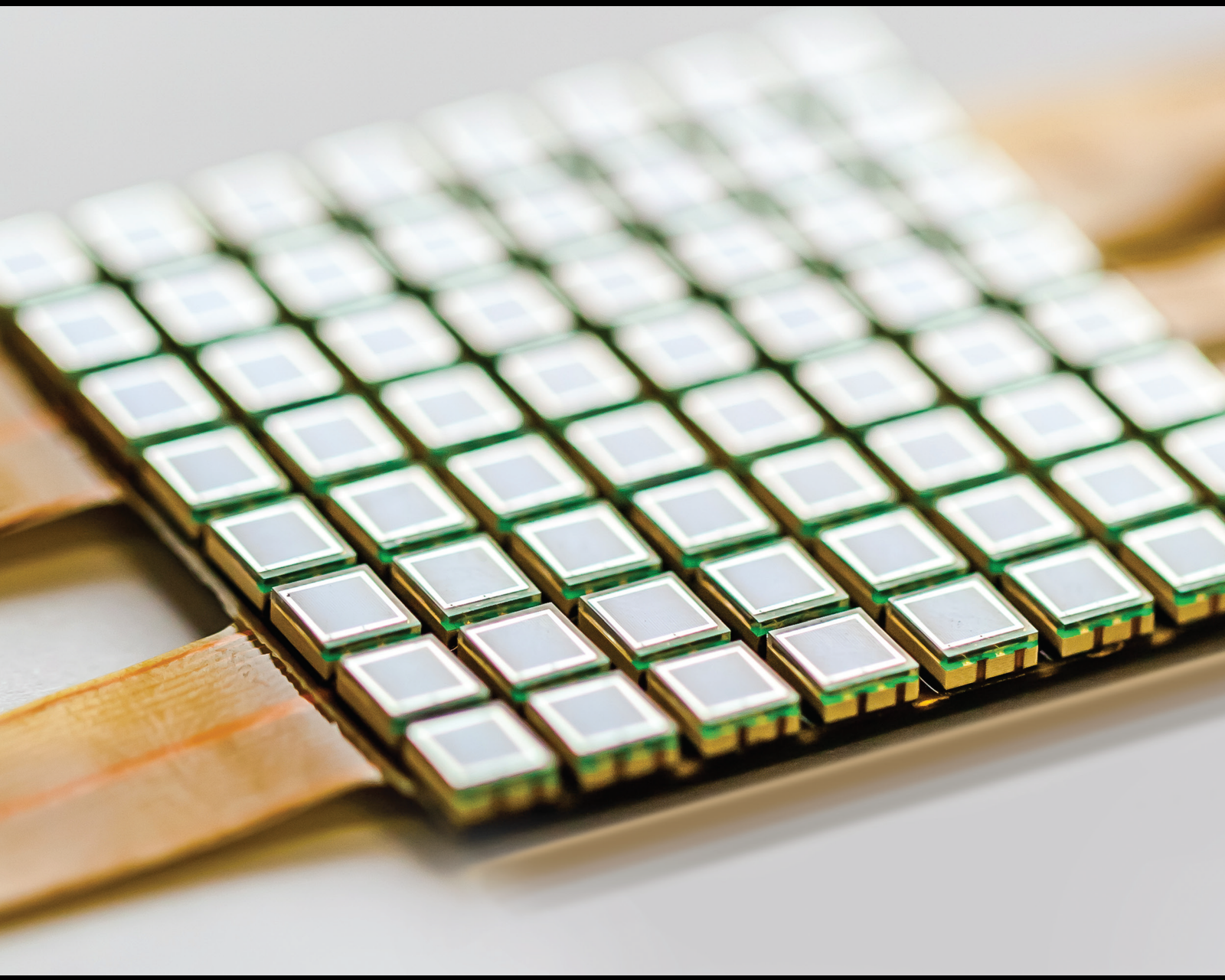
Journal of Sensors

Intelligent Sensors and Internet of Medical Things for Smart Healthcare

Lead Guest Editor: Dong Chen
Guest Editors: Balachandran Manavalan and Xiucai Ye



Intelligent Sensors and Internet of Medical Things for Smart Healthcare

Journal of Sensors

Intelligent Sensors and Internet of Medical Things for Smart Healthcare

Lead Guest Editor: Dong Chen
Guest Editors: Balachandran Manavalan and Xiucai Ye

Contents

Hindawi
Journal of Sensors
Volume 2023, Article ID 1282309, 12 pages
https://doi.org/10.1155/2023/1282309

Research Article

eMeD: An Experimental Study of an Autonomous Wearable System with Hybrid Energy Harvester for Internet of Medical Things

Yung-Wey Chong,[1] **Widad Ismail**,[2] **and Kok-Lim Alvin Yau**[3]

[1]*National Advanced IPv6 Centre, Universiti Sains Malaysia (USM), Penang 11800, Malaysia*
[2]*School of Electrical and Electronic Engineering, Universiti Sains Malaysia (USM), Nibong Tebal 14300, Malaysia*
[3]*Lee Kong Chian Faculty of Engineering and Science, Universiti Tunku Abdul Rahman (UTAR), Kajang 43200, Malaysia*

Correspondence should be addressed to Yung-Wey Chong; chong@usm.my

Received 9 September 2022; Revised 3 January 2023; Accepted 5 April 2023; Published 25 April 2023

Academic Editor: Abdellah Touhafi

We propose and experimentally validate a hybrid energy harvester embedded in a wearable system used to measure real-time information, such as body temperature, heartbeat, blood oxygen saturation (SpO_2), and movement (or acceleration) of human body in real time. This hybrid energy harvester, or in short eMeD, has a unique design that can improve the energy efficiency of the overall wearable system and extract more energy from ambient sources. Specifically, the wearable system is integrated with a hybrid photovoltaic-radio frequency (RF) energy harvester as the power source to prolong its lifetime and reduce the dependence on battery energy. Experimentally, the current consumption of the wearable system with load switching and event management algorithm improved from 31 mA to 18.6 mA. In addition, the maximum conversion efficiency is 14.35%. The experimental results illustrate a sustainable and long-term monitoring operation for Internet of Medical Things systems.

1. Introduction

Wearable system constitutes a new technological paradigm for Internet of Medical Things (IoMT). According to Cisco Systems [1], the number of connected wearable devices is expected to increase from 395 million in 2018 to 1,105 million in 2022. Further, IoMT growth of 21% from $72.5 billion in 2020 to $188.2 billion by 2025 has been estimated by Markets and Markets [2]. IoMT enables new creations and opportunities in the medical domain to improve the quality of people's lives while reducing healthcare cost. The creation of new application areas has changed the process in healthcare services such as data acquisition [3], clinical decision-making, and patient record management [4].

One of the critical issues in the development of IoMT is power consumption and supply in the long-term use of wearable devices [5]. To ensure user adoption, it is crucial that next-generation power sources increase the wearable device lifetime while greater functional capabilities and comfort are provided. The stagnated battery technology has prompted scientists to find ways to get the most out of their battery-powered devices. Scavenging energy from the environment is important from the viewpoint of wearable devices to reduce their dependence on battery energy [6–8]. The combination of wearable devices and the energy-harvesting technology has created opportunities. According to Brandessence, the energy-harvesting market is valued at $467.1 million in 2020 and it is expected to reach $881.7 million by 2027 [9]. In short, the challenges of IoMT are to achieve lower power requirements, use tiny devices that blend into the body, and reduce maintenance.

To provide a reliable power source containing a large energy density, the power management system of the wearable device must optimize energy conversion and deliver high efficiency and reliable energy. For outdoor applications, solar energy generally offers the best achievable energy density, but it is too dependent of the environment whereby it has energy shortage during night time. With the upsurge

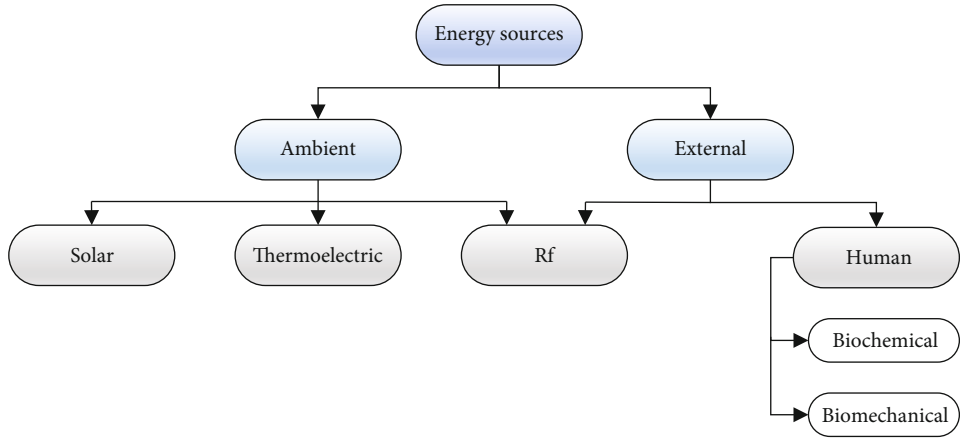


Figure 1: Energy sources for wearable systems [10].

of radio frequency (RF) applications, there has been a strong interest in the RF energy-harvesting technique. Because solar and RF sources are not available at a stable level at all times, it is necessary to design the power management system properly so that the wearable devices can draw power from multiple sources to increase its power availability.

The focus of this research is to design an effective energy-harvesting solution extracting energy from hybrid sources for wearable systems. The combination of hardware and software (i.e., the power management system) can form high-efficiency and sustainable energy supply for wearable systems. Specifically, this research integrates the low-power and state-retentive shutdown mechanism, aggressive power management, and high-efficiency environmental energy conversion into a single wearable system.

There are two main contributions in this research:

(1) An autonomous wearable system powered by hybrid energy. Solar and RF energy harvesters are used to maximize energy harvested so that the wearable system can be used in both indoor and outdoor environments. Due to the novelty of the proposed method, users do not need to consider the illumination effect of the environment

(2) An event-driven sensor management algorithm to preserve energy and meet the requirement of IoMT such as real-time tracking of patients' vital sign

2. Related Work

An energy harvester converts and stores energy from the ambient source to provide continuous power supply to electronic systems, including wearable systems. The energy source can be converted to electrical energy using transducers or converters. The energy sources can be classified into two broad categories, namely, ambient and external sources, as shown in Figure 1 [10]. Ambient sources, such as solar power, thermoelectric power, and RF are available in the surroundings at almost no cost. The characteristics of various ambient sources are unique in terms of predictability, controllability, and conversion efficiency [11]. This is because these sources are affected by time, location, and weather conditions. In contrast, external sources emit energy to the environment with the intent for the energy to be harvested by energy-harvesting system. Such energy sources are predictable and controllable because they are deployed explicitly in the environment.

To add the diversity of energy sources, several hybrid energy-harvesting systems have been proposed in the literature.

Saraereh et al. [12] propose an energy-harvesting protocol that harvests energy from 2.4 GHz RF energy and thermoelectric energy for a network of healthcare devices. The hybrid energy harvester has shown to increase network lifetime by 24% compared to a network without using any energy harvester. Colomer et al. [13] propose the integrated power conditioning circuit, which is designed using the 0.13 μm technology and a full-wave NMOS rectifier, to harvest biomechanical and solar energy. Simulation results show that the rectifier achieves an average efficiency of around 70%.

Mohsen et al. [14] propose a photovoltaic-thermoelectric hybrid energy harvester for wearable systems. The harvester provides energy to power temperature, heartbeats, and accelerator sensors. Nevertheless, the system uses the Bluetooth Low Energy (BLE) module as wireless communication, which is power hungry, to increase the lifetime of the sensor system up to 46 hours only. Zhang et al. [8] propose a hybrid RF-solar energy-harvesting systems that utilise transparent multiport micromeshed antennas for indoor applications. The integration of transparent antennas with photovoltaic (PV) cells has shown to enhance transparency and conductivity. The transparent approach allows space saving since the antenna can be integrated with solar panel and at the same time achieve good antenna efficiency.

Yu et al. [15] propose a flexible transparent antenna with flexible transparent rectifying circuit and an amorphous PV cell. The flexible antenna shows an impedance matching bandwidth of 3.5-3.578 GHz and 4.79-5.09 GHz. The transparent stacked structure enables the wearable device to achieve a higher RF-to-DC conversion efficiency at 13 dBm RF input power. Although it is designed for wearable system, the proposed system has not been tested

Table 1: Comparison between different wearable systems with its characteristics.

Reference	Methodology	Energy sources	Power management	Energy preservation algorithm
[12]	Using simulator to simulate performance of hybrid energy-harvesting protocol	RF and thermoelectric	Conceptual (simulated)	NA
[13]	Integrated power conditioning using 0.13 μm technology and full-wave NMOS rectifier to harvest biomechanical and photovoltaic energy	Biomechanical and photovoltaic	Conceptual (simulated)	NA
[14]	Wearable system that measures vital signs and powered using hybrid energy	Photovoltaic and thermoelectric	Simple DC-DC boost converter	NA
[8]	Using transparent multiport micromeshed antennas	Photovoltaic and RF	Parallel DC-DC boost converter	NA
[15]	Flexible transparent antenna with transparent rectifying circuit and photovoltaic cell	Photovoltaic and RF	ADP5091 only	NA
Proposed eMeD	Wearable system that measures vital signs and powered using hybrid RF and photovoltaic energy	Photovoltaic and RF	ADP5091 with load switch	Yes

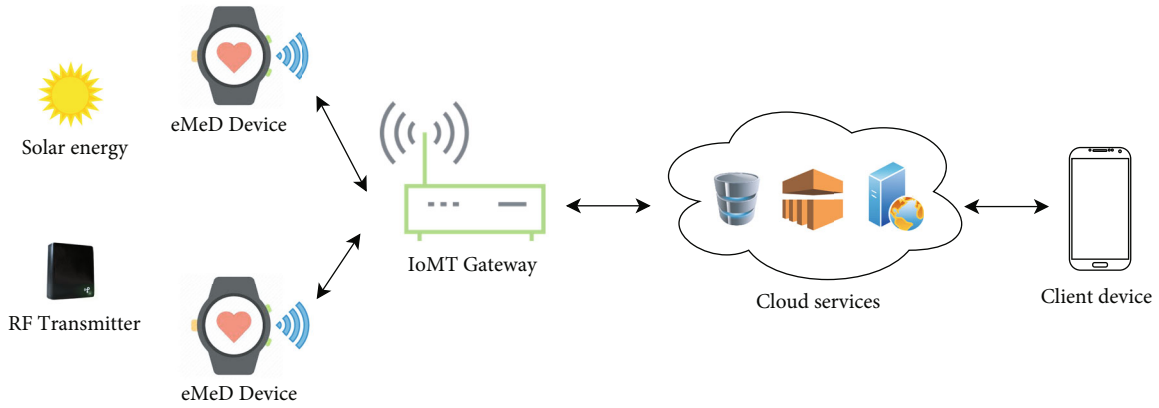


Figure 2: The overall eMeD system.

in actual IoMT environment. The design does not include wearable sensors, and the RF electromagnetic power density is artificially adjusted.

Table 1 shows the comparison between different wearable system with its characteristics.

3. Proposed System Design

Figure 2 presents the overall eMeD system. The eMeD system consists of wearable eMeD devices embedded with vital sign sensors, such as heart rate, temperature, and accelerometer. The eMeD devices are embedded with sensors with a low-power wireless communication module and a hybrid energy harvester. One of the innovations of this system is that the energy harvester is robust against changes of the environment. Specifically, the system consists of a PV panel to harvest energy from the irradiance of sunlight and a RF submodule to harvest energy from external RF. The combination of solar energy and RF energy is exploited to provide a stable energy source to the wearable system from predictable RF energy source and unpredictable solar energy source. In the power management unit, the maximum power point tracking and closed-loop voltage feedback control are used to achieve the highest possible total output power from different energy-harvesting subsystems. Specifically, the input impedance of two different converters is tuned dynamically to optimize the output power transferred from different energy-harvesting sources to the wearable device under a wide range of operating conditions. In addition, the power management unit, which operates at an input voltage spanning from 80 mV to 3.3 V, provides efficient conversion. Specifically, it has a minimum operation threshold whereby a specific condition can trigger a boost shutdown. For example, when solar energy harvester detected low light condition, an efficient power management helps to prolong the battery life and increase the life of wearable devices.

To maximize power efficiency, task execution is optimized in wearable eMeD devices using software. The eMeD device transmits sensing outcomes to an IoT gateway using the ZigBee technology. Conventionally, wearable devices are directly connected to mobile phones via BLE, but it consumes a high amount of energy, making it unsuitable for IoMT applications. The fast duty cycling is implemented in ZigBee, making the eMeD device to consume low power. There are two different profiles, namely, *sleep end nodes*

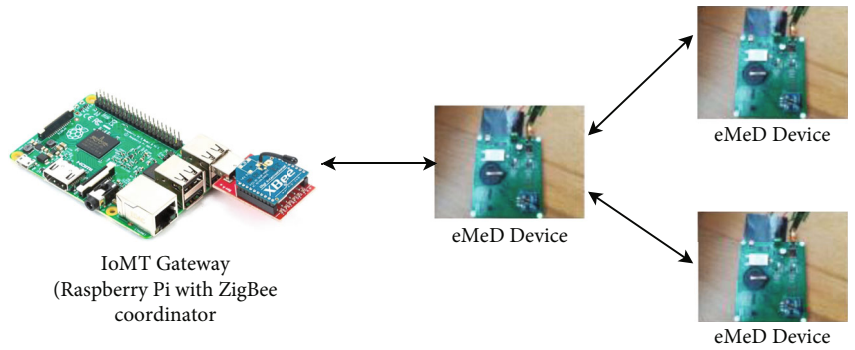


Figure 3: Wireless sensor network in eMeD.

