An Affordable 3D-printed Open-Loop Prosthetic Hand Prototype with Neural Network Learning EMG-Based Manipulation for Amputees

Sinchhean Phea,[a] Mark Ikechukwu Ogbodo,[b] and Abderazek Ben Abdallah[c]

The University of Aizu, School of Computer Science and Engineering, Adaptive Systems Laboratory, Japan.

[a]*Corresponding author: m5251138@u-aizu.ac.jp*
[b]*Electronic mail: d8211104@u-aizu.ac.jp*
[c]*Electronic mail: benab@u-aizu.ac.jp*

Abstract. Despite the advancement of prosthetic hands, many of the conventional products are difficult to control and have limited capabilities. Even though these limitations are being pushed by many state-of-the-art commercial prosthetic hand products, they are often expensive due to the high cost of production. Therefore, in the Adaptive Neuroprosthesis Arm (NeuroSys) project, we aim to develop a low-cost prosthetic hand with high functionalities that let users perform various gestures and accurate grasp. This paper mainly focuses on the sEMG signal recognition and control for a prototype 3D printed prosthetic hand model. In this work, we have considered the prosthetic hand to operate from a non-intrusive sensor, surface Electromyographic signal (sEMG). The signal used to control the prosthetic hand is received from a low-cost, 8-channel sEMG sensor, Myo armband. The sensor is placed around a person's upper forearm under the elbow, and the signal is sent wirelessly to a computer. After the signal is received, a neural network is used to recognize and classify the intention of the signals. The network model is designed for specific individuals to increase the controllability of the prosthetic hand. Also, to mimic the real-world usage, evaluation on two different sessions is conducted. With the use of Recurrent Neural Networks (RNNs) family, sEMG data recognition can reach around 85% of accuracy. While Gated Recurrent Units (GRUs) and Long Short Term Memory (LSTM) have similar results, simple RNN unit produces very low accuracy. Also, the more session the sample data is taken, the more robust the recognition system can be. Using the Myo armband sensor, sEMG signal data during a steady state with force or no force can affect the accuracy performance of the decoding hand gestures. In terms of real-world usage, however the constant force must be applied, otherwise, the system fails to classify the gestures. Also, the variation of sensor placement can affect the deep learning model. Although, there is a trade-off between accuracy and delay, optimal window size can be explored. Using the mentioned method, a prototype of an affordable 3D printed prosthetic hand controlled using sEMG is realized, although it is still far from real-world usage.

Keywords: Artificial Neural Network, Prosthetic Hand, Control System, Electromyography

INTRODUCTION

According to the statistics, there are more than 540,000 and 82,000 upper-limb amputees in the United States and Japan, respectively [1]. Worldwide, it is estimated that at least 1.5 million people are living with an absence of upper limbs [2]. Hands are one of the most vital body parts of humans. Unfortunately, some lose their hands or arms due to accidents or diseases. The progress of prosthetic arms and human-like robot hand research and development have been improved incessantly. There are many approaches to realize a prosthetic hand. In most recent years, using electromyographic (EMG) or surface electromyographic (sEMG) signals has been one of the most popular approaches in the field. These signals are used as a source to control the prosthetic hand [3, 4]. With the help of machine learning, the number of controls and movements has been increasing compared to only close and open movements in the past. To improve the development cost and maintainability of prostheses, recently the 3D printing techniques are used to design the hardware components of prosthetic hands [5].

While many machine learning algorithms and deep learning help improve the functionality of the prosthetic hand [6], there are still limitations and challenges when it comes to real-world usage [7]. For instance, the accuracy of the decoded movements is under-satisfying. Also, slow reaction of the prosthetic hand occurs even in some commercial products. The high cost of these sophisticated prosthetic hand products is also a reason why they are not being used widely [8, 9].

To tackle the aforementioned limitations, Adaptive Neuroprosthesis Arm (NeuroSys) project is deployed to build a highly functional and affordable neuroprosthetic hand. In this work, we are going to focus on hand gesture recognition based on the sEMG signal. The goal is to investigate a reliable approach for building a real-time sEMG-based control system for neuroprosthetic hand.

