eHealth Internet of Things on Low Power Wide Area Network of Hospital

Wiraphon Manatarinat[1], **Panjai Tantatsanawong**[2], **Suvit Poomrittigul**[3†], and **Sakol Udomsiri**[1], Non-members

ABSTRACT

Nowadays, the number of older adults is increasing due to medical advancements and attention to the health of the elderly. Effective healthcare and safety are essential. Therefore, the researcher has an idea to develop a system that can support the care of patients and the elderly safely by detecting falls. However, there are limitations in the contact with the public health system and other technical issues such as high power consumption, cost, and reliability. In this article, we would like to introduce a fictional wearable fall detection system. Algorithms based on threshold are specifically integrated with the Internet of Things (IoT) and a Low-Power Wide Area Network (LPWAN), with nodes monitoring and parsing the data. The server part consists of an application server and a UDP server. The application server is responsible for accessing the data. At the same time, the UDP server is responsible for receiving data and monitoring abnormal data packets sent by the Narrowband Internet of Things (NB-IoT) to develop an efficient and secure data transmission mechanism. For healthcare professionals, eHealth is designed from the server side, which comprises a server application and a client-server architecture. The eHealth system is a digital technology tool and ICT service that connects healthcare providers and citizens to access health services effectively, comprehensively, equitably, and safely for elderly patients and healthcare providers for algorithm validation. We recruit volunteers for events. Daily life and falls the experimental results showed that our presented algorithm could achieve accuracy, indicating the efficiency of our system.

Keywords: eHealth System, Fall Detection, NB-Internet of Things, LoRaWAN

1. INTRODUCTION

The Internet of Things (IoT) is suitable for application in the medical industry. IoT communication can use Wi-Fi, 3G, 4G, and 5G. However, there are limitations, such as the distance problem when using 3G, 4G, and 5G, as well as expense-related issues. To address the expense problem, the researchers had an idea to apply LoRaWAN technology for data transmission from the elderly's houses to the hospital by studying the behavior of patients, especially the elderly, to develop the technology designed to assist them during emergencies. By utilizing the Gyroscope principle, the researchers' model system was designed to detect falls among older individuals. The model was specifically designed to identify falls from a long distance, particularly for the elderly who just had a problem after having knee surgery. If this technology is widely adopted, it is expected to bring benefits to the medical industry [1], [2]. Therefore, the adoption of this technology may help decrease the frequency of falls among older people. Consequently, fall-related issues among older people remain a significant problem for both the elderly themselves and their relatives.

This research is divided into two parts: IoT technology and Long-Range Wide Area Networks (LoRaWAN) [3], [4]. The researchers utilize LoRaWAN devices for data transmission, as LoRaWAN is specifically designed for energy efficiency. Data is sent through the UDP protocol. However, UDP cannot guarantee data loss prevention during the transmission process.

The researchers have developed an architecture that not only increases the efficiency of data transmission and detects lost data but also ensures stability in data protection. This is crucial for protecting medical data. Nevertheless, the coverage of the LoRaWAN network still needs improvement. Therefore, the researchers have decided to use NB-IoT Shield, a module that facilitates device connectivity with the Narrowband Internet of Things (NB-IoT) [3], [5], and [6]. Essentially, NB-IoT functions as a protective armor or additional board that can be inserted into a compatible Arduino or micro-controller board. It requires cellular modules, related circuits, and antennas to communicate with the NB-IoT network. The antennas enable the device to transmit and receive data through the NB-IoT network, which can also integrate with the IoT system and applications.

NB-IoT Shield for Fall Diagnostic refers to the module hardware that combines NB-IoT with the sensor to detect and report falls in real-time. This shield is designed to

Manuscript received on May 26, 2023; revised on June 3, 2023; accepted on June 23, 2023. This paper was recommended by Associate Editor Piya Kovintavewat.

[1]The authors are with Electrical Engineering Department, Pathumwan Institute of Technology, Bangkok, Thailand.

[2]The author is with Computer Technology Department, Silpakorn University, Bangkok, Thailand.

[3]The author is with School of Information Technology, King Mongkut's Institute of Technology Ladkrabang (KMITL) Bangkok, Thailand.

[†]Corresponding author: suvit@it.kmitl.ac.th

Digital Object Identifier: 10.37936/ecti-eec.2023213.251460

be compatible with an Arduino board, making it easy to integrate with other IoT applications. Generally, NB-IoT for fall detection consists of an accelerometer sensor to detect falls. NB-IoT modules are used for wireless communication, and their microcontroller are responsible for processing data and sending notifications. The sensor can detect sudden movements and unexpected changes in acceleration that may indicate falls. The microcontroller processes this information and sends alerts to a central monitoring system or administrator through the NB-IoT network. The NB-IoT Shield detects falls, serving as a tool that enhances safety for the elderly or vulnerable individuals. It provided an additional layer of protection in the event of a fall or accident, enabling real-time monitoring and response. This method makes it possible to prompt intervention and potentially save lives.

This research aims to design and develop a system of small and energy-efficient fall detection devices, including mechanisms for effective data transmission and security systems. The device should be capable of detecting falls, determining the nature of falls, and transmitting medical data from fall detection using the IoT sensor and Low Power Wide Area Network (LoRaWAN) from the patient's home to the hospital.

2. SYSTEM DEVELOPMENT AND DESIGN

2.1 System Framework

The system architecture comprises three main components: 1) a wearable device on the user side; 2) decision support on the server; and 3) an interactive user interface on the client side, as illustrated in Fig. 1. NB-IoT wireless communication is implemented due to its advantages of low power consumption, low cost, and comprehensive coverage [7], [8]. NB-IoT is utilized to facilitate communication between the wearable device and the server. Monitoring nodes, such as wearable devices, consist of microprocessors and wireless communication modules. These monitoring nodes collect fall detection data and respond in real time [2], [9], and [10]. In case of server failure, the system consists of Platform One and the Server Part. Firstly, Platform One is dedicated to wireless communication, where nodes review and parse the data. The server part comprises application and UDP servers. The application server is responsible for data access. At the same time, the UDP server is responsible for receiving data and checking for abnormal data packets sent by NB-IoT to develop an efficient and secure transmission mechanism.

2.2 Main System Modules

1) NB-IoT and UDP Server and UDP Protocol

Narrowband Internet of Things, or NB-IoT, is a Low Power Wide Area Network (LPWAN) radio technology designed to improve communication efficiency. NB-IoT operates on authorized cellular frequencies, enabling greater distances and coverage compared to other IoT-

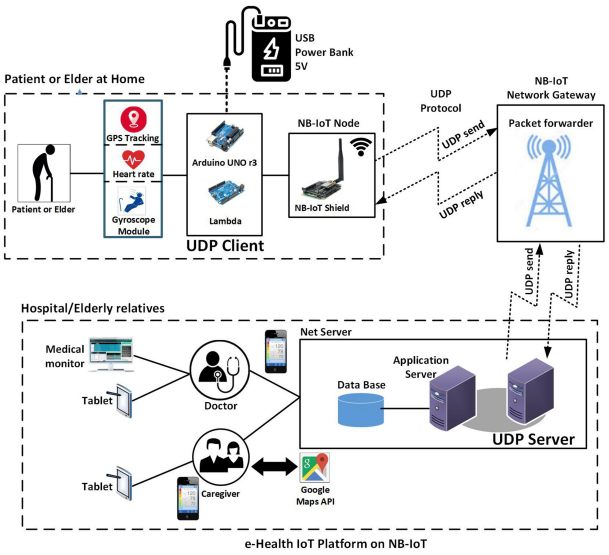

***Fig. 1**: Framework of the fall detection system.*

connected options like Wi-Fi or Bluetooth. It is also optimized to support numerous devices by saving energy and utilizing efficient hardware. This technology is widely used for smart cities, smart agriculture, asset tracking, and industrial automation applications [11].