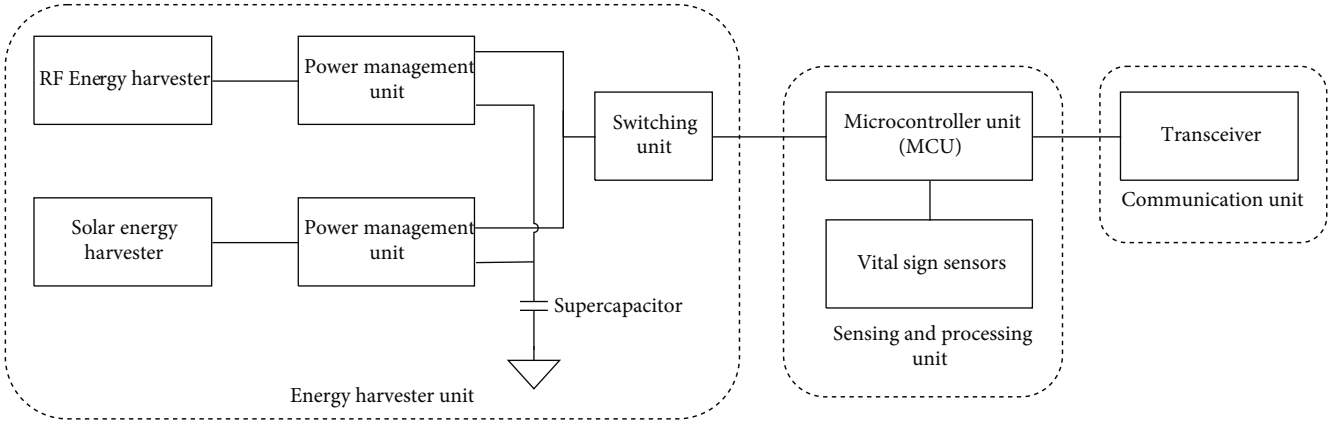


Figure 4: eMeD hardware design.

and *line-powered nodes*. Firstly, a sleep end node performs power management by means of duty cycling with two main states, namely, *hibernation* and *transmission*. A node is inactive in the hibernation state. A node is active in the transmission state. During the transmission state, the hardware abstraction layer is activated, and peripherals such as sensors and microcontroller are enabled for data collection. The node's radio uses carrier collision avoidance to ensure that a channel is available, sends data, and then switches to the sleep mode. Secondly, a line-powered node does not perform power management by means of duty cycling because the node is always awake to act as both parent and end node. Instead, the node uses software optimization to minimize the data transmission time, reduce power consumption, and optimize the power generated by energy harvesters. In contrast to many existing systems in the market, eMeD focuses on achieving an ultralow power in self-sustainable devices. WSN is integrated into eMeD devices so that they can send data to neighboring nodes, which can relay it to the IoT gateway, when patients move to different rooms or areas in a building and become disconnected from the IoT gateway.

In the eMeD system, an IoMT gateway is implemented using Raspberry Pi to handle IoMT devices and filter, preprocess, and encrypt data in the local stage as illustrated in Figure 3. Simple actions are performed at the IoMT gateway because this can reduce the bandwidth required for transmitting a large volume of data to the cloud server. An integrated application layer at the cloud server is also implemented to store a large amount of data and process it so that the data can be visualised in the mobile and web applications. This enables medical practitioners to interact with the platforms. A lightweight messaging protocol (MQTT) is used in the application layer because of its capabilities in providing different quality of service (QoS) levels, message persistence, and multicasting.

This paper presents the embodiment of technologies, including hardware and software, for building a new embedded system as part of the Internet of Medical Things. eMeD makes up for the shortcomings of using numerous cables and cords in the hospital while improving communication efficiency between medical practitioners and patients.

3.1. Hardware Design. eMeD has three subsystems, namely, the sensing and processing unit, the communication unit, and the energy harvester unit, as illustrated in Figure 4.

3.2. Sensing and Processing Unit. The sensing and processing unit consists of a microcontroller unit (MCU) and three (3) on-board sensors, namely, temperature sensor, heart rate sensor, and accelerometer. The core of the sensing and processing unit is the ATMEGA3028P-based MCU that is used to collect and process sensor data. The MCU is also used to run the event-driven sensor management algorithm to reduce the overall power consumption. The low-power, low-cost, and high-performance MCU is

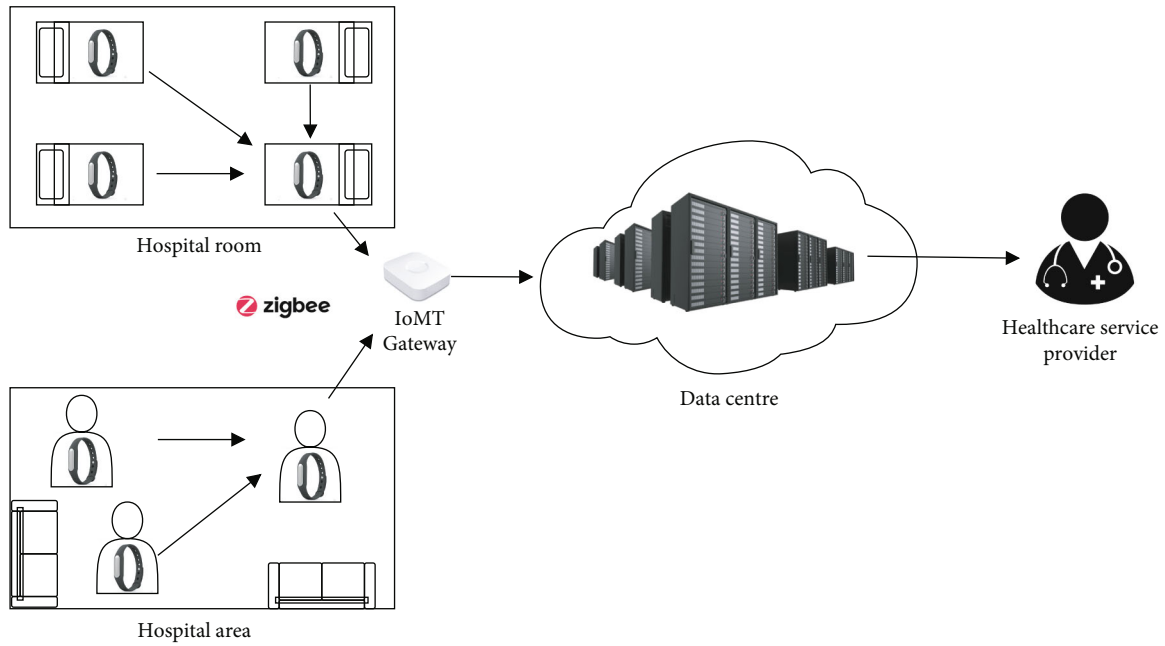


Figure 5: eMeD system deployment in IoMT setting.

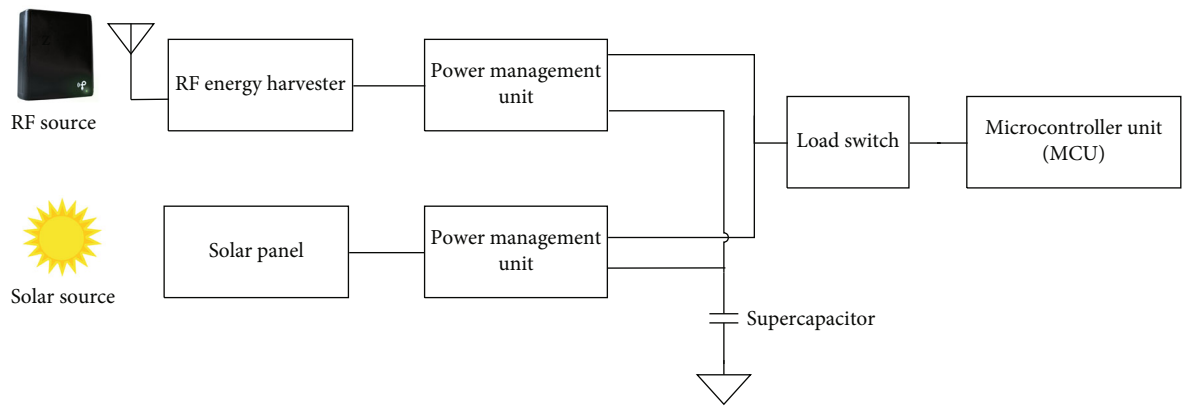


Figure 6: Energy harvester unit embedded in eMeD.

configured to 8 MHz at 3.3 V, and it can be throttled down to 1 MHz at 1.8 V for extremely low-power applications. The MCU has a 32 KB flash memory with a 2 KB SRAM, 6 analog input pins, and 14 digital I/O pins, which are connected to on-board sensors. It also comes with 16 MHz oscillators onboard.

Low-power temperature sensor (MAX30205) [16] is integrated to the eMeD device to measure skin temperature, which is one of the most fundamental monitoring measurements of patients. Intrinsic factors, such as infection or metabolic disturbances, can cause temperature instability in patients. The temperature data is sent to the MCU via the I^2C serial interface. It has a very low power consumption with a typical operating current of 600 μA and supply voltage of 3.0 V. A photoplethysmograph (PPG) heart rate sensor (MAX30102) is also connected to the eMeD device to monitor a patient's heart rate and peripheral capillary oxygen saturation (SpO_2). The PPG sensor measures changes in blood vessel contraction and expansion using LEDs and photodiodes. As arterial pulsations fill the capillary bed, the volumetric changes of the blood vessels modify the absorption, reflection, or scattering of the incident lights, such that the resultant reflective and transmittal lights represent the timing of cardiovascular events, including heart rate and oxygen saturation [17].

A 3-axis microelectromechanical system (MEMS) accelerometer (ADXL362) [18] is integrated into eMeD device so that it can capture motion rhythms used for artifact removal. ADXL362 is an ultralow power sensor that consumes (a) less than 2 μA when the output data rate is 100 Hz and (b) only 270 nA when it is in the motion-triggered wake-up mode, which has an adjustable threshold of the sleep/wake motion activation. It utilises capacitive sensing to sense the displacement of the proof mass, which is proportional to the applied acceleration. Therefore, the presence of acceleration above a threshold for a specified time period represents a motion (or activity), and the motion remains until there is a lack of acceleration above the threshold. In this project, the

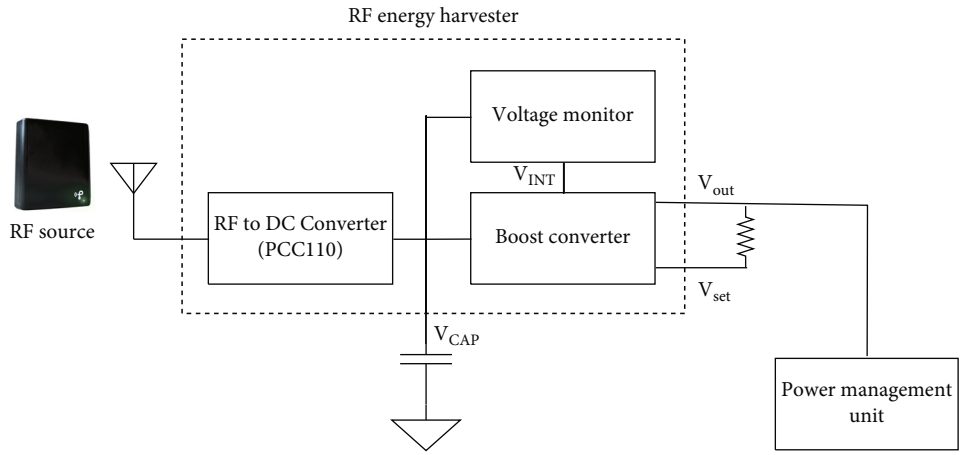


Figure 7: Block diagram of RF energy harvester in eMeD.

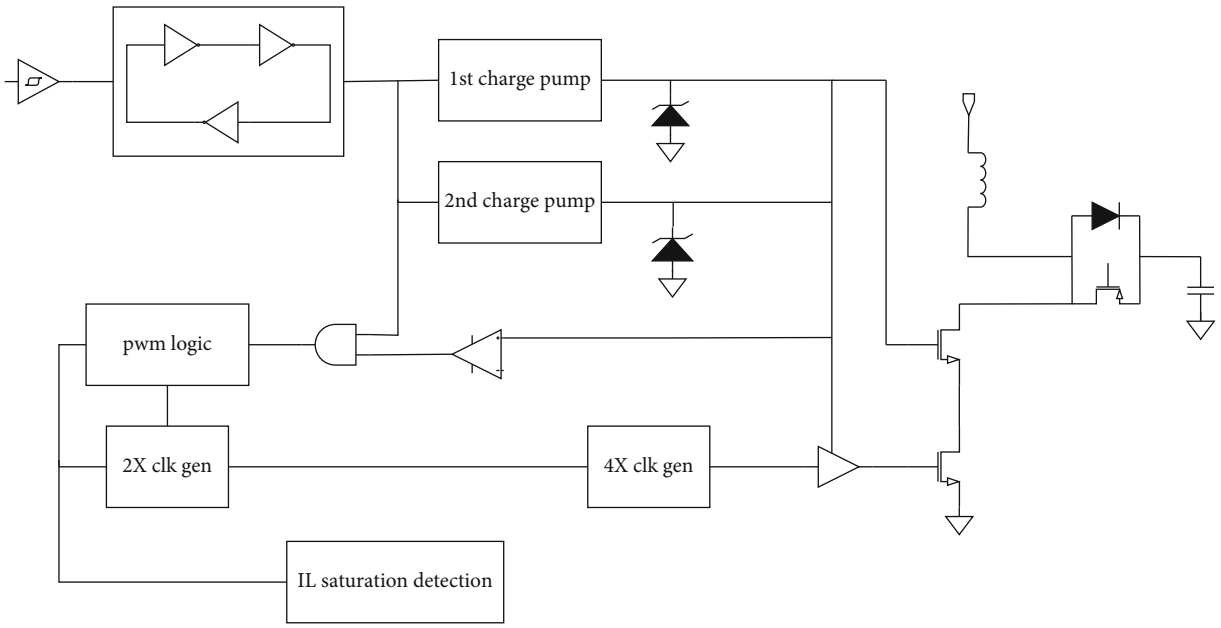


Figure 8: Power management unit [22].

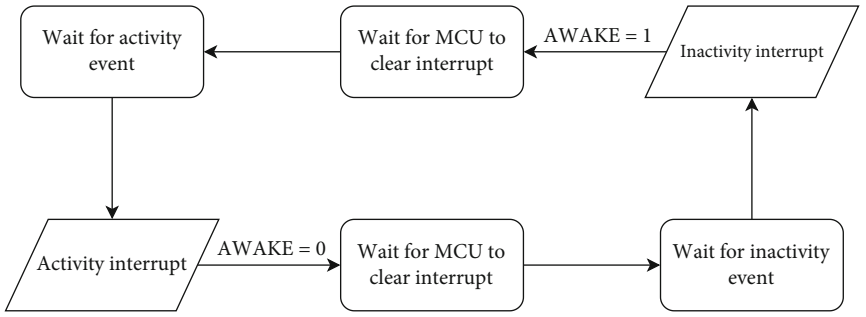


Figure 9: Flowchart illustrating activity and inactivity operation of accelerometer that is utilised in the event-driven sensor management algorithm.

motion-triggered wake-up mode is used to detect the presence of motion at extremely low power consumption of 270 nA at a supply voltage of 2.0 V. Using the motion-triggered wake-up mode allows the eMeD device to be switched off and powered down until activity is detected. The accelerometer measures acceleration only about six times per second to determine whether motion is present. When a motion is detected, the accelerometer (a) switches

into the full bandwidth measurement mode, (b) triggers an interruption signal called the AWAKE bit to MCU, and (c) wakes up downstream circuitry based on the configuration. In the wake-up mode, all accelerometer features are available except the activity timer. All registers can be accessed, and real-time data can be read and/or stored through FIFO. Such activity and inactivity detection using accelerometer can be used to trigger the wake-up mode of the heart rate and temperature sensors.

3.3. Communication Unit. To help the healthcare service provider in obtaining data from eMeD devices, a ZigBee transceiver transmits the sensor data of a patient to an IoMT gateway through wireless communication. The ZigBee-based module XBee-Pro, which follows the IEEE 802.15.4 standard for the RF module, enables a low-cost and low-power WSN. The ZigBee module is built with Silicon Labs EM357 system-on-chip [19] that operates in the 2.4 GHz ISM radio bands and offers low latency solutions with low power. The ZigBee module is also equipped with a NXP MC9S08QE32 [20] application processor that supports bootloader and controls the EM357 radio. The ZigBee module interfaces with the MCU through the universal asynchronous receiver/transmitter (UART) serial connection.

The eMeD system supports the tree configuration to increase the reliability and availability of the system. The transceiver of eMeD devices is set to the router mode whereas the transceiver of the IoMT gateway is set to the coordinator mode. The selection of different modes for the eMeD system allows the expansion of multipatient monitoring in a medical center as shown in Figure 5. eMeD devices can join an existing IoMT network and send, receive, and forward information. So, each eMeD device can act as a messenger for communication with other eMeD devices which are located outside the transmission range of the IoMT gateway. Nevertheless, to preserve the harvested energy, a separate router relays messages from one node to another node if patients are located outside the transmission range of the IoMT gateway. Unlike eMeD device that can go into the sleep mode, a ZigBee router is set up to always on mode so that it is always available for routing and forwarding packets. The IoMT gateway acts as a ZigBee coordinator to gather data from eMeD devices and forward it to the data center using TCP/IP sockets over an Ethernet network.

For data security, all data is encrypted using 128-bit symmetric encryption called AES. When new eMeD device is added to the network, AES authenticate and encrypt the data without interfering neighboring eMeD devices.

3.4. Energy Harvester Unit. To prolong IoMT operation, the energy harvester unit collects and stores a limited amount of energy extracted from two energy sources. Using two different kinds of energy sources reduces their effects to the environment condition. RF energy is used to charge the supercapacitor and operate eMeD devices when patients are inside the hospital building, and solar energy is used when patients are outside the hospital building. Such design reduces the dependence on a single energy source and the needs to manually recharge the eMeD devices. Using hybrid energy sources, the eMeD system can be deployed in both indoor and outdoor environments.

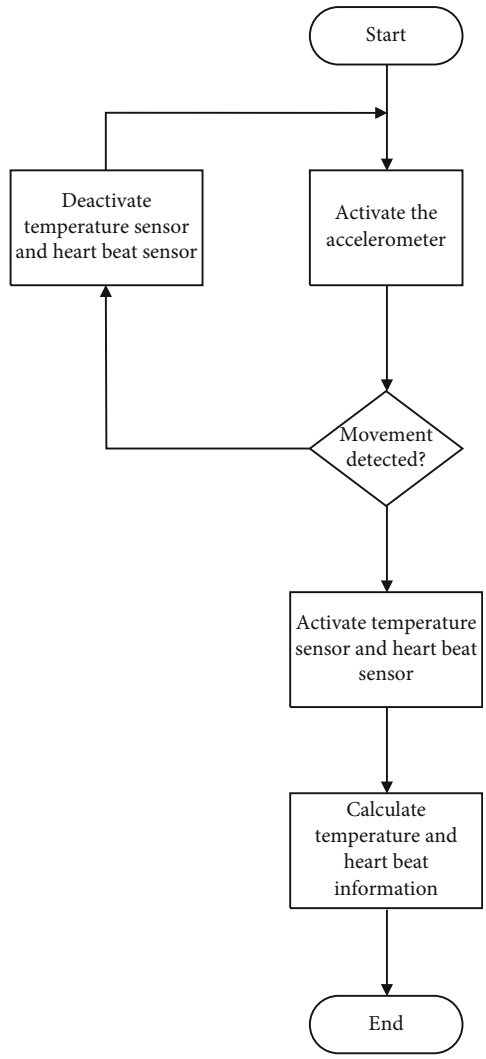


Figure 10: Event-driven sensor management algorithm to improve the accuracy of eMeD devices.

Figure 6 illustrates the block diagram of the energy harvester unit for eMeD devices. The energy harvester unit consists of two (2) power management units (PMUs), RF energy harvester, solar energy harvester, and load switch.