NeuroSys Overview

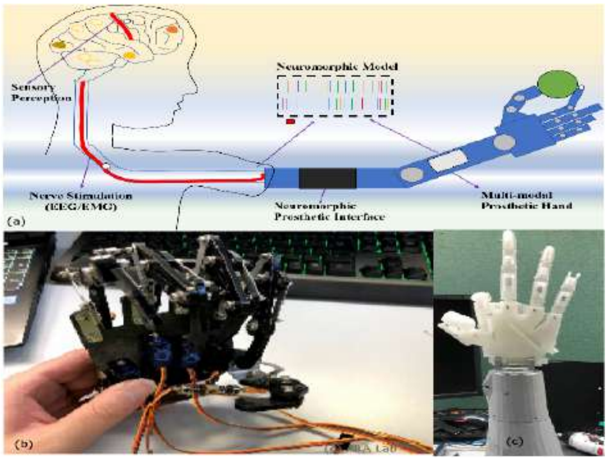

FIGURE 1. NeuroSys System Overview. (a) The EEG/EMG signals are used as input to the prosthetic hand's controller, (b) Mechanical prosthetic hand for testing, (c) 3D printed prosthetic hand.

Figure 1 shows the high-level overview of the NeuroSys project. We investigate advanced prosthetic hand with sensorimotor integration and tactile sensing based on biological signal discrimination with neuromorphic circuits to restore hand function movement for people with amputations or neurological disorders. Using our neuromorphic system, we aim to encode sensory information (such as pain or touch) as electrical stimulation pulses to restore natural sensory perception. Besides developing the prosthetic hand, we analyze neuromorphic agents (i.e., arm) integrated into cybernetic chains of action in which nervous systems engage in closed interaction with their bodies and environments. This study is a part of the NeuroSys project and aims to build a foundation for future research to realize the goal of the project.

METHODOLOGY

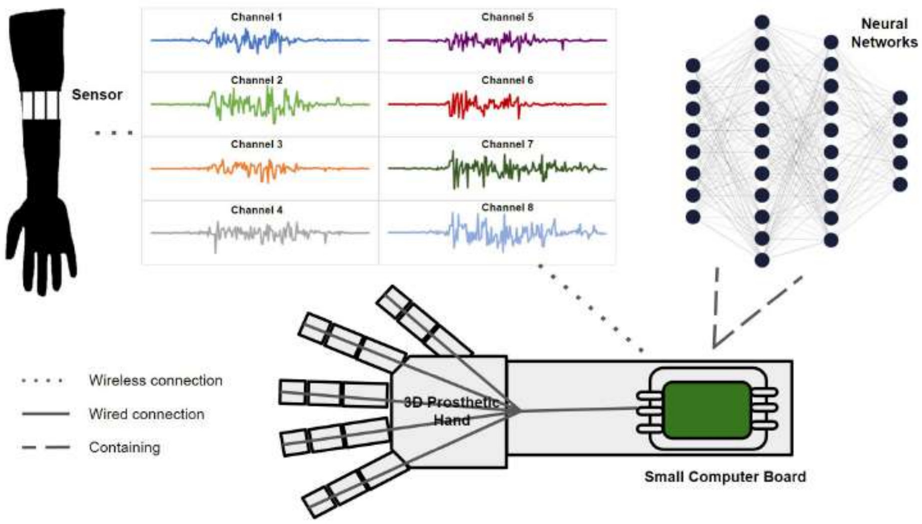

FIGURE 2. sEMG Recognition Architecture. The sEMG is sent wirelessly to a computer. Inside the computer, the data is then classified by neural networks. The 3D prosthetic hand is then moved according to the decoded movement.

In this section, the architecture of sEMG recognition is described. Figure 2 shows how the sEMG signal is taken and sent to a computer, which is then fed into a neural network to decode. The materials and characteristics of the data will be explained. Then the data recognition method is discussed. Finally, an affordable 3D printed hand model will be shown.

Electromyography Signal

Electromyography (EMG) is a signal of electrical activity produced by skeletal muscles. After the bio-electrical signal is generated by the brain, it is sent down to the neurons, and the alpha motor neurons cause the muscle to contract and release, which produces an EMG signal. Thus, EMG is captured when there is a movement of muscle activity.

There are two kinds of EMG. The first one is intramuscular EMG, which is a signal taken by implanting an electrode into the muscle; while accurate data can be taken, it is considered invasive. The non-invasive way is called surface Electromyography (sEMG), which can be taken by placing sensors or an array of electrodes on the part of the body (skin). Even though sEMG data is noisy compared to intramuscular EMG, it can be implemented and realized by researchers in laboratory. The sEMG can be further divided into two types: Sparse and high-density, as shown in Table I. We are going to use a sparse multi-channel sEMG sensor in this study, namely Myo Armband Sensor.

