SPRING / SUMMER 2023
VOLUME 15, NUMBER 1

Print ISSN: 2152-4157
Online ISSN: 2152-4165

WWW.IJERI.ORG

International Journal of Engineering Research & Innovation

Editor-in-Chief: Mark Rajai, Ph.D.
California State University Northridge

Published by the
International Association of Journals & Conferences

CUTTING EDGE JOURNAL OF RESEARCH AND INNOVATION IN ENGINEERING

Print ISSN: 2152-4157
Online ISSN: 2152-4165

www.ijeri.org

www.iajc.org

INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND INNOVATION

ABOUT IJERI:

- IJERI is the second official journal of the International Association of Journals and Conferences (IAJC).
- IJERI is a high-quality, independent journal steered by a distinguished board of directors and supported by an international review board representing many well-known universities, colleges, and corporations in the U.S. and abroad.
- IJERI has an impact factor of **1.58**, placing it among an elite group of most-cited engineering journals worldwide.

OTHER IAJC JOURNALS:

- The International Journal of Modern Engineering (IJME) For more information visit www.ijme.us
- The Technology Interface International Journal (TIIJ) For more information visit www.tiij.org

IJERI SUBMISSIONS:

- Manuscripts should be sent electronically to the manuscript editor, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For submission guidelines visit www.ijeri.org/submissions

TO JOIN THE REVIEW BOARD:

- Contact the chair of the International Review Board, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For more information visit www.ijeri.org/editorial

INDEXING ORGANIZATIONS:

- IJERI is indexed by numerous agencies. For a complete listing, please visit us at www.ijeri.org.

Contact us:

Mark Rajai, Ph.D.

Editor-in-Chief
California State University-Northridge
College of Engineering and Computer Science
Room: JD 4510
Northridge, CA 91330
Office: (818) 677-5003
Email: mrajai@csun.edu

www.tiij.org

www.ijme.us

International Journal of Engineering Research and Innovation

The INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND INNOVATION (IJERI) is an independent and not-for-profit publication, which aims to provide the engineering community with a resource and forum for scholarly expression and reflection.

IJERI is published twice annually (fall and spring issues) and includes peer-reviewed research articles, editorials, and commentary that contribute to our understanding of the issues, problems, and research associated with engineering and related fields. The journal encourages the submission of manuscripts from private, public, and academic sectors. The views expressed are those of the authors and do not necessarily reflect the opinions of the IJERI editors.

EDITORIAL OFFICE:

Mark Rajai, Ph.D.
Editor-in-Chief
Office: (818) 677-2167
Email: ijmeeditor@iajc.org
Dept. of Manufacturing Systems Engineering & Management
California State University-Northridge
18111 Nordhoff Street
Northridge, CA 91330-8332

The International Journal of Engineering Research and Innovation Editors

Table of Contents

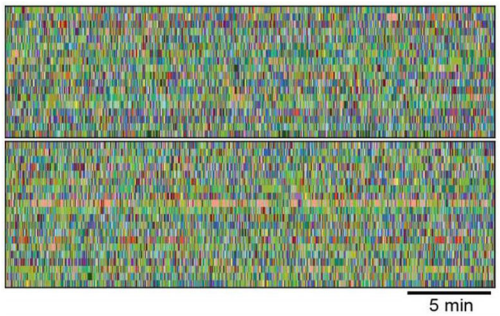
In This Issue (p.30): Machine Learning and Epilepsy Localization

Philip Weinsier, IJERI Manuscript Editor

The authors of the featured article in this issue (p.30) tell us that the rise of machine learning methodologies in recent years has seen great success in a variety of applications. However, this new paradigm is often utilized in limited ways through arbitrary selection of machine learning algorithms and static feature sets, particularly in the medical literature. To showcase the potential power from this expanded use of the machine learning paradigm, the authors applied this framework to the complex medical problem of epileptic seizure localization. Researchers funded by the National Institutes of Health used AI (artificial intelligence) technology to determine behavioral "fingerprints" in mice not found by the human eye. This machine learning approach not only automates behavior analysis but outperforms human assessment, at least in this case using animal (mice) models. Typically, researchers must wait for seizure events, which can oftentimes be rare, in order to make or support their diagnoses. With such automated behavioral phenotyping, however, even a relatively short one hour of video recording was all that was needed for the researchers to better study the disorder and identify potential treatments.

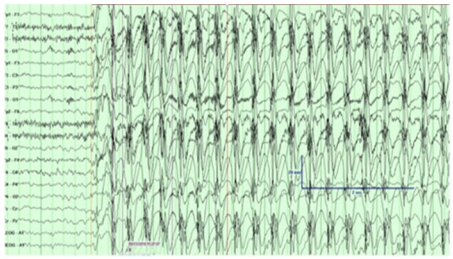
There are four types of epilepsy. Tonic, where muscles in the body become stiff. Atonic, where muscles in the body relax. Myoclonic, where short jerking in parts of the body occur. And clonic, marked by periods of shaking or jerking in parts of the body. If a doctor suspects epilepsy is causing a person's symptoms, he or she may order an electroencephalography, or EEG. EEG is the technology generally used to diagnose epilepsy and is the most specific test for diagnosing epilepsy, because it records the electrical activity of the brain. Idiopathic localization-related epilepsies, or ILRE, are treated with surgery such as laser ablation or temporal lobectomy, especially when the cause of the seizures is an abnormality in the brain such as mesial temporal sclerosis. Localization-related epilepsy is a useful tool for diagnosing and classifying various seizure types. Localization-related epilepsies, also known as focal epilepsies, refer to an abnormal neuronal activity arising from a localized focus and involve a limited portion of the cortex. A localized, or partial (focal) seizure happens when unusual electrical activity affects a small area of the brain. When the seizure does not affect awareness, it is known as a simple partial seizure.

Scientists found that this machine learning-assisted 3D video analysis outperformed the traditional approach in which analyses rely on human observation to label the behavioral signs of epilepsy in animal models during seizures. The labor-intensive process requires constant video monitoring of the mice over many days or weeks, while recording their brainwave activity with electroencephalography (EEG). The team led by Stanford researchers studied mice with acquired and genetic epilepsies. They found that machine analysis was better able to distinguish epileptic versus non-epileptic mice than trained human observers. The AI program also identified distinct behavioral phenotypes at different points in the development of epilepsy.

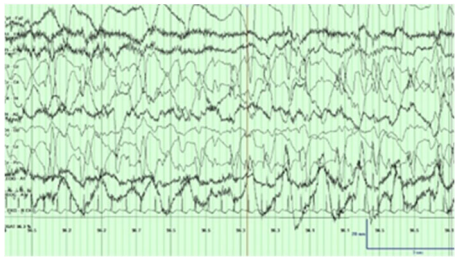


Researchers used a machine learning tool to analyze mouse behavior. Each row represents a sequence of behaviors or "syllables" of a particular mouse. Each syllable is assigned an individual color (controls in the top block; mouse models of epilepsy below). *Soltesz lab, Stanford University School of Medicine.* EEG Neonatal seizure. Contributed by Arayamparambil Anilkumar, MD

EEG showing the characteristic 3Hz spike and wave discharges seen in absence epilepsy. Contributed by Ana C. Albuja, MD



EEG - gen epileptiform. Contributed by Arayamparambil Anilkumar MD



EEG showing the characteristic 3Hz spike and wave discharges seen in absence epilepsy. Contributed by Ana C. Albuja, MD

Sharanya Ramakrishnan & Appaji Rayi. (2023). EEG Localization Related Epilepsies. National Institute of Health, National Library of Medicine (National Center for Biotechnology Information). StatPearls Publishing.

Editorial Review Board Members

Microcontroller-Operated pH and EC Reader

Ferhat Ozdemir, SafeKick Americas LLC; Abdulhamid Zaidi, Sam Houston State University;
Vajih Khan, Sam Houston State University; Iftekhar I Basith, Sam Houston State University

Abstract

The Microcontroller-operated pH (potential of Hydrogen) and EC (electrical conductivity) reader modules monitor and control the pH and EC of water to create a suitable environment for crop irrigation in a hydroponic system. The prototype module implemented in this study improved the stability of the water's pH and EC levels and should ensure long-term usage with minimum maintenance. The modules' microcontroller can support multiple communication protocols to communicate with other devices for long-distance data transmission. With the addition of ProfiNet/ProfiBus/ EthernetIP modules, the modules can support communication with PLCs. A pH sensor acted as a voltmeter and measured the voltage at milli-volt (mV) level. The pH sensor's voltage range was between -414 mV and 414 mV; this range was low enough to have interfered with electromagnetic interference (EMI). The length of the sensor cable needed to be kept as short as possible at the pH reading circuit for an accurate reading. The analog portion of the pH modules read the sensors and amplified the output, then sent it to the microcontroller to process. The microcontroller read the analog value and provided calculations for the pH scale. Also, the microcontroller allows users to perform digital calibration.

Introduction

In a hydroponic growing environment, the need to balance and maintain a proper pH level in the water system is critical. Without the right pH balance, the plant will receive much fewer nutrients and possibly die from lack thereof. In a tank that houses both plant and animal life, the pH level can affect the growth and development of unwanted bacteria and algae formations. The pH level also affects the absorption of various nutrients needed for plant life. Analyzing the chart for pH versus nutrient availability for various nutrients, it shows essential nutrients for plants and the pH level to which plants can absorb these nutrients. The thicker a line of nutrients is, the better plants can take them up. A plant's ability to absorb a few nutrients is severely affected when pH moves to either end. So, the ideal spot where plants can take up many minerals is in the middle, precisely between 6.0-6.5; this is also the range suggested for most common hydroponic crops, such as pepper, tomato, cucumber, eggplant, lettuce, etc.

The recommended pH range above is for hydroponics. Soil growing may require a slightly higher pH. Even though plants may require different pH levels, most plants are safe within this range. Most plants also prefer to live in a somewhat acidic growing environment. In a commercial greenhouse for hydroponic crops, the pH range is maintained at a tighter range of 5.8-6.0, which is controlled by the automatic recirculating systems that fill acids or bases to adjust the pH level to the correct level upon pH change.

pH is the "Potential Hydrogen" in a solution. The pH of pure water around room temperature is about 7 and is considered "neutral" because the concentration of hydrogen ions (H+) is equal to the concentration of hydroxide (OH-) ions produced by dissociation of the water. Increasing the concentration of H+ in relation to OH- results in a pH of less than 7, at which point the solution is considered acidic. Decreasing the concentration of H+ in relation to OH- produces a solution with a pH above 7, and the solution is considered alkaline or "basic" (Meirose, 2020). The pH range of different solutions varies from 0 to 14. A solution with a pH less than 7 is acid, while greater than 7 is base. A pH value equal to 7 is considered neutral. The pH value follows a logarithmic scale, each increment of pH has 10 times more hydrogen ions than the previous pH. For example, the substrate with a pH of 5.0 has one hundred times more hydrogen ions than the substrate with a pH of 7.0 (Kingsta, Saumi & Saranya, 2019).

When a glass membrane separates two solutions containing different concentrations of H+ ions, a voltage potential is developed across the membrane (sensing electrode). A voltage potential is also generated from the reference electrode. The pH meter measures the voltage potential difference (in mV) between the sensing electrode and the outside sample (reference electrode) and displays a pH value via an algorithm. In this current study, two 9-volt batteries powered a high-input-impedance operational amplifier, such as an LMP7702. The pH probe of the meter was connected to the non-inverting input. The output voltage (V_{out}), which is directly proportional to pH, was read with a voltmeter. Theoretically, a pH probe produces about 59 millivolts (mV) per pH unit, and at pH 7 (neutral pH), the probe produces 0 volts. Acid pHs produce negative voltages. Basic pHs produce positive voltages (Building the Simplest Possible pH Meter, n.d.). The pH Meter acts as a voltmeter and translates electrode potential (in mV) to the pH scale. An Arduino was used to store the calibration curve, adjust for temperature changes and electrode slope, and signaled when the reading was stable.

Design

The goal of this current project was to produce a microcontrolled, analog pH reader module that would improve stability and long-term usage with minimum maintenance. Most pH readers require manual calibration and have limited functions. A pH probe is a susceptible sensor that

works at the millivolt level; hence, the sensor must be well isolated from internal (circuit design)/external noises. Additionally, the authors aimed to provide multiple communication protocols, such as serial, SPI, and I2c for custom circuit integration. Furthermore, the module was intended to support field bus protocols, such as Profibus, Modbus, or RS485 to communicate with any other field devices. The module was controlled through commands that were received from the main processor for all of the sensor's operations.

The overarching goal was to be able to successfully construct the pH meter described above. The pH sensor was used to monitor the pH levels of a water solution. The plan was to build prototypes for each section and integrate the modules together to test and achieve communication with multiple protocols to ensure the success of the project. Materials and components consisted of the following:

- Charge pump, negative voltage generator—TPS60400DBV
- Voltage reference—REF3030
- pH probe output amplifier—LMP7202
- Analog/digital converter—MCP3221
- Power supply connection
- Charge pump, negative voltage generator

Analog Devices' family of regulated inverting charge pumps were used to invert an input voltage to a regulated output voltage. These are useful for systems with split-rail positive and negative input supplies. By eliminating the inductor, these switched-capacitor converters offer alternatives to switching-regulator topologies, providing a small solution footprint and a simple design. Key features include low quiescent current; multimode operation with automatic mode switching to maintain regulation; output currents of up to 500 mA; low noise, constant frequency operation; burst-mode operation; and over-temperature, fault, and short -circuit protection (Saaid, Sanuddin, Ali & Yassin, 2015).

The REF3030 is a precision, low noise, low-dropout voltage reference used to provide the level shift needed. The REF30xx series operates with supplies within 1mV of output voltage under zero-load conditions. The REF3030 was chosen based on its low dropout, small size, and low power consumption, as used in the portable and battery-powered project on the SLVS324C datasheet (2020). The output of pH electrodes is usually large enough that they do not require much amplification; however, due to their very high impedance, the output of a pH electrode needs to be buffered before it can go to an analog to digital converter (details about the ADC is covered in the next section). Because most ADCs are operated on a single supply, the output of the pH electrode also needed to be level shifted [SBVS032H datasheet, 2018]. Amplifier A1 buffered the output of the pH electrode with a moderate gain of +2, while A2 provided the level shifting. The voltage output at the A2 terminal is given by Equation 1:

$$\begin{aligned} V_{out} &= -2V \times pH + REF \times 2 \\ &= -2V \times (-177) + 541 \times 2 \\ &= 728mV \end{aligned} \quad (1)$$

where, V_{out} is the output voltage of A2, *pH* is the calculated *pH* value from the *pH* probe voltage reading, and REF is the voltage reference (541 mV).

Microchip's MCP3221 is a successive approximation A/D converter (ADC) with a 12-bit resolution. Available in the SOT-23 package, this device provides one single-ended input with very low-power consumption [SNOSAI9I datasheet, 2015]. Communication to the MCP3221 is performed using a 2-wire, I2C compatible interface. Standard (100 kHz) and Fast (400 kHz) I2C modes are available with the device. An on-chip conversion clock enables independent timing for the I2C and conversion clocks. The device is also addressable, allowing up to eight devices on a single 2-wire bus [DS20001732E datasheet, 2001]. The digital output of the MCP 3221 is equal to the voltage of the MCP out terminal, which equates to the voltage of the op amp output multiplied by 4096 and divided by the reference voltage before being converted to milli-volts, as in Equation 2:

$$\text{MCP3221 digital output} = \nu\text{MCPOut} = \frac{\nu\text{OpAmpOut} \times 4096}{V_{ref} \times 10^3} \quad (2)$$

where, V_{MCPOut} is the voltage read on the analog input and converted to digital, $V_{OpAmpOut}$ is the output voltage of the op amp, 4096 is used to convert the value to 12 bits, and V_{ref} is the reference voltage for analog conversion.

The Nernst equation was used to calibrate the PH sensor and calculate the voltage potential. The relationship between potential and pH is given by Equation 3:

$$E = E_0 - \frac{2.3RT}{nF} \times pH \quad (3)$$

where,
E = measured potential
E_0 = reference potential
R = universal gas Constant
T = temperature (at 25°C)
n = valency of ion (1 for hydrogen)
F = Faraday's constant
pH = 7.0 (neutral level)
Slope = RT/nF = 59.16mv at 25°C

The percentage of slope is the change in mV value divided by the Nernstian theoretical value of 59.2 mV, the expected change in mV per pH unit at 25°C. This means that when the pH sensor is calibrated, the electrode's slope is determined as it relates to the theoretical slope defined by the Nernst equation (Meirose, 2020). Because temperature can have a significant effect on pH measurements, the soft-

ware must be calibrated to take temperature measurements into effect. To find $V_{OpAmpOut}$ use Equation 4:

$$vOpAmpOut = -2 \times vProbe + vOpRef \times 2 \quad (4)$$

where, V_{probe} is the voltage reading of the probe and V_{OpRef} is the reference voltage of the op amp.

Equation 2 shows that the Digital output of the MCP3221 is equal to V_{MCPOut}, which equals $V_{OpAmpOut}$ times 4096 divided by V_{ref} times 10 kilovolts. The REF3030 V_{ref} output equals 3 volts. So V_{out} to the pH raw measurement is given by Equations 5 and 6:

$$vProbe = vOpRef \times 2 - V_{out} \quad (5)$$

$$\text{MCPAnalog to } V_{out} = \frac{\text{MCPAnalog}}{4096} \times V_{ref} \times 10^3 \quad (6)$$

So, the theoretical slope is 59.16 and

$$pH\,4.01 = 160mV$$

$$pH\,7.01 = -15mV$$

$$S = pH\,4.01 - (-pH\,7.01) = 160 - (-15) = 175mV$$

Therefore, the slope is 175/3 = 58.33 mV or, as a percentage, (58.33/59.16) * 100 = 98.6%. A slope value between 95% and 105% is valid for pH measurement and calculation. The percentage in this study was used to show the accuracy of the measurements. The pH conversion formula is Equation 7:

$$7.00 - \frac{\text{Probe Reading}}{\text{Slope}} \quad (7)$$

Design Phase and Validation: PCB Version 1.0

Due to the LMP7702 package limitation, the printed circuit board (PCB) was designed for surface mount devices (SMD). The first PCB prototype was engraved with a computerized numerical control (CNC) milling machine; a successful PCB was completed after several attempts. After the components were soldered, the traces and pads were inspected for possible short circuits; none were found. The circuit was then connected to an Arduino Nano board with the required power. However, there was no voltage output on the other side of the isolator chip. The datasheet was checked multiple times, but such information could not be found. Another 5V was provided on the other side of the isolator chip and measured other component voltages. The voltage on the operational amplifier (OPAMP) was good, except for the REF3030 regulator IC. The reading was 4.3V at the output of REF3030. Version 1.0 had some design errors.

