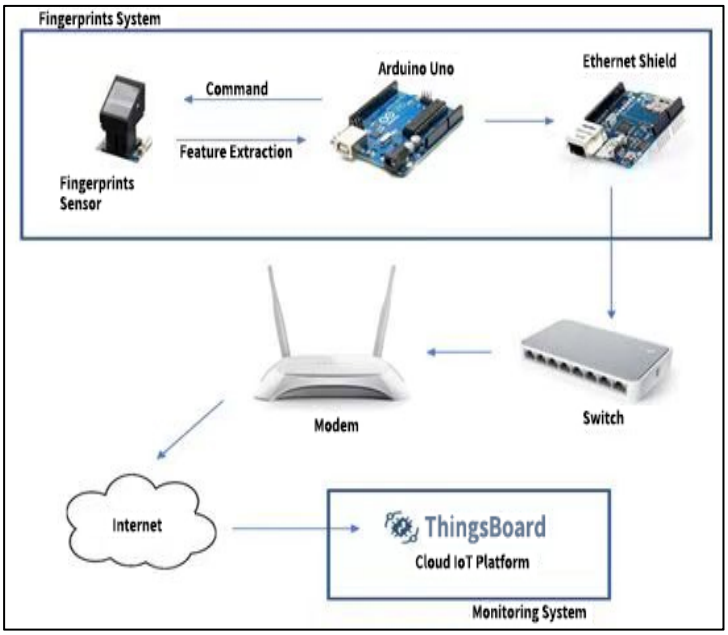


Figure 1. Architecture Design

Arduino Uno master is used to process the results of fingerprint identification from the fingerprint sensor. The LCD module is used to display the conditions that occur in the system. When Arduino Uno performs a fingerprint scan and gets a return from the fingerprint module, Arduino Uno will give a command (as written in the sketch) to the LCD module to display the scan results. For additional interaction, the buzzer component is also used as an additional notification for the user.

The fingerprint data that has been processed by the Arduino Uno Master, which has passed the matching procedure, will be sent to the ThingsBoard server via serial communication to the Arduino

Slave. On the Arduino Uno Slave, the fingerprint data will be converted into JSON format and sent directly with the MQTT protocol to the ThingsBoard telemetry server. For this reason, the Arduino Uno Slave is integrated with an ethernet shield. The Ethernet shield provides network facilities that Arduino Uno can use to access local networks or the internet or other Arduino Uno modules utilizing an ethernet cable for data transmission. Because it is in a shield form, the Ethernet Shield module can be installed directly on the Arduino Uno module board. To find out the connection status of the ethernet module and the connection to the ThingsBoard IoT server, a 128x64 OLED module is used can be seen in Figure 2.

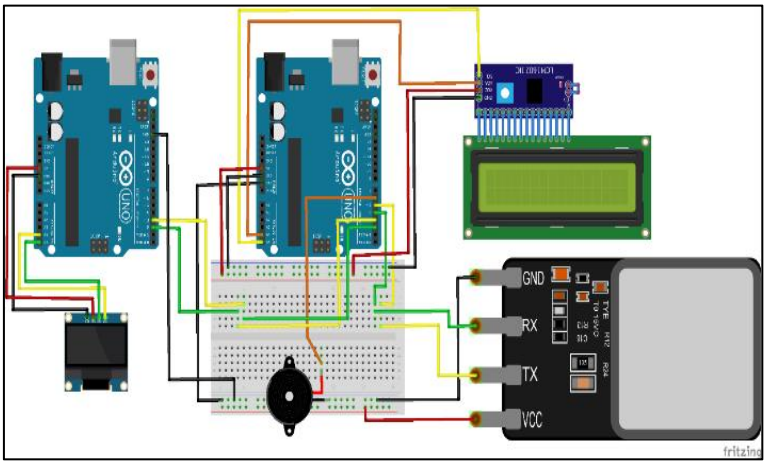


Figure 2. IoT-Based Fingerprint Time Attendance Diagram Using Arduino Uno and ThingsBoard

From the wiring diagram above, it can be explained that (1) Arduino Uno is the main component in the tool built because it functions as the control centre for all components used. There are two Arduino Uno used in this study. The first Arduino Uno (Arduino Master) controls fingerprint sensor components, LCDs, and active buzzers. The second Arduino (Arduino Slave) controls the Ethernet shield and OLED display. This is because the SRAM capacity of Arduino Uno is 2KB in size, so it requires 2 Arduino Uno to handle all the necessary components, (2) Ethernet Shield functions to connect Arduino Uno to the internet network with ethernet cable (LAN) media. Ethernet shield based on Wiznet W5100 chip and using the library in writing code for its interaction with Arduino Uno [40], (3) Fingerprint sensor is used to identify fingerprints for enrolment or scanning purposes, which is used in this study is the FPM10A DY50 model. The fingerprint sensor (4) LCD Display used in this study has been integrated with the IIC module to save pin usage. LCD with similar techniques will require six pins, whereas, with IIC, LCDs only require four pins. LCDs are used as display outputs for systems made, (5) OLED displays are used as network status display outputs, (6) The buzzer used in this study is the active buzzer type as a sound source for notification of certain events, (7) Breadboard, with utilizing breadboard, electronic components used will not be damaged due to the soldering and disordering process.