	Sparse multi-channel sEMG	High-density sEMG (HD-sEMG)
Pros	Cheaper hardware	More data
Cons	Sensitive to domain change of sEMG signal	Expensive hardware and complex data processing

TABLE I. Comparison Between Two Types of sEMGs.

There are challenges when working with sEMG data. First, sEMG signals are different for each person. Second, sEMG data taken from the same person at different sessions can be inconsistent. These characteristics make high accuracy gesture recognition a difficult task to achieve. In many medical real implementations on patients, a gesture recognition system is designed solely for one patient to increase and optimize prosthetic control functionalities.

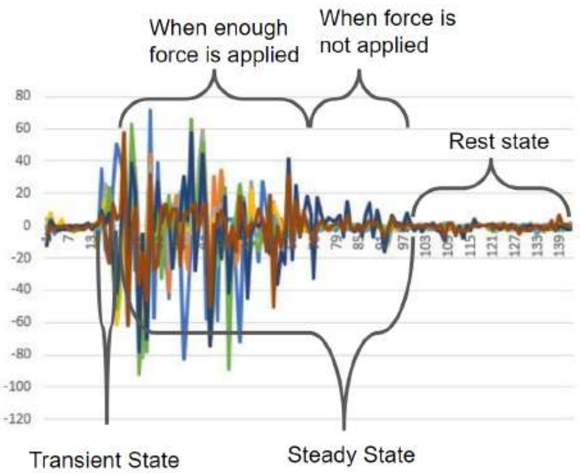

FIGURE 3. An Example of sEMG Taken From My Armband Sensor. Different states are labeled.

When a muscle is contracted and enough force is applied, there will be a big change in signal amplitudes. During rest state, when there is no muscle contraction or no force applied, the signal stays at around zero. There are two states during hand movements: transient state and steady state. When hand gesture changes from one to another, the first stage is called transient state. If the gesture continues to be held after, it is called a steady state. When enough force is applied, the signal amplitudes can easily be observed. However, when little to no force is applied, the signal is barely or not even distinguishable from the rest state. The different states are shown in Figure 3. (This problem might or might not be limited to the sensor)

Myo Armband Sensor

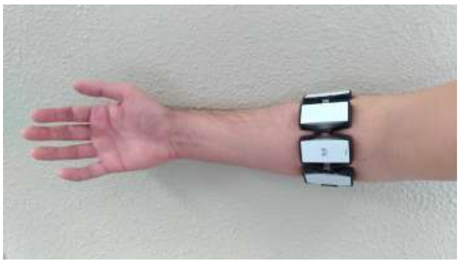

FIGURE 4. Myo Armband Sensor Worn on Right Arm. There are, in total, eight white blocks in total.

Myo armband is a non-invasive sparse multi-channel sEMG sensor. It is usually worn on the forearm just under the elbow, but not limited to this part. The Myo armband sensor used in this work can take a sampling frequency of up to 200Hz (However due to processing and communication delay it could take only around 100-150Hz sampling frequency) with eight channels. Each white block corresponds to one sensor which can capture one channel of sEMG data. The placement of the sensors is important. For example, as shown in Figure 4, if the sensor block with a mark is placed in the middle of the front part of the forearm, it should always be placed at the same position when taking new data or during testing. Slight variations can produce different data even with the same gestures or hand movements. After the sEMG data is captured, it is sent wirelessly to a computer that is connected to the sensor. Then, processing and data recognition happen in a computer.

Myo Armband Sensor

sEMG signal has both temporal and spatial characteristics. The signal is time series, and there are multiple channels. There are many different approaches to realize sEMG recognition but only a few effective ones for real-world usage [10, 11].

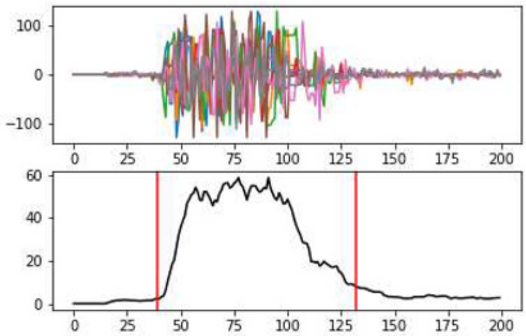

FIGURE 5. Motion Detection in a Sample Data Using Standard Deviation.