After the test results, and reviewing the schematic and PCB design, it was realized that REF3030's PCB package had incorrect pin assignments. The reason behind this problem was using a different PCB footprint of the chip. The schematic footprint was correct; however, the PCB footprint was expected to be good as well, but it turned out that skipping checking was a mistake. The isolator chip (SI8600) was assumed to be damaged because of REF3030 pin assignments, and it was concluded that this could be a possible reason for no voltage output from the isolator chip. The main reason to this conclusion was that the isolator chip came with special handling instructions due to electrostatic sensitiveness. The pins could have been fixed by changing REF3030 orientation; however, the copper-clad lines would be corroded very quickly due to acid-based soldering paste. The PCB design was revised and the PCB re-engraved. Figure 1 shows the design of PCB version 1.0.

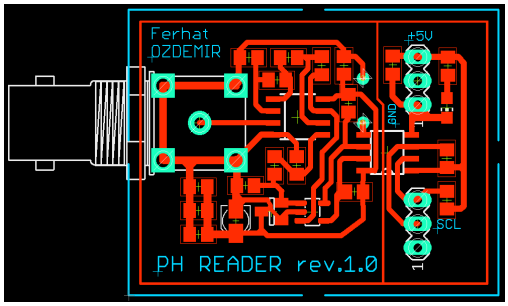
For version 1.1, the same Arduino setup was used; however, no voltage output was read at the other side of the isolator chip again. By providing 5V on the other side of the isolator chip as before, the REF3030 output was still 4.3V, and the chip was quickly getting hot. There was another design error on the PCB version 1.1—it was an incorrect pin assignment on REF3030. After a few design failures, the design finally worked. However, there were two major problems:

1. Access to the MCP3221 using the manufacturer's address assignments failed. The I2C address for the MCP3221 on the manufacturer's datasheet was incorrect. After searching the internet, a sample Arduino sketch for the chip with a different address was found and the authors attempted to access the chip with that I2C address. Surprisingly, communication was then established. The chip I2C address in the datasheet was 110, but the address in the sample sketch was 78. In this way, the problem was solved.
2. How to provide power to the isolated circuit. The authors found that the chip would not isolate the power and that the datasheet might have been overlooked. For an immediate solution, the isolator was removed from the design. After all of these revisions, the entire schematic and PCB were redesigned.

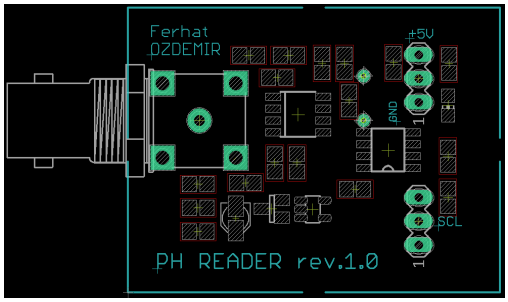
After all these experiments, a successful version of the circuit was finally engraved and developed—PCF version 1.2. As mentioned about the isolator chip, it was removed from this design's version. Figure 2 shows PCB version 1.2.

Final Prototype

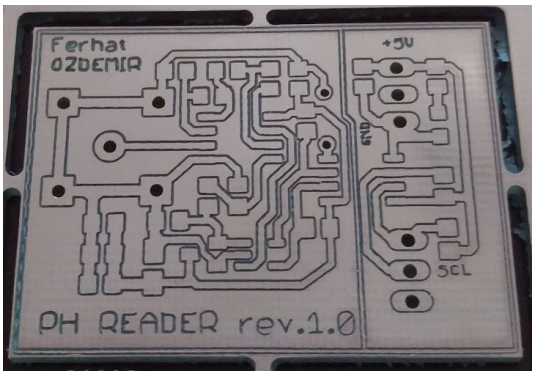
Figure 3(a) shows the final hardware design (version 1.2) with a BNC connector for the pH sensor and an Arduino board used for serial communication. Figures 3(b-c) show a test connection and the test water, respectively. C language was used in the Arduino development environment, which helped to achieve the goal with available Arduino libraries, such as the "MCP3221.h" for a 12-bit ADC microprocessor.



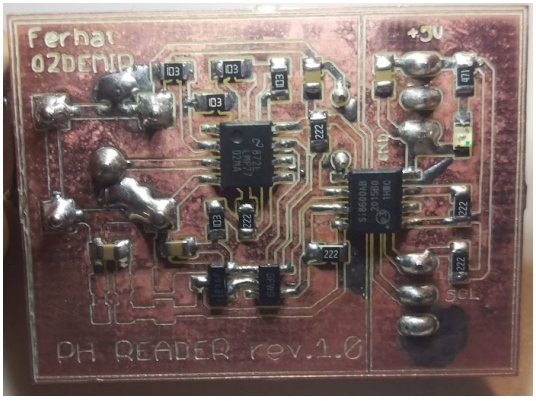
(a) Top view



(b) Components

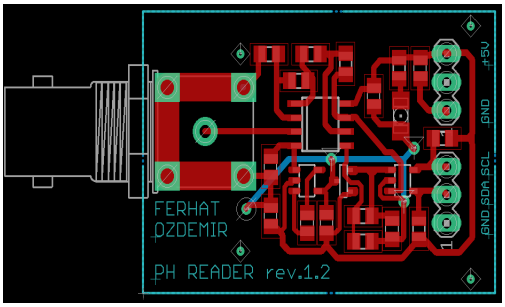


(c) Engraved components

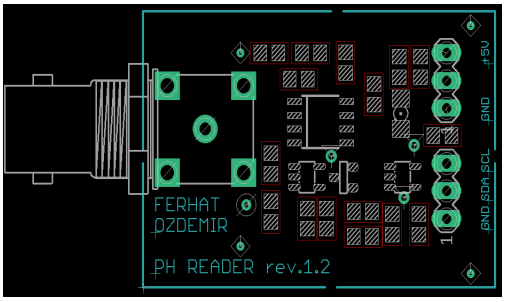


(d) Soldered components

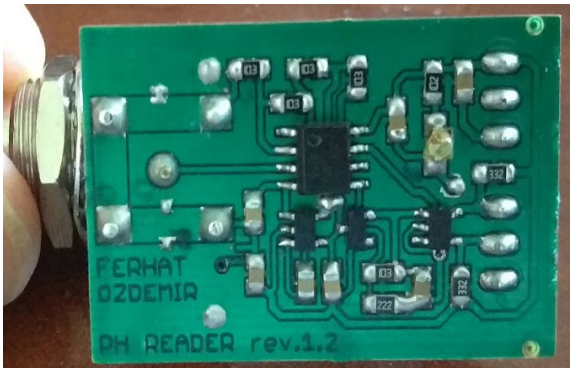
Figure 1. PCB Version 1.0.



a) Top view



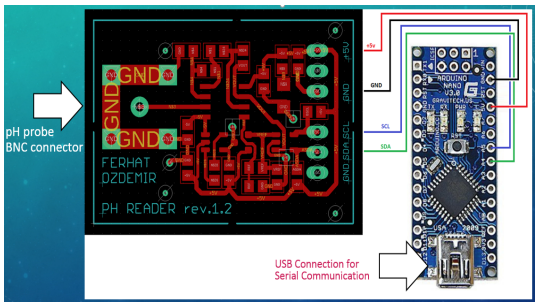
(b) Engraved components



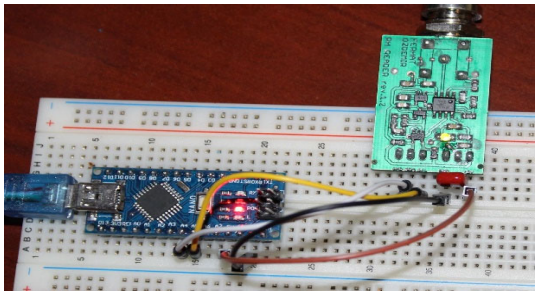
(c) Soldered components

Figure 2. PCB Version 1.2.

The SPI.h Library is for the SPI communication protocol. The stdio.h was imported for a C library for data formatting. For the computer interface, LabVIEW was used, which is a development environment for a visual programming language from National Instruments, to develop a program. Figure 4 shows the main panel of the computer interface that gives the user a real-time reading of the pH levels and monitors the device's status. Figure 5 shows how the current calibration settings could be read and which performed the calibration, if necessary. The calibration panel allows for manual temperature adjustment, if the user does not have a sensor, as well as setting low-, medium-, and high-pH buffers. The interface allows users to see the pH level and easily perform calibration. Figure 6 shows the logic flowchart used in designing the software.



(a) Wiring



(b) Arduino testing

(c) pH calibration solutions and test water

Figure 3. PCB Version 1.2.

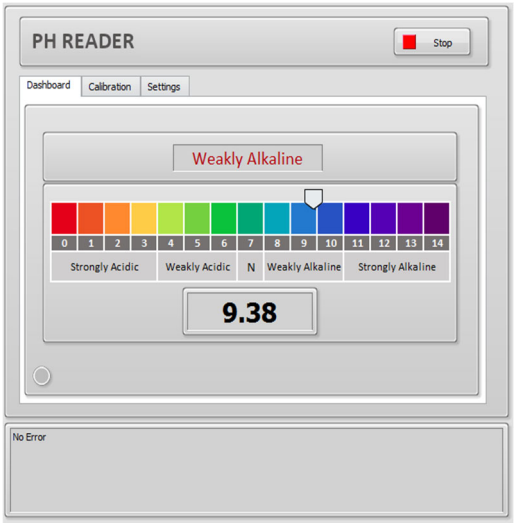


Figure 4. Computer interface main panel design.

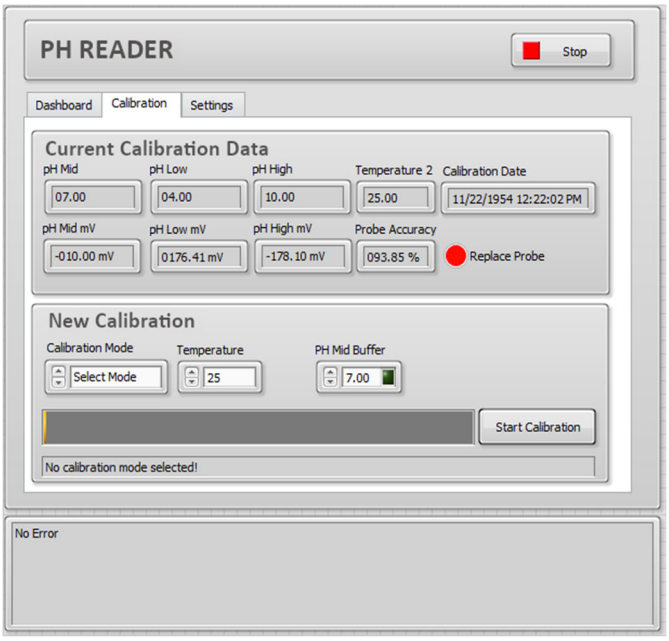


Figure 5. Calibration screen.

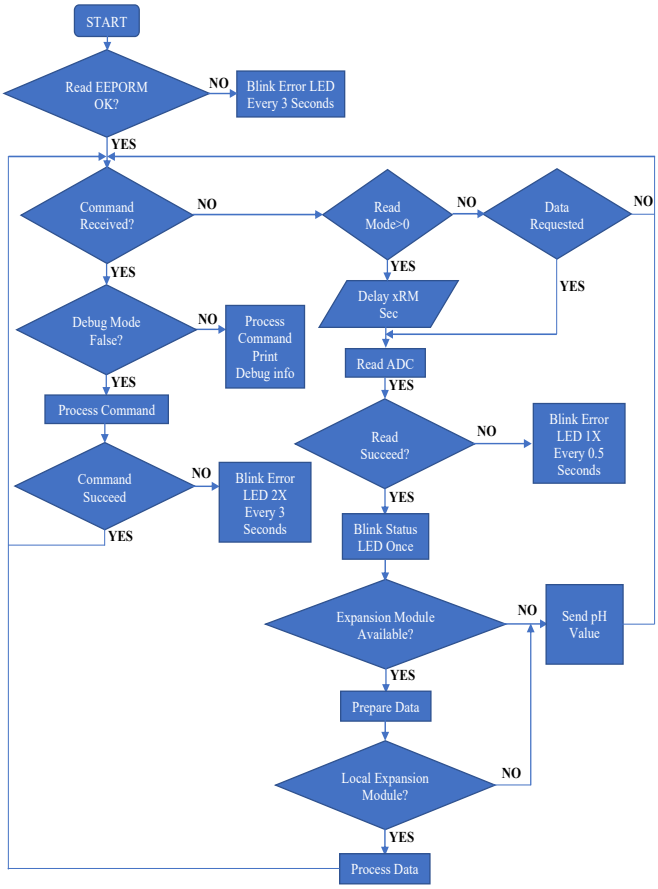


Figure 6. pH reader version 1.2 flowchart.

Table 1 is the variable list for electrically erasable programmable read-only memory (EEPROM) data to store. These variables were also used in a struct model to save to the EEPROM, or load from the EEPROM simultaneously. Table 1 further shows the EEPROM data map used in the project. Table 2 shows the raw command reference table. This is a list of commands for the user to manage the circuit. The default data were stored in the codes in case of resetting EEPROM data.

Table 1. EEPROM data map.

Command	Sample Data	Description	Writable	Command Options
R	6.75	Return current pH level value		
V	v1.0	Return firmware version		
/C00,?	X	Returns selected system mode		0: Normal mode 1: Debug mode
/C01,?	25	Returns default temperature rate	/C01,XX	10-50°C
/C02,?	1	Returns selected expansion module	/C02,x to change module id	0: No expansion (UART only) 1: NRF24L01 Wifi module 2: Ethernet module 3: Relay mode 4: Display mode
/C03,?	9600	Return configured serial baud rate	/C03,x to change baud rate	9600, 19200,…
/C04,?	10/06/2020	Return latest calibration date	n/a	
/C05,?	1234	Returns raw analog pH data	n/a	
/C06,?	Returns calibration point values	Set probe calibrations	/C06,M,7.0 /C06,L,4.0 /C06,H,10.0	M: Calibrate pH 7 L: Calibrate Low-Point number H: Calibrate High-Point number
/C07,?	0	Reading mode	/C07,x to change mode	0: Read on demand 1: Continues mode every 1 sec.
/C08,RESET	n/a	Set to default settings	/C08,RESET	n/a
/C09,?	99.5%	Show pH probe accuracy	n/a	n/a
/C10,X	Data	Sets returning data per by tab selection	/C10,0 /C10,1 C10,2	0: Response pH value continuously 1: Response calibration data continuously 2: Response settings continuously
/C11,X	n/a	Sets calibration mode	/C11,0 /C11,1 /C11.2	0: Calibration mode disable 1: Single point calibration enabled 2: Three points calibration enabled

Table 2. Raw command reference table.

Index	Data Type	Length	Description
0	Float	4 Byte	Version Number
4	Integer	1 Byte	System Mode
5	Integer	1 Byte	Reading Mode
6	Integer	1 Byte	Default Expansion Module ID Number
7	Long	4 Byte	Serial port baud rate
11	Long	4 Byte	Last Calibration date
15	Long	4 Byte	NRF24L01 Address
19	Byte	1 Byte	IP Address Part I
20	Byte	1 Byte	IP Address Part II
21	Byte	1 Byte	IP Address Part III
22	Byte	1 Byte	IP Address Part IV
23	Byte	1 Byte	Subnet Address Part I
24	Byte	1 Byte	Subnet Address Part II
25	Byte	1 Byte	Subnet Address Part III
26	Byte	1 Byte	Subnet Address Part IV
27	Byte	1 Byte	Gateway Address Part I
28	Byte	1 Byte	Gateway Address Part II
29	Byte	1 Byte	Gateway Address Part III
30	Byte	1 Byte	Gateway Address Part IV
31	Float	4 Byte	pH Down Relay Settings: pH Value
35	Float	4 Byte	pH Down Relay Settings: Threshold
39	Float	4 Byte	pH Up Relay Settings: pH Value
43	Float	4 Byte	pH Up Relay Settings: Threshold
47	Integer	1 Byte	Continues mode interval

Conclusions

The development of a micro-controlled, analog pH reader module was successful. Figure 4 shows the data in the computer interface, which are the test's real-time pH readings. Two points need to be mentioned. First, the calibration date on the calibration screen is incorrect. LabVIEW did not convert integer date and time to the standard format. Due to time limitations, fixing this bug was ignored. Second, the new pH probe was not received on time, so the module was tested with a used pH probe. Therefore, the probe accuracy was 93.85%, and the "Replace Probe" indication was illuminating.

Acknowledgements

The authors want to thank the College of Science and Engineering Technology at Sam Houston State University for the internal grant (2021-2022) to complete this project.