LPWAN stands for Low Power Wide Area Network, a wireless network that supports battery-powered, low-bandwidth devices over long distances. It is well-suited for Internet of Things (IoT) applications, where devices need to communicate with small amounts of data over a wide area while consuming minimal power. LPWAN technologies include LoRaWAN, NB-IoT, Sigfox, and others. These technologies differ in terms of range, bandwidth, and power consumption [11], [12].

NB-IoT operates on a licensed cellular network and works with other cellular technologies, such as 2G, 3G, 4G, and 5G. It offers several advantages over LPWAN technology, such as LoRaWAN and Sigfox. The transmission characteristics of NB-IoT devices involve transferring data in packets using the UDP protocol. UDP packet transmission consists of a head and payload, where the header contains information such as the port number and destination [13].

Moreover, its duty also includes checking the integrity of the packet. In comparison, the payload consists of the data being sent. These data may include sensors, updates, status, or other data from NB-IoT devices.

A UDP (User Datagram Protocol) server is a program or computer that runs a network application that uses the UDP protocol to receive and process data from UDP clients. UDP is a connectionless protocol, meaning a UDP server does not establish a dedicated communication channel with a client before exchanging data. Instead, the client sends UDP packets directly to the server without the need for a "handshake" process. The server receives the packets, processes them, and may respond to the client using UDP packets. UDP servers are commonly

used for real-time applications, such as video streaming or online gaming, in which low latency and high-speed data transfer are essential.

The data transmission characteristics between the NB-IoT device and the UDP server are generally one-way. The NB-IoT device sends data to the server without expecting a response, and the data is sent as a UDP packet consisting of a header and a payload.

The UDP server is a program that listens for inbound UDP packets and processes them according to the application logic. In the context of NB-IoT, a UDP server can receive data sent from NB-IoT devices and perform various operations with that data, such as storing it in a database, triggering a notification, or forwarding the data to another system or service.

Additionally, the UDP server is commonly used in IoT applications that require real-time data processing and low latency. It can be set up to receive data from multiple devices simultaneously and handle large volumes of data with minimal cost.

In the case of NB-IoT, a UDP server can be used as an endpoint for data transmission from NB-IoT devices. The NB-IoT devices can send data packets to the UDP server, which then processes the packets and performs the necessary actions based on the received data.

The researchers used the socket library by adopting methods that can create and communicate with sockets. Firstly, the researchers use Python code to send UDP packets to a server. Then, the researchers determine the IP address and port number of the UDP server that they want to connect to. After that, the researchers define the message they want to send to the server. In this case, it is "Hello, server!". Finally, the researchers created a socket object using the function 'socket.socket()', specifying the address family as 'socket.AF_INET' and the socket type as 'socket.SOCK_DGRAM', which pointed out that we are indeed using a UDP.

Regarding the 'sendto()' command of the socket object to send messages to the server, the 'sendto()' method requires two arguments. Therefore, the message sent to the server is a tuple that includes the address, IP number, and port. After sending the messages, UDP sockets can be closed using the 'close()' method of the socket object. To receive data from a UDP server, the 'recvfrom()' method of the socket object is used. An example of code that demonstrates how to retrieve data from a UDP server using Python is below.

```
# Set up the UDP socket
sock = socket.socket(socket.AF_INET, socket.SOCK_DGRAM)

# Bind the socket to a specific IP address and port
server_address = ('0.0.0.0', 1234)
sock.bind(server_address)

# Listen for incoming messages
while True:
    data, address = sock.recvfrom(4096)
    print(f'Received len(data) bytes from address: data')
```

This code defines the IP address and port number of the UDP server to bind to. Then, it creates a socket object using the function 'socket.socket( )' and specifies the address family as 'socket.socket( )', 'socket.AF_INET' and the socket type as 'socket.SOCK_DGRAM' and uses the 'bind()' method of the socket object to bind the socket to the local address and port, allowing it to listen for incoming data from that address and port.

Finally, the 'recvfrom( )' method of the socket object is used to receive data from the client. The 'recvfrom()' method takes the buffer size as an argument, which determines the maximum amount of data to be received simultaneously. The received data is stored in the 'data' variable, and the address of the client is stored in the 'addr' variable.

2) GY-521 IMU 3-axis Accelerometer/Gyroscope Module (MPU6050)

In the research, fall detection utilized gyroscope sensors to detect the rotation of various devices. It is a 3-axis (3-axis) sensing system paired with an accelerometer to detect object movement. Both of these sensors provide accurate results regardless of the user's posture.

The accelerometer/gyroscope module is a carrier board based on the MPU-6050 sensor. It contains a micro-electromechanical system (MEMS) and gyro MEMS in a single chip, which is very accurate (16 bits) due to the presence of analog-to-digital conversion hardware for each channel. Consequently, the x, y, and z channels can be captured simultaneously. The sensor connects to the microcontroller using the I2C bus.

3) Arduino UNO R3

The Arduino UNO R3 is a microcontroller board based on the ATmega328P microcontroller. It has 14 digital input and output pins, six analog inputs, and a 16 MHz quartz crystal. A USB connection, a power jack, an ICSP header, and a reset button are the most popular boards in the Arduino family. They are widely used for prototyping and DIY electronics projects. The board can be programmed using the Arduino Integrated Development Environment (IDE). Besides, it also supports various programming languages, including C and C++.

The Arduino UNO R3 is a board whose digital pins can be set to output or read voltage. The analog pins can read voltages between 0 and 5 volts. The microcontroller can be programmed using a USB connection to a computer.

The researcher uses the Arduino board to develop IoT devices connected to the NB-IoT network, leveraging various communication interfaces such as UART, I2C, and SPI to connect with the NB-IoT module. NB- IoT Arduino boards can be programmed in the Arduino IDE to collect data from sensors and transmit it over the NB-IoT network to a remote server. Overall, Arduino can be a cost-effective tool for building NB-IoT devices.

Regarding power banks, they can fully charge the device approximately five times before requiring manual charging. It is important to note that this estimation is approximate, and the actual charging capacity may vary based on factors like the efficiency of the charging circuit and the power requirements of the device being charged.