When working with sEMG data recognition, many studies suggest that using feature extraction techniques can improve the performance [12, 13]. In this study, we are going to use the method done by [10]. First, the standard deviation for multiple channels is applied to detect where the hand gesture movement begins and stops within the data. A threshold value is chosen; any values higher than the threshold are considered to be the gesture movement, otherwise rest. This acts as the signal amplitude label for sEMG data. This method is not one hundred percent accurate but acceptable labeling can be achieved. Figure 5 shows where gesture motion is detected and where there is no motion or rest state. For machine learning model, we are going to use Recurrent Neural Network (RNN), Gated Recurrent

Units (GRUs) [14] and Long Short Term Memory (LSTM) [15] to accomplish this task . This deep learning model is influenced heavily by the labels of the data which are generated from standard deviation as mentioned. However during data collection, if the subject does not apply force when making a gesture, the signal at that time period will look similar to that of the rest state and it will be labeled as rest even though in reality it is not. For the RNN family, a simple RNN unit has only tanh activation function, whereas LSTM, and GRUs are much more complex. As research [16] has shown a simple RNN has vanishing gradient problem, causing it to be less reliable than the more complex units. It is expected LSTM and GRUs would perform better than the simple RNN. The architecture of RNN, LSTM, and GRU units are shown in Figure 6.

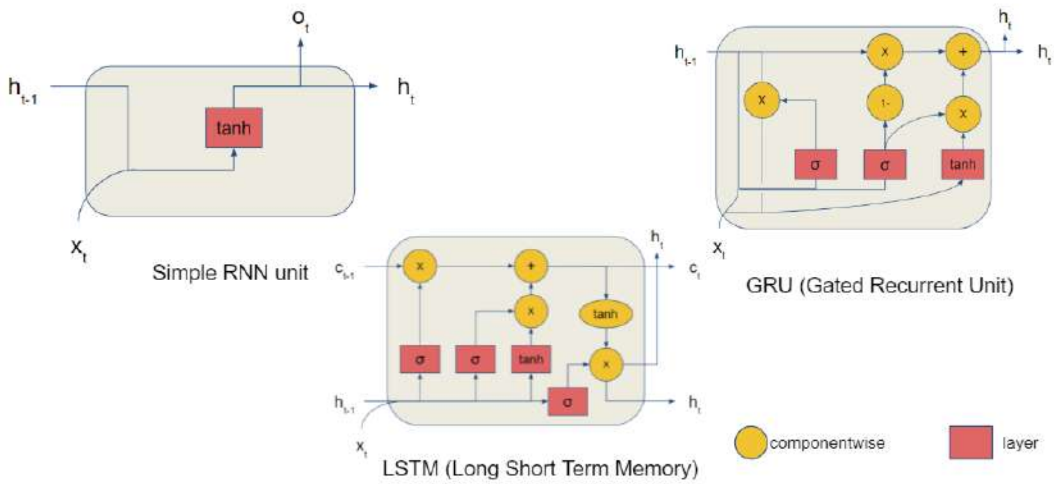

FIGURE 6. RNN, LSTM, and GRU units.

3D Printed Prosthetic Hand

In addition, an open-source inmoov hand model [17] is 3D printed to study and test if the sEMG recognition system does work in real-world usage. Figure 7 (A) illustrates all the parts needed to build the hand model including from fingers to palm until the base for motors. A low-cost material, Poly Lactic Acid (PLA) filament is used to print all these parts. Each finger is connected to a motor separately via a string, which is used to move the fingers. The five motors are connected to a Raspberry Pi. The Raspberry Pi acts as a controller for the prototype prosthetic hand. The trained model is converted to tensorflow lite (32-bit) version model for the small board computer. The Raspberry Pi uses this trained model do inference and control the motor accordingly. Figure 7 (B) demonstrates the controlling part, motors connecting to a Raspberry Pi. Figure 7 (C) shows the assembled hand model.

In terms of computational power, Raspberry Pi is considered relatively slow in exchange for low-power consumption. However, by using 3D printing and an open source hand model together with the Raspberry Pi, a low-cost prosthetic hand prototype could be realized.