3.4.1. Supercapacitor. Unlike the conventional rechargeable battery-based energy storage systems that (a) require frequent periodic maintenance and (b) have limited number of recharge/discharge cycles, eMeD utilises a supercapacitor to store the harvested energy. The supercapacitors have unlimited charging cycles; thus, it can minimize the need to change the battery once it reaches the maximum number of recharge/discharge cycles. In addition, supercapacitors can minimize the environmental impact when batteries are not disposed of properly.

3.4.2. RF Energy Harvester. The RF energy harvester harvests energy from RF signals, and it consists of a RF-to-DC converter, voltage monitor, and boost converter. The RF-to-

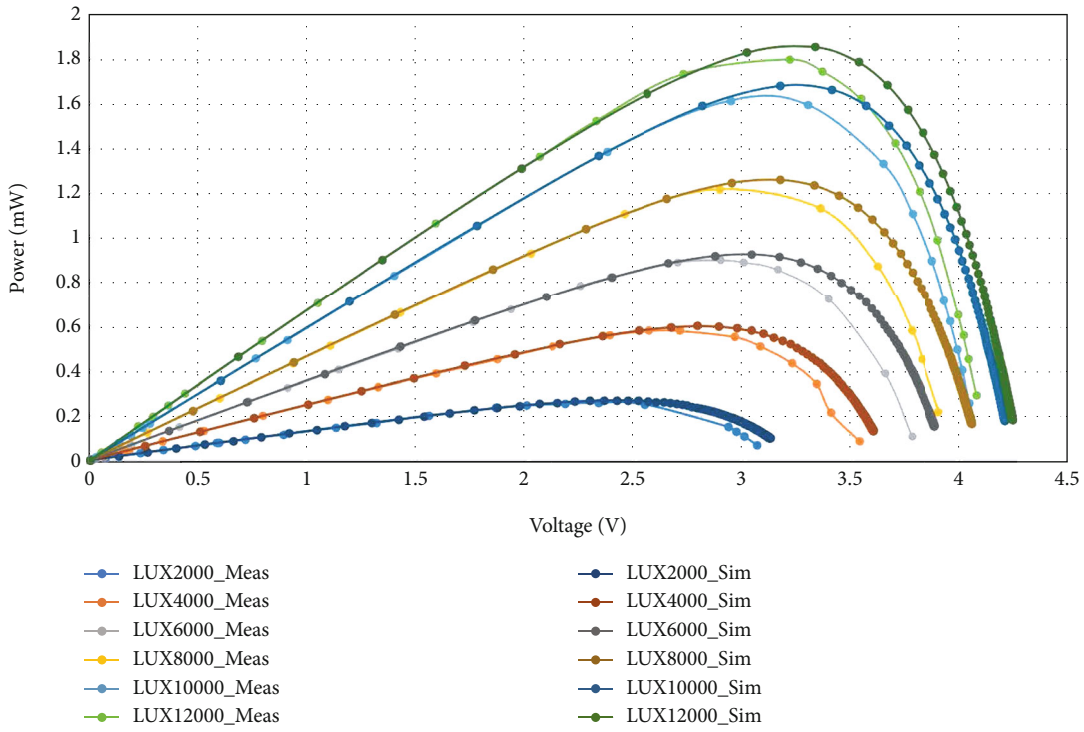


Figure 11: Output power versus output voltage using different light intensities.

Table 2: Conversion efficiencies of the photovoltaic cell.

Lux	2000	4000	6000	8000	10000	12000
Lux in mW/m^2 conversion	15.8	31.6	47.4	63.2	79.0	94.8
Input power (mW/cm^2)	0.0608	0.1217	0.1825	0.2433	0.3042	0.3650
Output power (mW/cm^2)	0.0069	0.0156	0.0241	0.0324	0.0436	0.0482
Conversion efficiency	11.40	12.81	13.18	13.31	14.35	13.19

DC-integrated circuit Powercast P2110 is used to capture energy from a 915 MHz-centered RF transmitter. RF energy reaches the RF-to-DC converter IC (PCC110) of an eMeD device as illustrated in Figure 7. Energy is stored in a capacitor connected to V_{cap} or fed to the boost converter (PCC210). The DC-DC boost converter is used to boost and regulate the output voltage of the RF rectifier. The boost converter has a default output of 3.3 V, and it can be adjusted by adding resistors to V_{set}. The impedance matching circuit maximizes the received power at the antenna, the RF-to-DC converter transforms the AC RF signals into DC voltage, and the DC-to-DC converter amplifies the DC voltage level from the RF-to-DC conversion unit to allow ultralow voltage operations. When V_{cap} reaches its maximum value, a digital line V_{INT} is set to high by the voltage monitor and the boost converter turns on, so V_{out} rises to the predetermined output voltage of the DC-DC boost converter. The supercapacitor then discharges until it reaches its minimum value, causing the voltage monitor to set the V_{INT} value to low. This causes the boost converter to switch off until the supercapacitor charges up to its maximum value. This strategy allows the RF energy harvester unit to be decoupled from the load (e.g., sensors and microcontroller) and, successively, to decouple the load from the harvester, giving the voltage regulator enough energy to be manipulated. Generally, the conversion efficiency is determined for a defined input power signal level, at a given frequency once the load impedance is specified. To have good values of this characteristic, the input RF power matching is one of the more delicate aspects in the design of the antenna-rectifier circuitry.

3.4.3. Load Switch. The load switch limits the current drawn from the supercapacitor from the harvester to ensure the current is not discharged from the supercapacitor during when energy is not harvested. Since the output voltage from the RF energy harvester depends on the amount of traffic in cellular communications or on the DTT power in the scavenging area, there is a need to control the output voltage delivered to precharge supercapacitor from the front-end circuit. The minimum voltage that the precharge capacitor needs to achieve to initiate the procedure from the buck-boost circuit is 1.8 V, whereas the maximum voltage must be no more than the voltage the supercapacitor bank is

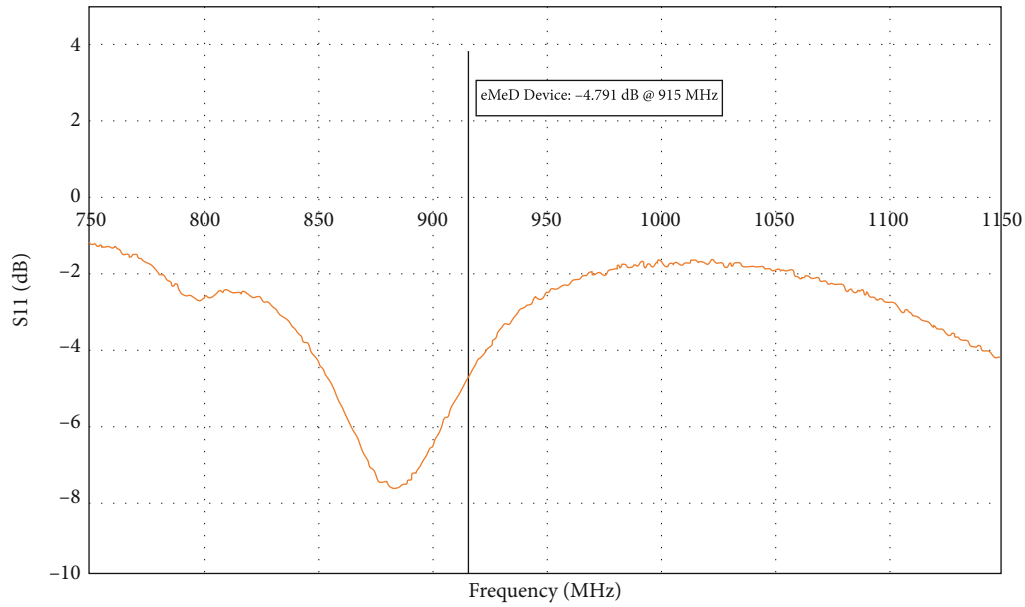


Figure 12: Return loss of eMeD device.

designed for. In terms of voltage supply, the buck-boost circuit takes a maximum tolerance of ~5%, while the maximum output ripple will be at most 2%. The RF energy is fed into the device where it reaches the RF-to-DC converter IC. There, the energy is stored in a capacitor connected to V_{CAP}.

3.4.4. Power Management Unit. An integrated circuit PMU by Analog Devices, ADP5091 [21], is connected to PV panel and RF energy harvester submodules to extract the maximum possible energy from both energy sources. The PMU integrates the maximum power point tracking circuit (MPPT) to continuously adapt the input impedance of the connected PV panels so that the maximum power can be extracted and efficiently transferred to the supercapacitor. The MPPT control keeps the input voltage ripple within a fixed range to maintain a stable DC-to-DC boost conversion. The dynamic sensing mode and no sensing mode, both programming regulation points of the input voltage, allow extraction of the highest possible energy from the harvester. It provides efficient conversion of the harvested limited power from a 6 μW to 600 mW range with submicrowatt operation losses.

An efficient energy-harvesting method is required to generate subthreshold voltage levels because the internal voltage-voltage conversion circuitry must be energized. The cold start-up circuitry in the PMU contains two charge pumps to allow the regulator to start operating at an input voltage of 380 mV as shown in Figure 8. The first charge pump controls cascading devices to protect the voltage threshold start-up switch which the second charge pump is used to make sure that the output voltage is good. The inductor saturation detection is used to minimize the start-up current. Such cold start-up circuitry design can bootstrap conversion circuits when the voltage level falls below the minimum level. As input voltage rises, the PMU disables the start-up converter and allows operations to proceed with the boost converter, which is very helpful in suboptimal situations.

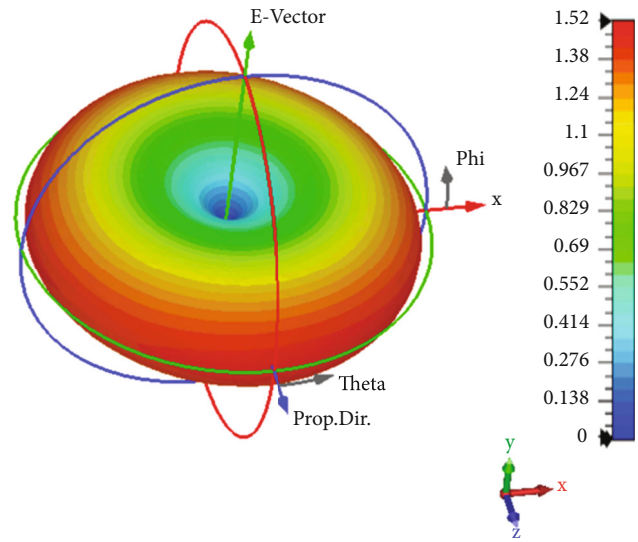


Figure 13: 3D linear radiation pattern of the designated antenna.

3.5. System Software Design. Besides hardware design, the software design plays an important role to preserve the energy of the eMeD system comprised of eMeD devices, the IoMT gateway, and WSN. The software design is separated into two (2) segments, namely, an event-driven sensor management algorithm implemented in MCU and the communication software stack in IoMT.

In eMeD devices, event-driven sensor management is implemented in the MCU. The benefit for such design is twofold. Firstly, the event-driven sensor management

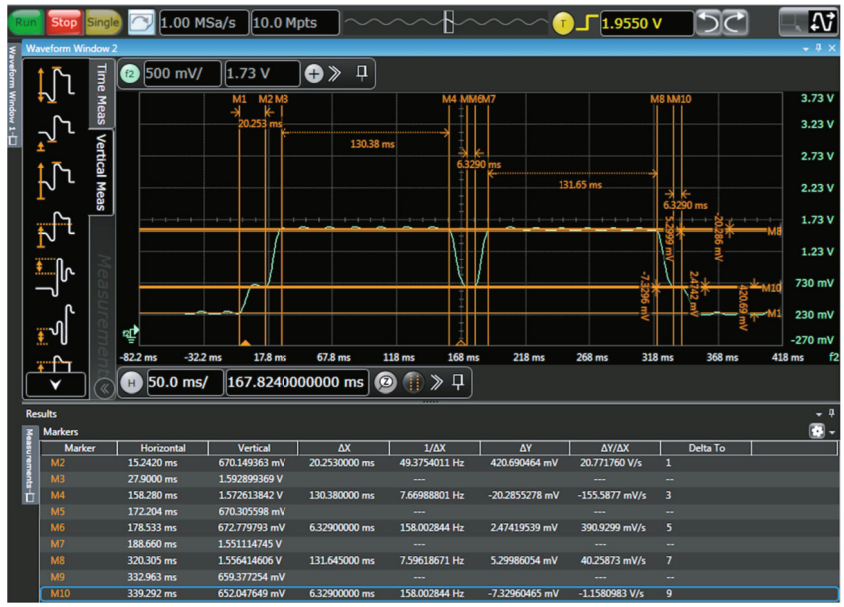


Figure 14: Current consumption of eMeD during active and idle modes.

Figure 15: Voltage of eMeD during active and idle modes with event-driven management algorithm.

algorithm assigns a minimum set of active components at any time and invokes new components only when state transitions happen. The ZigBee module spends majority of its time in the sleep mode to minimize power consumption and wakes up only when data transmission. To minimize the amount of time ZigBee wake-ups, the data acquisition and the transmission are made subsequent of each other. This can prolong the lifetime of the energy storage.

Secondly, before data can be acquired from the patient, the sensor data must be accurate. The event-driven sensor management algorithm fuses data from multiple sensors in managing sensors and providing accurate readings. PPG and temperature sensors are commonly worn on fingers which provide the strongest signal strength. Nevertheless, motion artifact due to body movements may cause temperature and heart noise estimation error [23]. Since eMeD is designed to be worn by patient, large magnitude accelerations will occur due to human movement, causing noise in the heart rate data.

In the event-driven sensor management algorithm, eMeD utilises 3-axis accelerometer to automatically recognise user's movement state before capturing vital sign data. The activity and inactivity detection functions in accelerometer are used concurrently and configured to process in the

linked mode. In the linked mode, activity and inactivity detection are linked to each other such that only one of the functions is enabled at any given time. As soon as an activity is detected, the device is assumed to be moving. Inactivity detection operates when inactivity is detected, and the eMeD device is assumed to be stationary as shown in Figure 9.

Heart rate and temperature sensors are activated when the accelerometer detects the user is in the static mode as illustrated in Figure 10. The processor operates in the active mode, but it does not activate the heart rate and temperate sensors because human movement may affect the readings of the heart rate.

4. Experimental Results and Analysis

Figure 11 shows the power-voltage characteristics of the energy harvester unit under different light intensity. The output power and output voltage are measured using a digital multimeter and light sensor to measure the lux of the light. The system is able to perform maximum power point tracking (MPPT) with the power curves achieving peak near the solar panel output voltage of 3.3 V. The maximum power point (MPP) of the solar panel is between 2.5 V and 3.5 V depending on the light intensity. This shows that the output voltage should be fixed at 3.3 V so that the maximum possible output power can be harvested from the solar panel. With the outdoor conditions being 8,000 to 10,000 lux, the maximum electrical power that the solar panel can harvest ranges from 1,200 to 1,800 μW. Figure 11 illustrates the simulated value track closely with measured value.

Table 2 shows measured conversion efficiencies of the photovoltaic cell at different input power values with solar energy harvester area of 38.5 cm^2. As shown in the table, the maximum conversion efficiency is 14.35%. The conversion efficiency is high as compared to existing work by Mohsen et al. [14] due to the efficient power management design.

As shown in Figure 12, S_{11} for an eMeD device at 915 MHz is -4.791 dB. Based on these results, the actual RF energy harvested by the embedded eMeD system is 66.9% of the incident power.

Figure 13 shows the far-field 3D radiation pattern of the designed dipole antenna at 915 MHz that is used in the hybrid energy harvester. The antenna is perfectly omnidirectional with a linear gain of 1.52 and a peak realized gain of 1.82 dBi at 915 MHz. The radiation from the RF antenna decreases to zero along its axis, as is typical for any dipole antenna. The radiation pattern envelope of points with identical radiation intensities for a doughnut-like form, with the antenna axis flowing through the hole in the middle of the doughnut, which is comparable to an omnidirectional antenna, may be seen in the three-dimensional graphic. When the source outputs unbalanced power, the sleeve balun in the antenna removes impedance mismatches.

Oscilloscope and current sense amplifier are used to verify the overall current consumption. The sensing resistor value is 0.1 Ω. For this measurement, the sensing resistor value of 0.1 Ω was selected to minimize voltage drop across it, and the selected current sense amplifier has gain of 500 V/V. In this case, the constant multiplier is equal to 0.1*500 = 50 V/A. The oscilloscope plot without event-driven sensor management algorithm is show in Figure 14. The idle current is 245.92 mV/50 V/A = 4.92 mA, busy current = 707.28 mV/50 V/A = 14.14 mA, and transmit current = 1.55 V/50 V/A = 31 mA.

When event-driven management algorithm is used, voltage for the eMeD system dropped as illustrated in Figure 15. Since the oscilloscope probe used for this testing is N2873A 10:1, the actual reading is attenuated by 10 times. Thus, the constant multiplier reduced to 5 V/A. During the idle state, the current consumption is 32.25 mV/5 V/A = 6.45 mA, busy current = 84.49 mV/5 V/A = 16.90 mA, and transmit current = 93 mV/5 V/A = 18.6 mA. This shows that the power consumption improved when event-driven management algorithm is used, making it attractive for IoMT application.

5. Conclusions

This research presents a sustainable photovoltaic-radio frequency PV-RF hybrid energy harvester, and it has been successfully developed and implemented for a self-powered IoMT system. The proposed harvester is implemented to enable eMeD to operate autonomously. The experimental results illustrate the sustainable and long-term monitoring operation for Internet of Medical Things systems through monitoring body temperature, heart rate, SpO_2, and acceleration. The system can be used in both indoor and outdoor environments, making it ideal for IoMT applications.

Data Availability

The experimental data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declared that they have no conflicts of interest regarding this work.

Acknowledgments

The research project is funded by the USM RUI grant (1001/PNAV/8014078) and USM external grant (304/PNAV/6501023/U157).

References

[1] Cisco, "Cisco annual internet report (2018-2023)," 2020, https://www.cisco.com/c/en/us/solutions/collateral/executive-perspectives/annual-internet-report/white-paper-c11-741490.html.

[2] Markets & Markets, "IoT in healthcare market by component, application, end user, and region global forecast to 2025," 2020, https://www.marketsandmarkets.com/Market-Reports/iot-healthcare-market-160082804.html.

[3] K. Saleem, I. S. Bajwa, N. Sarwar, W. Anwar, and A. Ashraf, "IoT healthcare: design of smart and cost-effective sleep quality monitoring system," *Journal of Sensors*, vol. 2020, Article ID 8882378, 17 pages, 2020.

[4] F. Al-Turjman, M. H. Nawas, and U. D. Ulusar, "Intelligence in the Internet of medical things era: a systematic review of current and future trends," *Computer Communications*, vol. 150, pp. 644–660, 2020.

[5] A. A. Thabit, F. A. Al-Mayali, and Q. H. Abbasi, "WBSN in IoT health-based application: toward delay and energy consumption minimization," *Journal of Sensors*, vol. 2019, Article ID 2508452, 14 pages, 2019.