3. METHODOLOGY

3.1 Fall Detection Algorithm

The algorithm that is used in fall detection relies on the available data. The fundamental concept involves collecting sensor data from accelerometers, gyroscopes, and fall detection. The data can be stored continuously or periodically, depending on the user's requirements. The relevant properties are extracted from the sensor data, such as acceleration magnitude, direction, and velocity, to differentiate between normal movements and falls.

Machine learning or pattern recognition algorithms are utilized to classify the sensor data into different activities such as walking, sitting, and falls. This is achieved by comparing the sensor data with a predefined fall template and analyzing changes in acceleration and direction during a fall event. When a fall is detected, an alert is generated to notify caregivers or emergency services. Signals can be acquired through a mobile application, enabling prompt response to fall alerts, such as contacting authorities or dispatching emergency services [14].

3.2 Feature Extraction

NB-IoT devices establish a connection to the network, generate data, and collect it to send it to a UDP server. The UDP server creates a UDP socket on the NB-IoT device, specifies the IP address and port number of the UDP server, and converts the data into UDP packets to be sent to the UDP socket server. Then, it waits for an acknowledgment or response from the UDP server. Upon receiving the acknowledgment or response, it processes the information as needed and repeats the process. It will process the information as needed and repeat. If there is no response within a certain period of time, or if a response is received, the device collects or generates data to send to the UDP server. When the specified time has elapsed, data packets are sent again. Finally, the UDP socket is closed, and the device disconnects from the NB-IoT network when finished.

3.3 Selection of Multilevel Thresholds

The main component used in this system is the MPU-6050 sensor. It is a 6-axis motion tracker that integrates a 3-axis gyroscope and a 3-axis accelerometer into a single package. The sensor is generally used in wearable devices and other applications that require motion-sensing capabilities. It is capable of detecting changes in orientation and movement and providing information about rotational acceleration. The MPU-6050 sensor combines a micro-electromechanical system (MEMS) accelerometer and a MEMS gyro on a single chip, ensuring high accuracy with a 16-bit resolution. Each channel of the sensor has dedicated hardware for analog-to-digital conversion, allowing simultaneous capture of data from the x, y, and z axes. The sensor communicates with the microcontroller through the I2C bus interface.

In fall detection, the values acquired from the MPU-6050 accelerometer for the x, y, and z axes are denoted as A_x, A_y, and A_z, respectively. Similarly, the gyroscope readings for the x, y, and z axes are represented as G_x, G_y, and G_z. Accelerometer and gyroscope values provide information about the acceleration and rotation experienced by the device during fall detection as Eqs. (1) and (2).

$$A_{axis} = \frac{scale}{Araw_{axis}} \tag{1}$$

$$G_{axis} = \frac{scale}{Graw_{axis}} \tag{2}$$

To calculate the fall detection value, the sensor's readings from the accelerometer and gyroscope need to be calculated first. Then, the magnitude and direction of the acceleration can be determined using the following Eqs. (3) and (4)

Sum Vector Magnitude of Acceleration (SVM)

$$\text{SVM-A} = \sqrt{A_x^2 + A_y^2 + A_z^2} \tag{3}$$

$$\text{SVM-G} = \sqrt{G_x^2 + G_y^2 + G_z^2} \tag{4}$$

where Eq. (3) magnitude and direction values were obtained from the accelerometer sensor. Eq. (4) magnitude and direction values were obtained from gyroscope sensors.

Finding Pitch, Roll, and Yaw for greater accuracy in finding fall detection values using Eqs. (5) – (7).

$$\text{Pitch} = \left(\frac{180}{\pi}\right) \times \text{atan}\frac{A_x}{\sqrt{A_x^2 + A_z^2}} \tag{5}$$

$$\text{Roll} = \left(\frac{180}{\pi}\right) \times \text{atan}\frac{A_y}{\sqrt{A_x^2 + A_z^2}} \tag{6}$$

$$\text{Yaw} = \left(\frac{180}{\pi}\right) \times \text{atan}\frac{A_z}{\sqrt{A_x^2 + A_z^2}} \tag{7}$$

where A_x, A_y, and A_z are the values obtained from the Accelerometer sensor.

A model that presents a fall detection system for patients or the elderly. The system can detect many types of falls. The hardware consists of an Arduino UNO R3 forum server equipped with a GY-521 MPU6050 Accelerometer and Gyroscope sensor, as shown in Figs. 1 and 2. To send data, NB-IoT is used, and the software is UDP Server and Arduino IDE v.1.8. .9 for programming the Arduino board. The system developed by the researcher has three functions:

1) Input Section: The Arduino board receives input values (x, y, and z axis values) from the GY-521 MPU6050 Accelerometer and Gyroscope sensors.
2) Process Section: The processing involves executing commands in different components:
   - The Arduino UNO R3 board processes the x, y, and z axis values obtained from the GY-521 MPU6050

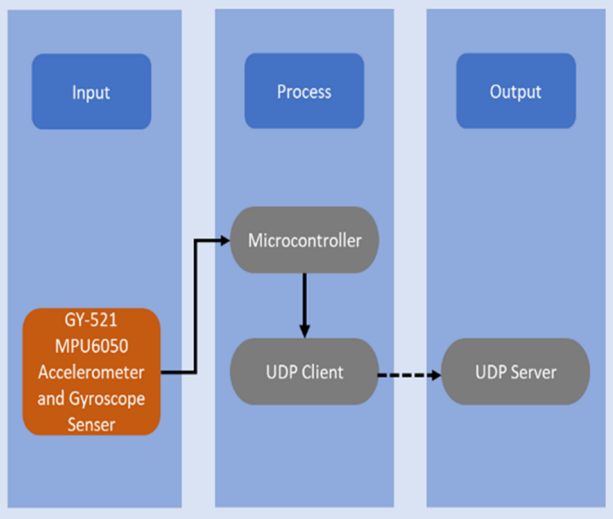

***Fig. 2**: A Schematic Diagram of the System.*

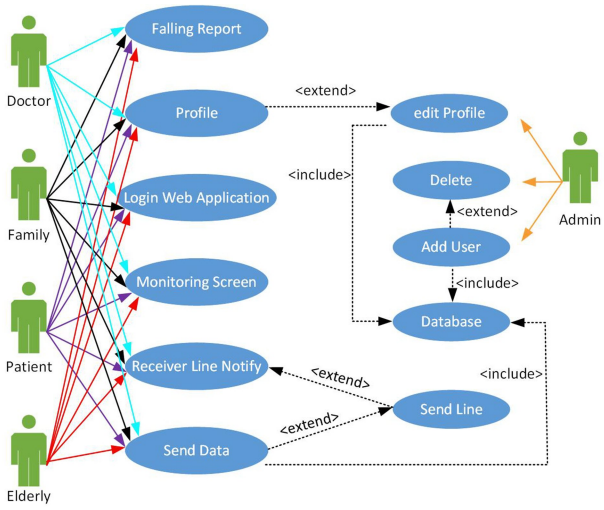

***Fig. 4**: Use Case Diagram.*

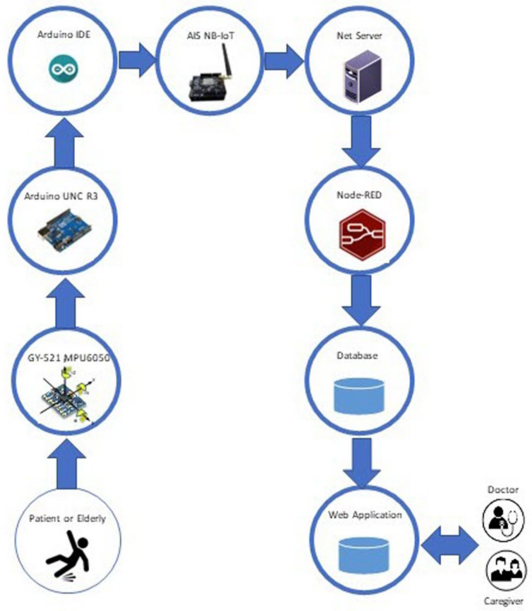

***Fig. 3**: A System Structure.*

Accelerometer and Gyroscope Sensor using the Arduino IDE.