Because many modules are integrated, and there are many libraries to be used, 2 Arduino Uno modules are needed to accommodate all the modules to be integrated. This is because the memory usage for the final sketch using one Arduino Uno module reaches 83% of the total memory available can be seen in Figure 3.

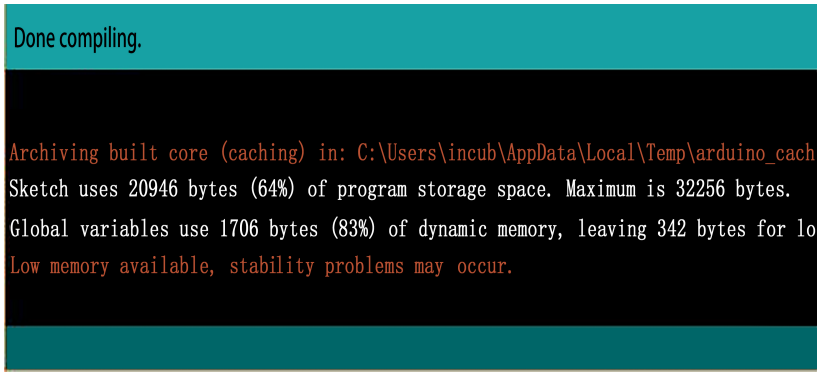


Figure 3. Log Performance

From this, the authors share the integration of the device using two Arduino Uno's. The Arduino Uno, which the author made as a master, is integrated with the fingerprint, buzzer, and LCD module. Meanwhile, the Arduino Uno I made as a slave integrates with the Ethernet Shield and OLED modules for integration between the Arduino Uno master and Arduino Uno slave using serial pins.

3.1 Experimental Setup

At the implementation stage, the design that has been prepared beforehand is built into a series of finished circuits and is supported by code written in the Arduino IDE environment as a rule that runs a series of tools. The programming language used is C as a compiler language. The finished circuit of the built device can be seen in Figure 4.

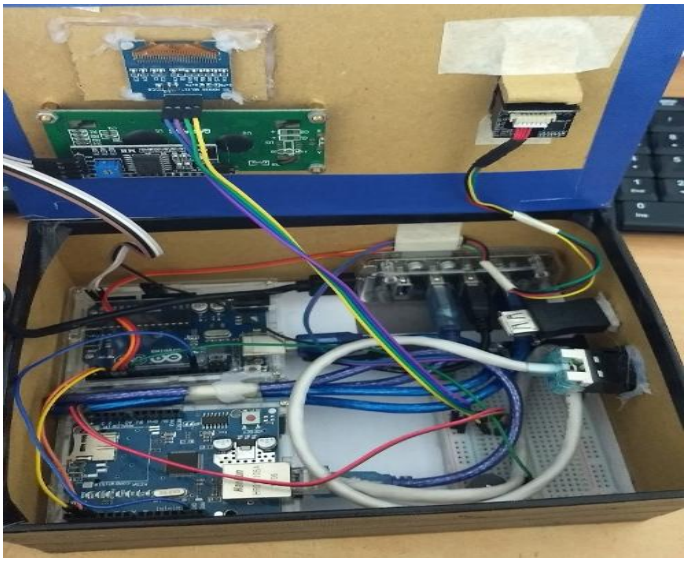


Figure 4. The Fingerprint Attendance Machine Series

Because this study uses the ethernet shield as a network connection module, a modular RJ45 connector as a connecting port between the LAN cable from the network server and the LAN port of the ethernet shield module. For connectors between non-shield-shaped components, male-to-male and female-to-male jumper cables are used, both for those leading to the Arduino Uno module and those heading to the breadboard.

4. Results and Discussion

4.1 Fingerprint Sensor Testing

The fingerprint sensor in this project uses the FPM10A DY50 model. The power used is 3.3v. Integration with the Arduino Uno module using the Adafruit_Fingerprint. Library. A fingerprint sensor is detected when the device is running. The system will proceed to the next procedure if a sensor is found. If a sensor is not found, the system will continue scanning until the fingerprint sensor is found. The program code used for the matching and fingerprint enrolment process follows the sample sketch in the Adafruit_Fingerprint. Library with modifications according to the procedures that have been designed.

4.2 Attendance Mode

By default, the fingerprint sensor is programmed in attendance mode when the device is on. In this mode, the results of fingerprint scanning will go through matching the fingerprint image that was successfully captured with a fingerprint template stored in the fingerprint sensor flash memory. If similarities are found, the scan has been successfully carried out. For attendance purposes, fingerprints whose ID numbers have been identified will be matched back to the attendance log in the form of arrays that have been prepared on the side of the code. If the log for the fingerprint that has been identified is not found, then the attendance status for the fingerprint is "In." If a log for the identified fingerprint is found, the attendance status for the fingerprint is "Out," and the system will delete the log for the fingerprint from the array.