[6] S. Zhu, Z. Fan, B. Feng et al., "Review on wearable thermoelectric generators: from devices to applications," *Energies*, vol. 15, no. 9, p. 3375, 2022.

[7] S. Niu, X. Wang, Y. Fang, S. Z. Yu, and L. W. Zhong, "A universal self-charging system driven by random biomechanical energy for sustainable operation of mobile electronics," *Nature Communications*, vol. 6, no. 1, p. 8975, 2015.

[8] Y. Zhang, S. Shen, C. Y. Chiu, and R. Murch, "Hybrid RF-solar energy harvesting systems utilizing transparent multiport micromeshed antennas," *IEEE Transactions on Microwave Theory and Techniques*, vol. 67, no. 11, pp. 4534–4546, 2019.

[9] Brandessense, "Energy harvesting system market by technology (light energy harvesting, vibration energy harvesting, electromagnetic/radio frequency (RF) energy harvesting, thermal energy harvesting), by component (transducers, PMIC, secondary batteries), by application (building & home automation, consumer electronics, industrial, transportation, security), industry analysis, trends, and forecast, 2022-2028," 2021, https://brandessenceresearch.com/energy-and-mining/energy-harvesting-market-size.

[10] Y.-W. Chong, W. Ismail, K. Ko, and C.-Y. Lee, "Energy harvesting for wearable devices: a review," *IEEE Sensors Journal*, vol. 19, no. 20, pp. 9047–9062, 2019.

[11] M. L. Ku, W. Li, Y. Chen, and K. J. Ray Liu, "Advances in energy harvesting communications: past, present and future challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1384–1412, 2016.

[12] O. A. Saraereh, A. Alsaraira, I. Khan, and B. J. Choi, "A hybrid energy harvesting design for on-body internet-of-things (IOT) networks," *Sensors*, vol. 20, no. 2, p. 407, 2020.

[13] J. Colomer, J. Brufau, P. Miribel, A. Saiz-Vela, M. Puig, and J. Samitier, "Novel autonomous low power VLSI system powered by ambient mechanical vibrations and solar cells for portable applications in a 0.13 μ technology," in *2007 IEEE Power Electronics Specialists Conference*, pp. 2786–2791, Orlando, FL, USA, 2007.

[14] S. Mohsen, Z. Abdelhalim, and Y. Khaled, "A self-powered wearable wireless sensor system powered by a hybrid energy harvester for healthcare applications," *Wireless Personal Communications*, vol. 116, no. 4, pp. 3143–3164, 2021.

[15] B.-Y. Yu, Z.-H. Wang, L. Ju et al., "Flexible and wearable hybrid RF and solar energy harvesting system," *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 3, pp. 2223–2233, 2021.

[16] I. Maxim, "Human body temperature sensor," 2016, https://datasheets.maximintegrated.com/en/ds/MAX30205.pdf.

[17] I. Maxim, "High-sensitivity pulse oximeter and heart-rate sensor for wearable health," 2015, https://datasheets.maximintegrated.com/en/ds/MAX30102.pdf.

[18] D. Analog, "Micropower, 3-axis, ±2 g/±4 g/±8 g digital output MEMS accelerometer," 2018, https://www.analog.com/media/en/technical-documentation/data-sheets/ADXL362.pdf.

[19] L. Silicon, "EM351/EM357 high-performance, integrated ZigBee/802.15.4 system-on-chip," 2013, https://www.silabs.com/documents/public/data-sheets/EM35x.pdf.

[20] S. Freescale, "MC9S08QE32 series covers: MC9S08QE32 and MC9S08QE16," 2011, https://www.nxp.com/docs/en/data-sheet/MC9S08QE32.pdf.

[21] D. Analog, "Ultralow power energy harvester PMUs with MPPT and charge management," 2017, https://www.analog.com/media/en/technical-documentation/data-sheets/ADP5091-5092.pdf.

[22] B. Chen, "Introduction to energy harvesting transducers and their power conditioning circuits," in *Low-Power Analog Techniques, Sensors for Mobile Devices, and Energy Efficient Amplifiers*, pp. 3–12, Springer, 2019.

[23] J. Shin and J. Cho, "Noise-robust heart rate estimation algorithm from photoplethysmography signal with low computational complexity," *Journal of Healthcare Engineering*, vol. 2019, Article ID 6283279, 7 pages, 2019.

Hindawi
Journal of Sensors
Volume 2023, Article ID 5997304, 9 pages
https://doi.org/10.1155/2023/5997304

Research Article

Identification of Network Traffic Intrusion Using Decision Tree

Dasheng Chen, Qi Song, Yinbin Zhang, Ling Li, and Zhiming Yang

Research Institute of China Telecom Corporation Limited, Guangzhou, 510660 Guangdong, China

Correspondence should be addressed to Dasheng Chen; chends4@foxmail.com

Received 8 September 2022; Revised 17 October 2022; Accepted 24 November 2022; Published 15 April 2023

Academic Editor: Dong Chen

Network technology plays an increasingly important role in all aspects of social life. The Internet has brought a new round of industrial revolution and industrial upgrading. The arrival of the "Internet" era is accompanied by a large-scale increase in network applications and the number of netizens. At the same time, the number and severity of cyberattacks continue to increase. Therefore, intrusion detection systems (IDSs) have become an important part of the current network security infrastructure in various industries. Anomaly detection of network traffic data is an effective method for network protection. In order to better realize the detection of network traffic anomalies, several algorithms have been successfully applied. Most of them come from artificial intelligence (AI), but there is a general problem of excessive model execution processing time and low detection rates. And through a lot of research, it is found that most models do not pay enough attention to the data processing in the early stage. Therefore, in this paper, we optimize the data normalization process through a series of experiments and combine the PCA feature selection method to propose an optimized MaxAbs-DT classifier model. To train and measure the performance of the model, we used the NSL-KDD dataset, which is the benchmark dataset for most network anomaly detection models. The experimental results show that MaxAbs-DT outperforms other existing models and validates the effectiveness of the method. In addition, its execution time is greatly reduced compared to many models.

1. Introduction

At the beginning of the development of the Internet and industrial networks, network security issues have attracted much attention. Although network administrators have deployed various security mechanisms over time, such as packet encryption technology and firewalls, most networks are still not immune to attacks because security breaches are proliferating [1]. Among the many defense methods, intrusion detection systems (IDSs) are one of the best solutions for protecting systems from malicious attacks [2]. Commonly used methods are based on anomaly detection and misuse detection [3]. Misuse detection performs comparison and detection based on predefined abnormal features, which is equivalent to a blacklist mechanism. Anomaly detection mainly detects abnormality based on normal behavior, which is equivalent to a whitelist mechanism, and everything outside the whitelist is abnormal. Now, intrusion detection plays a vital and increasingly important role in network defense, most notably to allow network security administrators to warn of intrusions, attacks, and malicious behaviors such as malware. Having an IDS system is a mandatory line of defense to protect critical internal networks from these ever-increasing malicious intrusion activities. Therefore, in order to propose better intrusion detection systems, the research in the field of intrusion detection has been vigorously developed in recent years. The network traffic detection part, which aims to maintain network security through the analysis of network traffic data, is the most important part of IDSs. In order to implement IDS, class methods are required. It is the process of classifying objects with knowledge extracted from a set of data during the learning step.

In the early days, based on the principle of network communication, we focused on some specific characteristics of network traffic data for abnormal detection and classification, and a relatively complete intrusion detection system appeared. Subsequently, network traffic data anomaly detection methods that combine network information such as network status, transmission packets, frequency of occurrence of specific data emerge in an endless stream, and various anomaly detection technologies based on supervised,

unsupervised, and clustering methods have also been successfully used in network traffic. Data anomaly detection. However, with the rapid development of network technology, most of the classical methods are no longer suitable for today's network traffic data with more complex structures and can no longer meet the huge demand for abnormal detection of network traffic data. At present, most of the research work aims to improve the accuracy and reduce the consumption of time resources. Because machine learning and deep learning can well meet the accuracy requirements, and the development of computer hardware has also greatly reduced the model calculation time, in recent years, most relevant scholars have focused on exploring network traffic data anomalies based on machine learning or deep learning. It is well known that data preprocessing and feature selection are an important step in the training of intelligent classifiers, but many research works have not done enough in-depth research here.

Since the NSL-KDD dataset [4] we used is labeled, we employ a supervised classification approach throughout the work. Each piece of data in this dataset contains 41 attributes, some of which are more informative and more important for classification prediction, while some unimportant features interfere with the detection process. Feature selection is a primitive preprocessing stage in which irrelevant features are ignored. To accomplish this task, we use the dimensionality reduction method by PCA (principal component analysis) to select suitable features.

This paper proposes a network intrusion detection (NIDS) method using decision tree classifier. In this study, we developed MaxAbs-DT, a computational predictor for predicting abnormal network traffic using machine learning. In this predictor, we choose the optimal data preprocessing method as well as the PCA feature selection method, tune the decision tree, and finally train the DT-based prediction model. Our main goal is to maximize detection rates and reduce false alarms while considering processing time, a very important factor in real-time applications. The rest of the paper is organized as follows: Section 2 presents related research work on different types of NIDs. Section 3 describes the datasets used in experiments, the feature processing method, and the proposed method. Section 4 summarizes the obtained results and comparisons. Section 5 gives conclusion and outlook that are drawn.

2. Related Works

In order to determine whether the traffic samples belong to the intrusion behavior, the researchers explored different binary classification algorithms to obtain better detection results.

The paper [5] introduced a number of methods for network anomaly detection using decision trees (DT), support vector machines (SVM), and Naive Bayesian networks (NB). Kevric et al. [6] pointed out that combining two tree algorithm models can achieve better performance than separate tree classification models, the best combination they reported was random tree and NB tree, the model was tested on the KDD dataset, and an accuracy of 89.24% was obtained. In the field of machine learning classifier research, feature selection is a key component that selects important features from a dataset. Alazzam et al. [7] proposed a feature selection method based on a wrapper method using a pigeon-inspired optimizer that selects informative attributes from a feature set. This method was tested on UNSW-NB15, KDDCUP 99, and NLS-KDD, respectively, and good experimental results were successfully obtained. Support vector machines have been widely used to detect intrusions in networks, Bachar et al. [8] stated that using support vector machines to detect network anomalies, the model was tested on the UNSW-NB15 dataset, which achieved an accuracy rate of 94%, and compared it with some machine learning classifiers such as MLP, REPTree, and RF.

Some IDS classifiers utilize ANN as a pattern recognition technique. ANNs are implemented using feedforward propagation, and during the learning step, their parameters are optimized so that the output matches the corresponding input model. The authors [9] proposed an artificial neural network (ANN) model, proposed a hybrid model that improves detection performance by combining different state-of-the-art algorithms, and achieved 81.2% accuracy for the NSL-KDD dataset. Gautam and Om [10] proposed two neural network classification models for host ID (HIDS), namely, generalized regression neural network (GRNN) and multilayer perceptron (MLP), and obtained relatively good results. Autoencoder (AE) methods commonly used for feature extraction are now widely used in the first stage of mixture models, existing as preprocessing work for downstream classifiers. It generates an efficient compressed representation of the original input by removing noisy features ([11, 12]. Al-Qatf et al. [13] successfully combined upstream AE and downstream SVM, and the model obtained 84.96% binary classification accuracy when tested on KDD dataset. Niyaz et al. [14] proposed a sparse autoencoder method for feature learning combined with a neural network classifier, which finally achieved 88.39% accuracy on KDD dataset.

3. Materials and Methods

In this section, we first describe the NSL-KDD dataset used to train the model and then introduce the classic decision tree classifier. Then, we introduce other variants of decision tree. In this paper, we only carry out the experimental process of binary classification and introduce several classical classifier performance indicators in the binary classification model.

3.1. Datasets. The NSL-KDD dataset [15, 16] generated in 2009 is widely used in intrusion detection experiments. It is an enhanced form of the KDD Cup 1999 dataset. The dataset covers the $KDDTrain^{+}$ dataset as the training set and $KDDTest^{+}$ dataset as the testing set, which has different normal records and four different types of attack records, as shown in Table 1.

In addition, in order to make intrusion detection more realistic, the test data set includes many attack categories that do not appear in the training set. In the several data sets

Table 1: Breakdown of traffic records in the NSL-KDD.

Dataset	Total	Normal	Abnormal
KDDTrain$^+$	125973	67343	58630
KDDTest$^+$	22544	9711	12833

used, in addition to the 22 attack types in the training set, there are 17 different attack types in the test set.

The NSL-KDD dataset contains 41 features, including 3 nonnumeric features and 38 numeric features (i.e., protocol_type, service, and flag), as shown in Table 2. It has a classification label, which is divided into two categories (abnormal and normal) for binary classification. For multiclass classification experiments, labels can be divided into five categories (i.e., normal, denial of service (DoS), user-to-root (U2R), remote-to-local (R2L), and probe), but in this paper, we only do binary classification experiments.

3.2. Evaluation Metrics. Before introducing the evaluation indicators, we first introduce several concepts: TP, FP, TN, and FN. TP represents the actual positive example, and the prediction is positive example; FP represents that the actual is a negative example, and the prediction is a positive example; TN represents the actual negative example, the prediction is the negative example, FN represents the actual positive example, and the prediction is the negative example. After understanding these concepts, we will then introduce the concepts of ACC, recall, precision, F1-score, MCC, kappa, and AUC. Their definitions are given below:

Accuracy (ACC): this is the ratio of the number of correctly detected intrusions to the total number of traffic records:

$$\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}}. \tag{1}$$

Recall: it refers to the ratio of the number of intrusion records correctly detected as an intrusion to the overall anomaly:

$$\mathrm{Recall} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \tag{2}$$

Precision: this is the ratio of true anomaly records to total traffic records identified as intrusions

$$\mathrm{Precision} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FP}}. \tag{3}$$

F1-score: it refers to the harmonic average of accuracy and true positive rate and is a relatively comprehensive evaluation mark

$$\mathrm{F}_1 - \mathrm{Score} = \frac{2 * \mathrm{recall} * \mathrm{precision}}{\mathrm{recall} + \mathrm{precision}}. \tag{4}$$

Matthews correlation coefficient (MCC): it will return a value between -1 and+1. Its meaning is to describe the correlation coefficient between the actual classification and the prediction classification, and the value range is -1 to 1. A value of 1 indicates the perfect prediction of the tested object, and a value of 0 indicates that the prediction result is not as good as the random prediction result; -1 means that the prediction classification is completely inconsistent with

Table 2: Features of NSL-KDD dataset.

No.	Feature	Type
1	duration	Numeric
2	protocol_type	Nonnumeric
3	service	Nonnumeric
4	flag	Nonnumeric
5	src_bytes	Numeric
6	dst_bytes	Numeric
7	land	Numeric
8	wrong_fragment	Numeric
9	urgent	Numeric
10	hot	Numeric
11	num_failed_logins	Numeric
12	logged_in	Numeric
13	num_compromised	Numeric
14	root_shell	Numeric
15	su_attempted	Numeric
16	num_root	Numeric
17	num_file_creations	Numeric
18	num_shells	Numeric
19	num_access_files	Numeric
20	num_outbound_cmds	Numeric
21	is_host_login	Numeric
22	is_guest_login	Numeric
23	count	Numeric
24	srv_count	Numeric
25	serror_rate	Numeric
26	srv_error_rate	Numeric
27	rerror_rate	Numeric
28	srv_rerror_rate	Numeric
29	same_srv_rate	Numeric
30	diff_srv_rate	Numeric
31	srv_diff_host_rate	Numeric
32	dst_host_count	Numeric
33	dst_host_srv_count	Numeric
34	dst_host_same_srv_rate	Numeric
35	dst_host diff_srv_rate	Numeric
36	dst_host_same_src_port_rate	Numeric
37	dst_host_srv_diff_host_rate	Numeric
38	dst_host_serror_rate	Numeric
39	dst_host_srv_serror_rate	Numeric
40	dst_host_rerror_rate	Numeric
41	dst_host_srv_rerror_rate	Numeric

the actual classification

$$\mathrm{MCC} = \frac{\mathrm{TP} * \mathrm{TN} - \mathrm{FN} * \mathrm{FP}}{\sqrt{(\mathrm{TP} + \mathrm{FN}) * (\mathrm{TP} + \mathrm{FP}) * (\mathrm{TN} + \mathrm{FP}) * (\mathrm{TN} + \mathrm{FN})}}. \tag{5}$$

Kappa: kappa coefficient is an index for consistency test, which can be used to measure the effect of classification. For classification problems, the so-called consistency is whether the predicted results of the model are consistent with the actual classification results. The calculation of kappa coefficient is based on the confusion matrix, and the value is between -1 and 1, usually greater than 0, where P_O is the overall accuracy and P_e is the accidental consistency error

$$\mathrm{kappa} = \frac{P_O - P_e}{1 - P_e}. \tag{6}$$

Area under the curve (AUC): ROC (receiver operating characteristic) is a curve connected by the points of FPR and TPR. The horizontal axis is fpr, which means the proportion of negative samples is wrongly classified as positive samples in all positive samples, and the vertical axis is TPR, that is, Sn mentioned above, which means the proportion of positive samples is correctly classified in all positive samples. The often mentioned AUC value is the area under the ROC curve. In other words, AUC is the area formed by the ROC "curve" and the straight line $x = 0$ and $y = 1$. AUC is mainly used to measure the performance or generalization ability of algorithms in binary classification problems. As a numerical value, we can intuitively evaluate the quality of the classifier. The larger the value, the better

$$\begin{aligned} \mathrm{FPR} &= \frac{\mathrm{FP}}{\mathrm{FP} + \mathrm{TN}}, \\ \mathrm{TPR} &= \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}}. \end{aligned} \tag{7}$$

TP (true positive), TN (true negative), FP (false positive), and FN (false negative) are the number of true positive, true negative, false positive, and false negative samples, respectively. TP means true positive, that is, the number of positive cases predicted; TN indicates true positive, that is, the number of positive cases predicted as negative cases; FP indicates false positive, that is, the number of negative cases predicted to be positive; and FN means false negative, that is, the number of positive cases predicted as negative cases.

3.3. Experimental Method

3.3.1. Crossvalidation. In this study, tenfold crossvalidation was selected to measure the performance of the model. The process of this verification method is described as follows: first, the entire data set will be randomly divided into ten copies, and the contents will not be repeated. Then, a single subset is randomly selected and retained as validation data to test the model, while the remaining 9 are used as training data to train the prediction model. This process goes through 10 times; that is, every piece of data will be used as a test data. Finally, the 10 results are averaged to obtain the final prediction results. Crossvalidation is often used in machine learning model evaluation, and it is rarely used in the field of deep learning. Because it is expensive to train and verify many times in the process of deep learning, it can be used when the amount of data is small.

3.3.2. Independent Testing. Compared with crossvalidation, independent testing is time-consuming and logically simple. First, the algorithm is trained on the training set. Secondly, the parameters of the model are adjusted by observing the performance of the model according to the evaluation indicators each time. At the same time, independent testing is also a method to test the effect of the model. Generally, independent test sets are used to verify the effect of the model in the end of experiments. The specific way is to use independent test sets as common data to compare with other methods. The above two experimental methods have been applied in this study. Generally, crossvalidation and independent testing experiments at the same time will make the experimental results more convincing.