- The UDP server processes the data received from the Arduino board to prepare it for further use.

3) Displaying Crash Information: The crash information obtained from the Arduino board is displayed on the application web page. Additionally, notifications are sent to the Line application of the patient's relatives or relevant individuals [15].

The fall detection system for patients or the elderly involves installing a device on the person. As the person moves, the device collects X, Y, and Z axis values. These values are then analyzed to determine if the person has experienced any of the four types of falls according to the predefined schedule.

Once the values are analyzed, in the event of an actual fall, the data is sent from the NB-IoT device to the UDP Server and Application Server. The data is stored in the database, allowing it to be displayed on web applications for further reference. Additionally, an alert is sent to the Line application to notify individuals involved with the patient or elderly person, enabling them to provide immediate assistance, as shown in Fig. 4 [16].

In NB-IoT (Narrowband Internet of Things), UDP is one of the transport layer protocols. That is supported for sending data between devices and servers. Because NB-IoT is designed for low-power and low-bandwidth devices, UDP is therefore a good choice for sending small amounts of data quickly and efficiently. However, because UDP has no reliability or error detection, applications that use it must therefore be designed to handle packet loss or corruption.

The nature of data transmission between NB-IoT devices and UDP servers is ideal for NB-IoT devices with low bandwidth and low power consumption with low latency. It is designed for the transmission of small, modular data and data packets.

3.4 NB-IOT data transmission with UDP protocols and detection of connection contracts between NB-IOT devices and UDP servers

1) NB-IOT data transmission with UDP protocols

NB-IoT devices establish a connection to the network. Collects and generates data to send to the UDP server. The UDP server creates a UDP socket on the NB-IoT device. Specifies the IP address and port number of the UDP server. To convert the data into UDP packets and send them to the UDP server, use the socket, which waits for an acknowledgment or response from the UDP server upon receipt of an acknowledgment. It processes the data as needed and reproduces it; for example, it collects or generates data to send to a UDP server. When the

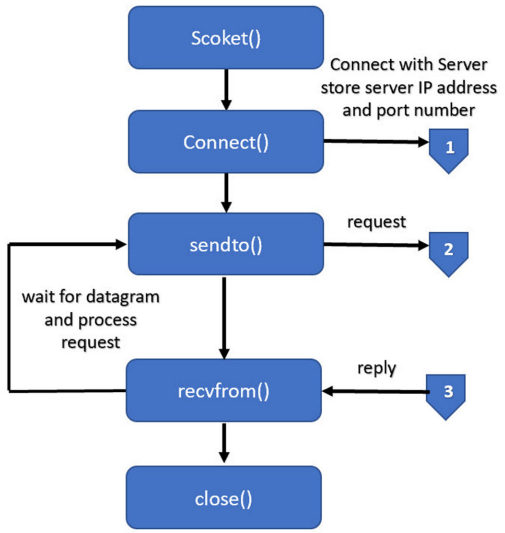

***Fig. 5**: Arduino and NB-IoT Device with C++.*

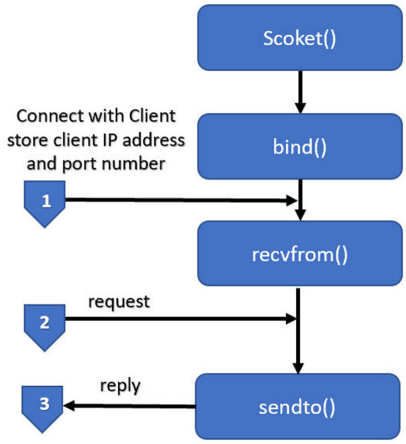

***Fig. 6**: Arduino and NB-IoT Device with Python.*

specified time has elapsed, data packets are sent again, closing the UDP socket and disconnecting from the NB-IoT network when finished.

In Fig. 3, it consists of an Arduino Uno R3, NB-IoT. The data from the sensor is sent via UDP protocol to the UDP server. Arduino, NB-IoT, C ++, and UDP Server are written with Python language programs.

Arduino and NB-IoT devices with C++ and connect them to a UDP server created with Python.

This code sets up an Arduino connected to an NB-IOT network and a UDP server to listen for incoming messages on port 22000 when a message is received. Then, it prints out the content of the message and sends a response back to the UDP server, as shown in Fig. 5.

This code creates a UDP socket and links it to port 22000 from the UDP server, which listens for incoming messages and sends a response to the sender as shown in Fig. 6.

The Arduino device, the NB-IOT, and the UDP Server's Python program run, sending messages from the Arduino to the Python program using the UDP protocol. The Python program has been received.

2) Detection of connection signals between NB-IOT devices and UDP servers

The use of error detection methods in data transmission between the NB-IoT and the UDP Server because the User Datagram Protocol (UDP) does not have a mechanism for detecting or correcting errors. Therefore, to calculate the transmission error rate between an NB-IoT device and a UDP server, an algorithm generates a unique value for each piece of data sent using checksum techniques. The receiver can calculate a checksum of the received data and compare it with the expected checksum. If the two values do not match, it indicates that an error has occurred.

To calculate the error rate using a checksum, the formula for calculating the error rate is:

$$\text{Error rate} = \frac{\text{Number of packets with errors}}{\text{Total number of packets transmitted}} \times 100 \tag{8}$$

The researchers employed a method that involved testing the signal between the NB-IoT device and the UDP server using specific server settings. In order to receive data packets from the NB-IoT device via UDP, the UDP server needed to listen to a designated IP address and have a unique port number assigned to the NB-IoT device. The NB-IoT device was configured to use the same IP address and port number as the UDP server, and security or authentication settings were applied as required. Test data was then sent from the NB-IoT device to the UDP server to verify successful transmission and processing. This test data could be in the form of a standard message or any other information suitable for confirming receipt and processing on the UDP server side. To ensure the reliability of the connection and verify the functionality of all UDP servers, the received data from the NB-IoT devices can be logged and analyzed using network analyzers. By sending multiple data packets from the NB-IoT device, the researchers were able to test the connection's reliability and ensure that all UDP servers passed the test. Additionally, to evaluate the stability of the connection over an extended period, the researchers conducted tests to monitor the connection's performance, including identifying any hidden issues or occurrences of packet loss. Network monitoring tools were utilized to review performance metrics and detect potential problems. With these procedures and using network monitoring tools, the researchers were able to test and ensure the reliability of the connection between the NB-IoT devices and UDP servers, thereby validating the performance of the system.