References

Building the Simplest Possible pH Meter. (n.d.). https://www.66pacific.com/ph/simplest-ph.aspx

Kingsta, R. M., Saumi, A. S., & Saranya, P. (2019). Design and Construction of Arduino Based pH Control System for Household Wastewater Reuse. Paper presented at the 3rd International Conference on Trends in Electronics and Informatics (ICOEI), Tirunelveli, India, (1037-1041). doi: 10.1109/ICOEI.2019.8862752

Meirose, T., 2020. Essentials Of Ph Measurement. ThermoFisher Scientific, pp.3-5. http://www.ohiowea.org/docs/1500_pH_Seminar_Meirose.pdf

Microchip Technology. (n.d.) Low-Power 12-Bit A/D Converter with I2C Interface. DS20001732E datasheet. https://ww1.microchip.com/downloads/en/devicedoc/20001732e.pdf

Saaid, M. F., Sanuddin, A., Ali, M., & Yassin, M. S. A. I. M. (2015). Automated pH controller system for hydroponic cultivation. IEEE Symposium on Computer Applications & Industrial Electronics (ISCAIE), Langkawi, (186-190). doi: 10.1109/ISCAIE.2015.7298353

Texas Instruments. (n.d.). TPS6040x Unregulated 60-mA Charge Pump Voltage Inverter. SLVS324C datasheet. https://www.ti.com/lit/ds/symlink/tps60402.pdf

Texas Instruments. (n.d.). REF30xx 50-ppm/°C Max, 50-µA, CMOS Voltage Reference in SOT-23-3. SBVS032H datasheet. https://www.ti.com/lit/ds/symlink/ref3030.pdf

Texas Instruments. (n.d.). LMP770x Precision, CMOS Input, RRIO, Wide Supply Range Amplifiers. SNOSAI9I datasheet. https://www.ti.com/lit/ds/symlink/lmp7701.pdf

Biographies

FERHAT OZDEMIR graduated from the Department of Engineering Technology at Sam Houston State University in 2021. He is currently working at SafeKick Americas LLC. Mr. Ozdemir may be reached at Ferhat_kL@yahoo.com

ABDULHAMID ZAIDI is a visiting assistant professor in the Department of Engineering Technology at Sam Houston State University since the fall of 2021. Dr. Zaidi may be reached at amz024@shsu.edu

VAJIH KHAN is an adjunct instructor in the Department of Engineering Technology at Sam Houston State University. Mr. Khan may be reached at vxk009@shsu.edu

IFTEKHAR IBNE BASITH is an assistant professor in the Department of Engineering Technology at Sam Houston State University since the spring of 2017. Dr. Basith may be reached at iib002@shsu.edu

On the Surface Finish of 3D-Printed Jewelry: An Undergraduate Research Project

Hani Saad, Eastern Washington University; Heechang Bae, Eastern Washington University;
Roman Semivrazhnov, Eastern Washington University; David Teterin, Eastern Washington University

Abstract

In this study, the authors analyzed the surface finish of 3D-printed precious-metals samples. The research was carried out by undergraduate mechanical engineering students in their senior year. Different samples of 3D-printed jewelry pieces were analyzed to determine the quality of their surface finish. The exact composition was unknown, as these were provided by a company that wished to keep it secret. However, these samples were created using direct metal laser sintering (DMLS) with atomized powder metal. The main objective was to ameliorate the surface finish through a series of surface treatments. The students were not told what processes to consider, but they were provided access to shops and machines and were responsible for designing all the necessary setups and jigs as well as running the experiments.

In addition, limited resources and access to equipment, due to COVID restrictions, forced them to improvise, which made the project even more challenging. Polishing was done in three ways, hand sanding, sand blasting, and tumbling. To ensure accurate results, hand sanding was automated by creating a mechanical jig capable of applying a constant pressure and identical conditions for each sample. The results obtained were analyzed and compared to the values obtained before polishing. The results showed that the greatest increase in the quality of surface finish was achieved by sanding, followed by sand blasting and finally tumbling. In some instances, such as in tumbling, the surface finish did not show improvement, but rather quite a bit of deterioration. In other instances, it was difficult reaching some areas such as in hand sanding. The major part of the effort was in deciding what methods to use, how to properly implement them, and how to draw the proper conclusions from the data obtained, given all the constraints present.

Introduction

Additive manufacturing (AM) is a process where material is added to form the object rather than removing material or using some form of a mold. Subtractive manufacturing (SM) requires that material be removed to form the desired part, utilizing processes such as cutting, milling, grinding, drilling, etc. Requiring a multitude of machines and the expertise to use each one individually further limits the capabilities for rapid prototyping with SM. Historically, AM has been used primarily to fill the role of rapid prototyping, but it is expanding and increasingly used to create end-user products (Caulfield, McHugh & Lohfeld, 2007). The use of AM has allowed an increase in custom or specialized products, while reducing waste and manufacturing time (Gibson, Rosen & Stucker, 2015; Wang, Jiang, Zhou, Gou & Hui, 2017; Qin, Qi, Scott & Jiang, 2019; Laureijs, Roca, Narra, Montgomery, Beuth & Fuchs, 2017; Korium, Roozbahani, Alizadeh, Perepelkina & Handroos, 2021). The product obtained will have mechanical and physical properties that depend on many factors, including printing parameters such as layer thickness and build orientation (Hanon, Dobos & Zsidai, 2021).

Among many different AM processes, direct metal laser sintering (DMLS) is widely used in building metal products from a CAD file by selectively fusing metal powder into thin layers (Xometry, 2022). In DMLS, layer-by-layer printing allows multiple parts to be combined during the printing process (Korium et al., 2021). Working with precious metals can be challenging for more than one reason; the most obvious being cost. This makes manufacturing very expensive and parts difficult to obtain. In addition, technically challenging issues result partly from the high thermal conductivities of metals such as gold and silver. Platinum is another metal that is used as well. It is very difficult to cast, but some companies, such as Cooksongold (n.d.) were able to produce a density of 99.9% using AM compared to 99.2% for cast platinum, and they currently have a wide range of platinum filaments available.

Current DMLS capabilities have limitations, due to the use of the sintering process (Cooper, 2015). Sintering is a thermal process of converting loose fine particles into a solid mass by applying heat and/or pressure without fully melting the particles to the point of liquefaction. The sintering process deals with densification of a powder by bonding individual powder particles via solid-state diffusion (Groza & Shackelford, 2007). This process involves the atoms in materials diffusing across the particle boundaries and fusing together into one piece. Sintering occurs naturally in mineral deposits and is used as a manufacturing process for materials that include ceramics, metals, and plastics (Popovich & Sufiiarov, 2016; Agarwal, Sarkar, Das & Dixit, 2016). The surface finish is a property of DMSL that is limited by the sintering process. The sintering process does not form a continuous and smooth surface finish. Instead, it consists of ridges and voids of material. Such surface texture is undesirable for jewelry properties where a smooth mirror surface is usually desired. This significantly impacts the quality and the surface finish of the product. For those reasons, it will be beneficial for the jewelry market if the DMLS technology and post-surface treatments are exploited efficiently.

In this current research project, the authors explored the effects of surface treatments on the surface quality and roughness of 3D-printed jewelry pieces using DMLS technology. Five samples of DMLS 3D-printed jewelry samples were obtained, although their specific compositions and printing parameters were not provided. These were secured through a fellow researcher working on the feasibility of using additive manufacturing for creating jewelry pieces. This was part of his master's degree at ESCP (Ecole Superieure de Commerce de Paris), London. The exact sample compositions and printing parameters were considered trade secrets and were not divulged. It is known, however, that the parts were created using DMLS (direct metal laser sintering) by Cooksongold.

The students were then asked to investigate the effects of the three following processes on the surface finish: sanding, sandblasting, and tumbling. They had to make decisions on which processes were possible and likely to yield results. They also had to design and build any jigs they might need in the process. This research project was performed during COVID lockdown, when support and resources for students and faculty were at a minimum. This included shop access, machine maintenance, parts ordering, etc. As a result, students had to improvise and adjust to these constraints. Additional methods, including waterjet and laser treatment, were considered but had to be dismissed due to a lack of support and maintenance.

Experimental Procedure

Surface finish measurements were performed using a calibrated Mitutoyo surface roughness tester SJ – 410 and surface profiles were recorded with a cutoff length of 0.8 mm. Figures 1 and 2 show the setup and samples, respectively. Figure 3 shows how measurements were performed on a number of locations on the samples and in different directions. The arithmetic average height (Ra) and total height of roughness profile (Rt) measurements were used as the basis for comparison. If the surface is imagined as having peaks and valleys, the Ra value is the average mean of the absolute values of the peaks and valleys of the surface, while the Rt value represents the difference between the deepest valley and highest peak. Three different processes were applied to all samples in order to improve surface finish quality.

Figure 1. Profilometer setup used in the experiment. Mitutoyo surface roughness tester SJ – 410.

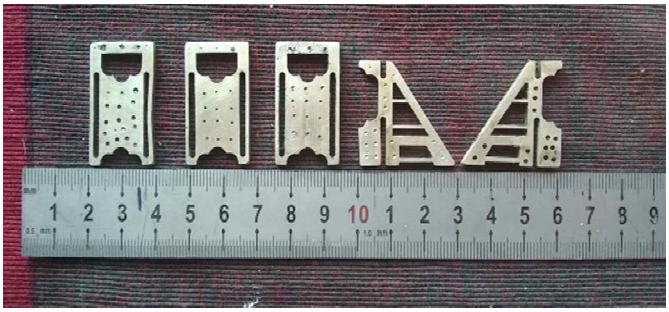


Figure 2. Samples 1-5 (left to right).

The first process used was sanding at 1500 grit. Surface epoxy was used to mount the samples to a 0.875-lb rod during sanding. Figure 3 shows how sanding was done in a figure-8 pattern and run 150 times. Care was taken to ensure that the samples were flat against the sandpaper. After sanding, Dykem was used to check the flatness of the samples. Figures 4 and 5 show the sanding apparatus and the results of using Dykem.

(a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Sample 4. (e) Sample 5.

Figure 3. Surface roughness test directions. Arrows represent the direction along which the different measurements were performed.

(a) Sanding process.

(b) Mounting the sample to a 0.875-lb rod and sanding in a figure-8 pattern.

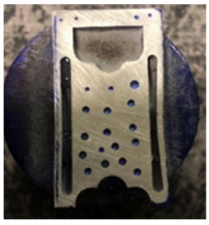
Figure 4. The sanding apparatus and the results of using Dykem.

The second process was sandblasting using 70-grit, black aluminum oxide as the blasting media. Each sample was sandblasted for 30 seconds using a back-and-forth motion to ensure a uniform process. The choice of the time was a rough estimate. Due to the limited number of samples available, no additional testing at different times was possible. Thirty seconds was long enough for sandblasting to treat the surfaces, and short enough not to remove excess material. Figure 6 shows the sandblasting setup.

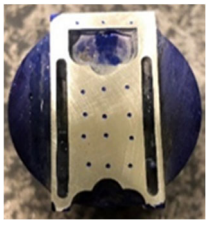
(a) Sandblasting setup used.

(a) Sample preparation.

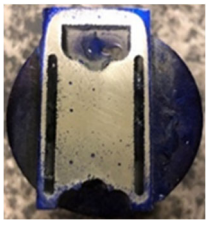
Figure 6. Sandblasting setup and sample preparation.



(a) Sample 1.



(b) Sample 2.



(c) Sample 3.

(d) Sample 4.

(e) Sample 5.

Figure 5. Flatness check of the sanded parts using Dykem.

The final process was tumbling. The same black aluminum oxide used for sandblasting was also used as the abrasive material for tumbling, and the parts were tumbled for 48 hours. Figure 7 shows the setup. Samples were sanded on one side entirely, as it was very difficult to perform sanding on part of the area and not all of it. But for tumbling and sand blasting, it was possible to protect half the area, while the other half was treated. Figure 6 shows that this was done using heavy-duty tape. It was important to apply only one process per area so that its effects could be compared effectively to the untreated one. Due to the sample sizes, it was not possible to consider additional processes or to modify some parameters on the current ones.

Experimental Data

Tables 1-3 show the results obtained for sanding, sandblasting, and tumbling, respectively. Table 4 shows the values of the standard deviations of the Ra and Rt values measured for each sample. Figures 8-12 show the values obtained for Ra for each specimen and for each process. Figures 13-17 show the average Ra values and their corresponding standard deviations.

Figure 7. Tumbling setup and media used.

Table 1. Average surface roughness values for sanding.

Average Surface Roughness Values for Sanding						
	Untreated		Sanding		Percent Change	
Sample #	Ra (µm)	Rt (µm)	Ra (µm)	Rt (µm)	Ra (µm)	Rt (µm)
Sample 1	0.343	2.720	0.219	3.107	-36.2%	14.2%
Sample 2	0.450	4.370	0.151	1.729	-66.4%	-60.4%
Sample 3	6.727	43.690	0.186	2.803	-97.2%	-93.6%
Sample 4	0.340	2.687	0.146	1.595	-57.1%	-40.6%
Sample 5	10.145	67.112	0.153	1.998	-98.5%	-97.0%

Table 2. Average surface roughness values for sandblasting.

Average Surface Roughness Values for Sandblasting						
	Untreated		Sandblasting		Percent difference	
Sample #	Ra (µm)	Rt (µm)	Ra (µm)	Rt (µm)	Ra (µm)	Rt (µm)
Sample 1	0.343	2.720	1.646	12.965	379.9%	376.7%
Sample 2	0.450	4.370	1.596	14.021	254.7%	220.8%
Sample 3	6.727	43.690	2.138	15.799	-68.2%	-63.8%
Sample 4	0.340	2.687	1.419	11.348	317.4%	322.3%
Sample 5	10.145	67.112	3.793	30.979	-62.6%	-53.8%

Table 3. Average surface roughness values for tumbling.

Average Surface Roughness Values for Tumbling						
	Untreated		Tumbling		Percent difference	
Sample #	Ra (µm)	Rt (µm)	Ra (µm)	Rt (µm)	Ra (µm)	Rt (µm)
Sample 1	0.343	2.720	0.408	4.032	19.0%	48.2%
Sample 2	0.450	4.370	0.540	5.587	20.0%	27.8%
Sample 3	6.727	43.690	6.354	38.348	-5.5%	-12.2%
Sample 4	0.340	2.687	0.703	7.809	106.8%	190.6%
Sample 5	10.145	67.112	8.182	53.151	-19.3%	-20.8%

Table 4. Standard deviation values of the surface roughness results.

Standard Deviation Values								
	Untreated		Sanding		Sandblasting		Tumbling	
Sample #	Ra (μm)	Rt (μm)	Ra (μm)	Rt (μm)	Ra (μm)	Rt (μm)	Ra (μm)	Rt (μm)
Sample 1	0.054	1.751	0.096	1.677	0.145	1.983	0.086	1.325
Sample 2	0.202	1.141	0.032	0.391	0.195	2.733	0.114	2.351
Sample 3	1.086	8.340	0.097	1.808	0.545	4.946	0.979	8.212
Sample 4	0.039	0.464	0.040	0.410	0.128	3.574	0.470	4.270
Sample 5	2.191	8.666	0.021	0.955	0.834	7.595	1.573	14.688
Avg.	0.714	4.072	0.057	1.048	0.369	4.166	0.644	6.169

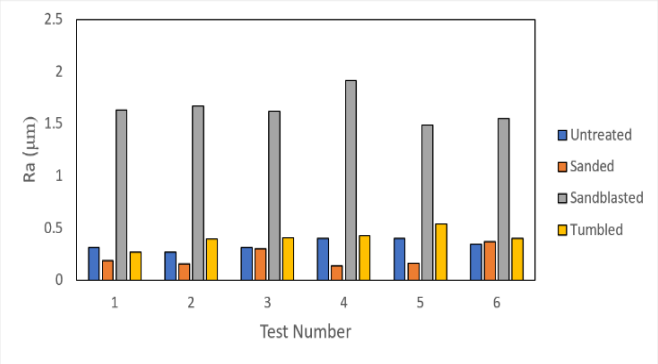


Figure 8. Sample 1 surface roughness data.

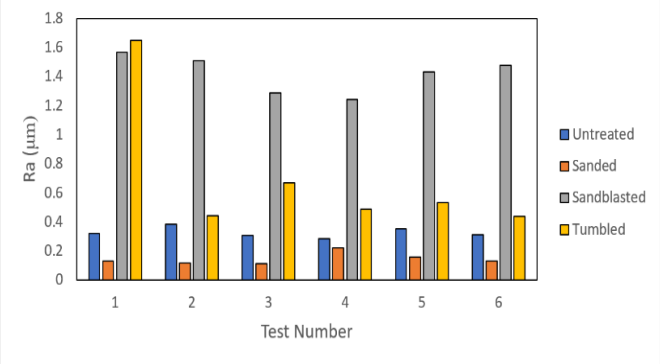


Figure 11. Sample 4 surface roughness data.

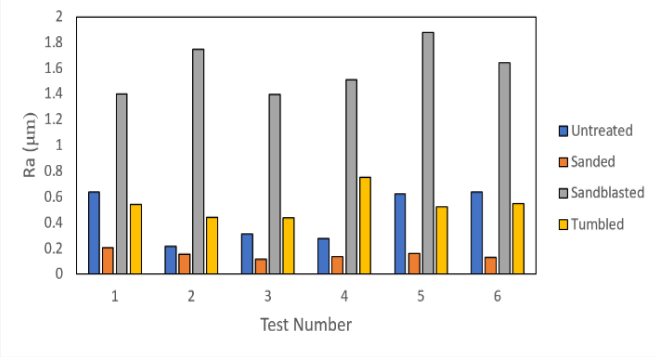


Figure 9. Sample 2 surface roughness data.

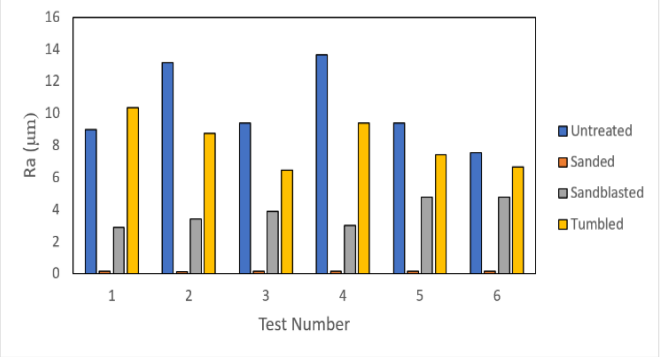


Figure 12. Sample 5 surface roughness data.