3.4. The Proposed Predictive Framework. The prediction process can be concluded as two stages: (1) model training and (2) prediction. In the training phase, training samples are encoded and integrated by the feature representation algorithm. Then, the features are optimized to obtain the best feature subset and then fed into the decision tree algorithm to train the prediction model MaxAbs-DT. In the prediction stage, given the uncharacteristic traffic samples, we follow a similar process to encode the samples and use a trained model to predict whether the query sequence is an abnormal sample. The decision tree model gives a score for each traffic sample to measure the probability of its normal traffic. If the score is higher than 0.5, it is considered as a normal sample, otherwise, no.

3.5. Classification Algorithm. The decision tree [17] is converted into an if-then rule: a rule is constructed from each path from the root node of the decision tree to the leaf node; the characteristics of the internal nodes on the path correspond to the conditions of the rule, and the class label of the leaf node corresponds to the conclusion of the rule. It is also the final result of the decision.

The basic principle of the construction of the decision tree is the following strategy. A recursive process from root to leaf is looking for a "partition" attribute at each intermediate node.

(1) Start: build the root node, all training data are placed at the root node, select an optimal feature, divide the training data set into subsets according to this feature, and enter the child nodes

(2) All subsets are recursively divided according to the attributes of internal nodes

(3) If these subsets have been basically correctly classified, then construct leaf nodes and assign these subsets to the corresponding leaf nodes

(4) Each subset is assigned to leaf nodes; that is, it has a clear class, thus generating a decision tree

The depth of the tree structure is a very important parameter. Generally speaking, the deeper the tree structure, the better the fitting effect on the data, but it may also lead to overfitting. In this study, we implemented the DT algorithm (version 2.7.15) using the DT library in Python. In this study, we finally set the depth of the tree to 15 through a series of experiments, which can better capture the information of the data. The classification algorithm optimization results can be seen in "Classifier Optimization."

3.6. One-Hot Feature Representation. In many experimental studies, using one hot coding, the value of discrete features of data can be extended to European space, and a value of discrete features will correspond to a point in European space. In this way, if the discrete features are encoded by one-hot, the distance calculation between features will be more reasonable and easier to understand. The one hot feature extraction method [18] is often used in the field of sequence recognition and NLP and other related fields. In most cases, it can obtain excellent experimental results.

As mentioned above, the NSL-KDD dataset has 38 numeric and 3 nonnumeric features. Like many models, the proposed decision tree model only deals with numerical data input. Therefore, we need to convert all nondigital features into digital representation. Features (protocol_type, service, flag) are nondigital features that needs to be converted to digital form in NSL-KDD dataset. In the training set, features, protocol_type, service, and flag, have 3, 70, and 11 categories, respectively. In the test set, features, protocol_type, service, and flag, have 3, 64, and 11 categories, respectively. For the consistency of data, we uniformly set protocol_type, service, and flag as 3, 70, and 11 categories, respectively, and then perform the one-hot encoding process.

3.7. Feature Scaling. After successfully converting these features into digital form, the next appropriate thing is feature scaling. Feature scaling ensures that the dataset is in a normalized form. The values of some features (such as "src_bytes" and "dst_bytes") in NSL-KDD dataset are unevenly distributed; so, it is necessary for us to use feature scaling technology. In this way, we can ensure that our classifier will not produce biased results. There are several feature scaling methods as follows:

Z-score normalization: The Z-score method is standardized based on the average value (mean) and standard deviation of the original data. The average data after processing is 0, and the square difference is 1, which meets the standard normal distribution. The main purpose is to unify different dimensions of data into the same order of magnitude and measure the calculated Z-score value uniformly to ensure comparability between data. The formula is as follows

$$x_{\text{normalization}} = \frac{x - \mu}{\mathrm{O}}. \tag{8}$$

Among them, x represents the original data, μ represents the mean value of the original data, O represents the standard deviation of the original data, and $x_{\text{normalization}}$ represents the data after the normalization process.

min-max standardization: min-max standardization refers to the linear transformation of the original data, and the value is mapped between [0, 1]

$$x_{\text{normalization}} = \frac{X - X_{\min}}{X_{\max} - X_{\min}}. \tag{9}$$

MaxAbs normalization: it is usually used for sparse matrices. Using this method for standardization, the data can fall into the specified range of [-1, 1], and the original structure of the data will not be damaged. The formula is as follows

$$x_{\text{normalization}} = \frac{X}{|X_{\max}|}. \tag{10}$$

Robust scaler normalization: for data with outliers, if the Z-score method is used for standardization, the characteristics of outliers are often lost after standardization; so, the standardized data is not ideal. In this case, the robust scaler method can be considered. Robust scaler has a standardized processing method for outliers, which has stronger parameter control for data centralization and data scaling robustness

$$x_{\text{normalization}} = \frac{X - X_{\text{median}}}{\mathrm{IQR}}. \tag{11}$$

X_{median} is the median of the sample, and IQR is the interquartile distance of it.

3.8. Feature Selection. PCA (principal component analysis) is a commonly used data analysis method. PCA transforms the original data into a set of linearly independent representations of each dimension through linear transformation, which can be used to extract the main feature components of the data, and is often used for dimensionality reduction of high-dimensional data.

4. Results and Discussion

4.1. Classifier Optimization. To achieve the best performance, we conducted the following experiments to optimize the DT classifier. We performed parameter optimization on the depth of the decision tree in order to find the optimal max-depth value. Figure 1 shows the classifier evaluation index curves under different decision tree depths. Next, we need to determine which depth of the decision tree model is best for our dataset. Therefore, we compared the performance of the three cores. We can observe in Figure 1 that when the max depth of the tree goes from 15 onwards, all performance indicators basically do not change, and all the curves are in a parallel state. Consequently, the DT with max depth 15 was used to train this model in our predictor.

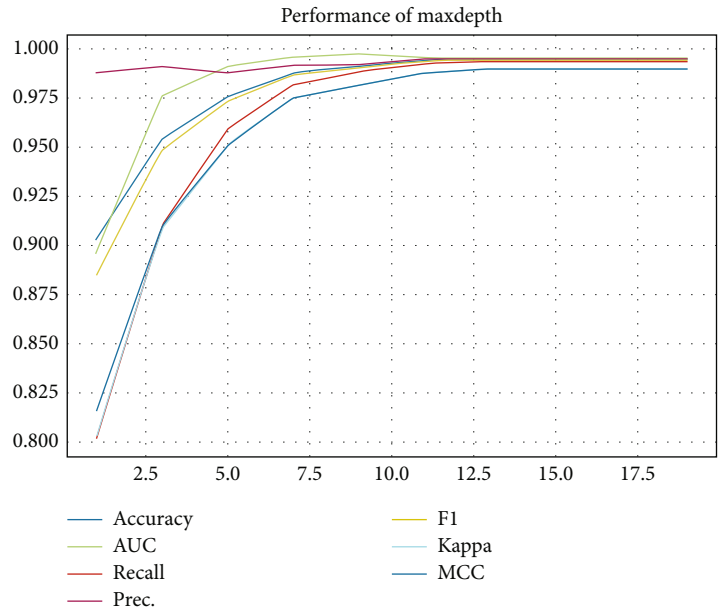


Figure 1: Evaluation indicator curves of different max depths in DT.

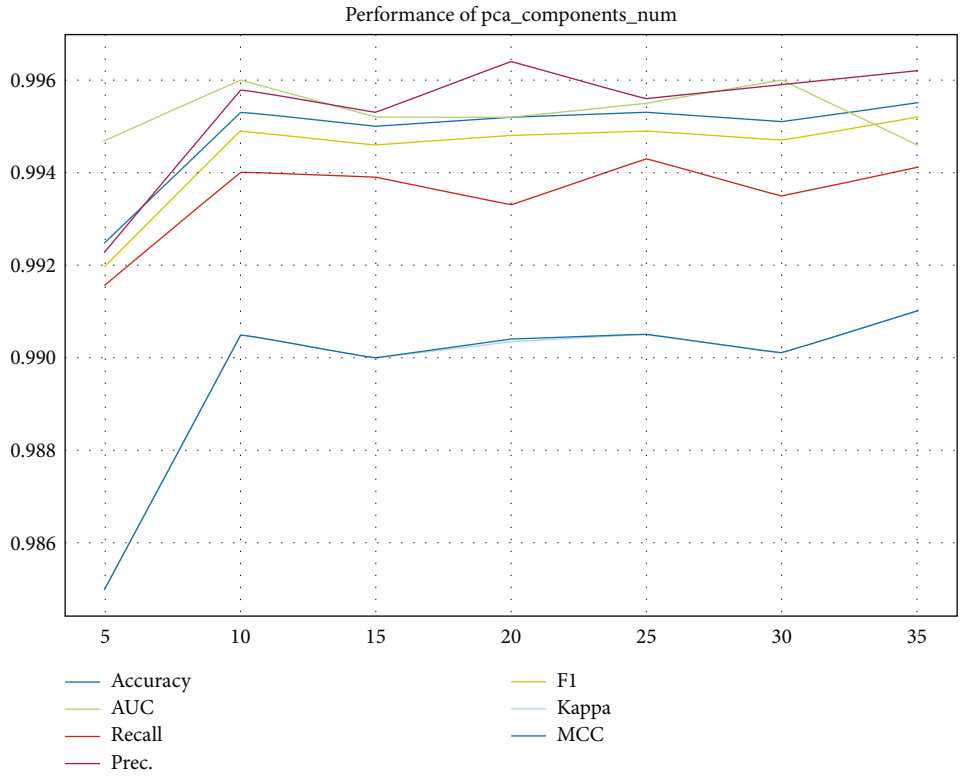


Figure 2: Evaluation indicator curves of different pca_components_nums in PCA.

In order to obtain the optimal feature subset, we conduct the following experiments to optimize the features. We perform parameter optimization on DT classifier using the PCA feature selection method to find the best number of features to keep. Figure 2 shows the classifier evaluation index curves under different numbers of retained features. We can observe in Figure 2 that when the number of features is at the position of 10, all the performance indicators basically reach the highest level. Consequently, the number of features to retain is set to 10 by the PCA method in our predictor.

4.2. Scale and Transform. There are generally four methods for data normalization. In order to get the optimal normalization method, we carried out the following experiments to

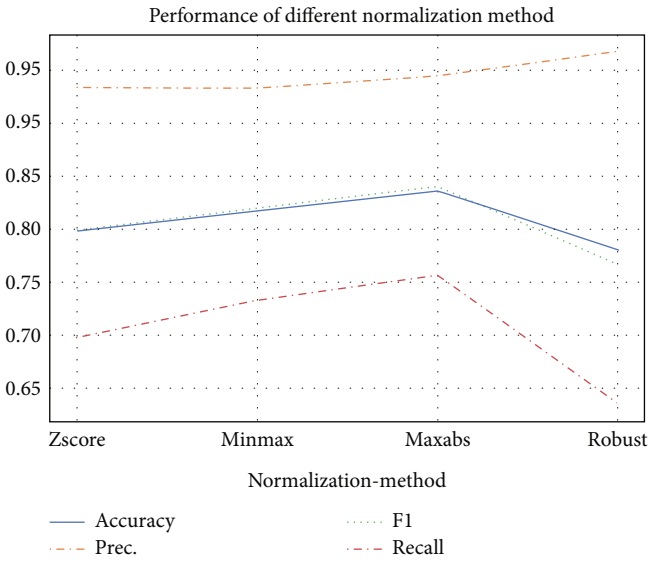


Figure 3: Evaluation indicator curves of different normalization methods in DT.

Table 3: Performance of different normalization methods.

Classifier	Accuracy	Prec.	F1	Recall
Z-score	0.7999	0.9344	0.7988	0.6975
min-max	0.8182	0.9332	0.8211	0.733
MaxAbs	0.8364	0.9452	0.8404	0.7565
Robust	0.7807	0.9682	0.7675	0.6357

Table 4: Performance of different classifiers.

Classifier	Accuracy	Prec.	F1	Recall
et	0.7919	0.9675	0.7822	0.6565
rf	0.7616	0.9681	0.7416	0.601
rbfsvm	0.7847	0.9233	0.7819	0.6781
knn	0.772	0.9231	0.7656	0.654
dt	0.8364	0.9452	0.8404	0.7565
gbc	0.7796	0.9684	0.766	0.6335
ada	0.7819	0.9225	0.7786	0.6735
svm	0.7569	0.9178	0.7466	0.6292

determine. We use 4 data normalization methods, namely, *Z*-score, min-max, MaxAbs, and robust scaler on the test dataset to compare the metrics of the models to determine the best data normalization method. Figure 3 shows the classifier evaluation index curves under different normalization methods. We observed in Figure 3 that when the normalization method is MaxAbs, all performance indicators basically reached the highest value, specifically ACC, F1, and recall obtained the highest value and obtained the second highest value on Prec. The results are listed in Table 3. As can be seen, amongst the four individual normalization methods, the method called MaxAbs performs the best than the other three. This indicates that the MaxAbs method is more useful for NSL-KDD abnormal prediction.

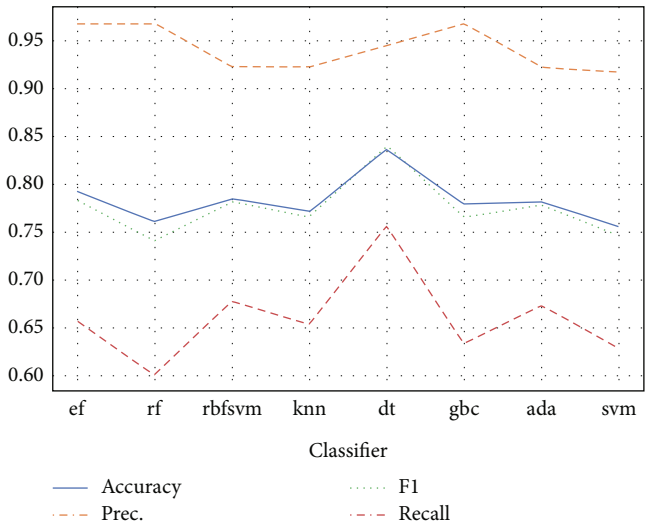


Figure 4: Evaluation indicator curves of different classifiers on KDDTest$^+$.

4.3. Comparison with Different Classification Algorithms. To measure the effectiveness of our proposed work, we compared its performance with several well-known classifiers, such as the extra trees classifier [19], random forest (rf) [20], SVM with RBF core (rbfsvm) [21], SVM with linear core (svm) [22], *K*-nearest Neighbor (knn) [23], gradient boosting classifier (gbc) [24] and AdaBoost classifier (ada) [25]. For a fair comparison, we trained classifiers on the same train dataset KDDTrain$^+$ with our feature set and then fine-tuned the classifiers one by one to achieve the optimal performance. The classifiers are also evaluated by tenfold crossvalidation, and the evaluation results on the same test dataset KDDTest$^+$ are presented in Table 4. We can see that DT achieved the highest value in three indicators, namely, ACC reached 81.86%, F1 reached 81.68%, and recall reached 71.04% outperforming other classifiers in three of the four metrics. Specifically, the performance of DT classifier on ACC, F1, and recall was higher than the second-place classifier ET by 2.67%, 3.46%, and 5.39%, respectively. The results presented in Figure 4 show that compared with the other seven classifiers in this study, the DT classifier has better discriminative ability to distinguish abnormal traffic from normal traffic.

As shown in Figure 5, in terms of the consumption time of the classification model, which is an important consideration in the actual industrial application, our proposed work takes the least time compared to other classifiers and has high practical significance. The specific values are shown in Table 5.

4.4. Comparison with Existing Classifiers. To better demonstrate the performance of our proposed MaxAbs-DT model, we compared its performance with seven existing intrusion detection techniques. As shown in Table 6, we compared the performance with other techniques mentioned in Tavallaee et al. [16] and Yin et al. [26]. The results prove that the MaxAbs-DT classifier is superior in detecting network anomalies. From Table 6, the MaxAbs-DT model improves

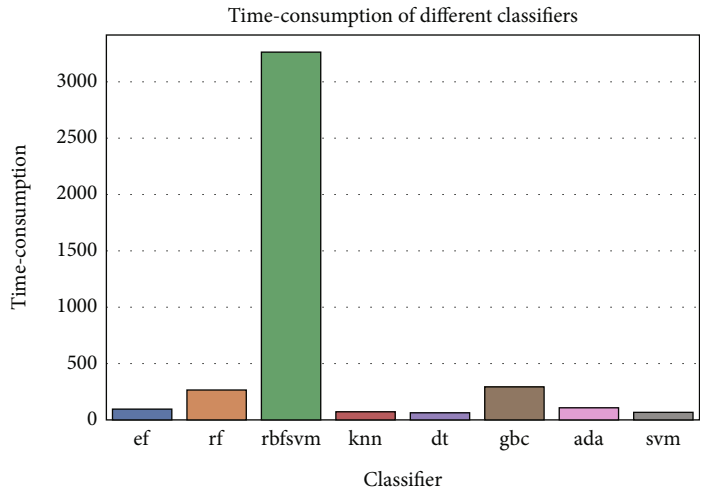


Figure 5: Time consumption histogram of different classifiers.

Table 5: Time consumption of between different classifiers.

Classifier	Time consumption (sec)
et	99.1
rf	270.448
rbfsvm	3265.208
knn	75.077
dt	67.582
gbc	294.056
ada	111.741
svm	69.829

Table 6: Comparison between existing methods using KDDTest$^+$ for binary classification.

Classifier	Accuracy
SVM [16]	69.52%
Naïve Bayes [16]	76.56%
Multilayer perceptron [16]	77.41%
Random tree [16]	80.67%
J48 [16]	81.05%
NB tree [16]	82.02%
RNN [26]	83.28%
DT (this study)	83.64%

the detection accuracy of the RNN model by 0.44% on the KDDTest$^+$ datasets.

The results clearly demonstrate that the detection performances of the proposed model are more effective. Although the processing time of decision tree classifier is short, the overall classification accuracy is not significantly increased. On the contrary, deep learning may take a long time, but it may have a better ability to capture other potential features of data. For this, it is worth further exploring in the future research work. For scenarios where the response time is required to be as short as possible, the classifier proposed in this paper may be more useful.

5. Conclusions

This work proposed the application of an improved decision tree classifier to detect network intrusions. The proposed work showed good performance and achieved nice results. To validate the performance of our model, we train and validate the model using the NSL-KDD dataset, which is currently widely used as a benchmark dataset for intrusion detection by most researchers. After the experiment, the MaxAbs-DT model obtained a higher accuracy, recall, and F-score and less execution time than other models, and it also outperforms other existing models on the accuracy metric in literature. However, there are many deep learning technologies that may have potential optimization capabilities for this research. In the future, we plan to integrate some optimal feature selection methods with novel deep learning models, such as graph convolutional neural networks, to explore better network anomaly classifier performance.