The researcher chose a network analysis tool for packet or protocol analysis. It is an open-source program for this research. It can capture and analyze network traffic to determine whether data packets from

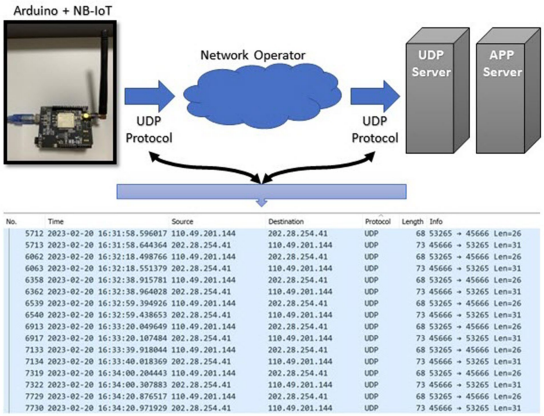

***Fig. 7**: UDP Protocol for Sending Data between NB-IOT and UDP Server.*

NB-IOT devices are received at UDP servers. Such network analyzers are developed according to international standards. For example, a wire shark supports it. Numerous protocol decoding and analysis of many protocols are defined by international standards such as the Internet Engineering Task Force (IATF), the International Organization for Standardization (ISO), and the Institute of Electric and Electronic Engineers for Software Development and Quality Assurance, such as the GNU General Public Service (GPL) and Software Engineering Knowledge (SVB). Common citations follow academic standards such as the APA or MLA STYLE GUIDE, and specific citations will depend on the context and purpose of the research and the effectiveness of the journal or publication to which the research is submitted.

The researchers installed the Network Analyzer on a computer connected to the same network as the UDP server, selected a network interface connected to the UDP server's network, and applied a filter to capture UDP traffic only in Wireshark. Send packet packet data to the IP address and port number of the UDP server delivered to the NB-IoT device and monitor the network traffic's network traffic to the UDP server sending data packets to the network [17]. Analyzer to verify that the analyzed UDP Packet Server UDP has been received. Packets to ensure that the data transmitted by the NB-IoT device is correct and there are no transmission errors, as shown in Fig. 7.

3.5 Security in data transmission

In conducting this research, UDP is a protocol that transmits data between NB-IoT and a UDP server. However, the User Datagram Protocol (UDP) does not have a reliable mechanism for error detection. Therefore, applications that use UDP must be designed or implemented to accommodate packet loss or corruption. This means the responsibility for ensuring the integrity and reliability of data transmitted over UDP falls on the application layer. It's not a transport class.

Defining credibility and ensuring data integrity. The researcher uses the following techniques:

- Validation of UDP packets, but this validation Does not guarantee that all errors will be detected. Therefore, the application undertaken by the researcher adds validation to the submitted data. To provide an extra layer of error detection.
- Using a checksum application to add a checksum to the submitted data to detect errors. Verification is an algorithm that generates a unique value for each piece of data sent. The recipient side can calculate the checksum of the received data and compare it with the expected checksum. And if the two values do not match, an error has occurred.
- Packet loss detection, it instructs the application to use the packet sequence number to determine if a packet has been lost in transmission. This allows the sender to track which packets have been sent and wait for a response from the receiver before sending the next packet. If the sender does not receive an acknowledgment within the specified period. It can be assumed that the packet was lost and retransmitted.

Using this technique mentioned above, applications can detect and handle errors in traffic between NB-IoT devices and UDP servers, although UDP itself has no error detection or reliability mechanism.

The researcher has adopted an encryption method for authentication for access control, which is a procedure for verifying the identity of the sender and receiver. Includes encryption methods for inspecting traffic between NB-IoT devices and UDP servers.

Using a shared secret method for authentication to encrypt authentication using Python

```
# Define the shared secret key
shared_secret = b'mysecretkey'
# Define the message to be authenticated
message = b'hello, world'
#Generate the message authentication code (MAC)
mac = hmac.new(shared_secret, message, hashlib.sha256).digest()
# Send the message and MAC to the receiver
```

Use public key encryption. Adopt public key infrastructure (PKI) to authenticate the sender and receiver. The sender and recipient each have a public and private key pair. The sender uses the recipient's public key to encrypt the message. And the recipient uses the private key to decrypt the message and verify the identity of the sender.

```
# Define the user attributes
user_attributes = {          'user_id': '123',
    'role': 'admin',
    'location': 'THAILAND'
}
# Define the resource attributes
resource_attributes = {          'resource_id': '456',
    'sensitivity': 'high',
    'location': 'THAILAND'
}
# Generate a JSON Web Token (JWT) containing the user and resource attributes
```

```
jwt_token = jwt.encode({'user': user_attributes, 'resource': resource_attributes}, 'mysecretkey', algorithm='HS256')

# Send the JWT to the receiver
```

The 'jwt' module is used to generate a JWT with 'user_attributes' and 'resource_attributes.' Sign the JWT using a shared secret key. It then sends the JWT to the recipient. The authenticity of the JWT can be verified by verifying the signature using the same shared secret. The receiver can also decode the JWT to retrieve user and resource attributes to determine access control.

Access control defines processes that have access to system resources. Define encryption settings for access control in traffic between NB-IoT devices and UDP servers and use user and resource (ABAC) attributes to define access. Assign attributes to users. According to the job position, and set the level of security. Therefore, access is granted or denied based on the combination of user and resource attributes.

4. EXPERIMENTS

4.1 Fall Detection for Patients or the Elderly

This section analyzes and validates the feasibility and efficiency results of our proposed multilevel threshold algorithm approach for fall detection and data transmission. Security during transmission was the objective of this research.

The experiment was done by installing the device on the experimenter, as shown in Fig. 8. Then, the fall data was collected by four falling characteristics: falling forward, falling backward, falling left, and falling right. The daily life data was collected for sleeping, walking, and sitting. A wearable device contains an Accelerometer sensor, a Gyroscope sensor, an Arduino UNO R3 board, and an NB-IoT (AIS) operator. The direction data of the sensor Accelerometer and Gyroscope will be collected before calculating various values according to the equation and analyzing the value during the fall to find the minimum, maximum, and average values to bring the data to the system to determine the fall conditions.

Summarize data on falling into four types: falling forward, falling backward, falling left, and falling right, shown in Table 1; the daily use of lying, walking, and sitting, shown in Table 2; and the experimental results from Table 3.

Table 1 presents a summary of the data for all four types of falls. The average value of SVM-A is approximately 1.47, with a minimum of 1.25 and a maximum of 1.76. The average value of SVM-G is approximately 167.88, with a minimum of 143.68 and a maximum of 187.25. The average, minimum, and maximum values for pitch are -3.25, -60.13, and 49.59, respectively. The average, minimum, and maximum values for Roll are -1.028, -56.91, and 43.18, respectively. Lastly, the average, minimum, and maximum values for Yaw are -22.40, -37.36, and -4.40, respectively.

