Assessing Lower Limb Strength using Internet-of-Things Enabled Chair and Processing of Time-Series Data in Google GPU Tensorflow CoLab

Hudson Kaleb Dy
Student Researcher, Walnut High School

Chelsea Yeh
Student Researcher, Walnut High School

Abstract—This project describes the application of the technologies of Machine Learning (ML) and Internet-of-Things (IoT) to assess the lower limb strength of individuals undergoing rehabilitation or therapy. Specifically, it seeks to measure and assess the progress of individuals by sensors attached to chairs and processing the data through Google GPU Tensorflow CoLab. Pressure sensors are attached to various locations on a chair, including but not limited to the seating area, backrest, hand rests, and legs. Sensor data from the individual performing both sit-to-stand transition and stand-to-sit transition provides a time series dataset regarding the pressure distribution and vibratory motion on the chair. The dataset and timing information is fed into a machine learning model to estimate the relative strength and weakness during various phases of the movement. They are compared with previously collected data to determine the efficacy and progress of the rehabilitation or therapy. This project created IoT-connected pressure and motion sensors. These were then used to capture the individual’s weight distribution and motion on the chair and send the data to a central server. This data is then made available to be processed in Google GPU Tensorflow CoLab. A future multi-disciplinary research project will collect data for training and modeling the Tensorflow machine learning algorithm. (Abstract)

Keywords— physical therapy, machine learning, internet-of-things, rehabilitation (keywords)

I. INTRODUCTION

During physical therapy, rehabilitation, or training, the measurement of lower body strength provides indications of the patient progress. They also provide lower limb strength and endurance assessments of older adults. These tests include the 30-second chair stand test (30CST) and 5 times sit-to-stand test (5xSTS) [1,2]. Internet of Things (IoT) allows for real-time remote collection and interpretation of data [3] that can be stored on a central database and processed with cloud-based multivariate time-series classification machine learning processes [4].
In this research, chairs equipped with pressure sensors will be able to capture the dynamic weight distribution and motion of the seat as the patients perform the test, and provide time-series data regarding the area of performance improvement and strengthening. The collected data was sent via custom web services where the data is stored and aggregated. This data is then fed into Google’s GPU Tensorflow CoLab where a machine learning model can be used. The goal of this research is to create a new tool to classify lower limb strength and endurance quantitatively and accurately by measuring the relative dynamic forces measured by the sensors on the chairs. In this way, the progress of the treatment or impairment can be assessed without using the current standard of Sit-to-Stand Tests.

II. SIT-TO-STAND ASSESSMENT

The current standard of assessing functional lower extremity strength in older adults is the Sit-to-Stand Test (STS).[16] The two most common variants of the Sit-to-Stand Test are the Five Times Sit-to-Stand Test (5TSTS) and the 30 Second Sit-to-Stand Test (30CST). The 5TSTS measures how many seconds a patient needs to complete 5 Sit-to-Stand motions while the 30CST counts how many times they can stand up from a sitting position repeatedly over the course of 30 seconds. Both assessments must be conducted by a professional physical therapist and/or physician.

In a previous work, one pressure sensor was attached to a medical chair and IoT was used to alert nurses and staff when a physically weakened individual stands from the chair and risks falls and injuries [5]. This allowed the investigators to become sufficiently familiar with working with microcontrollers and pressure sensors to embark on this new project which involves storing and processing time-series data for machine learning and artificial intelligence classifications of lower limb strength/mobility.

The goal is to create an automated system that will improve upon the existing standard of using a “30 Second Sit-to-Stand Test” that must be performed by a licensed medical practitioner.

The intent is to find measurable patterns and nuances in the time-series weight distribution data in the four corners of a chair based on a person’s lower limb strength. These nuances would be too minute for humans to recognize, but it may be sufficient for a deep learning convolutional network machine learning algorithm to properly classify.

The goal is to create a system that can accurately record time-series data and be able to graph these data in Google Tensorflow CoLab cloud environments. Doing so would allow future multi-disciplinary research with licensed physical therapists or physicians to collect and identify training data of patients.

The hope is that the IoT chair we developed would lead to future multi-disciplinary research that eventually creates a new modality in quantitatively assessing patient lower-limb strength and mobility using an automated machine learning process instead of the current standard of using Sit-to-Stand Tests that requires medical
professional intervention. The dense time series can be a valuable tool in the assessment and treatment of lower-body injuries, disease, and aging, and may have applicability in sports or fitness training.