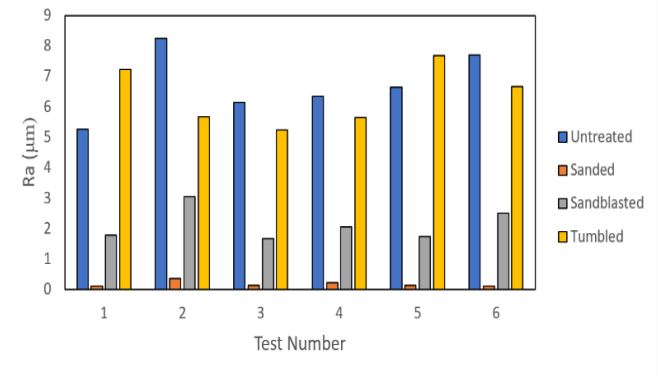
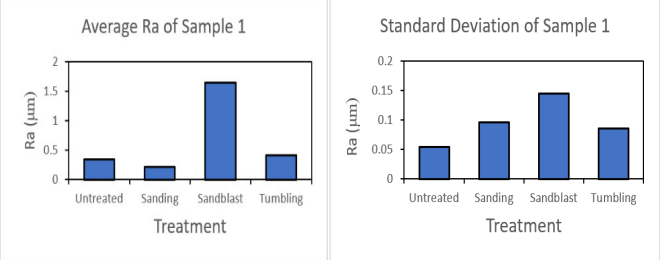
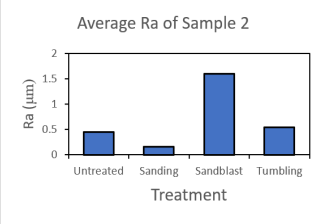


Figure 10. Sample 3 surface roughness data.

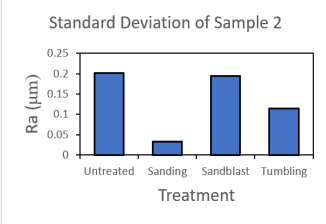


(a) Average Ra. (b) Standard deviation.

Figure 13. Sample 1.

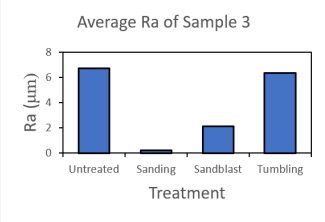


(a) Average Ra.

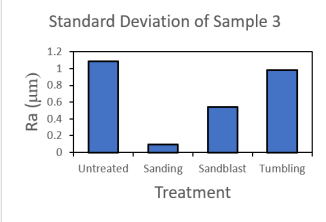


(b) Standard deviation.

Figure 14. Sample 2.

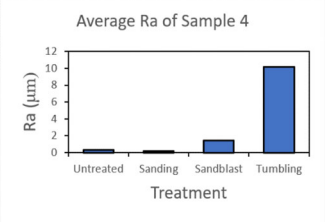


(a) Average Ra.

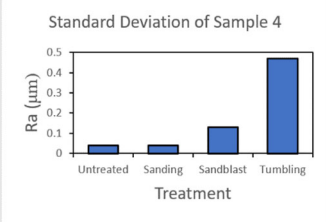


(b) Standard deviation.

Figure 15. Sample 3.

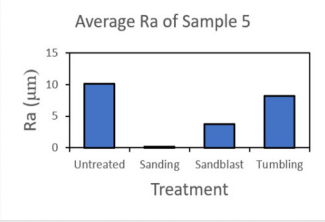


(a) Average Ra.

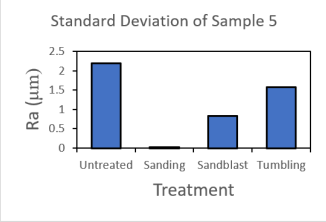


(b) Standard deviation.

Figure 16. Sample 4.



(a) Average Ra.



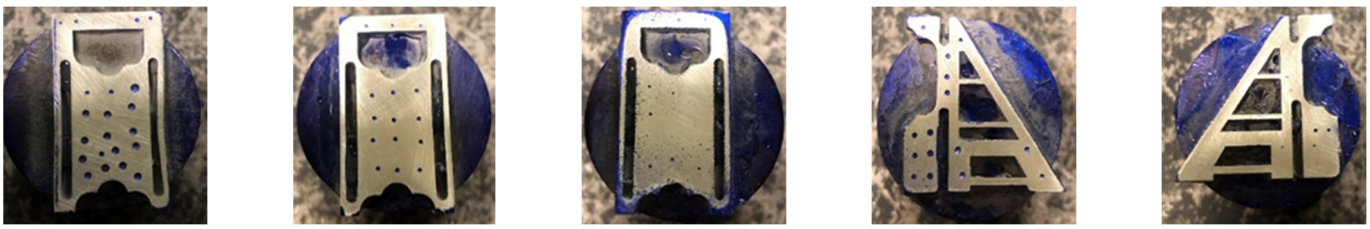
(b) Standard deviation.

Figure 17. Sample 5.

Discussion

Table 1 shows the average surface roughness values for sanding. For all of the samples, the surface roughness decreased substantially. The roughest samples were Sample 5 and 3, which saw 97% and 98% decreases in the Ra values, respectively. The Rt value for Sample 5 saw a 97% decrease and Sample 3 saw a 94% decrease in Rt value. The Rt value for Sample 1, however, experienced a 14% increase. On the third run of the surface measurement, abnormally high values were recorded for Ra and Rt. A visual inspection of the part showed a scratch in the area that was tested. This scratch was most likely made when handling the part in the lab. If this data run were not incorporated into the calculations, the Rt value would have decreased by 8%. Figures 18-20 show the surfaces of the samples for each of the processes performed. The second process performed was sandblasting. Table 2 shows the surface roughness values measured. For the parts that were relatively smooth (Samples 1, 2, and 4) the surface roughness increased substantially. The Ra value of Sample 1 increased by 380%, Sample 2 by 255%, and Sample 4 by 317%. For these three parts, sandblasting with the media selected was certainly not a viable option for increasing the smoothness. However, for the rough surfaces of Samples 3 and 5, it was a possibility. Both of these had a decrease in Ra and Rt values. After sandblasting, Sample 3 had a decrease of 68% for the Ra value and a decrease of 64% for the Rt value. Sample 5 experienced similar results.

This was due to the fact that these samples had a large number of sharp edges on the surface that were smoothed out after being sandblasted. Figure 19 shows that, after sandblasting, Sample 5 showed clear and severe pitting. The last process performed was tumbling the samples with media for two days. Table 3 shows the results of the surface finish. This process provided similar results to sandblasting. Samples 1, 2, and 4 all ended up with a rougher surface. Samples 1 and 2 both saw a 20% increase in Ra values and a 48.2% percent and 27.8% increase in the Rt values, respectively. Sample 4 saw the highest change of all the samples with a 106.8% increase in the Ra value and a 191% increase in the Rt value. As with sandblasting, the rougher finishes of Samples 3 and 5 were smoothed out. The Ra value of Sample 3 decreased by 5% and the Rt value decreased by 12%. Sample 5 saw a 19% decrease in the Ra value and a 21% decrease in the Rt value.



(a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Sample 4. (e) Sample 5.

Figure 18. Samples after sanding (note that Dykem was used to check for flatness).

(a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Sample 4. (e) Sample 5.

Figure 19. Samples after sandblasting.

(a) Sample 1. (b) Sample 2. (c) Sample 3. (d) Sample 4. (e) Sample 5.

Figure 20. Samples after tumbling.

For the Ra values, it was clear from these data that sanding provided the most uniform surface finish across all cases, including the untreated sample. For sand blasting and tumbling, all measurements except for Sample 3 yielded lower standard deviations. For the Rt values, the same was observed, except that for Sample 5 the difference was even more noticeable, the standard deviation went from a value of 0.95 for sanding to 7.95 for sandblasting to 14.69 for tumbling. Table 1 shows that Sample 5, in particular, had the roughest untreated surface. In order to create more uniform 3D-printed jewelry pieces that will keep their strength and their surface characteristics, more analysis should be conducted to try to relate the composition of the sample, the printing parameters, the heat dissipation rate to surface finish, and the general physical properties. The challenge was not only in obtaining these relationships but also procuring larger samples. These are usually difficult to procure and expensive to create.

Conclusions

The surface finish of five samples of 3D-printed jewelry pieces were analyzed by mechanical engineering students in their senior year. Sanding, sandblasting, and tumbling were performed on all samples and the resulting surface finish measured. Ra and Rt values were used as a reference. Sanding at 1500 grit showed the best results. In this case, the surfaces saw an improvement with a decrease in the Rt and Ra values in the range of 90%. Sandblasting showed mixed results with three of the five samples showing a drastic increase in the Ra and Rt values, up to 380%, while the other two showed about a 60% decrease in the Ra and Rt values. Tumbling yielded similar results to sandblasting and the Ra and Rt values increased for three samples up to 106% and 191%, respectively. Increases in these values were observed in the remaining three samples, up to 19% and 21% for Ra and Rt on one sample, respectively.

References

Agarwal, S., Sarkar, S., Das, M., & Dixit, A. R. (2016). Tribo-mechanical characterization of spark plasma sintered chopped carbon fibre reinforced silicon carbide composites. *Ceramics International*, *42*(16), 18283-18288.

Caulfield, B., McHugh P., & Lohfeld, S. (2007). Dependence of mechanical properties of polyamide components on build parameters in the SLS process. *Journal of Materials Processing Technology*, *182*, 477-488.

Cooksongold. (n.d.). *Cooksongold: Jewellery Making Supplies | UK Supplier*. https://www.cooksongold.com/

Cooper, F. (2015). Sintering and additive manufacturing: the new paradigm for the jewellery manufacturer. *Johnson Matthey Technology Review*, *59*(3), 233-242.

Gibson, I., Rosen, D., & Stucker, B. (2015). *Additive Manufacturing Technologies* (2nd ed.). New York, NY, USA: Springer-Verlag.

Groza, J. R., & Shackelford, J. F. (Eds). (2007). *Materials Processing Handbook*. CRC Press.

Hanon, M. M., Dobos, J., & Zsidai, L. (2021). The influence of 3D printing process parameters on the mechanical performance of PLA polymer and its correlation with hardness. *Procedia Manufacturing*, *54*, 244-249.

Korium, M. S., Roozbahani, H., Alizadeh, M., Perepelkina, S., & Handroos, H., (2021). Direct Metal Laser Sintering of Precious Metals for Jewelry Applications: Process Parameter Selection and Microstructure Analysis. *IEEE Access*, *9*, 126530-126540.

Laureijs, R. E., Roca, J. B., Narra, S. P., Montgomery, C., Beuth, J. L., & Fuchs, E. R. H., (2017). Metal additive manufacturing: Cost competitive beyond low volumes. *Journal of Manufacturing Science and Engineering*, *139*(8).

Popovich, A., & Sufiiarov, V. (2016). Metal powder additive manufacturing. In *New trends in 3D printing*. IntechOpen.

Qin, Y., Qi, Q., Scott, P. J., & Jiang, X. (2019). Status, comparison, and future of the representations of additive manufacturing data. *Computer-Aided Design*, *111*, 44-64.

Wang, X., Jiang, M., Zhou, Z., Gou, J., & Hui, D. (2017). 3D printing of polymer matrix composites: A review and prospective. *Composites Part B: Engineering*, *110*, 442-458.

Xometry. (2022). *Direct Metal Laser Sintering (DMLS) 3D Printing Service.* https://www.xometry.com/capabilities/3d-printing-service/direct-metal-laser-sintering/

Biographies

HANI SAAD earned his BS and MS degrees in mechanical engineering from Marquette University in 1997 and 1999, respectively. He received his PhD from Washington State University also in mechanical engineering in 2005. He is currently a professor at Eastern Washington University. Dr. Saad may be reached at hsaad@ewu.edu

HEECHANG BAE is an assistant professor in the Mechanical Engineering and Technology department at Eastern Washington University. His research focuses on producing numerical models and experimental optimization to improve the strength and durability of aerospace materials. His research interests include engineered materials, durability of materials, damage tolerance, FEA, and additive manufacturing. Dr. Bae received MS and PhD degrees in mechanical engineering from the University of Washington. Dr. Bae may be reached at hbae1@ewu.edu

ROMAN SEMIVRAZHNOV earned his BS in Mechanical Engineering from Eastern Washington University in 2021. Mr. Semivrazhnov may be reached at rsemivrazhnov@eagles.ewu.edu

DAVID TETERIN earned his BS in Mechanical Engineering from Eastern Washington University in 2021. Mr. Teterin may be reached at dteterin@eagles.ewu.edu

Internet of Things (IoT) with Embedded Intelligence for Smart Environments: A Case Study

Sangho Park, Central Connecticut State University

Abstract

The development of portable low-cost hardware/software systems has been considered increasingly important in experiential learning of technology and hands-on experiments in engineering research. The Internet of Things (IoT), implemented through the embedded systems approaches, is a useful domain for the purpose. In this paper, the author presents a case study that shows the design, implementation, and testing of an IoT system using low-cost off-the-shelf components. The proposed system consisted of multiple sensor nodes dispatchable to remote locations and a server computer that a user could control locally. The sensor nodes and the server computer were connected wirelessly for the transmission of sensory data and intelligent analytics information. The IoT system could function as a distributed sensor network with heterogeneous sensors that provided environmental monitoring capabilities for temperature, humidity, barometric pressure, geo-localization, and video streams from dispatched sites. Experimental evaluation showed that the system performed reliably and accurately. With the addition of intelligent analytics on the sensor node and server, the system could be deployed at various sites for creating smart environments.

Introduction

The development of portable low-cost hardware/software systems has been considered increasingly important in experiential learning of technology and hands-on experiments in engineering research. The Internet of Things (IoT), implemented through the embedded systems approaches, is a useful domain for the purpose. IoT can be defined as a network of physical devices, vehicles, home appliances, and other items embedded with electronics, software, sensors, actuators, and connectivity, enabling these objects to connect and exchange data (Shafiq, Gu, Cheikhrouhou, Alhakami & Hamam, 2022.) The IoT systems can vary in scale from personal health-monitoring systems (Baker, Xiang & Atkinson, 2017), to air quality monitoring systems at a house (Tryner et al., 2021), to city-scale smart transportation systems (Rego, Canovas, Jimenez & Lloret, 2018). The complexity of involved technologies can also vary. Health monitoring systems (Baker et al., 2017), for example, can use near-field communication (NFC) or Bluetooth (IEEE 802.15) communication between sensors and monitoring devices. The sensors in such systems typically consume very little energy that can be supplied by a battery or through the use of energy harvesting technologies.

Air-quality monitoring systems (Tryner et al., 2021) typically need specialized sensors to monitor the concentration of atmospheric gases as pollutants (e.g., $PM_{2.5}$, PM_{10}, CO_2, CO, NO_2, O_3, etc.) and can be deployed in enclosed environments such as residential homes or hospitals. In contrast, city-scale smart transportation systems (Rego et al., 2018) involve wide-area networks (WAN or cellular) with heterogeneous communication protocols and a plethora of sensors. Sensors in such systems can use high-power electricity for wide sensing coverage and reliability. A survey by Ang, Seng, and Wachowicz (2022) categorized the research and development domains of IoT into two phases. The first phase focused on developing the building blocks and enabling technologies, while the second phase focused on the addition of values to the application domain, such as smart environments and intelligent analytics. Oftentimes, the two phases are intertwined as new technologies emerge.

For a specific research group, both of the phases are necessary. Even if the general trend of IoT systems development is to combine more and more technologies, it is also true that specific application domains need a specific set of technologies optimized to application-level cases. As an example, consider the research by Liu (2016) in which the author reported a case study of designing and implementing an intelligent environmental control system specifically designed for the maintenance of a large-scale (8 meter × 50 meter footprint) greenhouse. In that study, multiple heterogeneous sensors were hard-wired and installed across the greenhouse, and the sensor readings were wirelessly transmitted by a data collection node to a remote web server for user access. The sensors and the data collection node were specially packaged for waterproofing.

It is important to consider the best choices among available technologies and options in cases with specific applications (Shirehjini & Semsar, 2017). From the embedded systems perspective (Lacamera, 2018), it is imperative to balance system performance with the cost of overall system operations. In this current study—the focus of this paper—the author used an embedded-systems approach to IoT to show a case study of designing, implementing, and evaluating a portable low-cost intelligent system for smart environments. In the literature, the majority of IoT papers reviewed (Ang et al., 2022; Nauman, Qadri, Amjad, Zikria, Afzal & Kim, 2020) were large-scale research studies and encompassed various application domains. Many new approaches and methods have been proposed and evaluated, some of which became standard technologies, while others remain in experimental phases. The case study presented in this paper

is based on these previous studies (Park, 2022) and mainly adopts standard technologies available to public and uses off-the-shelf devices for the implementation of the proposed system.

System Structure Design

Figure 1 shows the use-case diagram of the proposed systems for a smart environment, represented as a UML (Universal Markup Language) Use Case diagram. Smart environments (Cook & Das, 2004) have been envisioned as "a small world where different kinds of smart devices are continuously working to make inhabitants' lives more comfortable." A smart environment is a physical world that is richly interwoven with sensors, actuators, displays, and computational elements, embedded seamlessly in the environment and connected through a continuous network. The overall system structure is a server-client system with a wireless network connection. The "User" in Figure 1 is the human user of the smart environment, who accesses the server computer to obtain awareness of the environmental status. The human user may want to become aware of the situation, which involves interpreting, visualizing, analyzing, and accumulating sensor data transmitted from remote sensor nodes. The server computer, which is local to the user, is depicted by the dashed rectangle on top, while the group of remote sensor nodes are depicted by the dotted rectangle at bottom. Inside the group of sensor nodes is a single embedded system enclosed by the solid small rectangle that corresponds to a sensor node. The sensor node runs various sensors, packages sensor data, backs up the data, and sends it to the remote server via wireless communication. The server computer can communicate with multiple sensor nodes simultaneously.