Fingerprint sensor testing in the scanning process starts from the information displayed on the LCD after the loading process is complete; as explained at the implementation stage, the tool's default mode is attendance mode. The LCD will display the information "ATTENDANCE MODE, Press Finger ..." as information that the device is ready to be used for attendance, see Figure 5.

Figure 5. LCD Display Attendance Mode Display

When the user taps the fingerprint sensor and is verified as absent, the system will display information to the user via the LCD Display that the user has successfully logged in. When the user taps the fingerprint sensor and is verified as absent out, the system will display information to the user through the LCD Display that the user has successfully logged out. In other conditions, when the user's fingerprint fails to be verified, the system will display information to the user through the LCD Display that the fingerprint is not known, seen in Figure 6.

Finger 6. LCD Display Display Matching Process Fails

The accuracy of the project's fingerprint module scanning process can be seen in Table 1.

Table 1. Testing the Accuracy of the Fingerprint Module Scanning Process

ID USER	SCANNING									
	1	2	3	4	5	6	7	8	9	10
2	√	√	√	√	√	x	√	√	√	√
3	√	√	√	√	√	√	√	√	√	√
4	√	√	√	√	√	x	√	√	√	√
5	√	√	√	√	√	√	√	√	√	√
6	√	√	√	√	√	√	√	√	√	√
7	√	√	√	√	√	√	√	√	√	√
8	√	√	√	√	√	√	x	√	√	√
11	√	√	√	√	√	√	√	√	√	√

If the accuracy testing data above is presented in graphical form, it will look like Figure 7.

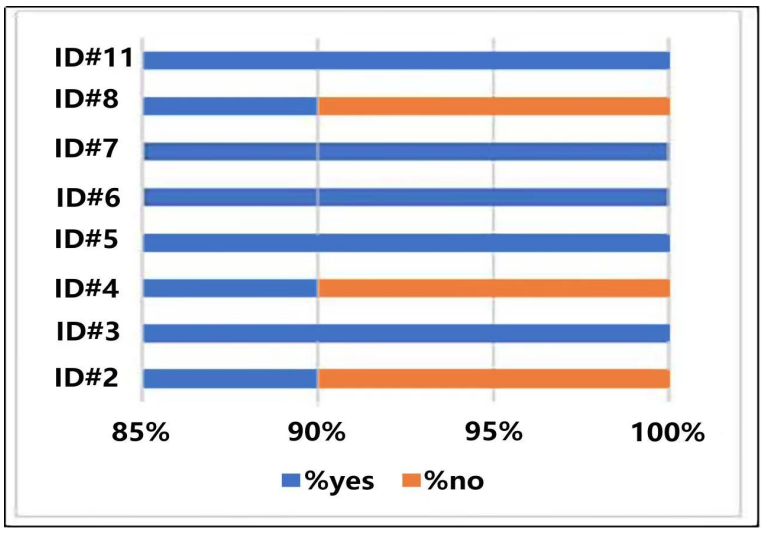


Figure 7. Graph of Testing the Accuracy of the Fingerprint Module Scanning Process

Based on the test data above, as many as three fingerprints could not be identified, so the accuracy level of the fingerprint module used in this project was 96.25%. The results obtained from the calculation are shown as formulation 3.1.

$$\frac{\text{Number of fingerprints identified}}{\text{Total amount of fingerprint test data}} \text{ x } 100\% \qquad (3.1)$$

4.3 Enrol Mode

For the enrollment process, a fingerprint template is needed as an admin. For this reason, before the final sketch is uploaded to the Arduino Uno module, the sample sketch enrollment available in the Adafruit_Fingerprint.h library is uploaded first to register one fingerprint as a fingerprint template, which will trigger the enrollment mode. The fingerprint template is admin for the registration process, done from the code side. The fingerprint enrollment procedure in this project is that if a fingerprint is registered as an admin during attendance mode, the system will activate the enrollment function for one cycle. If one fingerprint has been successfully registered, the admin fingerprint scanning is needed again to register another fingerprint. The registered ID is programmed to be auto incremented or automatic numbering according to the last ID on the fingerprint sensor flash memory. As explained earlier, when fingerprints registered as admin are detected (found ID # 1 with the confidence of 79), the system will go into enrollment mode and count the number of fingerprint templates stored (in a log reading ten templates) and provide the ID of the auto-increment result (waiting for a right finger to enroll as # 11) for the new fingerprint. The enrollment process in the log can be seen from the line "Image taken" to the line "Stored!", see Figure 8.