EVALUATION

In this section, sEMG data used in the experiments together with the detailed implementation of the deep learning model are shown. Also, the evaluation techniques with results will be discussed. Note that all the experiments are done using TensorFlow Keras.

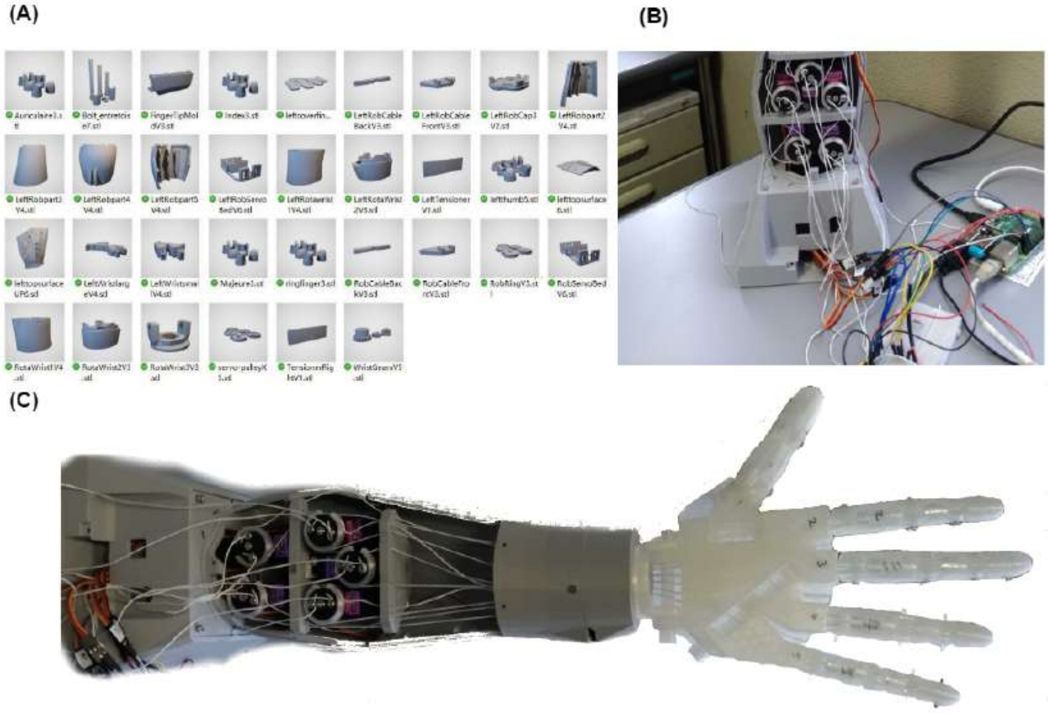

FIGURE 7. 3D printed prosthetic hand. (A) shows the parts of inmoov hand model, (B) illustrates the connection between Raspberry Pi and motors, and (C) is the assembled hand model.

Evaluation Methodology

Seven different hand gestures, including rest gesture, of data will be used in this study. In each data, there are signals of a gesture and a rest state signal. So, in one data, there are two classes. 8 illustrates the seven gestures with the sEMG signal visualized between -150 and 150 amplitude. The data is taken from three healthy male adults in two different sessions, so we can see the effect of sensor placement variation, which can happen during real-world usage. For each session, we have taken 20 times for one gesture. Each gesture data is 2 seconds long, so there are 200 signal points, where each signal point has a label, in one data. The data begins at rest state and transition to one of the gesture and then go back to the rest state again. However, for subject 3, after the hand gesture, the signal does not go back to the rest state.