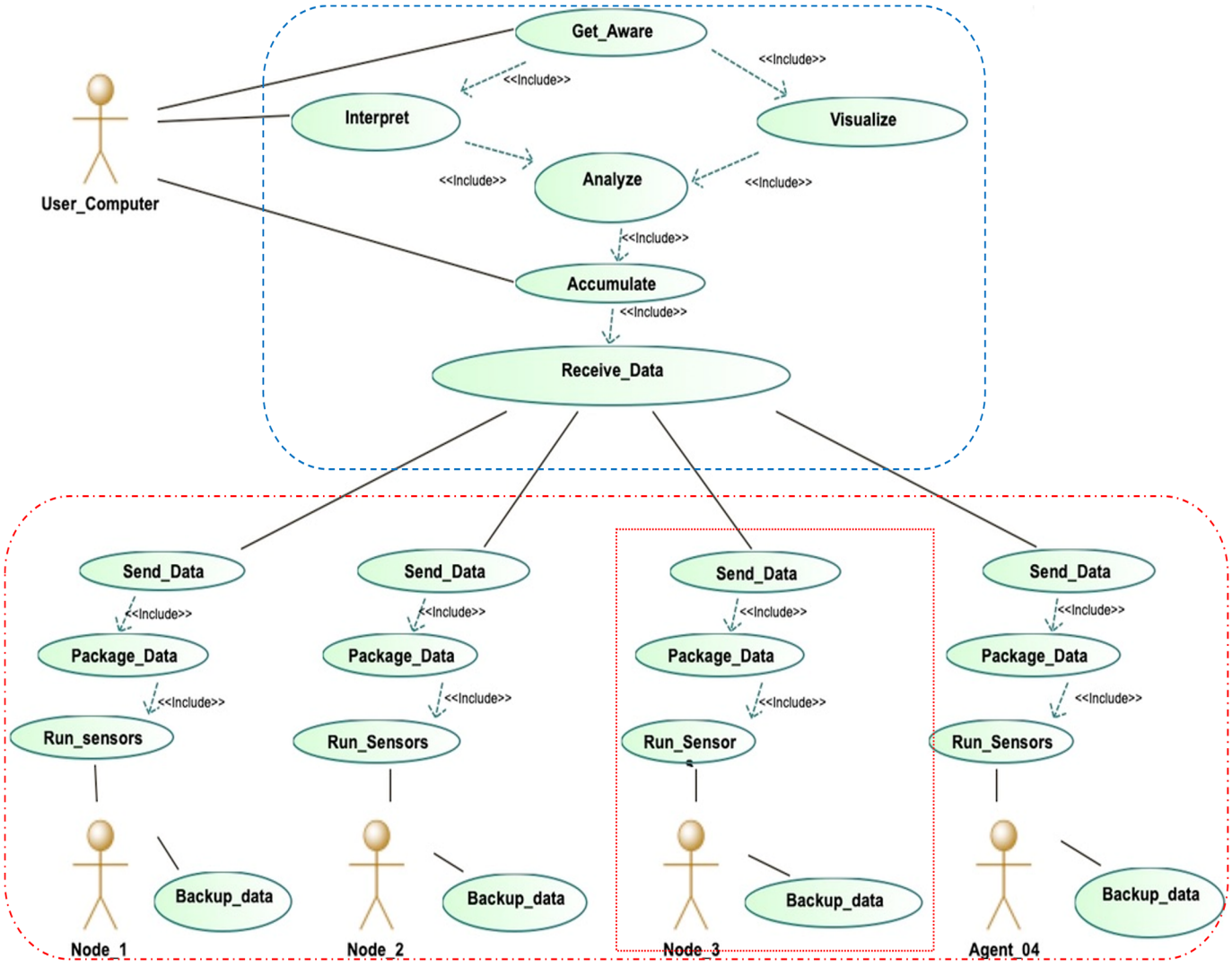


Figure 1. Use cases of the system.

Sensor nodes (see Figure 1: Node-1, Node-2, etc.) are independent, embedded systems with their own operating system (OS) and work as clients in the network. Communication between the server and the clients in the current study used a WiFi (IEEE 802.11) wireless network with a router, where the router provided a private network for the communication. It is possible to configure the router to support a public network, but it would need consideration of network security issues. Given that the sensor node clients are small-scale embedded computers, it is safer to configure the router for a private network.

Hardware Design

The implementation of the design in this current study used four Raspberry Pi embedded computers as the sensor node clients and an Apple *MacBook Pro* as the server. Any general-purpose computer (e.g., Windows PC, Apple Mac computer, Linux computer, etc.) will work as the server in the system. The sensor node used an ARM Quad-core Cortex-A72 chip as its CPU with a clock speed of 1.5 GHz. It also supported hardware interfaces via its general-purpose input and output (GPIO) pins that could be connected to various sensor devices. Each sensor device had its own communication interface, such as an inter-integrated circuit (I^2C) interface bus, a serial peripheral interface (SPI) bus, a universal asynchronous receiver/transmitter (UART) bus, or other such device. Figure 2 shows the overall system hardware diagram of the sensor node. The sensor node included BME280 combo-sensors—for temperature, humidity, and barometric pressure—a MAX31855 thermo-coupled temperature sensor, an MT3330 GPS sensor, and a camera with camera serial interface (CSI). Figure 3 shows the prototype implementation of the sensor node. Table 1 summarizes the specifications of the sensor chips used in this study.

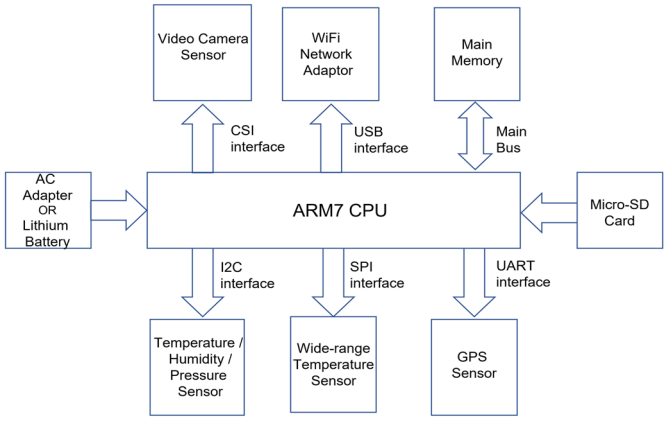


Figure 2. System diagram.

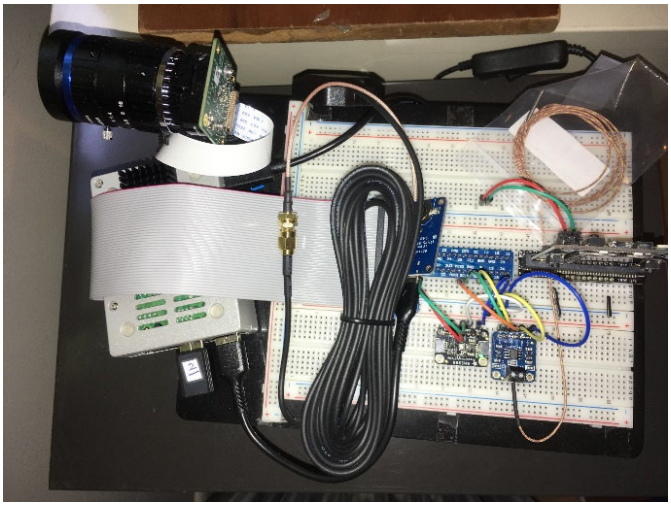


Figure 3. Prototype implementation of the sensor node.

Table 1. Sensor chip specifications.

Sensor chip	Make	Sensor Type	Interface	Voltage	Accuracy	Capacity
BME280	Bosch	Humidity / Temperature/ Pressure	I^2C	3.3 V	±3% / ±1°C / ±1 hPa.	0 % to 100 % / -40 °C to 80 °C / 300 to 1100 hPa
MAX31855	Analog Devices	Wide-range temperature	SPI	3.3 V	±2 °C to ± 6 °C	-200 °C to 1350 °C
MT3330	Mediatek	GPS	UART	3.3 V	3 meters	22 Satellite tracks
IMX477R	Sony	Camera	CSI	5 V	—	12.3 Mega pixels

The BME280 chip was used for heterogeneous sensing of humidity, temperature, and barometric pressure. The MAX31855 was used for wide-range temperatures, such as severe cold or flame-hot temperatures. In contrast, the BME280 mainly sensed atmospheric ambient temperature. The MT3330 chip was for GPS tracking and could search 66 satellites with an update rate of 1-10 Hz. The IMX477R chip was used as a CCD (charge-coupled device) sensor for the camera and supported 12.3 mega pixels of image resolution at a speed of 30 frames per second. The camera sensor was unique in the huge amount of video data it could capture as a series of image frames, whereas all the other sensors provides only simple scalar data (i.e., values.) The sampling rate of the scalar-data capture in this study was one second for all the sensors (except the camera) for evaluation purposes. For typical settings, it would not be necessary to sample temperature and humidity every second. The camera sensor supported video streaming as well as snapshot image capture. Image and video processing is highly CPU-intensive and causes CPU temperature to rise. The current implementation used a passive heat-dissipation method (i.e., aluminum diecasting case plus heat sink) for steady video streaming, but active cooling systems (e.g., cooling fan) may be needed for more CPU-intensive jobs such as image processing.

Software Design

Figure 4 shows the overall structure of software design for data processing and communication between a single sensor node and the server computer. Note that the server computer communicates with the four sensor nodes in the same fashion simultaneously. The dotted rectangle on the left represents the sensor-node processes, while the dotted, rounded rectangle on the right represents the server processes.

The CCD sensor in the camera generated a live video stream that required a fast refresh rate (i.e., typically 30 frames per second) and each frame occupied a huge amount of data, when compared to scalar sensor data; therefore, the video and scalar data were processed differently in the sensor node. Video data provides the user with rich information about the monitored scene for context awareness in smart environments (Park & Trivedi, 2008). Context awareness is one of the most salient features in intelligent environments (Roy, Roy & Das, 2006) and the realization of automatic context awareness requires advanced image processing algorithms, which are typically CPU-intensive. In the current study, the video data were mainly processed for the purpose of video streaming from the sensor node to the remote server computer via a wireless network for the user's context awareness in manual manner. The scalar data from the scalar sensors were time-series data, and they were processed mainly for data logging and data trend analysis. The sensor-node processes continuously sent scalar data to the remote server via the wireless communication, while backing up the data locally in the sensor node. The scalar data were buffered into a first-in first-out (FIFO) queue implemented on the main memory and backed up to a flash SD card sporadically. The FIFO queue also served as a data buffer for transmission.

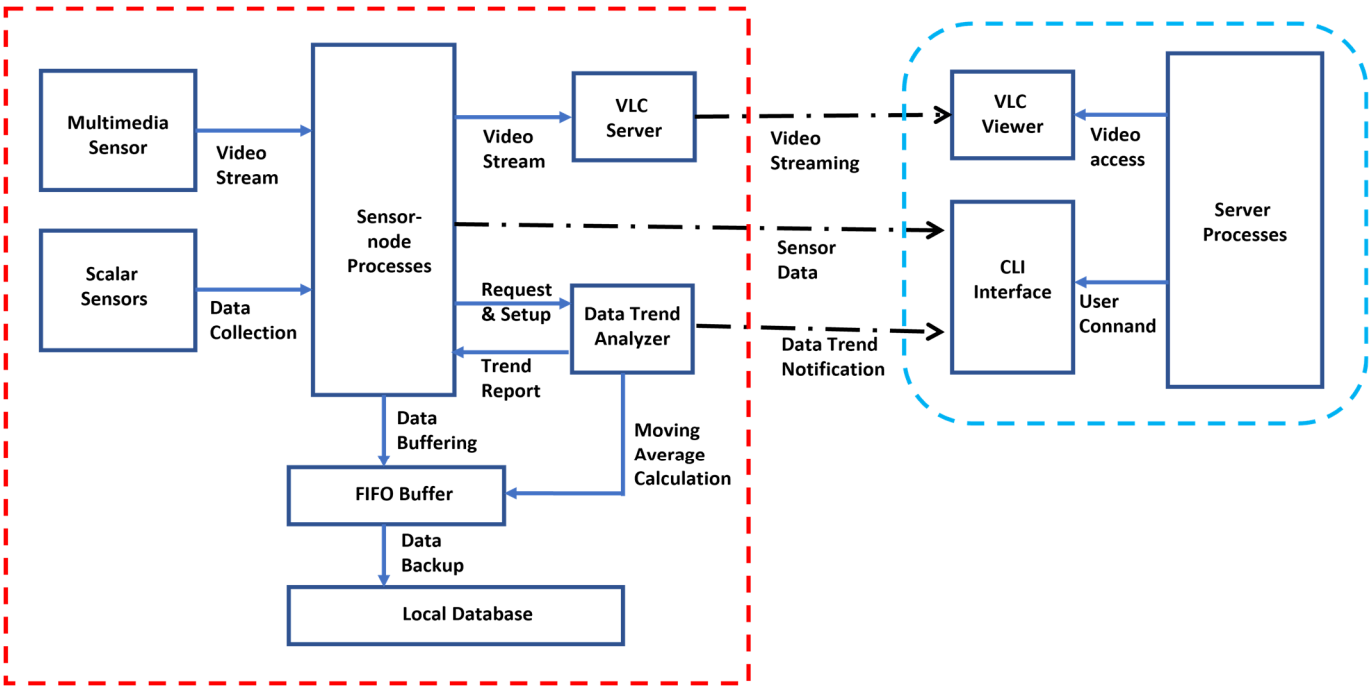


Figure 4. Software structure for data processing and communication.

The sensor-node processes also included the data trend analysis for detecting a rising or falling pattern of environmental variables (e.g., rising and falling of ambient temperature, humidity, etc.) Sensory data in general is noisy, due to hardware characteristics and electromagnetic influences, and it is not practical to compare two adjacent values to discern if the sensory value is rising or falling due to the noise; it is necessary to filter out the noise from the signal. Moving-average algorithms have been used to filter the spiky noises from data and to obtain reliable, smooth signal patterns (Fang, Xu & Zhu, 2014.) The moving-average value, m_t, at the t^{th} timestamp is defined in Equation 1 by summing the raw sample value r_t and N values around the raw value along the time as follows:

$$m_t = \frac{1}{N+1}\sum_{n=\frac{-N}{2}}^{\frac{N}{2}}\left(r_{t+n}\right) \tag{1}$$

The even number N determines how many nearby values are averaged. For example, if N=4, a total of five sample points including r_t are averaged. The value N needs to be empirically configured, depending on the characteristics of the sampled data pattern. The current study, the author extended the moving-average method to develop the data trend analyzer shown in Figure 5. The data trend analyzer accesses the FIFO buffer on the main memory to calculate the moving average values at every sampling moment and determines whether the data stream represents a rising or falling trend. The basic idea is similar with the time-series analysis in statistics (Hansun, 2013) or the technical analysis in stock market trading (Raudys, Lenčiauskas & Malčius, 2013.) It uses two moving averages of different scales, as follows. A long-term moving average with a large value of N is compared to a short-term moving average with a small value of N over time. The short-term moving average is more sensitive to the values adjacent to the current value, while the long-term moving average indicates overall value in a longer time duration. The data trend analyzer detects the moment of crossing between the two moving-average plots and reports if it is a rising trend or a falling trend. Figure 5 shows an example output of the proposed data trend analyzer that processes a random input signal as the test data.

The arbitrary time-series input data (Series 1 in Figure 5) was generated by Equation 2 with the constraint that the following value v_{t+1} at t+1 is randomly chosen within a boundary of the preceding value v_t at time t, as follows:

$$v_{t+1} = v_t + K \cdot rand\left(-1,1\right) \tag{2}$$

where, K is a constant (e.g., K = 5), and $rand$(-1,1) is a uniform-distribution, random real-number generator between -1 and 1.

The rectangular waveform (Series 2 in Figure 5) represents the trend analysis results in terms of binary values as follows: the output value of (+1) indicates a rising trend of the input data, while the output value of (-1) indicates a falling trend of the input data. The initial duration with the output value of zero (0) in Figure 5 indicates the initialization phase of the data trend analyzer. Note that the short-term and long-term average values are not shown in the figure for simplicity. Figure 5 further shows that the proposed data trend analyzer successfully interprets the time-series input data stream into semantic notion of rising versus falling trend segments. This is an example of sensor-based intelligence achieved at the edge-level computation and embedded in the sensor nodes in the context of IoT.

Each sensor node runs its own software for sensor reading and data backup, while all the sensor nodes should be communicating with the server in a corporative manner. In the current system, server-client communication was handled by a multi-socket connections approach, where the server opens/closes multiple sockets on demand requested from the clients and processes the communications via event-driven interrupts. Each sensor node had the following major jobs. It had to read data from multiple hard-wired sensors, convert the data format to byte streams, store the data temporarily on a local flash SD card, and wirelessly send the data to the server. Data communication was guaranteed by the TCP/IP protocol, which exchanges acknowledgement packets at every transaction.

The execution of programs on the sensor node can be autonomously run, according to a planned schedule via the crontab utility on Linux OS on the client. It also can be initi-

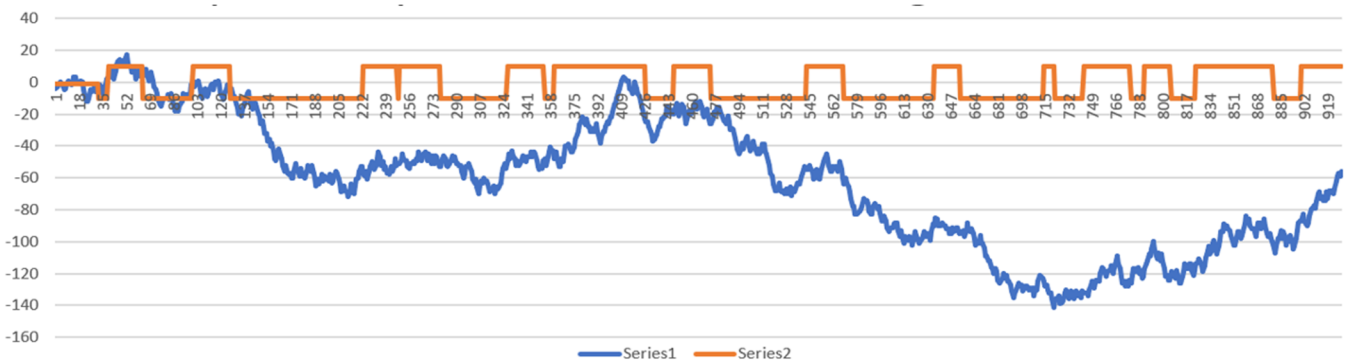


Figure 5. Data trend analyzer output from simulated input data.

ated and controlled by the user on the server side. A secure connection between the client and the server is maintained through the WiFi network via SSH or SFTP protocols. The overall software paradigm is well-suited with the system design in Figure 1. The current implementation of the software for the system was based on object-oriented design and used Python and C++ programming languages. For fast processing, C/C++ library modules are called from Python using C/Python a binding mechanism and running behind the Python code.