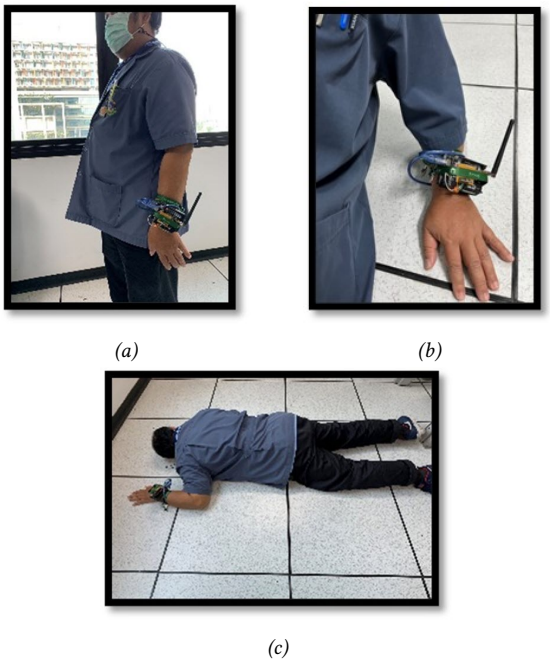

***Fig. 8**: Experimental Settings (a) Initial Setup (1 user, 2 monitoring nodes), (b) Collision with the Floor (1 user; 2 monitoring node), (c) Collision with the floor and stationary stage afterward.*

***Table 1**: Summary of the Minimum, Maximum, and Average of All Falls.*

	Avg-SVM-A	Avg-SVM-G	Avg Pitch	Avg Roll	Avg Yaw
Falling to the left	1.359	179.001	-10.119	-56.91	-4.404
Falling to the right	1.515	187.253	7.651	43.18	-21.051
Falling backwards	1.755	161.619	49.589	10.112	-37.359
Falling forward	1.250	143.680	-60.133	-0.496	-26.777
Avg	1.469	167.888	-3.253	-1.028	-22.397
Min	1.250	143.680	-60.133	-56.91	-37.359
Max	1.755	187.253	49.589	43.18	-4.404

Table 2 summarizes daily activities, including sleeping, sitting, and walking. The average value of SVM-A is approximately 0.97, with a minimum of 0.92 and a maximum of 1.06. The average value of SVM-G is approximately 2.49, with a minimum of 1.59 and a maximum of 3.51. The average, minimum, and maximum values for pitch are 39.88, 17.55, and 75.10, respectively. The average, minimum, and maximum values for Roll are 2.78, 2.39, and 3.426, respectively. Lastly, the average, minimum, and maximum values for Yaw are 5.98, -76.54, and 48.11, respectively.

Experiment to determine the accuracy of four types of falls. They are left-side falls, right-side falls, and backward falls. And those used in daily life are lying,

***Table 2**: Summary of Minimum, Maximum, and Average Values of Daily Activities.*

	Avg-SVM-A	Avg-SVM-G	Avg Pitch	Avg Roll	Avg Yaw
Laying down	1.062	3.512	75.907	2.514	-76.54
Sitting	0.916	1.594	17.55	2.387	46.368
Walking	0.925	2.373	26.195	3.426	48.111
Avg	0.967	2.493	39.884	2.775	5.979
Min	0.916	1.594	17.55	2.387	-76.54
Max	1.062	3.512	75.097	3.426	48.111

***Table 3**: Experimental Results of Fall Detection.*

Activity	Number of Trials	Correct Number	Accuracy
Falling to the left	20	18	90%
Falling to the right	20	17	85%
Falling backwards	20	17	85%
Falling forward	20	18	90%
Laying down	20	20	100%
Sitting	20	20	100%
Walking	20	20	100%

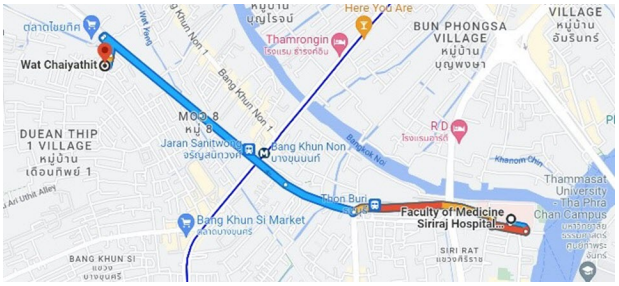

***Fig. 9**: The Location of the Patient was Located by Emergency Responders.*

walking, and sitting. The values collected from all 20 experiments are presented in Table 3. The experiment on the left has 90% accuracy; the correct drop experiment has an accuracy of 85%; the reverse fall test has an accuracy of 85%; the forward fall test has an accuracy of 90%; the sleep test has 100% accuracy; and the sitting test has 100% accuracy.

The walk-through is 100% accurate; however, it is important to note that the prototype installation may introduce experimental errors due to the nature of this proof-of-concept development. If the system is to be applied in real-world scenarios, further development is required to ensure 100% accuracy, as any errors could potentially impact the user.

4.2 Experimental results NB-IoT transmits data with UDP protocol and detects connection contract between NB-IOT device and UDP server

1) Experimental results Connection between NB-IoT device and UDP server

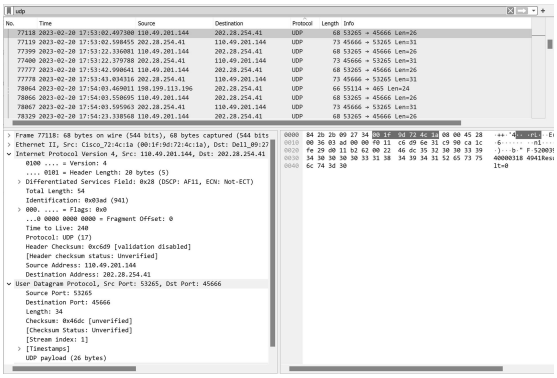

***Fig. 10**: Lists all UDP Packets and Traffic of a Specific IP Protocol using the UDP IP Filter to Inspect UDP Traffic.*

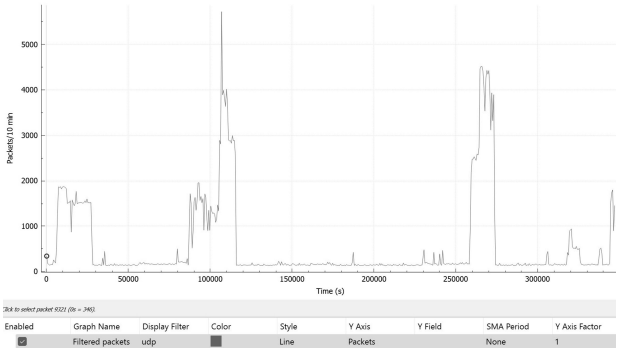

***Fig. 11**: A Graph Listing all UDP Packets being sent on the Network.*

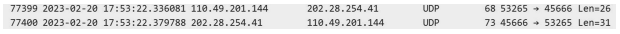

***Fig. 12**: The network source and destination IP addresses of NB-IoT devices and UDP servers.*

Specific connection results between NB-IoT devices and UDP servers. NB-IoT devices, when configured with network settings, APNs, IP addresses, and port numbers, NB-IoT devices establish a socket O packet connection. It is sent to the UDP server using the UDP protocol and port number 42302, and data packets are sent to the UDP server, which receives and processes the data. Record and analyze network traffic between the two endpoints to ensure successful transmission.