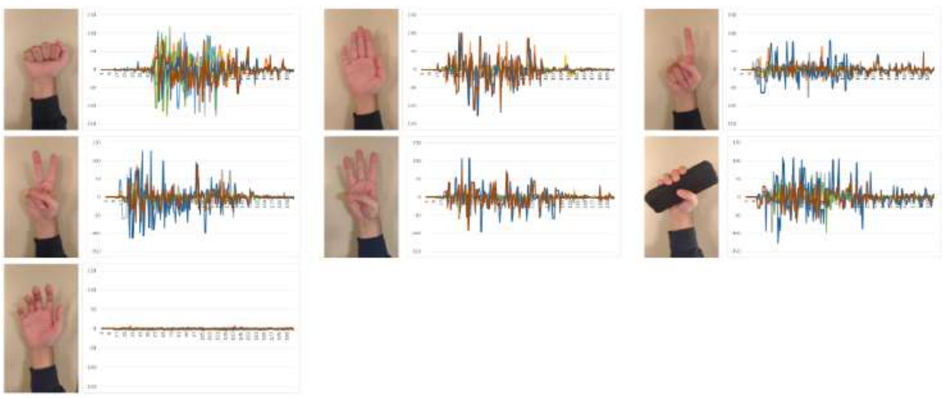

FIGURE 8. The Gestures Data Used in This Study. There are seven gestures including rest.

 https://doi.org/10.1051/shsconf/202213903004

For signal labeling standard deviation is used as mentioned. The window size is set to 10 time step at a time for calculation. The threshold is set to 8. Any standard deviation results more than the threshold are labeled as the gesture class, and results lower than the threshold are set to rest class.

The RNN model's architecture used in the experiment is presented in Figure. 9. First, the eight signal channels are input into 50 units of RNN/GRU/LSTM, followed by a linear layer. Because our sEMG signal has positive and negative values, the tanh activation function is used after the linear layer. After that, it goes through another 50 units of RNN/GRU/LSTM. And lastly, a linear layer with softmax function to classify into seven different classes, as we have seven gestures for our data. The parameters for training and testing is set similarly for the three models.

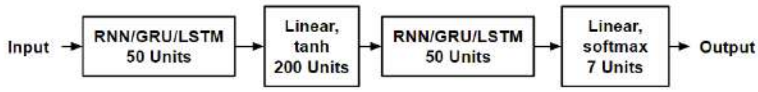

FIGURE 9. The RNN Architectures Used in This Study. There are three architectures using only RNN, GRU, or LSTM.

The offline evaluation is conducted. The evaluation focuses on the accuracy of decoding hand gestures. First, the two sessions of data are evaluated separately using five-fold cross-validation. Then, we combine the two data sessions by doing the same five-fold cross-validation but with more data. Also, the results of the combined data using RNN, GRU, and LSTM are compared. For all evaluations, the effect of different time-step window sizes on accuracy is observed. In addition, we also experiment the online evaluation on both software and the 3D hand model. For online evaluation of software, the best model is used. Excluding the rest gesture, at the start of a hand gesture, a rest state is held for 5 inference times. After the rest state, the gesture is held for 20 inference times. The process is repeated for the six gestures.

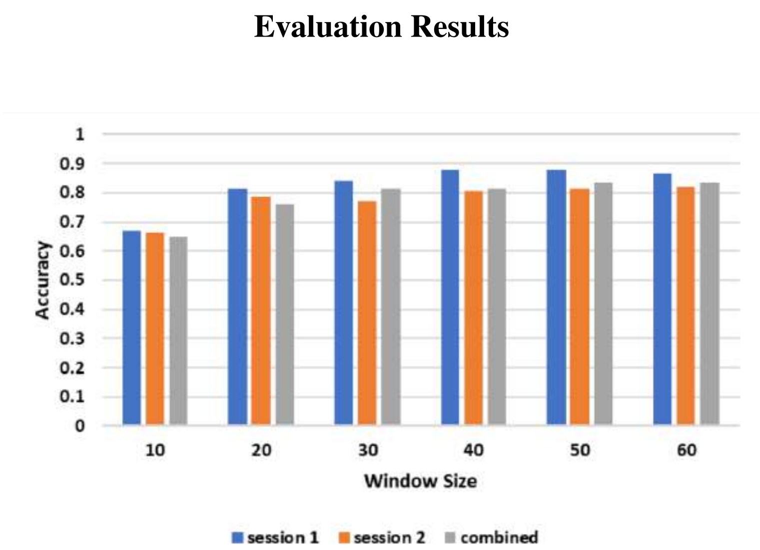

FIGURE 10. GRU Accuracy with Respect to Different Window Sizes of Subject 1. The accuracy is the average accuracy of 5 fold cross-validation.

From Figure 10, the blue, orange, and gray bars represent the accuracy of session 1, session 2, and combined data, respectively. In this case, the session 1 has shown the best accuracy but the session 2 has the lowest. Using data from two different sessions, the combined data accuracy is somewhat in between the session 1 and session 2 accuracy.