Experiment and Data Analysis

One of the motivations for this current study was to prototype a working system for a distributed mobile sensor network using off-the-shelf components and to evaluate the reliability of such a system. An experiment was conducted for basic data analysis that included a reliability test in the data-collection process, an accuracy estimation of sensory data, and an exploration of data profiling for descriptive statistics (i.e., mean, standard deviation, and curve fitting). The experiment was conducted over two consecutive days using four sensor nodes and one server computer. The sensor nodes were placed strategically on different sites within a two-story building. Node-1 was placed inside the basement near the entry door. Node-2 was placed inside a room on the ground floor near a window. Node-3 was placed inside a second-floor room with lots of windows. Node-4 was placed near the attic area. All windows and doors were closed during the experiment and no HVAC (heating, ventilation, and air conditioning) systems were on. Each node collected sensory data every second for two days, producing about 10 Mbytes of scalar data per node.

Figure 6 shows the temperature graphs from the four sensor nodes. The *y*-axis is temperature in degrees Celsius [°C] and the *x*-axis is sampling index in time. The temperature values of the [average ± standard deviation] of the four sensors were [23.2 ± 0.15]°C, [24.29 ± 0.5]°C, [28.27 ± 0.56]°C, and [28.56 ± 1.56]°C for sensors 1 through 4, respectively. The graphs show that the temperature rose from the lowest at the basement to the highest near the attic area. The standard deviations indicated that temperature variation becomes larger from the lower to higher levels in the building. The two dips in the temperature curves corresponded to nighttime, while the two peaks corresponded to daytime. The temperature variations between day and night became wider from the basement to the vicinity of attic area. This implied that the upper-level temperature near the attic was affected more by the ambient day/night temperature influence from outside the building.

Figure 7 shows the relative humidity graphs with the *y*-axis showing the percentage. The relative humidity values [average ± standard deviation] of the four sensors were [51.50 ± 2.19]%, [56.39 ± 1.57]%, [45.39 ± 1.25]%, and [44.39 ± 2.83]% for sensors 1 through 4, respectively. The relative humidity was inversely correlated with ambient temperature. The ambient temperature profiles and the relative humidity profiles also showed distinct patterns, depending on the location of the sensor node. The humidity was highest on the ground floor and lowest on the second floor near the attic.

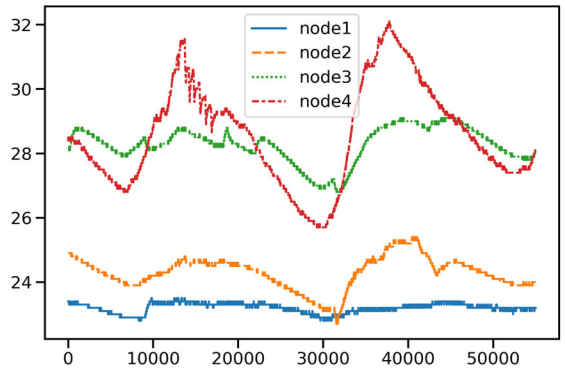


Figure 6. Temperature in degrees Celsius [°C].

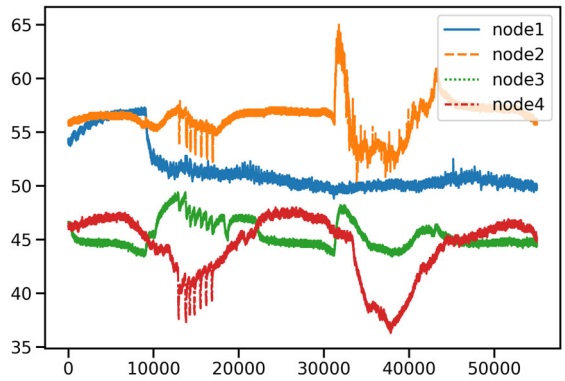


Figure 7. Relative humidity in percent [%].

Figure 8 shows the barometric pressure graphs in hPa. The barometric pressure values [average ± standard deviation] of the four sensors were [1003.45 ± 1.63]hPa, [1002.84 ± 1.59]hPa, [1003.30 ± 1.61]hPa, and [1002.87 ± 1.62]hPa for sensors 1 through 4, respectively. The four graphs show very similar patterns of variation over time. The barometric pressure patterns of the four sensor nodes closely resembled one another, which implied that the sensor reading was reliable. Figure 9 shows the GPS latitude coordinate values in degrees. The values [average ± standard deviation] of the four sensors were [31.7939 ± 0.00023]°, [31.7942 ± 0.00014]°, [31.7939 ± 0.00007]°, and [31.7940 ± 0.00006]° for sensors 1 through 4, respectively. Note that the actual GPS coordinate values of the latitude and the longitude were obfuscated by adding a random constant integer for privacy reasons. However, adding the constant number did not affect the standard deviation or the

temporal patterns. The high spikes as impulse noise in the graph of Node-1 showed signal fluctuation due to the insufficient number of tracked satellites. Despite the impulse noise, the [average ± standard deviation] of Node-1 was close to those of the other nodes. Such impulse noise can be easily removed by a moving-average filter.

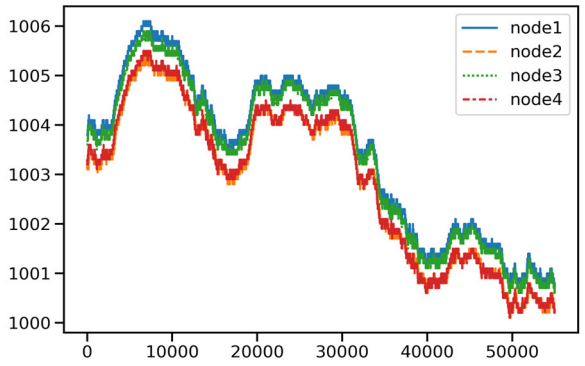


Figure 8. The barometric pressure graphs in hPa.

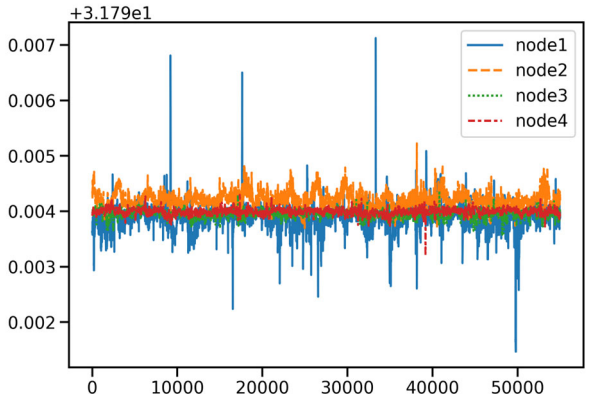


Figure 9. GPS latitude coordinate values in degrees.

Figure 10 shows the longitude coordinate graphs in degrees. The values [average ± standard deviation] of the four sensors were [-57.8859 ± 0.00024]°, [-57.8858 ± 0.00012]°, [-57.8859 ± 0.00009]°, and [-57.8857 ± 0.00005]° for sensors 1 through 4, respectively. The high spikes on the graph of Node-1 show signal fluctuation due to an insufficient number of tracked satellites. The fact that the high spike noises in Figures 9 and 10 can be easily filtered out by a Gaussian filter or a moving average filter indicates that GPS localization can be achieved reliably even at the basement level inside certain buildings (e.g., wooden buildings). Figure 11 shows the temporal graphs of the number of tracked satellites in integer count. The number of tracked satellites are integer numbers. For validation purposes, the values [average ± standard deviation] of the four sensors were [5.5 ± 1.43], [7.8 ± 1.64], [9.1 ± 1.65], and [10.4 ± 1.54] satellites for sensors 1 through 4, respectively. The number of tracked satellites directly influenced the reliability of the latitude/longitude calculations, as shown in Figures 9 and 10. The author also noticed that indoor environments can receive satellite signals, although they depends on the material and structure of the building.

The GPS coordinates of latitude and longitude were similar among the four sensor nodes. The random peak noises apparent on Node-1 at the basement level were caused by an insufficient number of tracked satellites, in and of itself due to the location of the MT3330 GPS sensor that was positioned at the basement level. Even at the basement level, the sensor received GPS signals from 2-10 tracked satellites. Figure 11 shows that as the locations of the sensor nodes changed—from the basement to the ground level, to the second floor, and to near the attic area—the number of tracked satellites increased from 5 to 7 to 9 and to 10. The fluctuation pattern in the number of tracked satellites was normal for GPS receivers in general situations. The MT3330 GPS sensor supports simultaneous searching and tracking of numerous satellites.

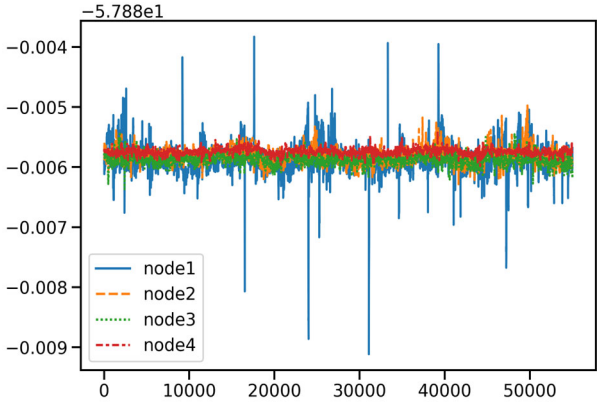


Figure 10. GPS longitude coordinate values in degrees.

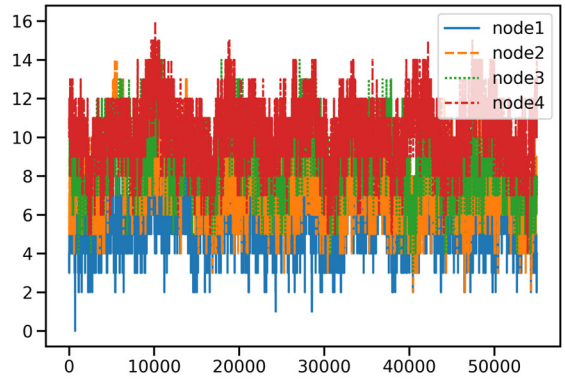


Figure 11. The number of tracked satellites [integer count].

Figures 12 and 13 show the moving-average plots overlaid on the raw data plot. “Raw” in the legend refers to the raw data plot; “MovAvg1” refers to the short-duration moving average plot; and, “MovAvg2” refers to the long-duration moving-average plot. Figure 12 shows that the long -duration moving average (MovAvg2) provided better a profile of the data pattern than the short-duration moving average (MovAvg1), while Figure 13 shows that the short-duration moving average performed better in data profiling.

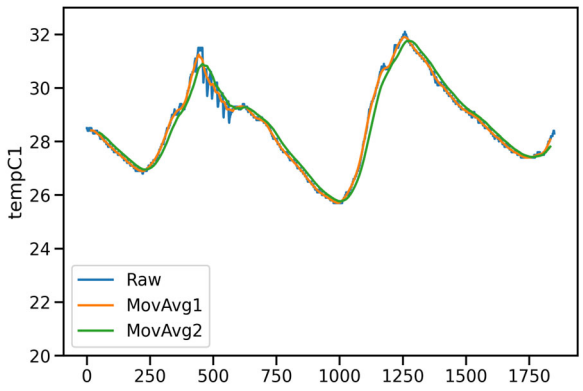


Figure 12. Temperature curves in raw versus moving averages.

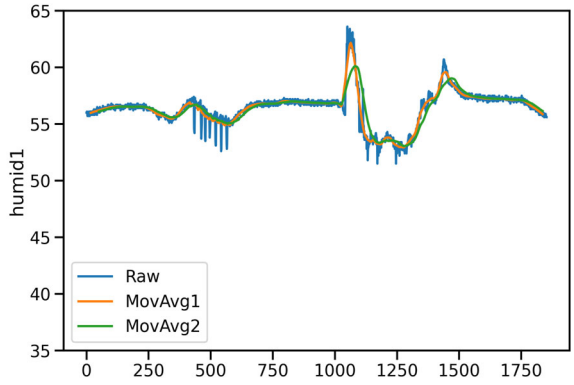


Figure 13. Humidity curves in raw versus moving averages.

Figures 12 and 13 also show comparisons of the short-term and long-term moving average plots. The short-term and long-term average patterns represent the trend of temperature change in the short term and the long term, respectively. It is likely that the next temperature in the near future will rise when the short-term moving average goes above the long-term moving average, and vise-versa. The curve-crossing moment was detected by the data trend analyzer on the sensor node, which notified the user on the server computer as a trend report. This trend report provided useful information to the user for decision making regarding the sensor-deployed environment.

Figure 14 shows snapshots of the streamed videos from the cameras on two sensor nodes (Node-2 and Node-3), respectively. The other sensor nodes (Node-1 and Node-4) had blocked camera views due to the deployed locations (i.e., basement and near attic, respectively.) Figure 14 shows the Node-2 camera view through the window glass and the Node-3 camera view of the building entrance area, respectively.

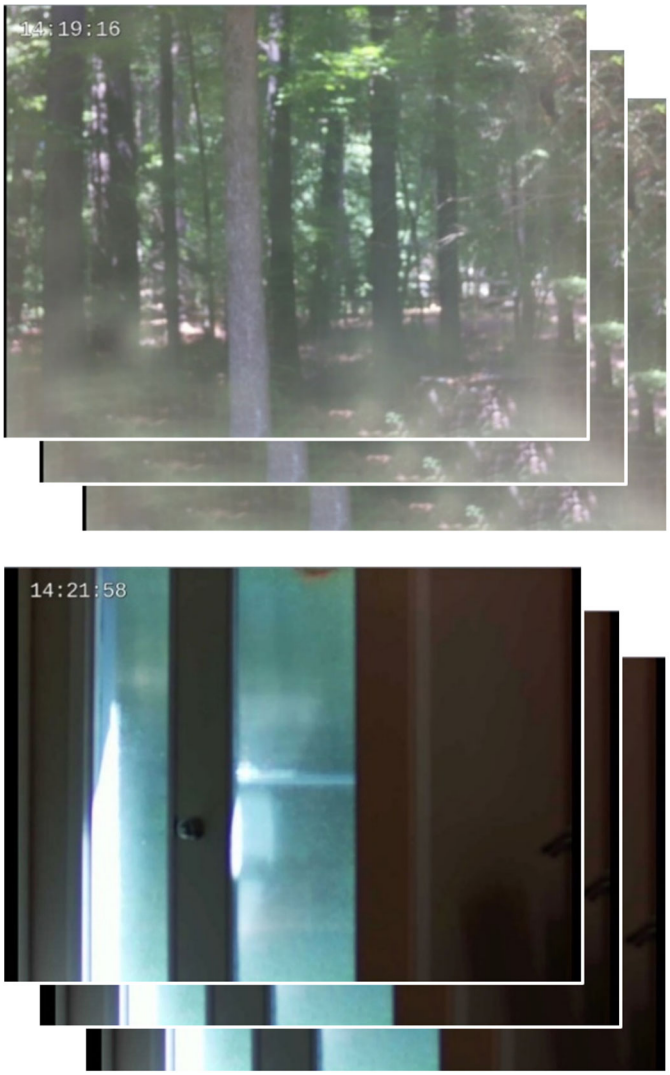


Figure 14. Snapshots of streamed videos from the sensor-node cameras.

The video streaming by the sensor nodes could be accessed from the remote server using the Virtual Network Computing (VNC) application. VNC supports simultaneous access to the multiple sensor nodes. Table 2 shows the CPU die temperature changes over time for specific processes. The temperature, while the CPU is idling, is the baseline. The CPU temperature stayed close to the baseline level while processing scalar sensors for one minute through one

hour. The CPU temperature rose significantly while streaming a live video, but did not go over 70°C even after an hour -long video streaming. The heat dissipation in the current implementation used a passive method (i.e., aluminum diecast case with heat sink). The aluminum case internally contacted the CPU and the memory chip via a heat-transferring adhesive tape. This passive heat dissipation method was enough to maintain the CPU temperature within a reasonable bound. For reference, the CPU temperature of the server computer (Apple *MacBook Pro*) was 64°C during the experiment. Active cooling systems (e.g., cooling fan) would significantly reduce the CPU temperature of the sensor node, but it would consume significant amounts of electric power for the deployed sensor node. Given the limited hardware resources of the embedded systems in general, it was not desirable to use active cooling methods.

Conclusions

The main objective of this study was to establish a baseline for determining whether future research on embedded intelligence using IoT is warranted in university engineering education. The contribution of this current study included empirical data that compared the performances of conventional scalar sensors to data-intensive video sensors. The author determined that video streaming is doable, but that image processing is not doable at the edge level, partly due to the excessive heat at the CPU die and the limited processing capabilities of the CPU of the sensor node. In addition, the author presented here the data-trend-analyzer algorithm and its implementation as an example of the embedded intelligence instantiated at edge-level computing. In the near future, observational studies will follow to investigate the learning effectiveness of the case study in engineering education.

The results of the study show that the development of a distributed sensor network is achievable using off-the-shelf components and low-cost, generic embedded system computers. Multiple sensor nodes, each of which was an independent embedded system with heterogeneous sensors, were simultaneously connected to a general-purpose notebook computer to provide environmental monitoring capabilities. For this study, such capabilities included temperature, humidity, barometric pressure, geo-localization, and video streaming, but more diverse sensing modalities (e.g., heterogeneous atmospheric gases such as CO, CO_2, O_3, NO_2, SO_2, etc.) could be added if bus interfaces are compatible. The current IoT system could be deployed to monitor various environments such as factories, schools, offices, hospitals, and homes.