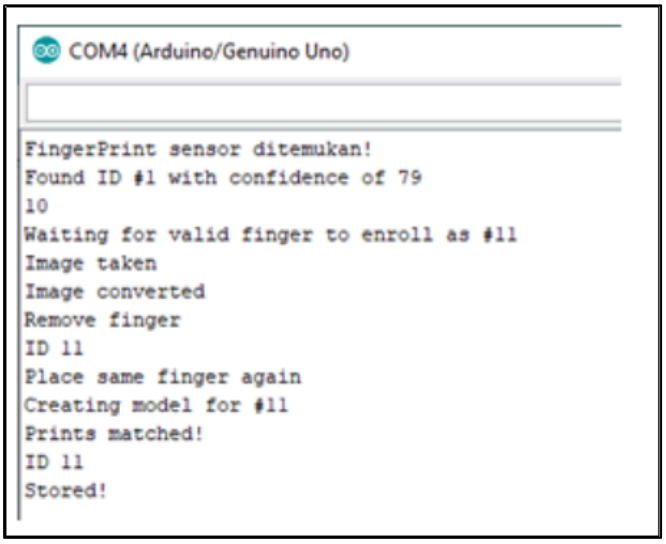


Figure 8. Serial Log Monitor Fingerprint Sensor Enroll Process

As explained earlier, if fingerprints are identified as admin, then the system will switch to enrol mode. The information on the LCD when the admin fingerprint is detected is as in Figure 9.

Figure 9. Display LCD When Admin Fingerprint is Detected

At the time of enrollment mode, each fingerprint identified will go through the process of conversion by the system into a fingerprint template and stored in a fingerprint flash memory module with the ID number printed on the LCD, as shown in Figure 10.

Figure 10. LCD when the Enroll Process is Successful

4.4 Testing the Ethernet Shield Module

In this study, the ethernet shield module has the role of providing an internet connection as a transmission medium for the Message Queueing Telemetry Transport (MQTT) protocol to send attendance data to the Thingsboard server as its data centre. The Ethernet Shield and Arduino Uno integration is done by calling the Ethernet. H and SPI.h libraries on the Arduino sketch. The network configuration used is DHCP. Arduino Uno connection procedure with the network is after the ethernet shield is initialized, the system will make a bind IP request to the DHCP server. If the binding is successful, the system will apply the IP provided by the DHCP server to the ethernet shield module. If the binding fails, the system will apply IP localhost (127.0.0.1) to the ethernet shield module and repeat the bind IP request process until the binding process is successful.

For MQTT transmission for sending data to the Halboard telemetry server, the PubSubClient.h library is used. The procedure for sending data with the MQTT protocol is that after the system successfully binds with the DHCP server and the MQTT protocol is initialized; the system will start the handshaking process with the Halboard telemetry server by using access tokens from devices already registered on the Halboard server. If the handshaking process is successful, the system will send data to the Thingsboard server. The data sent to the telemetry server is a JSON payload containing Name, Status, and Device ID. If the payload is successfully published, the data on the payload will appear in the "Latest Telemetry" panel on the Thingsboard server. If the handshaking process fails, the system will activate the reconnect function to check the internet connection status and re-handshake the Thingsboard server. The system will recall the reconnect function if the handshake process fails, even if the internet connection is online seen in Figure 11.

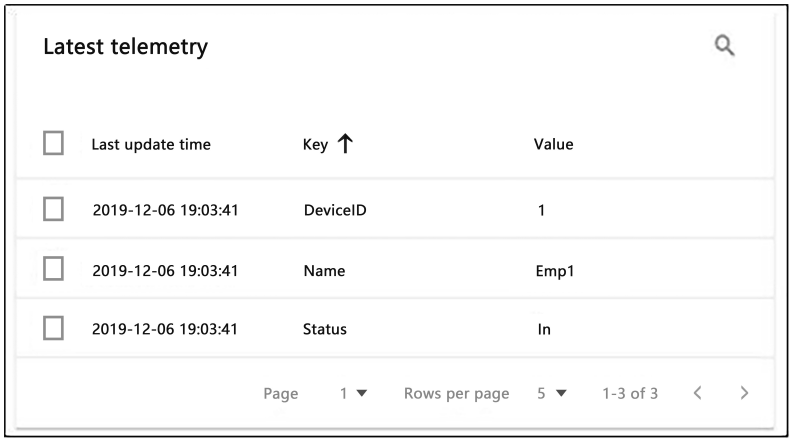
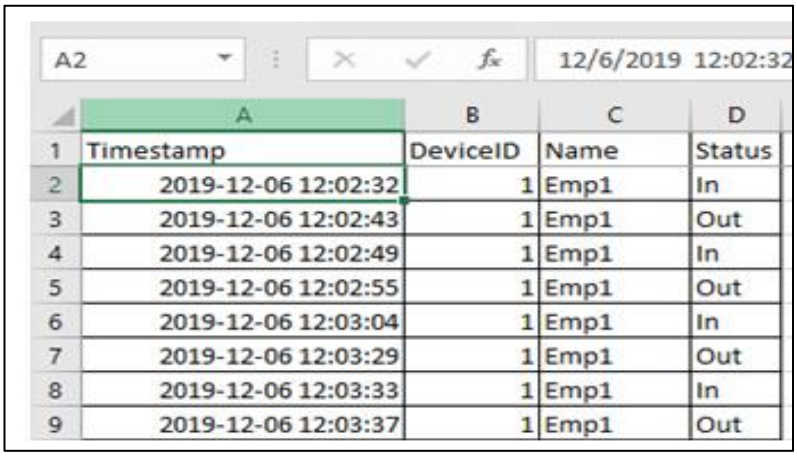