2) The result of detecting the connection signal between the NB-IoT device and the UDP server

The results of detecting traffic between NB-IoT and UDP servers and Wireshark will depend on the specific network settings and the data being sent. By analyzing packets captured in Wireshark, details of UDP traffic can be seen, including source and destination IP addresses, UDP sources, and destination ports and port data being sent. This can be used to verify that data is being successfully transmitted through NB-IoT devices and UDP servers, as shown in Fig. 10.

In Fig. 11, the results show that all the UDP packets have been sent on the network. Finding packets

containing the source and destination IP addresses of the NB-IoT device and UDP server, respectively, to verify that they are being transmitted between them.

Analysis of all UDP packets sent on the network source and destination IP addresses of NB-IoT devices and UDP servers. And set the sending and receiving data to be every 20 seconds and display only packets with the specified IP address by typing "ip.addr == x.x.x.x" in the filter box, where x.x.x.x is the specific IP address to filter, as shown in Fig. 12.

4.3 Security of data transmission between NB-IoT and UDP servers

Using a method to secure data transmission between NB-IoT devices and UDP servers, appropriate measures are taken to ensure reliable and secure communications. Some steps can be taken:

1) Optional error detection and correction mechanism: Since UDP does not have a built-in error detection or correction mechanism, the researcher uses tools at the application layer, where we use techniques such as sum checking. Cyclic Redundancy Check (CRC) and Forward Error Correction (FEC) codes are used to detect and correct errors in transmitted data.

2) Using packet size and transmission rate optimization methods to reduce the risk of packet loss or corruption We optimize the size and frequency of the data packets sent. Adjust packet sizes and transmission rates based on network conditions and application-specific requirements.

3) Network congestion control to prevent network congestion and reduce the risk of packet loss, such as packet buffering. Traffic generation and flow control can help manage data flow across the network. and guarantees that packets are delivered in a timely and efficient manner.

4) Use methods to determine security measure for the safety of data transmission through encryption, authentication, and access control. This will protect your data from unauthorized access, interception, or modification.

5) Perform regular testing and inspections. To ensure the stability and security of data transmission over a period of time. It is important to regularly test and verify network and communication protocols. This can help identify potential problems or vulnerabilities and fix them before they become serious.

When following the methods and procedures mentioned above, researchers were able to establish a secure communication system between NB-IoT devices and a UDP server that provides reliable and secure data transmission.

4.4 Comparison

Researching remote fall detection systems using NB-IoT principles and connected technologies for the Internet of Things (IoT) can be inspiring. Increasing safety and independence for seniors constitutes a significant health risk, especially for older people. Thus, with remote fall detection using NB-IoT, caregivers or family members can receive instant alerts when a fall occurs. This may help older people respond faster to signals and increase their safety. This can help older people have a better quality of life.

Compared to other jobs, our method has become more balanced over the past five years. And can meet the current requirements in terms of accuracy. For example, we are creating a secure data transmission system between NB-IoT and UDP servers using hospital data. Which results are reliable. In addition, our system works in real-time on local devices, which require higher-performance resources.

The researchers also compared our algorithm with other algorithms. Tao Xu, Wei Sun1, Shaowei Lu, Ke-ming Ma, and Xiaoqiang Wang [18]; and Faisal Hussain, Fawad Hussain, Muhammad Ehatisham-ul-Haq, Muhammad Awais Azam [19]; and Jian He, Chen Hu, and Xiaoyi Wang [20]. We used a 3-axis fall detection method. Accelerometer and Gyroscope sensor, and by using them in conjunction, it has high accuracy. In addition, we added a way to connect the server to the web application. And increase security. The limitation of the above research with this comparison is that it does not yet have a connection to the server or web application. It is also incompatible with the LINE application and was developed to provide notifications to the LINE application of related parties to increase efficiency [10].

5. CONCLUSION

This research aims to develop a prototype device that enhances the safety of patient and elderly care using Internet of Things principles and sensor technology. The device incorporates an accelerometer and gyroscope to improve the accuracy of fall detection and utilizes a low-power NB-IoT network for communication. The developed device is designed to detect four types of falls: left and right falls, as well as falls forward and backward. It distinguishes these falls from everyday activities such as lying down, walking, and sitting based on experimental measurements. The system has been verified to have an efficiency of 93%.

However, it is important to note that this research is still in the prototype development stage and has certain limitations. These limitations include:

1) One person can only use one set of equipment.
2) The device's accuracy may be affected if the patient's lying position prevents it from lying flat or falling properly.
3) The fall detection device requires a sufficient level of acceleration to detect a fall. If a fall occurs without significant acceleration, the device may not be able to detect it.
4) Occasionally, the device may send multiple values in succession due to processing time variations. This behavior should be considered during data analysis.
5) Since the research is based on a single-subject trial,

there may be discrepancies when applied to other individuals. Calibration and customization may be required to adapt the system to different individuals.

The implementation of data transmission security includes actions such as fault detection and remediation, optimization of packet size and transmission rate, network congestion control, and conducting regular testing and inspections. Researchers have successfully established a secure communication system between NB-IoT devices and a UDP server, ensuring reliable and secure data transmission.

NB-IoT typically utilizes the UDP protocol, which is straightforward and well-suited for NB-IoT due to its low resource consumption and the absence of the need to establish a connection for data transmission. However, it's important to note that when a UDP packet travels over the Internet, all packet data becomes visible to third parties. To ensure security, data is encrypted end-to-end using appropriate technologies, and the responsibility of verifying and decrypting the data lies with the cloud server. In summary, addressing these limitations through ongoing research and improvement, along with the implementation of data transmission security measures, will enhance the usability, accuracy, and overall effectiveness of the device. The results of detecting traffic between NB-IoT and UDP servers and Network Analyzer can be used for reliably detecting traffic between NB-IoT devices and UDP servers [3].

REFERENCES

[1] M. T. Buyukakkaslar, M. A. Erturk, M. A. Aydin, and L. Vollero, "LoRaWAN as an e-Health Communication Technology," *2017 IEEE 41st Annual Computer Software and Applications Conference (COMPSAC)*, Jul. 2017.

[2] P. Pierleoni, A. Belli, L. Palma, M. Pellegrini, L. Pernini, and S. Valenti, "A High Reliability Wearable Device for Elderly Fall Detection," *IEEE Sensors Journal*, vol. 15, no. 8, pp. 4544–4553, Aug. 2015.