Experimental evaluation showed that the developed IoT system was reliable for extended use and accurate in data sampling and transmission. The sensor nodes also realized locally embedded intelligence (e.g., data trend analysis) and autonomously triggered notifications to the server computer. The developed system could be used for portable deployment in diverse environments with extra intelligent analytics applications. Future studies will include weather-proof housing of the sensor node for various environments and the incorporation of more diverse sensor types, as noted above. Another important feature of future studies would be to investigate light-weight image processing onboard the sensor node for locally embedded intelligence.

Acknowledgements

This research was supported and funded by a CSCU-AAUP research grant from Central Connecticut State University.

References

Ang, L., Seng, K., & Wachowicz, M. (2022). Embedded intelligence and the data-driven future of application-specific Internet of Things for smart environments. *International Journal of Distributed Sensor Networks*, *18* (6). doi:10.1177/1550132922110237

Baker, S., Xiang, W., & Atkinson, I. M. (2017). Internet of Things for Smart Healthcare: Technologies, Challenges, and Opportunities. *IEEE Access*, 1-1. Doi:10.1109/ACCESS.2017.2775180.

Cook, D., & Das, S. (Ed.). (2004). *Smart Environments: Technology, Protocols and Applications*. Hoboken, NJ: J. Wiley & Sons.

Table 2. CPU die temperatures in degrees Celsius during various tasks.

Sensor Node	Idling	Processing_Scalar_Sensors			Streaming_Videos		
		1 minute	10 minutes	1 hour	1 minute	10 minutes	1 hour
Node-1	46	46	47	48	-	-	-
Node-2	48	48	47	49	60	67	69
Node-3	50	51	51	52	59	63	66
Node-4	54	54	55	55	-	-	-
Average	49.5	49.8	50.0	51.0	59.5	65.0	67.5

Fang, S. F., Xu, L. D., & Zhu, Y. Q. (2014). An integrated system for regional environmental monitoring and management based on internet of things. *IEEE Transactions on Industrial Informatics*, *10*(2).

Hansun, S. (2013). A new approach of moving average method in time series analysis. *Conference on New Media Studies (CoNMedia),* (pp. 1-4). Tangerang, Indonesia. doi: 10.1109/CoNMedia.2013.6708545

Lacamera, D. (2018). *Embedded Systems Architecture*. Birmingham, UK: Packt Publishing Ltd.

Liu, J. (2016). Design and implementation of an intelligent environmental-control system: perception, network, and application with fused data collected from multiple sensors in a greenhouse at Jiangsu, China. *International Journal of Distributed Sensor Networks*. Doi:10.1177/155014775056

Nauman, A., Qadri, Y. A., Amjad, M., Zikria, Y. B., Afzal, M. K., & Kim, S. W. (2020). Multimedia Internet of Things: a comprehensive survey. *IEEE Access*, *8*. Doi: 10.1109/ACCESS.2020.2964280

Park, S., & Trivedi, M. (2008). Understanding Human Interactions with Track and Body Synergies (TBS) Captured from Multiple Views. *Computer Vision and Image Understanding: Special Issue on Intelligent Visual Surveillance*, *111*(1), 2-20. Elsevier Inc.

Park, S. (2022). Sensor-based Intelligence for Smart Environments using Internet of Things (IoT): A Case Study. IAJC 2022 International Conference. Orlando, Florida.

Raudys, A., Lenčiauskas, V., & Malčius, E. (2013). Moving Averages for Financial Data Smoothing. In: T. Skersys, R. Butleris, & R. Butkiene, (Eds.) *Information and Software Technologies. ICIST 2013. Communications in Computer and Information Science*, *403*. Springer, Berlin, Heidelberg.

Rego, A., Canovas, A., Jimenez, J. M., & Lloret, J. (2018). An Intelligent System for Video Surveillance in IoT Environments. *IEEE Access*, *6*, 31580-31598.

Roy, N., Roy, A., & Das, S. (2006). Context-Aware Resource Management in Multi-Inhabitant Smart Homes: A Nash H-Learning based Approach. *The Fourth Annual IEEE International Conference on Pervasive Computing and Communications.*

Shafiq, M., Gu, Z., Cheikhrouhou, O., Alhakami, W., & Hamam, H. (2022). The Rise of "Internet of Things": Review and Open Research Issues Related to Detection and Prevention of IoT-Based Security Attacks. *Wireless Communications and Mobile Computing, 2022*.

Shirehjini, A. A. N., & Semsar, A. (2017). Human interaction with IoT-based smart environments. *Multimedia Tools and Applications, 76,* 13343-13365.

Tryner, J., Phillips, M., Quinn, C., Neymark, G., Wilson, A., Jathar, S. H. …Volckens, J. (2021). Design and testing of a low-cost sensor and sampling platform for indoor air quality. *Building and Environment,* 206, Elsevier Inc.

Biography

SANGHO PARK is an Associate Professor of Computer Electronics and Graphics Technology at Central Connecticut State University, where he has been teaching courses including circuit analysis, analog electronics, embedded systems, and capstone projects. His research interests include sensor-based intelligent systems, computer vision, image processing, and machine learning. Dr. Park may be reached at spark@ccsu.edu

A Case Study in a Machine Learning Framework Applied to Epilepsy Localization

Anthony D. Bowman, University of Alabama at Birmingham; Wesley Conwell, University of Alabama at Birmingham;
Leon Jololian, University of Alabama at Birmingham

Abstract

The rise of machine learning methodologies in recent years has seen great success in a variety of applications. However, this new paradigm is often utilized in limited ways through arbitrary selection of machine learning algorithms and static feature sets, particularly in the medical literature. The current authors previously published a framework that removes these artificial limiters, while laying the groundwork for parallel research and development tracks. To showcase the potential power from this expanded use of the machine learning paradigm, the authors applied this framework to the complex medical problem of epileptic seizure localization. Resting state EEG/MEG data were simultaneously collected from 22 patients prior to epilepsy surgery and retrospectively selected for analysis. Power spectral and coherence features were extracted from all sensor time series data. Sets and subsets of these features were used to train multiple machine learning algorithms for classifying epilepsy in different brain regions.

Models generated by a variety of algorithms—and trained by delta, theta, beta, and low gamma MEG and EEG features—were able to achieve an f-measure > 0.95, when distinguishing between left frontal epilepsy and bilateral extra-frontal epilepsy patients. Results showed that the artificial neural network also achieved this f-measure, but only when trained on the subset of features including beta and low gamma EEG features. Models generated by training the same algorithms and feature sets only achieved an f-measure of 0.818, when classifying right frontal epilepsy versus bilateral extra-frontal epilepsy. In this current study, using parallel applications of the machine learning paradigm, the authors were able to both improve on results seen in previous studies in classifying epilepsy and showcase the potential for meta-analysis across research tracks. This study provided additional insights into how research can be greatly expedited and expanded in scope through parallel exploration of topics that share overlapping feature or data sets.

Introduction

With the emergence of the machine learning paradigm in recent years, researchers in a wide variety of fields have sought out solutions generated by machine learning algorithms. While machine learning techniques continue to evolve, their adoption and usage is often carried out in an ad hoc manner, particularly by those whose expertise lies outside the field of computer science or related disciplines. One method to help bridge the gap between the non-expert and such an evolving methodology would be to introduce new development environments to aid in facilitating adoption of the machine learning paradigm. The current authors previously published a novel framework designed from the ground up with this specific purpose in mind, first as a framework for a course in machine learning and later as a framework for a development environment (Bowman & Jololian, 2021; Bowman, Prabhakar & Jololian, 2022). The authors encourage the reader to review these papers for a detailed description of the framework. In this current study, the authors applied this framework and methodology to a problem in the medical domain involving the localization of seizure onset zones in epilepsy patients.

Epilepsy is a neurological disorder characterized by abnormal neural activity, often manifesting as seizures (England, Liverman, Schultz & Strawbridge, 2012). A patient is diagnosed with focal epilepsy when their seizures are found to originate from a single region within the brain, deemed the seizure onset zone. To study, diagnose, and treat this disorder, researchers often analyze data gathered from electroencephalography (EEG) and magnetoencephalography (MEG), both non-invasive procedures (Tovar-Spinoza, Ochi, Rutka, Go & Otsubo, 2008). While EEG records electrical activity, MEG records measurements of the magnetic field created by the underlying electric current through neural tissue (Cuffin & Cohen, 1979). If medications fail to produce the desired control over seizure frequency and severity, epilepsy surgery is considered in order to remove the region of cortex containing the seizure onset zone.

To find an automated solution for localizing this zone in patients with focal epilepsy, previous studies explored the application of machine learning to neuro-imaging data with limited success. These included attempting to analyze magnetoencephalogram (MEG) data with a support vector machine to identify high-frequency oscillations thought to correspond to epilepsy activity (Guo et al., 2018). Others used the support vector machine to analyze graph theoretical features extracted from fMRI data to lateralize and localize seizures to the temporal lobes (Wu et al., 2018). Author in yet other studies attempted to train an artificial neural network and support vector machine on frequency domain features extracted from MEG data to discriminate between healthy and controls and epilepsy and between frontal focal epilepsy and generalized epilepsy (Aoe et al., 2019; Soriano, Niso, Clements, Ortín, Carrasco, Gudín & Pereda, 2017).

These studies are representative of a large portion of the medical literature reviewed, due to their use of either the support vector machine or artificial neural network. While these previous publications often did not discuss how they arrived at the algorithm used, the authors suspect that such discussions were not included, because those authors found that such algorithms had already been employed with some success in previous literature. This represents an almost arbitrary decision in their methodology. The current authors' proposed framework sought to correct this aspect of the literature by providing a more comprehensive approach toward predictive model construction and selection. Early results from the framework presented in this paper include some models with accuracy consistent with these previous studies, but also some models with higher accuracy, sometimes using the same classifier but with different features.

Applying this framework to the case study, the authors instantiated this three-layer architecture with each layer broadening the scope of their search through the solution space. Beginning from the research layer, the authors were able to generate multiple research questions that could be explored from the same data set. Data selection and preprocessing was then determined according to the needs of those research questions with one control layer generated to address each question. Within each control layer, the feature and classifier sets were then defined and feature extraction implemented. Of note was the ability for this architecture to allow for the researcher to explore the effect completely different features have in relation to each research question. Model construction and testing was then performed in every instantiation of the composite layer with a confusion matrix generated.

Methods

The authors collaborated with a neurologist who retrospectively selected patients with medically intractable epilepsy from a database (n = 22). All patients had previously undergone surgical resection and were seizure free for at least six months, thus confirming their epileptic locus was within the resected region. As part of their pre-surgical evaluation, all patients had an MEG study performed using the system described below. The study was approved by the Institutional Review Board at the University of Alabama at Birmingham. All MEG recordings were performed using a whole-head, 148-channel system housed within a magnetically shielded room (4D Neuroimaging, San Diego, CA). All patients were in a reclined position for the duration of the recordings. Multiple recordings were collected from each patient, each lasting 10 minutes and collected at a sampling rate of 508.63 Hz. Each data file was then preprocessed with in-house MATLAB scripts using Statistical Parametric Mapping software (SPM12b, http://www.fil.ion.ucl.ac.uk/spm). All electroencephalogram (EEG) recordings were gathered using the international 10-20 system of electrode placement and gathered concurrently with each MEG recording. All EEG data were gathered at a sampling rate of 2 kHz and down-sampled to 600 Hz using a low-pass filter. Only data from 25 EEG leads common across all patients were included for further analysis and feature extraction.

Feature extraction from both MEG and EEG recordings was performed using a combination of Brainstorm functions and in-house MATLAB scripting (MATLAB, 2018; Tadel, Baillet, Mosher, Pantazis & Leahy, 2011). Mean power spectrum density (Welch method) was computed using the Brainstorm function with the frequency bands slightly adjusted to the following: Delta band from 1–3 Hz, theta band from 3–8 Hz, alpha band from 8–12 Hz, beta band from 15–29 Hz, low gamma band from 30–59 Hz, and high gamma band from 60–90 Hz. Spectral coherence features between sensors were computed from both MEG and EEG time series data using the mean square coherence function in MATLAB 2018a.

Machine Learning Training and Testing

All classifier training and testing was performed using the Waikato Environment for Knowledge Analysis (WEKA) open-source machine learning software (Witten, Frank, Trigg, Hall, Holmes & Cunningham, 1999). Default parameters were used for each classifier unless otherwise noted. Unless otherwise noted, all classifiers were trained and tested using 10-fold leave-one-out cross validation with WEKA reporting the confusion matrix, precision, recall, and weighted f-measure for each trained model. Classification was performed using different combinations of features provided to each classifier to find the combination of classifier and feature set that produced the highest f-measure.

Results

In Table 1, the authors present some results with seizure localization to the left frontal lobe. Using the weighted f-measure reported by WEKA as the metric for this matrix, the authors were able to see a wide range in results, depending on algorithm and feature set. For this and future tables of results, the authors denote a calculated f-measure as "not a number" ("NaN") when the calculation involves division by zero. The maximum f-measure for each table is bolded. Maximum f-measure of 0.951 was achieved from models generated by logistic regression, stochastic gradient descent (SGD), simple logistic, support vector machine (SMV), and logistic model tree (LMT) algorithms trained by a feature set containing MEG power in the delta, theta, beta, and low gamma ranges. In the case of logistic regression and SVM, the achieved maximum f-measure decreased, when features extracted from EEG were included in addition to the aforementioned MEG features. The multilayer perception, otherwise known as the artificial neural network (ANN), was the only algorithm to match the maximum f-measure with fewer, albeit different features.

Table 1. Left frontal versus bilateral extra-frontal.

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	NaN	0.777	NaN	0.777	0.872	0.872
NaiveBayes	0.653	0.753	0.777	0.777	0.291	0.347
NaiveBayesMultinomial	NaN	NaN	NaN	NaN	NaN	NaN
Logistic Regression	0.727	0.727	0.818	0.753	0.951	0.836
SGD	0.777	0.777	0.909	0.777	0.951	0.951
Multilayer Perceptron	0.777	0.777	0.951	0.777	0.909	0.909
SimpleLogistic	0.753	0.777	0.889	0.777	0.951	0.951
SMO (SVM)	NaN	0.777	NaN	NaN	0.951	0.909
DecisionStump	0.753	0.753	0.777	0.753	0.872	0.872
J48	NaN	0.753	0.889	0.786	NaN	0.786
LMT (log tree)	0.753	0.777	0.889	0.777	0.951	0.951
Random Forest	NaN	0.777	NaN	NaN	0.777	0.852
Random Tree	0.700	0.727	0.909	0.727	0.852	0.818

Table 2. Right frontal versus bilateral extra-frontal.

Algorithm	Delta EEG	Theta EEG	Beta and low gamma EEG	Delta, theta, beta, low gamma EEG	Delta, theta, beta, low gamma MEG	EEG&MEG
BayesNet	NaN	NaN	NaN	NaN	0.805	0.805
NaiveBayes	0.084	0.364	0.611	0.261	0.287	0.287
NaiveBayesMultinomial	NaN	NaN	NaN	NaN	NaN	NaN
Logistic Regression	0.570	0.613	0.398	0.600	0.818	0.636
SGD	0.590	0.590	0.745	0.636	0.778	0.600
Multilayer Perceptron	0.513	0.484	0.422	0.600	0.727	0.579
SimpleLogistic	NaN	0.566	NaN	NaN	0.566	NaN
SMO (SVM)	NaN	0.590	NaN	0.642	0.611	0.566
DecisionStump	0.513	0.455	0.590	0.590	0.805	0.805
J48	0.540	0.540	0.513	0.485	NaN	0.485
LMT (log tree)	NaN	0.513	NaN	NaN	0.566	NaN
Random Forest	0.540	0.579	0.566	0.540	0.642	0.745
Random Tree	0.438	0.600	0.485	0.441	0.611	0.745

Table 2 shows the results from the same algorithms trained with the same feature sets, relabeled to data from patients with right frontal epilepsy. In this classification task, the highest-weighted f-measure of 0.818 was only achieved by the logistic regression algorithm trained using the combined MEG feature set with power in all stated frequency bands. Table 3 shows the results of training the same set of algorithms with coherence features extracted from MEG in the theta, alpha, beta, and low gamma frequency ranges (the far-right column being the combined feature set with all frequency bands included: "OOM" denotes an out-of-memory error when training the model). Here, the models were trained to discriminate between right and left temporal lobe epilepsy, also known as lateralization. In this task, the maximum weighted f-measure of 0.818 was achieved from the SimpleLogistic and LMT algorithms trained using only coherence features in the beta frequency band as well as the random tree algorithm using coherence features in the beta and low gamma bands.

Representing a single row in Table 3 in a different manner, Figure 1 graphically shows how the performance of the J48 decision-tree algorithm varied, depending on which feature set was used. The performance of this algorithm did not improve with beta and low gamma coherence features combined instead of the subset including only coherence in the beta frequency range. Also noteworthy is the lower performance when trained on the full feature set, including coherence features from all frequency bands.

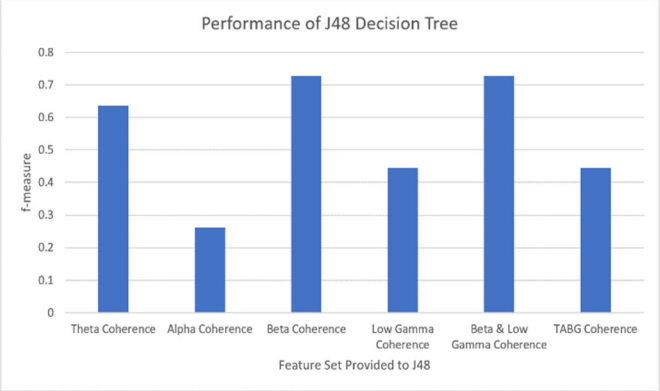


Figure 1. Performance of the J48 Decision Tree with various feature sets.

Discussion of Technical Results

From an engineering standpoint, this methodology presents a more thorough exploration through both the solution space associated with each research question as well as the capability to efficiently investigate multiple research questions. Whereas the traditional methodology would have arrived at one or two of these models, the framework employed here allows for a much broader view of the landscape from which researchers can select the maximum-weighted f-measure. Through this broader view, the authors see some models with tested f-measure approximately equal to those in previous studies, potentially confirming their results. Results also show that some models achieved higher f-measure, sometimes using the same algorithm as in previous studies but trained with a different feature set. In this case, the maximum was observed by multiple models, which presents the researcher with the opportunity to consider other metrics as well as generate additional research questions that may explore why different feature sets resulted in high accuracy.