Figure 11. The "Latest Telemetry" Panel of Thingsboard

4.5 ThingsBoard IoT Platform Integration Testing

Testing the integration of the IoT Thingsboard platform can be seen from whether the JSON data sent by the Arduino Uno module with the MQTT protocol can be received by the telemetry system of the IoT Thingsboard platform. From the test results, each fingerprint that was successfully identified when the system connection was online, the data was successfully entered into the telemetry server from Thingsboard. The Thingsboard telemetry server can read JSON data sent by the Arduino Uno module. The data read on the telemetry server are Name, Status, and DeviceID, according to the payload written on the Arduino Uno II (slave) sketch in Figure 12.



	A	B	C	D
1	Timestamp	DeviceID	Name	Status
2	2019-12-06 12:02:32	1	Emp1	In
3	2019-12-06 12:02:43	1	Emp1	Out
4	2019-12-06 12:02:49	1	Emp1	In
5	2019-12-06 12:02:55	1	Emp1	Out
6	2019-12-06 12:03:04	1	Emp1	In
7	2019-12-06 12:03:29	1	Emp1	Out
8	2019-12-06 12:03:33	1	Emp1	In
9	2019-12-06 12:03:37	1	Emp1	Out

Figure 12. Excel file (.xlsx) from Export Telemetry Data*

This study also utilizes the existing dashboard facilities on the Thingsboard Professional server for reporting purposes. The dashboard created can display all telemetry history received by the telemetry server. From the dashboard facility, existing telemetry data history can be downloaded in Excel format (* .xls or * .xlsx) and comma-separated values (* .csv).

4.6 Performance Testing on the Arduino Module

For Arduino Uno, I (master), which controls the fingerprint sensor, LCD, and buzzer, the sketch uploaded uses a storage program (flash ROM) of 11,360 bytes out of a total of 32,256 bytes or about 35% of the total flash ROM and uses SRAM as many as 1,452 bytes of a total of 2,048 bytes or around 70% of the total SRAM in Figure 13.

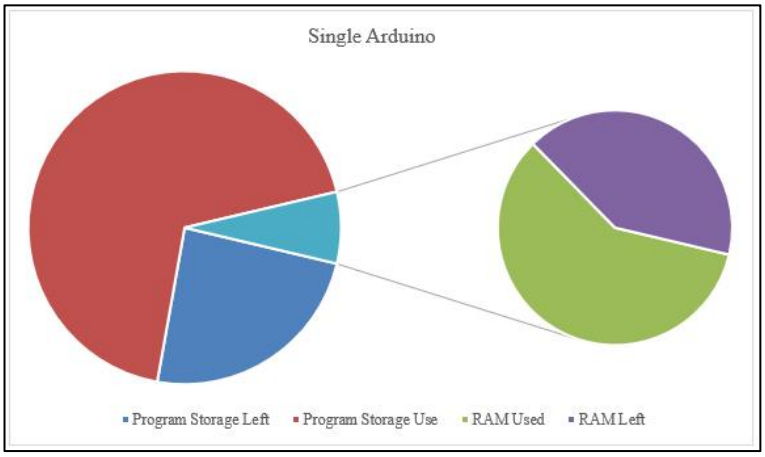


Figure 13. Memory Allocation for Single Arduino Use

As for the Arduino Uno II (slave), which controls the ethernet shield module and OLED display, the sketch uploaded uses 27,800 bytes of storage (flash ROM) out of a total of 32,256 bytes or around 86% of the total flash ROM. It uses SRAM as many as 1,456 bytes of 2,048 bytes, or about 71% of the total SRAM in Figure 14.

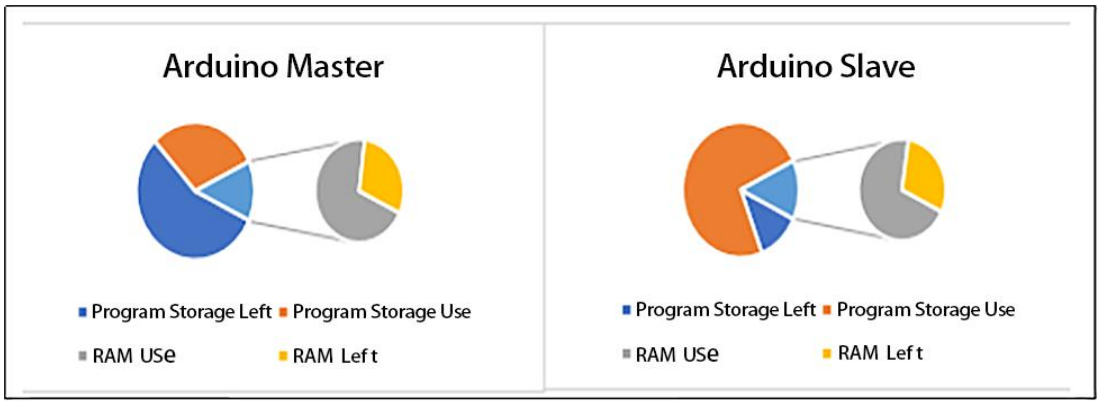


Figure 14. Memory Allocation for the Use of Two Arduino

4.7 Response Time Server Telemetri Thingsboard

Test conditions using an internet connection via a LAN cable (ethernet shield). The internet speed from the provider used is 20Mbps. The response time from the time lag at the fingerprint scan notification with the timestamp on the ThingsBoard server is in Table 2.