Data Availability

The datasets generated for this study can be found in https://github.com/Kevin-chen-sheng/MaxAbs-DT.

Conflicts of Interest

The authors declare that there are no conflict of interest.

Acknowledgments

The authors would like to give thanks for the financial support provided by the Research Institute of China Telecom Corporation Limited and encouragement of colleagues in the Research Institute.

References

[1] F. Masoodi, S. Alam, and S. T. Siddiqui, "Security & privacy threats, attacks and countermeasures in Internet of Things," *International Journal of Network Security & Its Applications*, vol. 11, no. 2, pp. 67–77, 2019.

[2] S. Alam, M. Shuaib, and A. Samad, "A collaborative study of intrusion detection and prevention techniques in cloud computing," in *International Conference on Innovative Computing and Communications*, Singapore, 2019.

[3] L. Lv, W. Wang, Z. Zhang, and X. Liu, "A novel intrusion detection system based on an optimal hybrid kernel extreme learning machine," *Knowledge-Based Systems*, vol. 195, article 105648, 2020.

[4] F. Z. Belgrana, N. Benamrane, M. A. Hamaida, A. M. Chaabani, and A. Taleb-Ahmed, "Network intrusion detection system using neural network and condensed nearest neighbors with selection of NSL-KDD influencing features," in *2020 IEEE International Conference on Internet of Things and Intelligence System (IoTaIS),.*, Bali, Indonesia, 2021.

[5] S. Agrawal and J. Agrawal, "Survey on anomaly detection using data mining techniques," *Procedia Computer Science*, vol. 60, pp. 708–713, 2015.

[6] J. Kevric, S. Jukic, and A. Subasi, "An effective combining classifier approach using tree algorithms for network intrusion detection," *Neural Computing and Applications*, vol. 28, Supplement1, pp. 1051–1058, 2017.

[7] H. Alazzam, A. Sharieh, and K. E. Sabri, "A feature selection algorithm for intrusion detection system based on pigeon inspired optimizer," *Expert Systems with Applications*, vol. 148, article 113249, 2020.

[8] A. Bachar, N. El Makhfi, and O. E. L. Bannay, "Machine learning for network intrusion detection based on SVM binary classification model," *Advances in Science, Technology and Engineering Systems Journal*, vol. 5, no. 4, pp. 638–644, 2020.

[9] B. Ingre and A. Yadav, "Performance analysis of NSL-KDD dataset using ANN," in *2015 international conference on signal processing and communication engineering systems*, Guntur, India, 2015.

[10] S. K. Gautam and H. Om, "Computational neural network regression model for host based intrusion detection system," *Perspectives in Science*, vol. 8, pp. 93–95, 2016.

[11] H. Liu and B. Lang, "Machine learning and deep learning methods for intrusion detection systems: a survey," *Applied Sciences*, vol. 9, no. 20, p. 4396, 2019.

[12] T. N. Sainath, B. Kingsbury, and B. Ramabhadran, "Auto-encoder bottleneck features using deep belief networks," in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, Kyoto, Japan, 2012.

[13] M. Al-Qatf, Y. Lasheng, M. Al-Habib, and K. Al-Sabahi, "Deep learning approach combining sparse autoencoder with SVM for network intrusion detection," *IEEE Access*, vol. 6, pp. 52843–52856, 2018.

[14] Q. Niyaz, W. Sun, and A. Y. Javaid, "A deep learning based DDoS detection system in software-defined networking (SDN)," *EAI Endorsed Transactions on Security and Safety*, vol. 4, no. 12, 2017.

[15] S. Revathi and A. Malathi, "A detailed analysis on NSL-KDD dataset using various machine learning techniques for intrusion detection," *International Journal of Engineering Research & Technology (IJERT)*, vol. 2, no. 12, pp. 1848–1853, 2013.

[16] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE symposium on computational intelligence for security and defense applications*, Ottawa, ON, Canada, 2009.

[17] A. J. Myles, R. N. Feudale, Y. Liu, N. A. Woody, and S. D. Brown, "An introduction to decision tree modeling," *Journal of Chemometrics: A Journal of the Chemometrics Society*, vol. 18, no. 6, pp. 275–285, 2004.

[18] P. Rodríguez, M. A. Bautista, J. Gonzalez, and S. Escalera, "Beyond one-hot encoding: lower dimensional target embedding," *Image and Vision Computing*, vol. 75, pp. 21–31, 2018.

[19] A. Sharaff and H. Gupta, "Extra-tree classifier with metaheuristics approach for email classification," in *Advances in Computer Communication and Computational Sciences*, pp. 189–197, Springer, 2019.

[20] Z. Lv, S. Jin, H. Ding, and Q. Zou, "A random forest sub-Golgi protein classifier optimized via dipeptide and amino acid composition features," *Frontiers in Bioengineering and Biotechnology*, vol. 7, p. 215, 2019.

[21] Y. Liu, D. Chen, R. Su, W. Chen, and L. Wei, "iRNA5hmC: the first predictor to identify RNA 5-hydroxymethylcytosine modifications using machine learning," *Frontiers in Bioengineering and Biotechnology*, vol. 8, p. 227, 2020.

[22] B. Liu, H. Wu, and K. C. Chou, "Pse-in-one 2.0: an improved package of web servers for generating various modes of pseudo components of DNA, RNA, and protein sequences," *Natural Science*, vol. 9, no. 4, p. 67, 2017.

[23] S. Zhang, X. Li, M. Zong, X. Zhu, and R. Wang, "Efficient kNN classification with different numbers of nearest neighbors," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 5, pp. 1774–1785, 2018.

[24] Z. Liao, S. Wan, Y. He, and Q. Zou, "Classification of small GTPases with hybrid protein features and advanced machine learning techniques," *Current Bioinformatics*, vol. 13, no. 5, pp. 492–500, 2018.

[25] T. K. An and M. H. Kim, "A new diverse AdaBoost classifier," in *2010 International conference on artificial intelligence and computational intelligence*, Sanya, China, 2010.

[26] C. Yin, Y. Zhu, J. Fei, and X. He, "A deep learning approach for intrusion detection using recurrent neural networks," *IEEE Access*, vol. 5, pp. 21954–21961, 2017.

Hindawi
Journal of Sensors
Volume 2022, Article ID 5227955, 11 pages
https://doi.org/10.1155/2022/5227955

Research Article

Analyzing Human Muscle State with Flexible Sensors

Zhiyong Chen,[1] Qingsuo Wang,[2] Yunce Bi,[1] Juncong Lin,[1] Wei Yang,[3] Chaoyang Deng,[1] Shihui Guo,[1] and Minghong Liao[1]

[1]*School of Informatics, Xiamen University, Xiamen 361005, China*
[2]*Beidahuang Industry Group General Hospital, Harbin, China*
[3]*School of Life Sciences, Xiamen University, Xiamen 361102, China*

Correspondence should be addressed to Shihui Guo; guoshihui@xmu.edu.cn

Received 12 May 2022; Revised 6 June 2022; Accepted 7 June 2022; Published 11 July 2022

Academic Editor: Dong Chen

Analyzing human muscle states has attracted extensive attention. EMG (electromyography) pattern recognition methods based on these works have been proposed for many years. However, uncomfortable wearing and high prices make it inconvenient for motion tracking and muscle analysis by using robotic arms and inertial sensors in daily life. In this study, we propose to use smart clothes integrated with flexible sensors to collect arm motion data, estimate the kinematic information of continuous arm motion, and predict the EMG signal of each arm muscle. Firstly, the neural network regression model integrated with the LSTM (long short-term memory) module is used to continuously estimate the sensor resistances collected by the smart clothes and the angles collected by Kinect. Then, six types of shoulder and elbow movements' angles and the corresponding EMG signals of 5 subjects are preprocessed and aligned. The stacked regression model based on extremely randomized trees (extra trees) is used for regression. Our experimental results show that the average estimation absolute error from the sensor resistances to the joint angle is 3.45 degrees, and the absolute percentage error from the joint angle to the EMG signal is only 1.82%.

1. Introduction

Human motion posture tracking and muscle analysis capture the continuous motion of bone joints by various types of sensors or multiple types of cameras to continuously estimate the value of muscle activity at the moving parts [1]. It is of great significance in medical rehabilitation, military and national defense, animated cartoon, and other fields. Take the postoperative rehabilitation of patients with cerebral stroke as an example, there are about 2 million new stroke patients in China every year, and up to 80 percent of stroke patients have the sequelae of dyskinesia [2]. The existing rehabilitation treatment is mainly concentrated in the hospital environment [3], under the guidance of professional rehabilitation therapists. In contrast, the rehabilitation training of patients in the home environment lacks effective and accurate monitoring methods. Unscientific rehabilitation training may lead to an abnormal movement, leading to poor recovery and even secondary physical impairment. So, it is of great value to accurately track the movement posture of these people and analyze the muscle exercise situation for the development of personalized rehabilitation programs.

A lot of early work is based on mechanical arm [4–6] mechanical inertial sensor (MEMS) [7–9] to capture human motion posture. The human motion posture tracking based on an inertial sensor unit (IMU) is the current mainstream, wearable method [10, 11]. IMU has the advantages of low cost, small size, no interference, etc. It is suitable for human motion posture tracking outdoors. Nevertheless, there are three main problems: foreign-body sensation when wearing, requiring calibration, and data drift. One reason for the widespread use of inertial sensor units (IMUs) is that wearing foreign body sensations is severely constrained, and researchers are looking for ways to collect physiological data in ways that make subjects more comfortable [12, 13]. In recent years, the use of intelligent wearable sensors with integrated flexible sensors to monitor human health state

has aroused widespread interest [14, 15]. They provide good user experience and comfortable wearing [16] with cheap price [17].

Our work proposes a method of human posture tracking based on a flexible sensor. The captured posture is calculated to obtain the joint angle of the upper limb, which is used to estimate the electromyographic signals of the upper limb movement continuously.

Our work is aimed at solving the problem of muscle analysis on a smart clothing platform. Muscle analysis using a smart jacket is defined as two key subproblems:

(1) Using flexible sensors to track human body posture so as to calculate upper limb joint angles

(2) Using the joint angle of the upper limb movement to estimate the myoelectric signal continuously

There are two main contributions to this study:

(1) We provide a complete human posture tracking and joint angle calculation solution. We use flexible sensors to minimize interference with the users' daily activities and ensure the best comfort possible in their experience. And the novelty of the method is that it is not affected by the light, and the user does not need to consider the illumination effect of the environment

(2) We introduce a stacked regression model based on the extreme random tree for continuous estimating joint angles and EMG signals. Compared with other traditional machine learning methods and regression networks with short and long memory modules, this regression model has a better regression effect

2. Related Work

2.1. Motion Tracking Based on Flexible Sensor. The main scope of monitoring human movement is divided into two kinds. One is a large range of individual position movement; the other is the individual joint movement, temperature, and other physiological indicators. Because of its physical characteristics, flexible sensors are mainly used to detect human body states. And flexible sensors used for human body state monitoring mainly convert the signals to be monitored into the stretching and bending of the sensor through mechanical deformation [18] or use the physicochemical mechanism driven by temperature, humidity, and chemical reaction to measure the changes in resistance or capacitance to achieve monitoring purposes [19, 20]. Its functions mainly include motion posture tracking and physical skin deformation. Representative works are listed as follows. The Massachusetts Institute of Technology (MIT) group uses high-density array-type pressure sensor gloves for object recognition [21]; Northwestern University research uses a wireless passive flexible vibrator to realize tactile feedback in virtual reality scene [22]; the Swiss Federal Institute of Technology in Zurich uses the flexible gloves on finger movement tracking [23]; the United States Dartmouth College team uses the flexible sheath on elbow motion tracking [24]; the National University of Singapore research uses the sensor fusion of leap motion controller and flexible sensor to track human finger using Kalman filter [25]; Tsinghua University researchers use the array-type pressure membrane to identify the interaction between the human body and the object, etc. [26]; Northwestern Polytechnical University research uses a small set of wearable sensors to estimate whole-body pose in human bicycle riding [27]; Chinese Academy of Sciences research uses micro flow on the skin surface tactile sensing [28]; Institute of electronics, Chinese Academy of Sciences research uses flexible pressure sensor to 3-dimension force recognition [29]; Shenzhen University research group realized noncontact human-computer interaction by using short-range capacitance sensor [30]; etc.

The human body motion posture tracking method based on flexible (nonfabric) sensors achieves the target of finger motion posture tracking [31] and arm deformation reconstruction [32]. In the above work, the researchers collected training data and established a deep neural network and then realized the prediction of joint posture and skin deformation. The human motion posture tracking method based on the fabric sensor realizes the tracking of the whole body [33], elbow [34], and back and shoulder [33]. The researchers used a neural network of short- and long-duration memory to interpret stretch sensor signals as body posture but only did error analysis for specific movements such as squatting and bending. Another work used stretch sensors to track the posture of upper body movements (back and shoulders). The researchers also used stretch sensors to track elbow motion and analyzed motion tracking errors for different arm circumferences and sensor position offset (up to 1 cm). The pliable and flexible characteristics of the flexible sensor provide a convenient way of wearing and a comfortable user experience and provide more possibilities for human motion posture tracking [17].

2.2. Muscle Analysis. According to the International Classification of Functioning, Disability, and Health [35], muscle power is the maximum power that can be released by a test muscle under certain limits [36, 37]. And muscle power controls the movement of our limbs. These forces must be estimated by indirect means since the direct measurement is usually neither possible nor practical. Therefore, many studies measure muscle strength through inverse kinematics [38, 39], including the fingers [40], upper limbs [41], and lower limbs [42–44]. In experimental studies of human movement, muscle strength tests are useful in assessing the recovery of stroke patients [45]. And EMG signals are measured to determine the electrical current generated by muscle contractions in neuromuscular activity [46]. Therefore, electromyography (EMG) is often used as a tool to determine muscle activity [47–50]. Bogey et al. [44, 51] recently developed a method to estimate force from EMG signals and based on normalization of activation during maximum voluntary contraction to record maximum muscle force. Heintz and Gutierrez-Farewik [52] adopted the numerical algorithm established based on the traditional optimization technology, that is, the constraint minimization technology

using the Lagrange multiplier method to solve constraints. Lloyd and Besier [53] used EMG to predict knee torques through inverse dynamic calculations under different dynamic contraction conditions. They used a modified Hill-type muscle model to better estimate living muscle strength during exercise tasks. Amarantini et al. [54] used two-step EMG and optimization methods to estimate muscle forces under dynamic conditions. This method has the ability to propose a method to account for agonist/antagonist cocontraction properly. In addition, the method can improve the confidence of muscle force estimation. Hashemi et al. [55] combined angle-based EMG amplitude calibration and parallel cascade identification (PCI) modeling for EMG-based force estimation in dynamic contractions, including concentric and eccentric contractions of the biceps and triceps, in order to enhance dynamic EMG-force estimation. Hsu et al. [47] used EMG sensors to study the sequence of muscle contractions in patients from sitting to standing (STS) after stroke. Kim et al. [56] estimated the muscle strength of nine muscle groups of the lower limbs using a static optimization method with inverse dynamics based on motion data and compared it with EMG signals. It is proved that establishing the relationship between EMG signal and muscle force calculated by inverse kinematics is a practical method to measure muscle strength in vivo.

3. Method

We built a sparse sensor network on the smart jacket to capture the sensor resistance value of human right upper limb movement to estimate the EMG signal continuously. The workflow of our algorithm is shown in Figure 1.

We split the task into two subtasks:

(1) Prediction from sensor signal to joint angle

(2) Continuous estimation from joint angle to EMG signal. We use a neural network with long short-term memory (LSTM) [57–59] module to regress the sensor data and joint angle data to solve the nonlinear and hysteretic problems of the sensor itself. Joint angle data and EMG data were fitted by stacking a regression model based on extreme random tree. Thus, a new method of continuous estimation from sensor signal to EMG signal is provided

3.1. Hardware Preparation. We prepared a smart jacket prototype (Figure 2(a)) with integrated, flexible sensors to complete our study. The smart jacket cloth has five flexible, stretchable sensors, four around the shoulder and one under the elbow, as shown in Figure 2(b). And the above parts contain 20 cm sensors, respectively, as the total cost is about 15.1488 dollars. Sensors are fixed to the garment by hand sewing. The clothes are tight tracksuits, ensuring the sensors fit snuggly and better capture shoulder and elbow movement. The fabric is made of 80% polyester fiber and 20% polyurethane fiber. The sensor we use is a conductive rubber wire stretch sensor manufactured by Adafruit [60, 61]. The sensor is 2 mm in diameter and made of carbon-black impregnated rubber. As for the traditional sensor, multi-point instruments need to be used; the production and use are complex; the experimental conditions are not convenient; also users cannot use them directly at home. In addition, traditional sensors are more expensive compared to flexible, stretchable sensors. As a result of fact, the conductive rubber wire stretch sensor is more suitable and practical, and for the sensor, in the relaxed state, the resistance is about 350 ohms per inch. The human body posture tracking is realized by monitoring the resistance changes of the five flexible sensors. At the elbow, for example, when the user bends the elbow, the sensor is stretched and its resistance increases. The resistance sampling frequency of the sensor is 32 Hz. The server receives sensor data through the Arduino UNO3 development board and is responsible for all subsequent calculations. The server comes with Intel Core I7 (6 cores), 16 GB of RAM, and NVIDIA GTX 1080Ti.

3.2. Sensor Resistance and Joint Angle Data Collection. For the participant, the experiment involved a 30-year-old doctoral student in the lab. The participant is familiar with the whole experimental design and the experimental process before the investigation. For the collection process, before the experiment, the participant got to know the purpose of the experiment, put on the intelligent jacket with integrated, flexible sensors; and took part in some guiding activities to get familiar with the collection system and experimental process. The participant started and stopped the data collection experiment by listening to voice instructions. He adjusted his position towards a Kinect camera that could collect depth information, allowing the camera to capture the full movements of his right upper limbs. If the participant gets tired during the collection process, he can terminate the collection process at any time.

The data collected each time is saved to the server. The sensor resistance data includes the collection time and resistance values of five sensors. The data collected by Kinect includes the collection time and three-dimensional position of each node of the right upper limb. The joint angles of each part were calculated according to ISB (International Society of Biomechanics) convention [62]. These angles are calculated by the following methods. First, create local coordinate systems based on marker points and then, decompose the rotation matrix into Euler sequences proposed by ISB for a particular skeleton. The data are in the y-axis upward direction, which is consistent with the joint angle data introduced in the next section.