III. COMPONENTS

The IoT lower body assessment system consists of the following elements:

1) A chair suitable for performing sit-to-stand testing that can be equipped with sensors
2) Network of pressure sensors and motion sensors suitable for mounting to a chair;
3) Internet-capable processing devices and software co-located and connected to the chair sensors, which collect, interpret, and transmit the sensor data
4) A Linux Apache web server that hosts custom PHP web services. These web services are used by the IoT chair to store data to a MySQL Server. It also performs post-processing and feeds the data to Google’s Tensorflow/Keras GPU Colab instances.
5) A MySQL server that stores the chair sensor data and processes them into the required format.
6) A machine learning model that is trained using the multiple time-series sensor data and provides estimates of lower limb strength and endurance or specific muscle group strength from test data.

A. Chair

The chair requirement dictates that pressure sensors will need to be mounted on different locations of the chair that will bear the full weight of the individual. To measure the dynamic weight distribution during the sit-to-stand transition, the accurate transmission of the forces to the sensors is required. To achieve this goal, the seat of the chair must be a flat hard surface. Any cushion on the seat will cause lag and spreading of the forces on the sensors. Sensors were mounted to be optimally distant from each other to provide the most independent measurement of the forces.

The sensor can be mounted in the following locations: 1) under the 4 legs of the chair, 2) between the legs of the chair and the frame, and 3) between the frame and the seat of the chair. Mounting the sensors under the legs of the chair can cause the chair to be unstable and unsafe for the user. For most chairs, including folding chairs, there is no way to disassemble the chair in such a way as to mount the sensor between the legs and the frame due to construction. A non-folding wooden chair was sourced from a DIY furniture store [6] that had a single hard seating surface that is screwed on the frame. This met all of the measurement requirements and provided a good mounting location for the sensor between the seat and frame.

B. Sensors

For this project, strain gauges were chosen to measure the dynamic forces during the sit-to-stand transition. Strain gauges are electrical resistive elements mounted to rigid metals pieces that deform (strain) under forces (stress). As the metal piece strains under the weight, the resistance changes and is measured electrically.
To measure human weight, strain gauges each with a maximum range of 50kg (110lb) were chosen. These gauges are commercially available and commonly used for bathroom scales [7].

The electrical resistance of the strain gauges is measured via a quarter bridge circuit that converts the change in resistance to a change in voltage. The voltage is measured digitally by an analog-to-digital converter (ADC). The ADC used was the HX711 integrated circuit produced by Avia Semiconductor [8]. Designed for human weight measurements, it has a relatively fine analog resolution of 24 bits, but a relatively slow speed of 10Hz or 80Hz.

The modules come pre-configured for 10Hz measurement. Initial tests were performed at 10 samples per second and the smaller quantity of data is adequate for measurements thus far. However, the modules have also been tested at 80Hz and work equally well at the higher sample rate.

The HX711 ADC module is designed so 1, 2, or 4 strain gauges can be connected to it, and the reading will be the sum of the readings of the strain gauges. For bathroom scales, four (4) strain gauges are connected to one (1) ADC to give a single weight measurement. For this application, multiple weight measurements on different locations of the chair are needed. The initial attempt was to connect one (1) strain gauge for each ADC. However, the measurement drifted over time due to changes in environmental conditions such as temperature. So, the investigation was modified to use 2 strain gauge per ADC configuration. Here the strain gauges are connected differentially, so the drift from one device is canceled out by the drift from another device. This effectively removed the drift problem.

Four strain gauges were mounted to the four corners of the chair. They are connected to 4 ADC modules. Four (4) additional strain gages are connected differentially to the ADC modules, do not bear any weight and are unused for the measurement, and are used solely for drift cancellation.

C. Microcontroller

The microcontroller and software running on it are responsible for the user interface, data collection, and data transmission. The sensors are connected via a compatible interface to the microcontroller.

The microcontroller is equipped with networking capabilities, preferably wireless, which will be used to transmit the collected data to a server. It is powered via AC power or batteries.

The microcontroller was chosen to be the ESP32 because it is powerful, low cost, and, most importantly, has integrated WiFi capabilities [9].

The HX711 ADC’s were connected to the ESP32 with a simple serial interface. WiFi networks are ubiquitous and secured with WPA2 [10]. The IoT devices are connected over a 2.4GHz WiFi band because it is more secure and has a longer range compared to the 5.0GHz band.