Table 2. Time Response

Timestamp	Device ID	Name	Status	Response Time
2019-12-06 12:02:32	1	Emp1	In	1s
2019-12-06 12:02:43	1	Emp1	Out	1s
2019-12-06 12:02:49	1	Emp1	In	1s
2019-12-06 12:02:55	1	Emp1	Out	1s
2019-12-06 12:03:04	1	Emp1	In	1s
2019-12-06 12:03:29	1	Emp1	Out	1s
2019-12-06 12:03:33	1	Emp1	In	1s
2019-12-06	1	Emp1	Out	1s

12:03:37				
2019-12-06 12:03:41	1	Emp1	In	1s
2019-12-06 12:05:55	1	Emp2	In	1s
2019-12-06 12:05:59	1	Emp2	Out	1s
2019-12-06 12:06:03	1	Emp2	In	1s
2019-12-06 12:06:16	1	Emp2	Out	1s
2019-12-06 12:06:16	1	Emp2	Out	1s
2019-12-06 12:07:49	1	Emp2	In	1s
2019-12-06 12:07:49	1	Emp3	In	1s
2019-12-06 12:07:49	1	Emp3	Out	1s
2019-12-06 12:08:10	1	Emp8	In	1s
2019-12-06 12:08:12	1	Emp8	Out	1s
2019-12-06 12:08:34	1	Emp8	Out	1s
2019-12-06 12:08:34	1	Emp9	In	1s
2019-12-06 12:09:02	1	Emp4	Out	1s

Based on the test results as shown in Table 2, the response time of each data is processed with accuracy as expected and the program runs successfully.

The IoT-based biometric attendance system leverages Arduino Uno and ThingsBoard to create a sophisticated method for tracking attendance. This system involves a fingerprint sensor for identification. An IoT-based biometric attendance system uses Arduino Uno as a hardware platform and ThingsBoard as a dashboard for visualization and data management [41]. The main hardware component is a fingerprint sensor integrated with Arduino Uno. This sensor scans fingerprints for identification during attendance marking. This result is supported by several researchers that before fingerprint identification is successful, attendance data is recorded and stored. ThingsBoard functions as an interface for monitoring and displaying attendance records on a dashboard. Projects often involve the use of libraries such as the Adafruit Fingerprint Sensor Library to facilitate communication between the fingerprint sensor and Arduino Uno.

5. Conclusion

This paper uses a prototype method to create a biometric attendance system using Arduino Uno and ThingsBoard with the MQTT protocol. The results obtained from all experiments indicate that testing of fingerprint detection uses methods with high accuracy. The system designed can detect fingerprints connected to a physical device whose data is centralized, allowing superiors to monitor employee absences in real time. Further studies can be carried out to examine the level of data security through other protocols. It is real-time and safe without the possibility of further data interruptions. Also, the use of other types of Arduino Uno with larger memory such as Arduino Mega to replace the use of two Arduino Uno modules, the use of other types of microcontroller boards such as NodeMCU, which has been integrated with the ESP8266 wireless chip for a more compact circuit model, better use of fingerprint sensors in terms of template storage capacity and sensor sensitivity, the use of MicroPython as an interpreter language, replacing C as a compiler language in Arduino Ide, the use of other IoT platforms besides ThingsBoard and the use of the

keypad panel for inputting custom User IDs, replacing the Auto Increment system used in this research.

References

[1] D. Patel, "Multimodal Biometric Systems: a Review," *International Journal of Advanced Res earch in Computer Science*, vol. 9, no. 2, pp. 361-365, 2018, doi: 10.26483/ijarcs.v9i2.5742.

[2] M. Sandhya and M. V. N. K. Prasad, "Securing fingerprint templates using fused Structures," *IET Biometrics*, vol. 6, no. 3, pp. 173-182, 2017, doi: 10.1049/iet-bmt.2016.0008.

[3] H. a Abdullah, "Fingerprint Identification System Using Neural Networks," *Al-Nahrain Journal for Engineering Sciences*, vol. 15, no. 2, pp. 234-244, 2012.

[4] K. V. Krishnam Raju, P. Nishmitha, P. Mounika, N. Ajeeth, V. Krishna Sandeep, and N. Kishore Raju, "Implementation of fingerprint recognition system using minutiae score matching," *International Journal of Recent Technology and Engineering*, vol. 8, no. 1, pp. 62-67, 2019.