3.3. Joint Angle and EMG Data Alignment Preprocessing. We used the data set published by Bolsterlee et al. [63], Shoulder Database, for continuous estimation from shoulder elbow angle to EMG signal. The data set contained data on six movements of five healthy young adults. No one has uncomfortable shoulder joints or has been specially trained. Kinematic data were collected using a four-device Optotrak system (Northern Digital, Inc., Waterloo, Ontario, Canada) to collect marker location information for six groups of markers in the subjects' chest, scapula, humerus, forearm, and hands. The acquisition frequency is 100 Hz. EMG data were collected using surface electrodes (Ambu N-00-SECG)

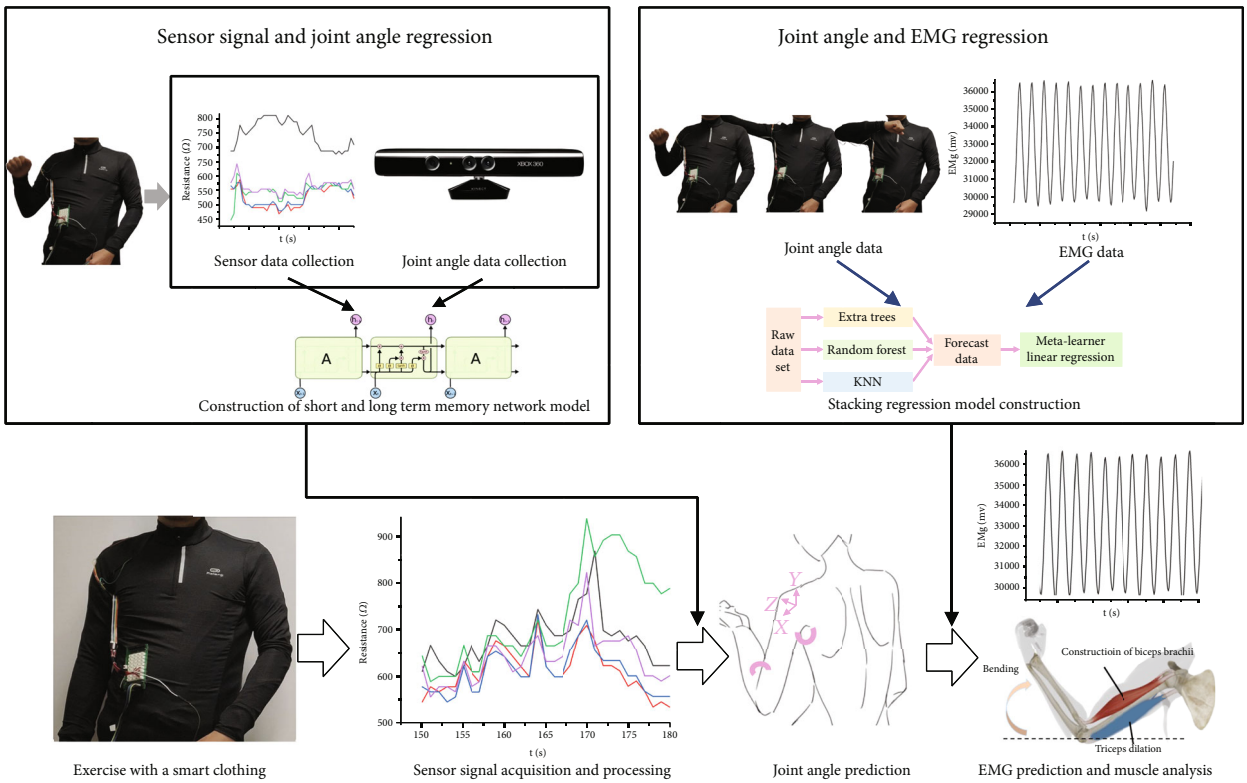


Figure 1: The flow chart.

Figure 2: Smart jacket with five flexible sensors: (a) smart jacket prototype; (b) sensor layout.

and a 16-channel Porti system (TMS International, Enschede, the Netherlands, Sampling frequency: 1000 Hz). EMG data were collected from 14 muscles, including the triceps and biceps (Figure 3). The electrodes are placed as recommended by SENIAM [64]. Since the frequency of kinematic data and EMG data are not the same, we need to preprocess the data for alignment. Due to the fast-sampling frequency of EMG data, we first performed linear interpolation on the joint angle data to obtain 1000 pieces of data per second and then aligned the joint angle data with EMG data through the acquisition time.

3.4. Regress from Sensor Signal to Joint Angle. From the description of the sensor manufacturer and our experiments, it is found that the relationship between the resistances of the sensors is nonlinear when subjected to tension. Moreover, the sensor has a lag problem, and it needs to be stationary for a period of time to return to the initial state. Therefore, we propose to use the LSTM model to regress sensor signals and joint angles in order to obtain more accurate angle prediction values.

LSTM is an artificial recursive neural network (RNN), which can effectively process temporal data. The network model we designed has an LSTM layer with five hidden cells, followed by a full connection layer with ten cells, and finally, a full connection layer with only one cell as output. The input to the network is a vector:

$$\vec{S} = \left\{ S_{t-(N_p-1)\delta t}, \cdots, S_{t-\delta t}, S_t \right\}, \quad (1)$$

where S_t is the sensor resistance at a specific time t, δt is the time step read by the sensor signal, and N_p is the number of sampling points. In our experiment, δt and N_p were set to

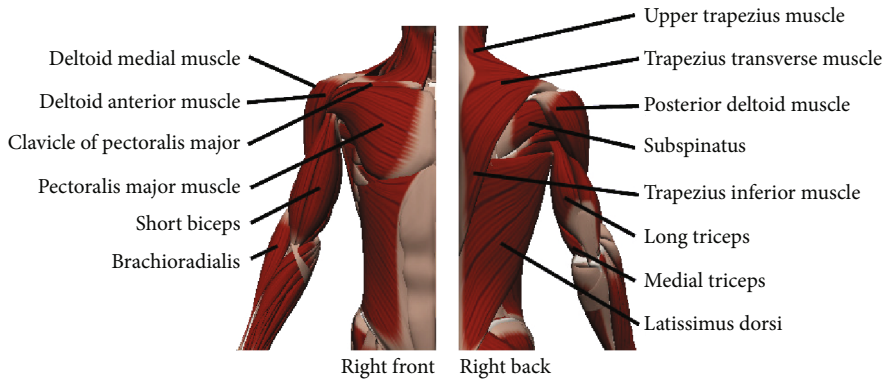


Figure 3: EMG signals were collected from 14 muscle blocks.

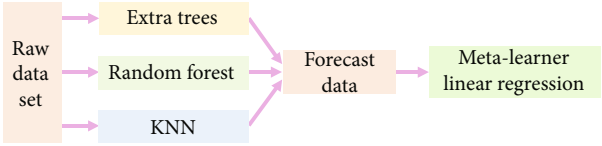


Figure 4: The stack model which includes extra trees, random forest, and KNN basic model.

0.286 and 50, respectively. The output of the LSTM network is the estimated joint angle. LSTM regression model is used to solve the nonlinear and hysteresis characteristics of the sensor, which realizes the precise prediction of the joint angle.

3.5. Regress from Joint Angle to EMG Signal. We chose to use a stacked regression model based on extra trees [65] to process from joint angle to EMG signal. The extreme random tree algorithm builds a set of nonrunning decision trees or regression trees based on the classical top-down process. The two main differences between it and other tree-based integration approaches are that it splits nodes by randomly selecting complete pointcuts and uses the entire learning sample to grow the tree. From the perspective of the bias difference, the basic principle behind the extreme random tree approach is that explicit randomization of pointcuts and attributes combined with integrated average should be able to reduce variance more effectively than the weaker randomization schemes used by other methods. Using the complete original learning sample rather than the boot stringing copy is aimed at minimizing bias. From a computational point of view, suppose there is a balanced tree, the complexity of the tree growth is order $N \log N$ with respect to the learning sample size, just like most other tree growth processes. Our stacked model (Figure 4) integrates extreme random tree, random forest [66–68], and K-nearest neighbor [69] algorithm as the basic model and uses a linear regression model in the metalearner [70]. The metamodel is trained based on the prediction results of the training samples output by the basic model. The stacked model can solve the problem of the prediction errors made by different models which are not correlated or have low correlations.

4. Result Analysis

4.1. Evaluation Criteria. The following six evaluation criteria were used to evaluate our experimental results: mean absolute error (MAE), mean square error (MSE), root mean square error (RMSE), R square (R^2), root mean square logarithmic error (RMSLE), and mean absolute percentage error (MAPE). The values of MAE, MSE, RMSE, and MAPE are in the range of $[0, \infty)$; when the predicted value is exactly the same as the true value, the value is 0; the larger the error, the larger the value. MAE and RMSE can roughly estimate the difference between the predicted value and the true value; when the predicted value is entirely consistent with the true value, the value is 1; when each predicted value of the sample is equal to the mean value, the value is 0; it may also be negative, and the regression effect is poor.

4.2. Regression Results of Short- and Long-Term Memory Network Regression Model. Sensor data is composed of resistance signals from five sensors numbered 1-5 (Figure 2(b)). Joint angle data is calculated by the spine points, shoulder points, elbow points, and wrist points collected by Kinect, including shoulder angle and elbow angle. We defined that the elbow angle has only one degree of freedom, and the shoulder angle has three degrees of freedom. Then, the LSTM regression model was used to perform training regression on sensor data and joint angle data. 70% of the collected data were trained in the front segment, and the rest was used for predictive testing.

We compare the LSTM regression model with the traditional machine learning regression model and polynomial regression model, and the true value is also involved. In this paper, the gradient lifting decision tree (GBR) with the best regression effect in the traditional machine learning regression model and the regression results of the fourth-order polynomial regression are selected for comparative display, as shown in Figure 5.

It can be seen from the experimental results that the LSTM regression model (red curve) fits the real value curve (black curve) well, and there is a smooth transition between each point. While the regression results of the gradient lifting decision tree (blue curve) are consistent with the real curve at some points, it fluctuates greatly at many points.

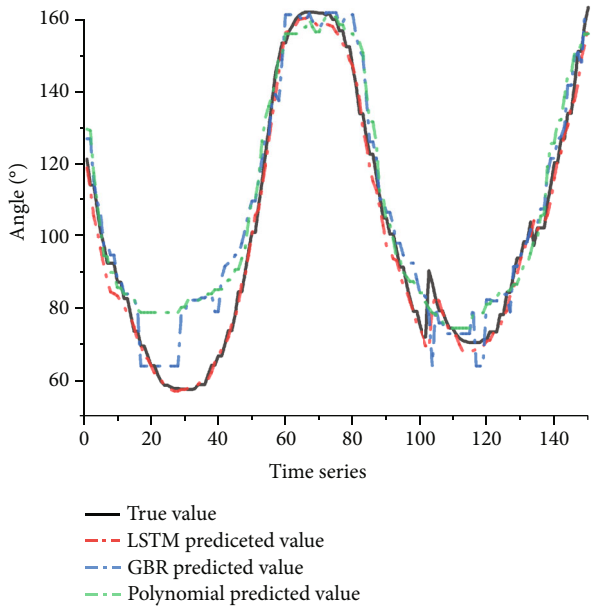


Figure 5: LSTM regression model is compared with the GBR regression model and fourth-order polynomial regression model.

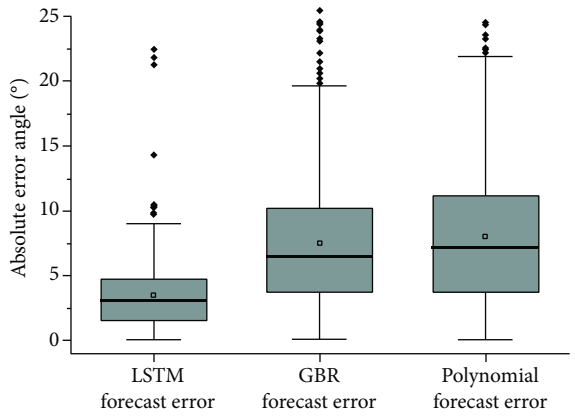


Figure 6: The absolute error statistics for STM regression model, GBR regression model, and polynomial regression model.

Although the fitting result of the fourth-order polynomial (green curve) is relatively smooth, the error between some values and the real values is huge.

We also conducted absolute error statistics for these three regression models, as shown in Figure 6. The mean absolute errors of the LSTM regression model, GBR regression model, and polynomial regression method are 3.45, 7.48, and 7.98, respectively. The LSTM regression model has the smallest overall regression error range.

Table 1 shows the statistics of LSTM regression model prediction results. It can be seen from the table that all error evaluation criteria are quite small, R^2 value reaches 0.9839, the absolute error value is only 3.45 degrees, and the average absolute error percentage is only 3.17%. Therefore, the LSTM regression model can get accurate angle prediction results.

Table 1: LSTM regression model prediction results statistics.

MAE	MSE	RMSE	R^2	RMSLE	MAPE
3.45	18.95	4.35	0.9839	0.0186	0.0317

4.3. Comparison of Regression Results between Extreme Random Tree Regression Model and Other Traditional Machine Learning Regression Models. In this section, we show the results of comparing the first subject's elbow's slow-motion data and EMG data when using extreme random tree regression models with other traditional machine learning regression models in the Regression Shoulder Database. The objective of regression was the EMG signal of the coracobrachialis. There are 18,729 pieces of data set shown in Figure 7(a). We select 70% of the data as the training set and the rest as the test set. The number of folds in cross-validation is 10. Data is normalized using Z-score.

The regression results are shown in Table 2. We show the three models, extreme random tree (ET), random forest (RF), and K-nearest neighbor (KNN) regression models, with the best regression effect among the traditional machine learning regression methods; the extreme random tree takes 4.06 seconds in training time. However, the final R^2 value of extreme random tree regression is the highest, and other error evaluation items are the smallest. We can intuitively see the estimated difference between the predicted value and the actual value through the mean absolute error (MAE) and root mean square error (RMSE). The extreme random tree regression error is the smallest, and it is almost 54% of the regression error of the random forest model, which has the second-best regression effect. Finally, Figure 7(b) shows the residual prediction results and residual distribution of the extreme random tree regression model. The residual is concentrated below the absolute value of 1000 (about 3% of the original EMG data).

Table 3 shows the prediction results of the extreme random tree regression model on the test set. The data are even better than the training results. Therefore, in the experiment, the extreme random tree regression model has the best regression effect and is much better than any other traditional machine learning regression model.

4.4. Stacked Regression Model. In order to further reduce the regression error, we designed a stacking model based on the extreme random tree (Figure 4), stacked the three regression models in Table 2 (including extreme random tree, random forest, and K-nearest neighbor regression model), and then trained and predicted the data.

The first row (beginning with T) of Table 4 shows the training results of the stacking model. The average absolute error and root mean square error are more than 100 smaller than the extreme random tree regression method in the traditional machine learning method (see Table 2), indicating that our stacking model effectively reduces the regression error. Moreover, from the second row of Table 4 (beginning with P), we can see that the stacked regression model also achieved 312 as the results of average absolute error and 457 as the root mean square error, which is about 22% less

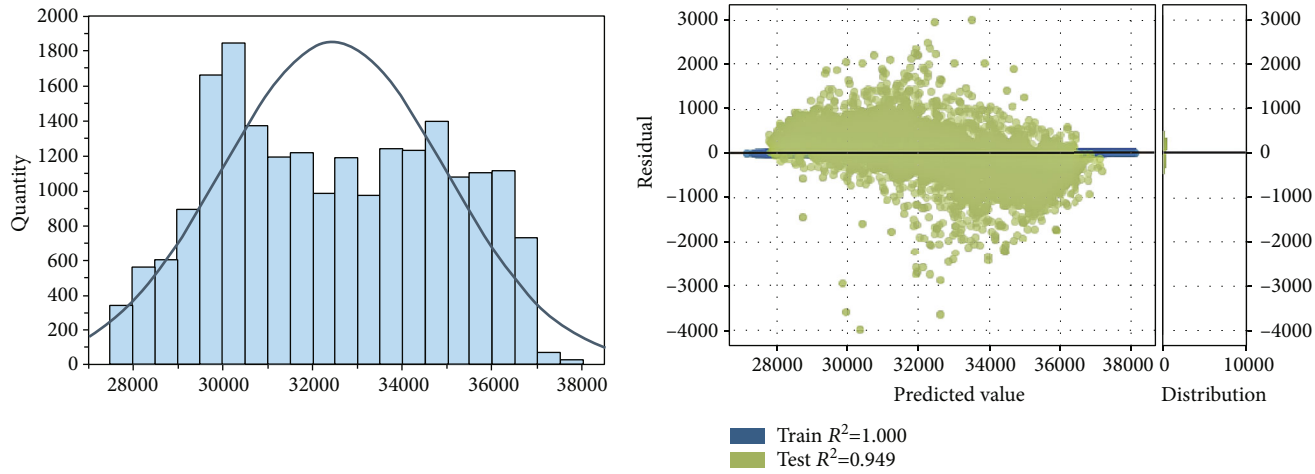


Figure 7: (a) EMG distribution of the first subject's elbow in slow motion; (b) the extreme random tree regression model predicts residual results and their distribution.

Table 2: Comparison of regression results among extreme random tree (ET), random forest (RF), and K-nearest neighbor (KNN) regression models.

Model	MAE	MSE	RMSE	R^2	RMSLE	MAPE	TT (sec)
ET	470.03	443522.74	665.71	0.9281	0.0205	0.0145	4.06
RF	867.68	1168076.13	1080.65	0.8108	0.0333	0.0268	4.37
KNN	1180.42	2146966.71	1465.03	0.6522	0.0451	0.0365	0.75

Table 3: Extreme random tree regression model prediction results statistics.

MAE	MSE	RMSE	R^2	RMSLE	MAPE
404.43	313466.77	559.88	0.9492	0.0172	0.0125

Table 4: Stack regression model training and predicted results (T: train; P: predicted).

	MAE	MSE	RMSE	R^2	RMSLE	MAPE
T	369.24	297341.21	544.81	0.9518	0.0168	0.0114
P	311.74	208847.27	457.00	0.9661	0.0140	0.0096

than the regression model with only extreme random tree. The average absolute percentage error is only 0.96%.

4.5. Stacked Regression Model Regression Results on Different Person Data. We used the stacked regression model to carry out the regression test on the joint angle and EMG signal of shoulder and elbow activity data of different people. The results are shown in Table 5. It can be seen from the average absolute error and root mean square error that the average difference between the predicted value and the real value is within 370; while MAPE showed that the absolute errors of the five subjects were about 1.14%, 0.08%, 0.08%, 2.61%, and 5.18% of the original EMG signal, respectively. The experimental results show that the stacking model can well regress the joint angle and EMG data. Due to the large individual differences of each person, there will be great differences in the relationship between each person's joint angle data and EMG data. Therefore, before using the smart coat for daily activity tracking and EMG signal prediction, users are required to collect standard angle data and corresponding EMG signal data in the hospital according to professional guidance [26] to build a stacked regression model. Therefore, everyone needs to train a unique stacked regression model to facilitate the subsequent regression and analysis of EMG signals.

4.6. Comparison between the Stacking Regression Model and LSTM Regression Effect. We compared the stacking model with the regression model with LSTM module. Table 6 shows the statistics of regression results of the LSTM regression model.

Compared to the regression prediction results of the stacking model in Table 4, better regression results cannot be obtained on this data set by the LSTM regression model than by the stacked regression model. The MAE value of the LSTM regression model reached 547.39, which was 235.45 higher than 311.94 of the stacked regression model. Moreover, the average absolute percentage error of the LSTM regression model is 13.23%, while that of the stacked regression model is only 0.96%. Combining the results of this section with the analysis in Section 4.2, the LSTM regression model is dominant in solving the problems of nonlinearity and hysteresis of flexible sensors, but not in

Table 5: The regression results of the stacked regression model for different people data.