[3] W. Ayoub, A. E. Samhat, F. Nouvel, M. Mroue, and J.-C. Prevotet, "Internet of Mobile Things: Overview of LoRaWAN, DASH7, and NB-IoT in LPWANs Standards and Supported Mobility," *IEEE Communications Surveys & amp; Tutorials*, vol. 21, no. 2, pp. 1561–1581, 2019.

[4] S. Popli, R. K. Jha, and S. Jain, "A Survey on Energy Efficient Narrowband Internet of Things (NBIoT): Architecture, Application and Challenges," *IEEE Access*, vol. 7, pp. 16739–16776, 2019.

[5] M. Kanj, V. Savaux, and M. Le Guen, "A Tutorial on NB-IoT Physical Layer Design," *IEEE Communications Surveys & amp; Tutorials*, vol. 22, no. 4, pp. 2408–2446, 2020.

[6] C. Chakraborty, B. Gupta, and S. K. Ghosh, "A Review on Telemedicine-Based WBAN Framework for Patient Monitoring," *Telemedicine and e-Health*, vol. 19, no. 8, pp. 619–626, Aug. 2013.

[7] F. Nasri, N. Moussa and A. Mtibaa, "Internet of Things: Intelligent system for healthcare based on WSN and android," *2014 World Congress on Computer Applications and Information Systems (WCCAIS)*, Hammamet, Tunisia, 2014, pp. 1-6.

[8] X. Ming, Zhan and C. Xu, "A Multimedia Telemedicine System in Internet of Things," *International Conference on Information and Multimedia Technology (ICIMT 2010)*, Dec. 2010, pp. 180-187.

[9] W. Manatarinat, S. Poomrittigul, and P. Tantatsanawong, "e-Health Internet of Thing Platform on Low Power Wide Area Network: A Case Study of Siriraj Hospital, Mahidol University," *International Technical Conference on Circuits/Systems, Computers and Communications (ITC-CSCC)*, Jul. 2018, pp. 216-219.

[10] S. Koonphak, W. Manatarinat, and P. Tantatsanawong, "Prototype Development of Emergency Medical service using Narrow Band Internet of Things," *Thai Medical Informatics Association (TMI)*, pp. 86-92, Nov. 2020.

[11] K. Chaikot, W. Manatarinat, and P. Tantatsanawong, "Prototype Development of Falling Tacking System of Patient or Eldery People using Internet of Things," *Thai Medical Informatics Association (TMI)*, pp. 79-85, Nov. 2020.

[12] A. Mdhaffar, T. Chaari, K. Larbi, M. Jmaiel and B. Freisleben, "IoT-based health monitoring via LoRaWAN," *IEEE EUROCON 2017 -17th International Conference on Smart Technologies*, Ohrid, Macedonia, 2017, pp. 519-524.

[13] W. Manatarinat, S. Poomrittigul and P. Tantatsanawong, "Narrowband-Internet of Things (NB-IoT) System for Elderly Healthcare Services," *2019 5th International Conference on Engineering, Applied Sciences and Technology (ICEAST)*, Luang Prabang, Laos, 2019, pp. 1-4.

[14] Y. YIN, Y. Zeng, X. Chen, and Y. Fan, "The internet of things in healthcare: An overview," *Journal of Industrial Information Integration*, vol. 1, pp. 3–13, Mar. 2016.

[15] X. Zhang, M. Zhang, F. Meng, Y. Qiao, S. Xu, and S. Hour, "A Low-Power Wide-Area Network Information Monitoring System by Combining NB-IoT and LoRa," *IEEE Internet of Things Journal*, vol. 6, no. 1, pp. 590–598, Feb. 2019.

[16] T. Xu, W. Sun, S. Lu, K. Ma, and X. Wang, "The real-time elderly fall posture identifying scheme with wearable sensors," *International Journal of Distributed Sensor Networks*, vol. 15, no. 11, Nov. 2019.

[17] [17] P. Chaudhary, V. Kashyap, N. Sonwal, P. Panwar, M. Dadheech, M. Bhatt, and M. Jain. "Analysis of Network Traffic Using Wireshark. Journal of Computer Networks," *International Conference on Muti-Disciplinary Application & Research Technologies (ICMART)*, vol. 10, May. 2023.

[18] F. Hussain, F. Hussain, M. Ehatisham-ul-Haq, and

M. A. Azam, "Activity-Aware Fall Detection and Recognition Based on Wearable Sensors," *IEEE Sensors Journal*, vol. 19, no. 12, pp. 4528–4536, Jun. 2019.

[19] J. He, C. Hu, and X. Wang, "A Smart Device Enabled System for Autonomous Fall Detection and Alert," *International Journal of Distributed Sensor Networks*, vol. 12, no. 2, Feb. 2016.

[20] S. M. Riazul Islam, Daehan Kwak, M. Humaun Kabir, M. Hossain, and Kyung-Sup Kwak, "The Internet of Things for Health Care: A Comprehensive Survey," *IEEE Access*, vol. 3, pp. 678–708, 2015.

Wiraphon Manatarinat completed his master's degree in engineering at Silapakorn University in Thailand. He holds a position as an engineer in the Information Technology Department at the Faculty of Medicine Siriraj Hospital, Mahidol University, Thailand. His works concerned with telemedicine system and he is also interested in researching about Internet of Things (IoT) for telemedicine. Currently, he is also a president of the Telemedicine Network Society Thailand (TENST) and holds a committee position in the Thailand Research Education Network Association (ThaiREN).

Panjai Tantatsanawong an Associate professor at Computing Department, Faculty of Science, Silpakorn University. He received Doctor of Technology in Computer Science, Asian Institute of Technology (AIT), Thailand in 2000. He received Master of Science in Computer Science, Chulalongkorn University, Thailand in 1992 and Bachelor of Science, Mahidol university, Thailand in 1984. His research interests are data and information processing, internet of things, computer network and artificial intelligent.

Suvit Poomrittigul received a D.Eng. in Information Science and Control Engineering from Nagaoka University of Technology, Japan in 2014. He received his M.Eng. in Computer Engineering from Chulalongkorn University and B.Eng. in Telecommunication Engineering from King Mongkut's Institute Technology Ladkrabang (KMITL), Thailand in 2009 and 2005, respectively. From, 2009 to 2022, he worked as a lecturer and an Assistant Professor at the Department of Software Engineering and Information System, Pathumwan Institute of Technology, Thailand. He is currently a lecturer at the School of Information Technology, KMITL, Thailand. His current research interests include Artificial Intelligence, Image Processing, Intelligent Transportation System, Internet of Things, Computer Simulation and Information System.

Sakol Udomsiri received the B.Eng. degree in Electrical Engineering from Mahanakorn University of Technology (MUT), Thailand in 1997, the M.Eng. degree in Electrical Engineering from King Mongkut's Institute Technology Ladkrabang (KMITL), Thailand in 2001. and the D.Eng. in Information Science and Control Engineering from Nagaoka University of Technology (NUT), Japan in 2014. He is currently an Associate Professor at the Faculty of Engineering, Pathumwan Institute of Technology (PIT), Thailand. His research interests include image signal processing, speech signal processing, signal processing for machine learning and deep learning and artificial intelligence.