[5] J. Zhu, B. Arsovska, and K. Kozovska, "Acupuncture treatment in osteoarthritis," *International Journal of Recent Scientific Research,* vol. 11, no. 02, pp. 37471-37472, 2020.

[6] H. S. Oluwatosin, "Client-Server Model," *IOSR Journal of Computer Engineering*, vol. 16, no. 1, pp. 57-71, 2014.

[7] D. Dinculeană and X. Cheng, "Vulnerabilities and limitations of MQTT protocol used between IoT devices," *Applied Sciences*, vol. 9, no. 5, pp. 1-10, 2019.

[8] G. Liao, X. Zhu, S. Larsen, L. Bhuyan, and R. Huggahalli, "Understanding power efficiency of TCP/IP packet processing over 10GbE," in *Proceedings - 18th IEEE Symposium on High Performance Interconnects, HOTI 2010*, 2010, pp. 32-39.

[9] G. S. Kuaban, T. Atmaca, A. Kamli, T. Czachórski, and P. Czekalski, "Performance analysis of packet aggregation mechanisms and their applications in access (E.g., iot, 4g/5g), core, and data centre networks," *Sensors*, vol. 21, no. 11, p. 3898, 2021.

[10] K. K. Patel, S. M. Patel, and P. G. Scholar, "Internet of Things-IOT: Definition, Characteristics, Architecture, Enabling Technologies, Application & Future Challenges," *International journal of engineering science and computing*, vol. 6, no. 5, pp. 1-10, 2016

[11] G. Misra, V. Kumar, A. Agarwal, and K. Agarwal, "Internet of Things (IoT) - A Technological Analysis and Survey on Vision, Concepts, Challenges, Innovation Directions, Technologies, and Applications (An Upcoming or Future Generation Computer Communication System Technology)," *American Journal of Electrical and Electronic Engineering*, vol. 4, no. 1, pp. 23-32, 2016, doi: 10.12691/ajeee-4-1-4.

[12] Z. Ajazmoharkan, T. Choudhury, S. C. Gupta, and G. Raj, "Internet of Things and its applications in E-learning," in *3rd IEEE International Conference on*, 2017, pp. 1-5.

[13] A. Protopsaltis, P. Sarigiannidis, D. Margounakis, and A. Lytos, "Data visualization in internet of things: Tools, methodologies, and challenges," in *Proceedings of the 15th international conference on availability, reliability and security*, 2020, pp.1-11.

[14] A. Arman, P. Bellini, D. Bologna, P. Nesi, and G. Pantaleo, "Automating IoT Data Ingestion Enabling Visual Representation," *sensors*, vol. 21, no. 8429, pp. 1-25, 2021.

[15] R. Krishnamurthi, A. Kumar, D. Gopinathan, A. Nayyar, and B. Qureshi, "An overview of iot sensor data processing, fusion, and analysis techniques," Sensors (Switzerland), vol. 20, no. 21, pp. 1-23, 2020, doi: 10.3390/s20216076.

[16] S. Traboulsi and S. Knauth, "Towards implementation of an IoT analysis system for buildings environmental data and workplace well-being with an IoT open software," *Procedia Computer Science*, vol. 170, no. 2019, pp. 341-346, 2020, doi: 10.1016/j.procs.2020.03.048.

[17] A. Rinaldi, "Biometrics' new identity—measuring more physical and biological traits," *EMBO Rep*., vol. 17, no. 1, pp. 22-26, 2016, doi: 10.15252/embr.201541677.

[18] C. Wendehorst and Y. Duller, "Biometric Recognition and Behavioral Detection," Ssrn.com, 2021. https://ssrn.com/abstract=4087455 (accessed Jan. 29, 2024).

[19] Y. Liu, "Identifying legal concerns in the biometric context," *J. Int. Commer. Law Technol*., vol. 3, no. 1, pp. 45-54, 2008.

[20] R. Ackerley, I. Carlsson, H. Wester, H. Olausson, and H. Backlund Wasling, "Touch perceptions across skin sites: Differences between sensitivity, direction discrimination and pleasantness," *Frontiers in behavioral neuroscience*, vol. 8, no. FEB, pp. 1-10, 2014, doi: 10.3389/fnbeh.2014.00054.

[21] P. K. Bose and M. J. Kabir, "Fingerprint: A Unique and Reliable Method for Identification," *Journal of Enam Medical College*, vol. 7, no. 1, pp. 29-34, 2017, doi: 10.3329/jemc.v7i1.30748.

[22] N. Kaushal and P. Kaushal, "Human Identification and Fingerprints: A Review," *Journal of Biometrics & Statistics*, vol. 02, no. 04, pp. 1-6, 2011.

[23] B. Fakiha, "How Technology has Improved Forensic Fingerprint Identification to Solve Crime," *International Journal of Advanced Science and Technology*, vol. 29, no. 05, pp. 746-752, 2020.

[24] S. Memon, M. Sepasian, and W. Balachandran, "Review of Finger Print Sensing Technologies," in *2008 IEEE International Multitopic Conference*, 2008, pp. 226-231.