Custom software was developed on the Arduino platform to collect the sensor data. Custom drivers were required to read each of the four sensors asynchronously. While each ADC operates nominally at 10Hz, they actually sample at slightly different rates. Additional sensors can be added
with minimal effort. The first software plots the sensor readings for chair setup and testing. The second software provides the user interface to calibrate the sensors, collect the data, and for the operator to label or classify the data for supervised machine learning training.

After each trial, initially selected for a 5-second data collection period at 10Hz, the data is packaged and sent via an HTTP POST request to the cloud-based data collection server.

D. Data Collection Server

The data collection server is comprised of two components:

1) A Linux Apache Webserver [11] that hosts the PHP web services. These web services are used by the IoT chair to store data in the database, with separate web services for Training and Test modes. Webservices are also used to pull data out of the SQL server by the machine learning processes, again with separate web services used for Training and Testing modes.

2) A MySQL [12] database that is used to store time-series data for both Training and Testing.

E. Machine Learning Model

The machine learning model used for the initial test was the Keras [13] module of TensorFlow [14] using cloud-based Google Colab with GPU [15]. This was selected due to its low cost (free). Pytorch is an alternative technology that can be used to perform time-series multivariate classification models.

A key advantage of this system is the distributed architecture. The data collected is accessible using separate standard web services and Tensorflow can easily be replaced with PyTorch or Amazon ML going forward.

IV. METHODS AND RESULTS

The prototype chair that was built is fully functional and has only been used to collect and store unit testing data by the two investigators performing sit-to-stand motions. We have successfully recorded the unit test data and retrieved them systemically in Google’s GPU CoLab where Tensorflow could be used to further perform machine learning training and classification in future multi-disciplinary research where more sufficient and relevant data are collected.

A complete demonstration of the system can be viewed at the following link: http://aichair.tk.

Figure 1 shows a photograph of the prototype chair.

Figure 2 shows a photograph of the prototype chair with the seat removed showing the strain gauge sensors, HX711 ADC’s, and ESP32 microcontroller.

Figure 3 shows the system architecture and data flow diagram.

Figure 4 shows a close-up view of the prototype board and weight sensor.

Figure 5, 6, 7 shows graphs of the actual data collected for sample user 1 performing a stand-sit-stand motion sequence plotted from Google Colab.

Figure 8, 9, 10 shows graphs of the actual data collected for sample user 2 performing a stand-sit-stand motion sequence plotted from Google Colab.
Figure 11, 12, 13 shows graphs of the actual data collected for sample user 3 performing a stand-sit-stand motion sequence plotted from Google Colab. Figure 14, 15 shows graphs of the actual data collected for sample user 4 performing a stand-sit-stand motion sequence plotted from Google Colab.

V. DISCUSSION

All software has been developed with the completed user interface to facilitate data collection of the Training and Testing data. Data can be collected at both 10Hz and 80Hz sampling rate, but initial unit testing does not show any appreciable differences in the plotted graphs by the higher sampling rates. Therefore, all unit tests have been performed at 10Hz to reduce the amount of data collected and speed up plotting performance.

The actual unit testing performed and shown on Figure 5 to 15 shows actual data collected at 10Hz from 4 different users. Multivariate time-series classification machine learning has not been performed as we do not have any data from actual patients with known lower-limb strength classified by medical professionals. The purpose of the different data collection shown as Figure 5 to Figure 15 are to validate that there is an appreciable difference in the displayed plots of the actual stand-to-sit and sit-to-stand data sequences from one user to another. This validation will be performed visually in this project but will be performed via machine learning in future projects (once there is enough data collected and properly classified by a medical professional to facilitate properly training the machine learning classification.)

Each user performed three stand-to-sit and sit-to-stand activity sequences on the chair and their results are shown at Figure 5-15. Looking at the graphs, we can clearly see that all the graphs for each person is clearly differentiated from one user to another (User 1: Figure 5, 6, 7; User 2: Figure 8, 9, 10; User 3: Figure 11, 12, 13; User 4: Figure 14, 15, 16).

![Fig 1. Picture of the Prototype.](image-url)
Fig 2. Prototype with seats removed showing sensors, ADC, and ESP32 module.