Figure 1 shows the variability in performance of a single classifier, reinforcing the need for researchers to explore the solution space through multiple machine learning algorithms and feature sets. Further experimentation may explore the effect that changing various hyper-parameters has on improving the performance of some algorithms, such as altering the number of hidden layers of the artificial neural network. Stepping back to take in a more abstract view of these results leads to additional questions relating to model performance relative to feature subset, in some cases decreasing substantially. This comparative meta-analysis is further expanded on by the opportunity to efficiently investigate parallel research questions (right and left frontal lobe epilepsy) and comparing the results from the same algorithms and feature sets. The authors were also afforded the opportunity to efficiently explore another research track in tandem by reusing the same data set, resulting in the investigation into the possible relationship between coherence features and focal epilepsy.

Table 3. Left versus right temporal lobe epilepsy, MEG feature sets.

Algorithm	Theta Coherence	Alpha Coherence	Beta Coherence	Low Gamma Coherence	Beta & Low Gamma Coherence	TABG Coherence
BayesNet	0.364	0.364	0.723	0.723	0.696	0.617
NaiveBayes	0.696	0.617	0.538	0.545	0.364	0.538
NaiveBayesMultinomial	0.091	0.636	0.364	0.455	0.538	0.445
Logistic Regression	0.445	0.331	0.723	0.445	0.636	0.455
SGD	0.331	0.261	0.538	0.636	0.723	0.723
Multilayer Perceptron	0.331	0.261	0.617	0.617	0.636	OOM
SimpleLogistic	0.140	0.195	0.818	0.445	0.727	0.331
SMO (SVM)	0.261	0.261	0.617	0.617	0.636	0.455
DecisionStump	0.331	0.140	0.727	0.445	0.727	0.331
J48	0.636	0.261	0.727	0.445	0.727	0.445
LMT (log tree)	0.140	0.195	0.818	0.445	0.727	0.331
Random Forest	0.331	0.364	0.808	0.455	0.696	0.636
Random Tree	0.445	0.455	0.723	0.455	0.818	0.538

Discussion of Domain-Specific Medical Results

From a medical standpoint, these results show an interesting disparity between the metrics achieved to discriminate between focal epilepsy in different brain regions. This is most easily seen when comparing the results in Tables 1 and 2: Investigating mirrored hypotheses, left frontal versus right frontal epilepsy, reveals classification of left frontal epilepsy using the defined power spectral features to be the "easier" of the two tasks. This may suggest that the underlying neurophysiological characteristics associated with right frontal lobe epilepsy consist of a more complex pattern than that which can be identified by these algorithms and feature sets for left frontal lobe epilepsy. This may imply that focal epilepsy originating from the right frontal lobe may be better characterized by changes in features beyond power spectra. The nature of these features may provide further insight into the processing and functional structure of the right frontal lobe and how electrical pathologies such as epilepsy disrupt normal function.

These results also show greater opportunity for success in such classification tasks with power spectral features over coherence features, although this observation may change with focal epilepsy in other cortical regions. Further research is needed with a larger data set to confirm this trend. Spectral coherence was chosen as a feature because of the connectivity exhibited between the temporal lobes and other regions of the brain (Haneef, Lenartowicz, Yeh, Levin, Engel Jr. & Stern, 2014; Spencer, 2002). Previous studies have also explored other methods of quantifying neural or cortical connectivity, such as transfer entropy, directed transfer functions, and graph theoretic metrics (Basu, Kudela, Korzeniewska, Franaszczuk & Anderson, 2015; Dai, Zhang, Dickens & He, 2012; Ursino, Ricci & Magosso, 2020; Wu et al., 2018). Mirroring the limited success seen in those studies, the results of this current study suggest that further research is needed to explore these different metrics for connectivity in combination or refined, perhaps with more narrowly defined coherence metrics than used here. The results of this research clearly show the need to explore the use of feature subsets to train multiple machine learning models for testing, as a subset may lead to higher performance than the complete feature set.

Conclusions

In this study, the authors employed a previously published framework for the development of machine learning solutions conducted in the context of a medical case study. Using this methodology to embrace the machine learning paradigm more fully, the authors were able to efficiently explore both the problem and solution spaces within the case study's domain. Implementation of the composite layer allowed for empirical identification of algorithm-feature set pairings with higher performance metrics than seen in previous studies. Implementation of the control layer in combination with the composite layer greatly improved the scope of this research, expanding the potential of the project to explore multiple, related domain-specific questions in rapid succession through the re-use of data and feature sets.

Acknowledgements

This work was supported by an institutional grant from the University of Alabama at Birmingham. The authors would also like to acknowledge the contributions of Dr. Ismail Mohamed and Jeff Killen in data collection, management, and selection.

References

Aoe, J., Fukuma, R., Yanagisawa, T., Harada, T., Tanaka, M., Kobayashi, M. ...Kishima, H. (2019). Automatic diagnosis of neurological diseases using MEG signals with a deep neural network. *Scientific Reports*, *9*(1), 1-9.

Basu, I., Kudela, P., Korzeniewska, A., Franaszczuk, P. J., & Anderson, W. S. (2015). A study of the dynamics of seizure propagation across micro domains in the vicinity of the seizure onset zone. *Journal of Neural Engineering*, *12*(4), 046016.

Bowman, A. D., & Jololian, L. (2021). A conceptual framework for an introductory machine learning course. *Journal of Computing Sciences in Colleges*, *37* (1), 78-83.

Bowman, A. D., Prabhakar, S. P., & Jololian, L. (2022, March). A Framework for an Automated Development Environment to Support the Data-driven Machine Learning Paradigm. In *SoutheastCon 2022* (pp. 329-331). IEEE.

Cuffin, B. N., & Cohen, D. (1979). Comparison of the magnetoencephalogram and electroencephalogram. *Electroencephalography and Clinical Neurophysiology*, *47*(2), 132-146.

Dai, Y., Zhang, W., Dickens, D. L., & He, B. (2012). Source connectivity analysis from MEG and its application to epilepsy source localization. *Brain Topography*, *25*(2), 157-166.

England, M. J., Liverman, C. T., Schultz, A. M., & Strawbridge, L. M. (2012). Epilepsy across the spectrum: Promoting health and understanding: A summary of the Institute of Medicine report. *Epilepsy & Behavior*, *25* (2), 266-276.

Guo, J., Yang, K., Liu, H., Yin, C., Xiang, J., Li, H. ...Gao, Y. (2018). A stacked sparse autoencoder-based detector for automatic identification of neuromagnetic high frequency oscillations in epilepsy. *IEEE Transactions on Medical Imaging*, *37*(11), 2474-2482.

Haneef, Z., Lenartowicz, A., Yeh, H. J., Levin, H. S., Engel Jr. J., & Stern, J. M. (2014). Functional connectivity of hippocampal networks in temporal lobe epilepsy. *Epilepsia*, *55*(1), 137-145.

MATLAB. (2018). *9.4.0.813654 (R2018a)*. Natick, Massachusetts: The MathWorks Inc.
Soriano, M. C., Niso, G., Clements, J., Ortín, S., Carrasco, S., Gudín, M., & Pereda, E. (2017). Automated detection of epileptic biomarkers in resting-state interictal MEG data. *Frontiers in Neuroinformatics*, *11*, 43.
Spencer, S. S. (2002). Neural networks in human epilepsy: evidence of and implications for treatment. *Epilepsia*, *43*(3), 219-227.
Tadel, F., Baillet, S., Mosher, J. C., Pantazis, D., & Leahy, R. M. (2011). Brainstorm: a user-friendly application for MEG/EEG analysis. *Computational Intelligence and Neuroscience*, *2011*.
Tovar-Spinoza, Z. S., Ochi, A., Rutka, J. T., Go, C., & Otsubo, H. (2008). The role of magnetoencephalography in epilepsy surgery. *Neurosurgical Focus*, *25*(3), E16.
Witten, I. H., Frank, E., Trigg, L. E., Hall, M. A., Holmes, G., & Cunningham, S. J. (1999). Weka: Practical machine learning tools and techniques with Java implementations. (Working paper 99/11). Hamilton, New Zealand: University of Waikato, Department of Computer Science.
Ursino, M., Ricci, G., & Magosso, E. (2020). Transfer entropy as a measure of brain connectivity: A critical analysis with the help of neural mass models. *Frontiers in Computational Neuroscience*, *14*, 45.
Wu, T., Chen, D., Chen, Q., Zhang, R., Zhang, W., Li, Y. ... Zhang, J. (2018). Automatic lateralization of temporal lobe epilepsy based on MEG network features using support vector machines. *Complexity*, *2018*.

Biographies

ANTHONY D. BOWMAN is a doctoral candidate pursuing a PhD in Computer Engineering at the University of Alabama at Birmingham. He earned both his MS in Biomedical Engineering (2017) and BS in Computer and Information Sciences (2012) from the University of Alabama at Birmingham. Mr. Bowman's research interests include machine learning, data science, and bioinformatics. Mr. Bowman may be reached at anbowman@uab.edu

WESLEY CONWELL is a PhD candidate of interdisciplinary engineering at the University of Alabama at Birmingham. He earned his BS in Electrical Engineering from the University of Alabama, MBA from the University of Alabama at Birmingham, and MS in Electrical Engineering from the University of Alabama at Birmingham. Mr. Conwell's research interests include smart cities, data analytics, big data, Internet of Things, and machine learning. Mr. Conwell may be reached at wlconwel@uab.edu

LEON JOLOLIAN is a professor of electrical and computer engineering at the University of Alabama at Birmingham. He earned his BS in Electrical Engineering from Manhattan College, his MS in Electrical Engineering from Georgia Institute of Technology, his MS in Computer Science from Polytechnic University, and his PhD in Computer Science from the New Jersey Institute of Technology. Dr. Jololian's research interests include machine learning, mobile computing, and the Internet of Things. Dr. Jololian may be reached at leon@uab.edu

INSTRUCTIONS FOR AUTHORS: MANUSCRIPT FORMATTING REQUIREMENTS

The INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH AND INNOVATION is an online/print publication designed for Engineering, Engineering Technology, and Industrial Technology professionals. All submissions to this journal, submission of manuscripts, peer-reviews of submitted documents, requested editing changes, notification of acceptance or rejection, and final publication of accepted manuscripts will be handled electronically. The only exception is the submission of separate high-quality image files that are too large to send electronically.

All manuscript submissions must be prepared in Microsoft Word (.doc or .docx) and contain all figures, images and/or pictures embedded where you want them and appropriately captioned. Also included here is a summary of the formatting instructions. You should, however, review the sample Word document on our website (http://ijeri.org/formatting-guidelines) for details on how to correctly format your manuscript. The editorial staff reserves the right to edit and reformat any submitted document in order to meet publication standards of the journal.

The references included in the References section of your manuscript must follow APA-formatting guidelines. In order to help you, the sample Word document also includes numerous examples of how to format a variety of scenarios. Keep in mind that an incorrectly formatted manuscript will be returned to you, a delay that may cause it (if accepted) to be moved to a subsequent issue of the journal.

1. **Word Document Page Setup:** Two columns with ¼" spacing between columns; top of page = ¾"; bottom of page = 1" (from the top of the footer to bottom of page); left margin = ¾"; right margin = ¾".

2. **Paper Title:** Centered at the top of the first page with a 22-point Times New Roman (Bold), small-caps font.

3. **Page Breaks:** Do not use page breaks.

4. **Figures, Tables, and Equations:** All figures, tables, and equations must be placed immediately after the first paragraph in which they are introduced. And, each must be introduced. For example: "Figure 1 shows the operation of supercapacitors." "The speed of light can be determined using Equation 4:"

5. **More on Tables and Figures:** Center table captions above each table; center figure captions below each figure. Use 9-point Times New Roman (TNR) font. Italicize the words for table and figure, as well as their respective numbers; the remaining information in the caption is not italicized and followed by a period—e.g., "*Table 1*. Number of research universities in the state." or "*Figure 5*. Cross-sectional aerial map of the forested area."

6. **Figures with Multiple Images:** If any given figure includes multiple images, do NOT group them; they must be placed individually and have individual minor captions using, "(a)" "(b)" etc. Again, use 9-point TNR.

7. **Equations:** Each equation must be numbered, placed in numerical order within the document, and introduced—as noted in item #4.

8. **Tables, Graphs, and Flowcharts:** All tables, graphs, and flowcharts must be created directly in Word; tables must be enclosed on all sides. The use of color and/or highlighting is acceptable and encouraged, if it provides clarity for the reader.

9. **Textboxes:** Do not use text boxes anywhere in the document. For example, table/figure captions must be regular text and not attached in any way to their tables or images.

10. **Body Fonts:** Use 10-point TNR for body text throughout (1/8" paragraph indention); indent all new paragraphs as per the images shown below; do not use tabs anywhere in the document; 9-point TNR for author names/affiliations under the paper title; 16-point TNR for major section titles; 14-point TNR for minor section titles.

11. **Personal Pronouns:** Do not use personal pronouns (e.g., "we" "our" etc.).

12. **Section Numbering:** Do not use section numbering of any kind.

13. **Headers and Footers:** Do not use either.

14. **References in the Abstract:** Do NOT include any references in the Abstract.

15. **In-Text Referencing:** For the first occurrence of a given reference, list all authors—last names only—up to seven (7); if more than seven, use "et al." after the seventh author. For a second citation of the same reference—assuming that it has three or more authors—add "et al." after the third author. Again, see the *sample Word document* and the *formatting guide for references* for specifics.

16. **More on In-Text References:** If you include a reference on any table, figure, or equation that was not created or originally published by one or more authors on your manuscript, you may not republish it without the expressed, written consent of the publishing author(s). The same holds true for name-brand products.

17. **End-of-Document References Section:** List all references in alphabetical order using the last name of the first author—last name first, followed by a comma and the author's initials. Do not use retrieval dates for websites.

18. **Author Biographies:** Include biographies and current email addresses for each author at the end of the document.

19. **Page Limit:** Manuscripts should not be more than 15 pages (single-spaced, 2-column format, 10-point TNR font).

20. **Page Numbering:** Do not use page numbers.

21. **Publication Charges:** Manuscripts accepted for publication are subject to mandatory publication charges.

22. **Copyright Agreement:** A copyright transfer agreement form must be signed by all authors on a given manuscript and submitted by the corresponding author before that manuscript will be published. Two versions of the form will be sent with your manuscript's acceptance email.

    Only one form is required. Do not submit both forms!

    The form named "paper" must be hand-signed by each author. The other form, "electronic," does not require hand signatures and may be filled out by the corresponding author, as long as he/she receives written permission from all authors to have him/her sign on their behalf.

23. **Submissions:** All manuscripts and required files and forms must be submitted electronically to Dr. Philip D. Weinsier, manuscript editor, at philipw@bgsu.edu.

24. **Published Deadlines:** Manuscripts may be submitted at any time during the year, irrespective of published deadlines, and the editor will automatically have your manuscript reviewed for the next-available issue of the journal. Published deadlines are intended as "target" dates for submitting new manuscripts as well as revised documents. Assuming that all other submission conditions have been met, and that there is space available in the associated issue, your manuscript will be published in that issue if the submission process—including payment of publication fees—has been completed by the posted deadline for that issue.

    Missing a deadline generally only means that your manuscript may be held for a subsequent issue of the journal. However, conditions exist under which a given manuscript may be rejected. Always check with the editor to be sure. Also, if you do not complete the submission process (including all required revisions) within 12 months of the original submission of your manuscript, your manuscript may be rejected or it may have to begin the entire review process anew.

THE LEADING JOURNAL OF ENGINEERING, APPLIED SCIENCE, AND TECHNOLOGY

Print ISSN: 2157-8052
Online ISSN: 1930-6628

www.ijme.us

www.iajc.org

INTERNATIONAL JOURNAL OF MODERN ENGINEERING

ABOUT IJME:

- IJME was established in 2000 and is the first and official flagship journal of the International Association of Journal and Conferences (IAJC).
- IJME is a high-quality, independent journal steered by a distinguished board of directors and supported by an international review board representing many well-known universities, colleges and corporations in the U.S. and abroad.
- IJME has an impact factor of **3.00**, placing it among the top 100 engineering journals worldwide, and is the #1 visited engineering journal website (according to the National Science Digital Library).

OTHER IAJC JOURNALS:

- The International Journal of Engineering Research and Innovation (IJERI)
  For more information visit www.ijeri.org
- The Technology Interface International Journal (TIIJ).
  For more information visit www.tiij.org

IJME SUBMISSIONS:

- Manuscripts should be sent electronically to the manuscript editor, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For submission guidelines visit
  www.ijme.us/submissions

TO JOIN THE REVIEW BOARD:

- Contact the chair of the International Review Board, Dr. Philip Weinsier, at philipw@bgsu.edu.

  For more information visit
  www.ijme.us/ijme_editorial.htm

INDEXING ORGANIZATIONS:

- IJME is indexed by numerous agencies. For a complete listing, please visit us at www.ijme.us.

Contact us:

Mark Rajai, Ph.D.

Editor-in-Chief
California State University-Northridge
College of Engineering and Computer Science
Room: JD 4510
Northridge, CA 91330
Office: (818) 677-5003
Email: mrajai@csun.edu

www.tiij.org

www.ijeri.org

IJERI

The International Journal of Engineering Research & Innovation (IJERI) is the second official journal of the International Association of Journals and Conferences (IAJC). IJERI is a highly-selective, peer-reviewed print journal which publishes top-level work from all areas of engineering research, innovation and entrepreneurship.

IJERI Contact Information

General questions or inquiry about sponsorship of the journal should be directed to:

Mark Rajai, Ph.D.

Founding and Editor-In-Chief

Office: (818) 677-5003

Email: editor@ijeri.org

Department of Manufacturing Systems Engineering & Management

California State University-Northridge

18111 Nordhoff St.

Room: JD3317

Northridge, CA 91330