ID	MAE	MSE	RMSE	R^2	RMSLE	MAPE
S1	369.24	297341.21	544.81	0.9518	0.0168	0.0114
S2	91.05	26058.68	161.07	0.9659	0.0014	0.0008
S3	127.61	47251.42	216.72	0.9974	0.0013	0.0008
S4	248.85	247110.06	495.31	0.8851	0.0586	0.0261
S5	48.72	7500.17	86.13	0.9965	0.1218	0.0518

Table 6: Extreme random tree regression model prediction result statistics.

MAE	MSE	RMSE	R^2	RMSLE	MAPE
547.39	462187.33	679.84	0.7065	0.0835	0.1323

the data set of this section, which requires us to select an appropriate regression model for the data from different sources. Our method also has some limitations. Because the training data is limited to the data collected by professional equipment in a short time, it is impossible to predict the irregular changes of myoelectric signals caused by muscle fatigue due to long-term exercise.

5. Conclusions

This paper presents a method for continuously estimating upper limb EMG signals by using an intelligent jacket with integrated, flexible sensors. Firstly, we use the long short-term memory network regression model to continuously estimate the sensor signals collected by the smart coat and the joint angle information collected by Kinect. Then, the shoulder elbow angle information and the corresponding EMG signals of five subjects were aligned and preprocessed, and the stacked regression model based on the extreme random tree was used for regression. The experimental results show that the long short-term memory network can effectively solve the nonlinearity and hysteresis of flexible sensors, and the stacking model can well regress the joint angle data and EMG signal data. This method provides more possibilities for home health monitoring and exercise guidance in the future; for example, users can achieve online medical treatment and online diagnosis based on this system. However, there are still some problems to be solved and studied, such as the sensor position offset and the deformation caused by clothing wrinkles, which will affect the accuracy of the prediction results. Therefore, we will continue to solve the above problems in the follow-up work and the proposed machine learning based pipeline can be applied to other fields, such as bioinformatics and computational biology [67, 71–74].

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Disclosure

The noted sponsors had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript; or in the decision to publish the results that appear herein.

Conflicts of Interest

The authors declare no conflict of interest.

Authors' Contributions

Zhiyong Chen and Qingsuo Wang contributed equally to this work as first authors.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (62072383, 61702433, and 62077039), the Fundamental Research Funds for the Central Universities (20720190006), and the Open Project Program of State Key Laboratory of Virtual Reality Technology and Systems, Beihang University (VRLAB2020B17).

References

[1] D. Maheshwari, S. K. Ghosh, R. K. Tripathy, M. Sharma, and U. R. Acharya, "Automated accurate emotion recognition system using rhythm-specific deep convolutional neural network technique with multi-channel EEG signals," *Computers in Biology and Medicine*, vol. 134, p. 104428, 2021.

[2] S. Li, G. E. Francisco, and P. Zhou, "Post-stroke hemiplegic gait: new perspective and insights," *Frontiers in Physiology*, vol. 9, p. 1021, 2018.

[3] J. M. Ferro, M. G. Bousser, P. Canhão et al., "European stroke Organization guideline for the diagnosis and treatment of cerebral venous thrombosis–endorsed by the European Academy of Neurology," *European Stroke Journal*, vol. 2, no. 3, pp. 195–221, 2017.

[4] Y. Tao, H. Hu, and H. Zhou, "Integration of vision and inertial sensors for 3D arm motion tracking in home-based rehabilitation," *The International Journal of Robotics Research*, vol. 26, no. 6, pp. 607–624, 2007.

[5] S. Filiatrault and A.-M. Cretu, "Human arm motion imitation by a humanoid robot," in *2014 IEEE International Symposium on Robotic and Sensors Environments (ROSE) Proceedings*, Timisoara, Romania, 2014IEEE.

[6] A. Diaz and D. Newman, "Musculoskeletal human-spacesuit interaction model," in *2014 IEEE Aerospace Conference*, Big Sky, MT, USA, 2014IEEE.

[7] E. Wilmes, C. J. de Ruiter, B. J. C. Bastiaansen et al., "Inertial sensor-based motion tracking in football with movement intensity quantification," *Sensors*, vol. 20, no. 9, p. 2527, 2020.

[8] H. Zhou, T. Stone, H. Hu, and N. Harris, "Use of multiple wearable inertial sensors in upper limb motion tracking," *Medical Engineering & Physics*, vol. 30, no. 1, pp. 123–133, 2008.

[9] J. W. Judy, "Microelectromechanical systems (MEMS): fabrication, design and applications," *Smart Materials and Structures*, vol. 10, no. 6, pp. 1115–1134, 2001.

[10] T. von Marcard, R. Henschel, M. J. Black, B. Rosenhahn, and G. Pons-Moll, "Recovering accurate 3D human pose in the wild using IMUs and a moving camera," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018.

[11] Z. Zheng, T. Yu, H. Li et al., "Hybridfusion: real-time performance capture using a single depth sensor and sparse IMUs," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018.

[12] E. Vavrinsky, V. Stopjakova, M. Donoval et al., "Design of sensor systems for long time electrodermal activity monitoring," *Advances in electrical and electronic Engineering*, vol. 15, no. 2, 2017.

[13] M. I. Mokhlespour Esfahani and M. A. Nussbaum, "Preferred placement and usability of a smart textile system vs. inertial measurement units for activity monitoring," *Sensors*, vol. 18, no. 8, p. 2501, 2018.

[14] M. Totaro, T. Poliero, A. Mondini et al., "Soft smart garments for lower limb joint position analysis," *Sensors*, vol. 17, no. 10, p. 2314, 2017.

[15] M. I. Mokhlespour Esfahani, O. Zobeiri, B. Moshiri et al., "Trunk motion system (TMS) using printed body worn sensor (BWS) via data fusion approach," *Sensors*, vol. 17, no. 12, p. 112, 2017.

[16] G. Ge, W. Huang, J. Shao, and X. Dong, "Recent progress of flexible and wearable strain sensors for human-motion monitoring," *Journal of Semiconductors*, vol. 39, no. 1, article 011012, 2018.

[17] A. Nag, S. C. Mukhopadhyay, and J. Kosel, "Wearable flexible sensors: a review," *IEEE Sensors Journal*, vol. 17, no. 13, pp. 3949–3960, 2017.

[18] X. Wang, Z. Liu, and T. Zhang, "Flexible sensing electronics for wearable/attachable health monitoring," *Small*, vol. 13, no. 25, p. 1602790, 2017.

[19] M. Ha, S. Lim, and H. Ko, "Wearable and flexible sensors for user-interactive health-monitoring devices," *Journal of Materials Chemistry B*, vol. 6, no. 24, pp. 4043–4064, 2018.

[20] Y. Liu, H. Wang, W. Zhao, M. Zhang, H. Qin, and Y. Xie, "Flexible, stretchable sensors for wearable health monitoring: sensing mechanisms, materials, fabrication strategies and features," *Sensors*, vol. 18, no. 2, p. 645, 2018.

[21] S. Sundaram, P. Kellnhofer, Y. Li, J. Y. Zhu, A. Torralba, and W. Matusik, "Learning the signatures of the human grasp using a scalable tactile glove," *Nature*, vol. 569, no. 7758, pp. 698–702, 2019.

[22] X. Yu, Z. Xie, Y. Yu et al., "Skin-integrated wireless haptic interfaces for virtual and augmented reality," *Nature*, vol. 575, no. 7783, pp. 473–479, 2019.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] R. Liu, Q. Shao, S. Wang, C. Ru, D. Balkcom, and X. Zhou, "Reconstructing human joint motion with computational fabrics," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 3, no. 1, pp. 1–26, 2019.

[25] G. Ponraj and H. Ren, "Sensor fusion of leap motion controller and flex sensors using Kalman filter for human finger tracking," *IEEE Sensors Journal*, vol. 18, no. 5, pp. 2042–2049, 2018.

[26] Y. Wang, J. Zhou, H. Li et al., "FlexTouch," *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies*, vol. 3, no. 3, pp. 1–20, 2019.

[27] Y. Zhang, K. Chen, J. Yi, T. Liu, and Q. Pan, "Whole-body pose estimation in human bicycle riding using a small set of wearable sensors," *IEEE/ASME Transactions on Mechatronics*, vol. 21, no. 1, pp. 1–174, 2015.

[28] T. Han, S. Bansal, X. Shi et al., "HapBead: on-skin microfluidic haptic Interface using tunable bead," in *Proceedings of the 2020 CHI Conference on Human Factors in Computing Systems*, New York, 2020.

[29] X. Sun, J. Sun, T. Li et al., "Flexible tactile electronic skin sensor with 3D force detection based on porous CNTs/PDMS nanocomposites," *Nano-micro letters*, vol. 11, no. 1, pp. 1–14, 2019.

[30] Y. Ye, C. He, B. Liao, and G. Qian, "Capacitive proximity sensor array with a simple high sensitivity capacitance measuring circuit for human–computer interaction," *IEEE Sensors Journal*, vol. 18, no. 14, pp. 5906–5914, 2018.

[31] O. Glauser, S. Wu, D. Panozzo, O. Hilliges, and O. Sorkine-Hornung, "Interactive hand pose estimation using a stretch-sensing soft glove," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 4, pp. 1–15, 2019.

[32] O. Glauser, D. Panozzo, O. Hilliges, and O. Sorkine-Hornung, "Deformation capture via soft and stretchable sensor arrays," *ACM Transactions on Graphics (TOG)*, vol. 38, no. 2, pp. 1–16, 2019.

[33] D. Kim, J. Kwon, S. Han, Y. L. Park, and S. Jo, "Deep full-body motion network for a soft wearable motion sensing suit," *IEEE/ASME Transactions on Mechatronics*, vol. 24, no. 1, pp. 56–66, 2019.

[34] M. I. M. Esfahani and M. A. Nussbaum, "A "smart" undershirt for tracking upper body motions: task classification and angle estimation," *IEEE Sensors Journal*, vol. 18, no. 18, pp. 7650–7658, 2018.

[35] World Health Organization, *IFC: International Classification of Functioning, Disability and Health*, WHO, 2001.

[36] R. W. Bohannon, "Considerations and practical options for measuring muscle strength: a narrative review," *BioMed Research International*, vol. 2019, Article ID 8194537, 10 pages, 2019.

[37] H. Yu, W. Xu, Y. Zhuang, K. Tong, and R. Song, "Wavelet coherence analysis of muscle coupling during reaching movement in stroke," *Computers in Biology and Medicine*, vol. 131, p. 104263, 2021.

[38] B. J. Borbély and P. Szolgay, "Real-time inverse kinematics for the upper limb: a model-based algorithm using segment orientations," *Biomedical Engineering Online*, vol. 16, no. 1, pp. 1–29, 2017.

[39] C. Pontonnier and G. Dumont, "Inverse dynamics method using optimization techniques for the estimation of muscles forces involved in the elbow motion," *International Journal*

on Interactive Design and Manufacturing (IJIDeM), vol. 3, no. 4, pp. 227–236, 2009.

[40] S. Kim, J. Lee, and J. Bae, "Analysis of finger muscular forces using a wearable hand exoskeleton system," *Journal of Bionic Engineering*, vol. 14, no. 4, pp. 680–691, 2017.

[41] J. Li, Q. Ye, L. Ding, and Q. Liao, "Modeling and dynamic simulation of astronaut's upper limb motions considering counter torques generated by the space suit," *Computer Methods in Biomechanics and Biomedical Engineering*, vol. 20, no. 9, pp. 929–940, 2017.

[42] H. Lim, B. Kim, and S. Park, "Prediction of lower limb kinetics and kinematics during walking by a single IMU on the lower back using machine learning," *Sensors*, vol. 20, no. 1, p. 130, 2020.

[43] G. M. Lozito, M. Schmid, S. Conforto, F. R. Fulginei, and D. Bibbo, "A neural network embedded system for real-time estimation of muscle forces," *Procedia Computer Science*, vol. 51, pp. 60–69, 2015.

[44] R. Bogey, J. Perry, and A. Gitter, "An EMG-to-force processing approach for determining ankle muscle forces during normal human gait," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 13, no. 3, pp. 302–310, 2005.

[45] A. W. Andrews and R. W. Bohannon, "Short-term recovery of limb muscle strength after acute stroke," *Archives of Physical Medicine and Rehabilitation*, vol. 84, no. 1, pp. 125–130, 2003.

[46] M. B. I. Reaz, M. S. Hussain, and F. Mohd-Yasin, "Techniques of EMG signal analysis: detection, processing, classification and applications," *Biological procedures online*, vol. 8, no. 1, pp. 11–35, 2006.

[47] W.-C. Hsu, C. C. Chang, Y. J. Lin, F. C. Yang, L. F. Lin, and K. N. Chou, "The use of wearable sensors for the movement assessment on muscle contraction sequences in post-stroke patients during sit-to-stand," *Sensors*, vol. 19, no. 3, p. 657, 2019.

[48] S. Lee, J. Yoon, D. Lee et al., "Wireless epidermal electromyogram sensing system," *Electronics*, vol. 9, no. 2, p. 269, 2020.

[49] R. P. Di Fabio, "Reliability of computerized surface electromyography for determining the onset of muscle activity," *Physical Therapy*, vol. 67, no. 1, pp. 43–48, 1987.

[50] K. R. Berckmans, B. Castelein, D. Borms, T. Parlevliet, and A. Cools, "Rehabilitation exercises for dysfunction of the scapula: exploration of muscle activity using fine-wire EMG," *The American Journal of Sports Medicine*, vol. 49, no. 10, pp. 2729–2736, 2021.

[51] R. Bogey, A. J. Gitter, and L. Barnes, "Determination of ankle muscle power in normal gait using an EMG-to-force processing approach," *Journal of Electromyography and Kinesiology*, vol. 20, no. 1, pp. 46–54, 2010.

[52] S. Heintz and E. M. Gutierrez-Farewik, "Static optimization of muscle forces during gait in comparison to EMG-to-force processing approach," *Gait & Posture*, vol. 26, no. 2, pp. 279–288, 2007.

[53] D. G. Lloyd and T. F. Besier, "An EMG-driven musculoskeletal model to estimate muscle forces and knee joint moments in vivo," *Journal of Biomechanics*, vol. 36, no. 6, pp. 765–776, 2003.

[54] D. Amarantini, G. Rao, and E. Berton, "A two-step EMG-and-optimization process to estimate muscle force during dynamic movement," *Journal of Biomechanics*, vol. 43, no. 9, pp. 1827–1830, 2010.

[55] J. Hashemi, E. Morin, P. Mousavi, and K. Hashtrudi-Zaad, "Enhanced dynamic EMG-force estimation through calibration and PCI modeling," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 23, no. 1, pp. 41–50, 2015.

[56] S. Kim, K. Ro, and J. Bae, "Estimation of individual muscular forces of the lower limb during walking using a wearable sensor system," *Journal of Sensors*, vol. 2017, Article ID 6747921, 14 pages, 2017.

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

[58] Z. Chen, X. Chen, Y. Ma, S. Guo, Y. Qin, and M. Liao, "Human posture tracking with flexible sensors for motion recognition," *Computer Animation and Virtual Worlds*, vol. 32, no. 5, article e1993, 2021.

[59] J. Chen, Q. Zou, and J. Li, "DeepM6ASeq-EL: prediction of human N6-methyladenosine (m6A) sites with LSTM and ensemble learning," *Frontiers of Computer Science*, vol. 16, no. 2, article 162302, 2022.

[60] "Conductive rubber cord stretch sensor + extras!," https://www.adafruit.com/product/519.

[61] B. Gao, A. Elbaz, Z. He et al., "Bioinspired kirigami fish-based highly stretched wearable biosensor for human biochemical–physiological hybrid monitoring," *Advanced Materials Technologies*, vol. 3, no. 4, p. 1700308, 2018.

[62] G. Wu, F. van der Helm, H. E. Veeger et al., "ISB recommendation on definitions of joint coordinate systems of various joints for the reporting of human joint motion–part II: shoulder, elbow, wrist and hand," *Journal of Biomechanics*, vol. 38, no. 5, pp. 981–992, 2005.

[63] B. Bolsterlee, A. N. Vardy, F. C. T. van der Helm, and H. E. J. (DirkJan) Veeger, "The effect of scaling physiological cross-sectional area on musculoskeletal model predictions," *Journal of Biomechanics*, vol. 48, no. 10, pp. 1760–1768, 2015.

[64] "Seniam," http://www.seniam.org.

[65] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely randomized trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[66] X. Q. Ru, L. H. Li, and Q. Zou, "Incorporating distance-based top-n-gram and random forest to identify electron transport proteins," *Journal of Proteome Research*, vol. 18, no. 7, pp. 2931–2939, 2019.

[67] L. Wei, J. Tang, and Q. Zou, "Local-DPP: an improved DNA-binding protein prediction method by exploring local evolutionary information," *Information Sciences*, vol. 384, pp. 135–144, 2017.

[68] L. Wei, P. Xing, G. Shi, Z. Ji, and Q. Zou, "Fast prediction of protein methylation sites using a sequence-based feature selection technique," *Ieee-Acm Transactions on Computational Biology and Bioinformatics*, vol. 16, no. 4, pp. 1264–1273, 2019.

[69] C. Wu, Q. Li, R. Xing, and G. L. Fan, "Using the Chou's pseudo component to predict the ncRNA locations based on the improved K-nearest neighbor (iKNN) classifier," *Current Bioinformatics*, vol. 15, no. 6, pp. 563–573, 2020.

[70] R. Su, X. Liu, G. Xiao, and L. Wei, "Meta-GDBP: a high-level stacked regression model to improve anticancer drug response prediction," *Briefings in Bioinformatics*, vol. 21, no. 3, pp. 996–1005, 2020.

[71] B. Manavalan, S. Basith, T. H. Shin, L. Wei, and G. Lee, "Meta-4mCpred: a sequence-based meta-predictor for accurate DNA 4mC site prediction using effective feature representation," *Molecular Therapy-Nucleic Acids*, vol. 16, pp. 733–744, 2019.

[72] B. Manavalan, S. Basith, T. H. Shin, L. Wei, and G. Lee, "mAHTPred: a sequence-based meta-predictor for improving the prediction of anti-hypertensive peptides using effective feature representation," *Bioinformatics*, vol. 35, no. 16, pp. 2757–2765, 2019.

[73] L. Wei, Y. Ding, R. Su, J. Tang, and Q. Zou, "Prediction of human protein subcellular localization using deep learning," *Journal of Parallel and Distributed Computing*, vol. 117, pp. 212–217, 2018.

[74] L. Wei, M. Liao, Y. Gao, R. Ji, Z. He, and Q. Zou, "Improved and promising identification of human microRNAs by incorporating a high-quality negative set," *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, vol. 11, no. 1, pp. 192–201, 2014.