Fig 3. System Architecture and Data Flow Diagram

Fig 4. Prototype board (top) and weight sensor (bottom).
Fig 5. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 1, Iteration 1

Fig 6. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 1, Iteration 2

Fig 7. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 1, Iteration 3

Fig 8. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 2, Iteration 1
Fig 9. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 2, Iteration 2

Fig 10. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 2, Iteration 3

Fig 11. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 3, Iteration 1

Fig 12. Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 3, Iteration 2
Fig 13 Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 3, Iteration 3

Fig 14 Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 4, Iteration 1

Fig 16 Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 4, Iteration 2

Fig 16 Sample Graph Of A 30 Second 10Hz Time-Series Dataset Plotting a Sit-To-Stand Motion with User 4, Iteration 3
VI. CONCLUSION AND FUTURE WORK

Our goal is to continue the project by partnering with a medical professional to find a sizeable number of individuals of varying known strength/mobility as classified by medical professionals. These data can then be collected and made available to properly train and optimize the TensorFlow model.

Future plans include reaching out to kinesiology departments of local universities and finding faculty members that are interested in working with us. Optimization of the multivariate time-series classification models and creation of a standardized quantifiable measurement of lower limb mobility and strength will follow successful training of the machine learning model.

Further work could also include the estimation of isolated muscles or muscle groups from additional isolated muscle strength data collected during testing and training the machine learning model. This way the specific muscle groups for additional intervention can be identified.

References

[1] Zhang, Q., Li, Y. X., Li, X. L., Yin, Y., Li, R. L., Qiao, X., ... & Hou, G. (2018). A comparative study of the five-repetition sit-to-stand test and the 30-second sit-to-stand test to assess exercise tolerance in COPD patients. International journal of chronic obstructive pulmonary disease, 13, 2833.

[2] Pozaic, T., Lindemann, U., Grebe, A. K., & Stork, W. (2016). Sit-to-stand transition reveals acute fall risk in activities of daily living. IEEE journal of translational engineering in health and medicine, 4, 1-11.

[3] Atzori, L., Iera, A., & Morabito, G. (2010). The internet of things: A survey. Computer networks, 54(15), 2787-2805.

[4] Zheng, Y., Liu, Q., Chen, E., Ge, Y., & Zhao, J. L. (2014, June). Time series classification using multi-channels deep convolutional neural networks. In International conference on web-age information management (pp. 298-310). Springer, Cham.

[5] Yeh, C., Lee, A., Dy, H., & Li, K. (2022) Internet-of-Things Management of Medical Chairs and Wheelchairs. Submitted to International Conference on Internet of Things, Big Data and Security.

[6] Ikea Jokkmokk chair. https://www.ikea.com/us/en/p/jokkmokk-table-and-4-chairs-black-brown-80394456/

[7] Straining Gauge Load Cell. https://www.digikey.com/en/products/detail/s parkfun-electronics/SEN-10245/5843757

[8] Avia Semiconductor HX711 ADC. http://en.aviaic.com/detail/730856.html

[9] ESP32. https://www.espressif.com/en/products/hardware/esp32/overview

[10] IEEE Standard for Information technology—Telecommunications and information exchange between systems Local and
metropolitan area networks—Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," in IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012), vol., no., pp.1-3534, 14 Dec. 2016

[11] Fielding, R. T., & Kaiser, G. (1997). The Apache HTTP server project. IEEE Internet Computing, 1(4), 88-90.

[12] Widenius, M., Axmark, D., & Arno, K. (2002). MySQL reference manual: documentation from the source. " O'Reilly Media, Inc."

[13] Gulli, A., & Pal, S. (2017). Deep learning with Keras. Packt Publishing Ltd.

[14] Abadi, M., Barham, P., Chen, J., Chen, Z., Davis, A., Dean, J., ... & Zheng, X. (2016). {TensorFlow}: A System for {Large-Scale} Machine Learning. In 12th USENIX symposium on operating systems design and implementation (OSDI 16) (pp. 265-283).

[15] Carneiro, T., Da Nóbrega, R. V. M., Nepomuceno, T., Bian, G. B., De Albuquerque, V. H. C., & Reboucas Filho, P. P. (2018). Performance analysis of google colaboratory as a tool for accelerating deep learning applications. IEEE Access, 6, 61677-61685.

[16] Yee, X.S., Ng, Y.S., Allen, J.C. et al. Performance on sit-to-stand tests in relation to measures of functional fitness and sarcopenia diagnosis in community-dwelling older adults. Eur Rev Aging Phys Act 18, 1