[25] A. Q. M. S. U. Pathan, K. K. Thakur, A. Chakraborty, and M. H. Kabir, "Fingerprint Authentication Security: An Improved 2-Step Authentication Method with Flexibility," *International Journal of Scientific & Engineering Research*, vol. 10, no. 1, pp. 438-445, 2019.

[26] W. Yang, S. Wang, J. Hu, G. Zheng, and C. Valli, "Security and accuracy of fingerprint-based biometrics: A review," *Symmetry*, vol. 11, no. 2, pp. 1-19, 2019, doi: 10.3390/sym11020141.

[27] P. Asabere, F. Sekyere, and W. K. Ofosu, "Wireless Biometric Fingerprint Attendance System using Arduino and Mysql Database," *International Journal of Computer Science, Engineering and Applications (IJCSEA)*, vol. 9, no. 4/5, pp. 1-11, 2019.

[28] C. O. Akanbi, I. K. Ogundoyin, J. O. Akintola, and K. Ameenah, "A Prototype Model of an IoT-based Door System using Double-access Fingerprint Technique," *Nigerian Journal of Technological Development*, vol. 17, no. 2, pp. 142-149, 2020.

[29] S. C. Hoo and H. Ibrahim, "Biometric-Based Attendance Tracking System for Education Sectors : A Literature Survey on Hardware Requirements," *Journal of Sensors*, vol. 2019, pp. 1-25, 2019.

[30] M. A. Muchtar, Seniman, D. Arisandi, and S. Hasanah, "Attendance fingerprint identification system using arduino and single board computer," in *2nd International Conference on Computing and Applied Informatics 2017*, 2018, pp. 1-8, doi: 10.1088/1742-6596/978/1/012060.

[31] S. Minz, A. Saha, and M. R. Dev, "Arduino Based Automatic Irrigation System," ADBU J. Electr. Electron. Eng., vol. 3, no. 1, pp. 31-36, 2019, [Online]. Available: https://media.neliti.com/media/publications/287656-arduino-based-automatic-irrigation-syste-d4f342de.pdf.

[32] T. Jain, U. Tomar, U. Arora, and S. Jain, "IOT based biometric attendance system," Int. J. Electr. Eng. Technol., vol. 11, no. 2, pp. 156-161, 2020, [Online]. Available: http://www.iaeme.com/IJEET/issues.asp?JType=IJEET&VType=11&IType=02.

[33] C. . Sarika, A. B. Malakreddy, and H. N. Harinath, "IoT-Based Smart Login Using Biometrics," in *International Conference on Computer Networks and Communication Technologies: ICCNCT 2018*, vol. 15, pp. 589-597, 2019, doi: 10.1007/978-981-10-8681-6.

[34] R. Sarmah, M. Bhuyan, and M. H. Bhuyan, "SURE-H: A Secure IoT Enabled Smart Home System," in *2019 IEEE 5th World Forum on Internet of Things*, 2019, pp. 59-63.

[35] A. A. Abdelhafez, O. Ismael, and H. Elkady, "IoT-Based Data Size Minimization Using Cluster-Based-Similarity- Elimination Elimination," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 15, no. 02, pp. 34-50, 2023, doi: 10.17762/ijcnis.v15i2.6151.

[36] I. Al-Sgir and W. Karamti, "Intelligent Agents Model with JADE for Scheduling Analysis and Correction of Real-Time Systems," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 15, no. 1, pp. 107-119, 2023, doi: 10.17762/ijcnis.v15i1.5762.

[37] J. Al-Saraireh and H. Joudeh, "An Efficient Authentication Scheme for Internet of Things," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 13, no. 3, pp. 416-430, 2021, doi: 10.54039/IJCNIS.V13I3.3422.

[38] W. Yang, S. Wang, N. M. Sahri, N. M. Karie, M. Ahmed, and C. Valli, "Biometrics for Internet - of - Things Security : A Review," *Sensors* , vol. 21, no. 6163, pp. 1-27, 2021, doi: 10.3390/s21186163.

[39] M. Sarma, A. Gogoi, R. Saikia, and D. J. Bora, "Fingerprint Based Door Access System using Arduino," Int. J. Sci. Res. Eng. Manag., vol. 4, no. 8, pp. 1-5, 2020, [Online]. Available: https://media.neliti.com/media/publications/287656-arduino-based-automatic-irrigation-syste-d4f342de.pdf.

[40] L. Louis, "Working Principle of Arduino and Using it as a Tool for Study and Research," *International Journal of Control, Automation, Communication and Systems* (IJCACS), vol. 1, no. 2, pp. 21-30, 2018.

[41] N. Taj, M. H. Zafar, S. A. Waqas, H. Rehman, M. O. Alassafi, and I. Khan, "Smart relay selection scheme based on fuzzy logic with optimal power allocation and adaptive data rate assignment," *International Journal of Communication Networks and Information Security (IJCNIS)*, vol. 11, no. 1, pp. 239-247, 2019, doi: 10.17762/ijcnis.v11i1.4049.