For the time step of 10, only 10 amplitude signals or around 100 ms of a signal is used to decode the intent, where the time step of 50, 50 amplitude signals, or around 500 ms of a signal is used. We can observe that bigger time step window sizes have higher accuracy performance than smaller ones. However, this does not hold true when the window size is too big. For more detailed view, the numeric results are shown in Table II.

TABLE II. Numeric Accuracy Results of GRU Respects to Different Window Sizes of Subject 1. The accuracy is the average accuracy of 5 fold cross-validation.

Window Size	Session 1	Session 2	Combined Sessions
10	0.74286	0.62143	0.68214
20	0.91429	0.77857	0.83571
30	0.94286	0.82143	0.83929
40	0.92857	0.85	0.84643
50	0.95	0.82142	0.85
60	0.92143	0.81429	0.85
70	0.91429	0.83571	0.84643
80	0.91429	0.82857	0.84643

From Figure 11, for all three subjects, simple RNN has the worst performance; accuracy of around 40% for S1 and S2. Because S3 data is taken without the rest state at the end, the accuracy could reach 60%. For LSTM and GRU, S1 and S2 have very similar results. For S3 however, GRU has a slight edge over LSTM.

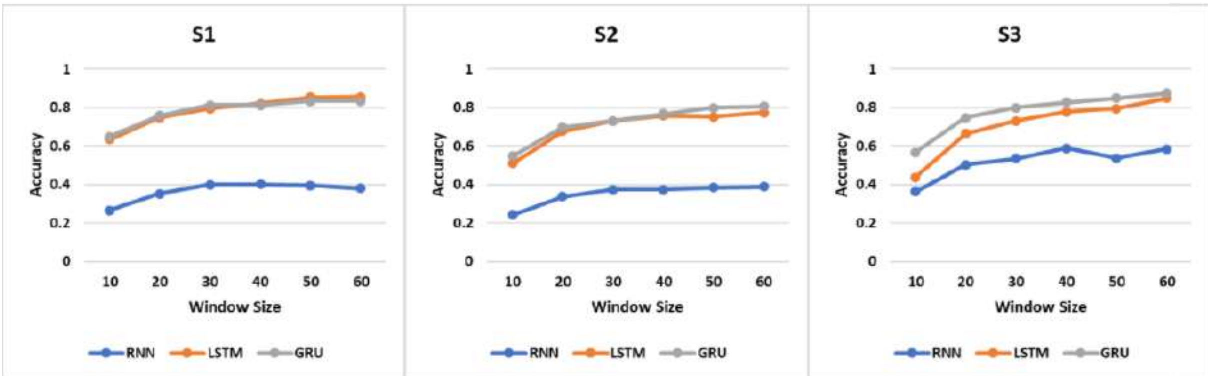

FIGURE 11. Average accuracy of combined data of the three subjects with RNN, LSTM, and GRU.

The RNN unit is the most simple one of the three however it has the worst results. This is due to the RNN gradient vanishing property. This limitation is overcome by using LSTM and GRU as the results have been shown. The detaled view of the results are shown in Table III.

TABLE III. Average accuracy of combined data of the three subjects with RNN, LSTM, and GRU.

Window Size	RNN			LSTM			GRU		
	S1	S2	S3	S1	S2	S3	S1	S2	S3
10	0.26786	0.24286	0.36429	0.63571	0.51071	0.43929	0.65	0.55	0.56786
20	0.35357	0.33929	0.50357	0.75	0.67857	0.66428	0.76071	0.7	0.75
30	0.4	0.375	0.53571	0.79643	0.73214	0.73214	0.81429	0.73214	0.8
40	0.40357	0.375	0.58929	0.82143	0.76071	0.77857	0.81429	0.76786	0.82857
50	0.39643	0.38571	0.53929	0.85357	0.75357	0.79643	0.83571	0.8	0.85
60	0.38214	0.38929	0.58571	0.85714	0.775	0.85	0.83571	0.80714	0.875

Using the best trained model, an online evaluation is conducted. Figure 12 shows the results of the online evaluation using software. For distinct gestures, the system has high accuracy, however for similar signal gestures, especially one finger, two fingers, and three fingers, the prediction shows poor results. For the rest state however, the timing delay causes some wrong prediction. The average accuracy is around 83%.

Due to the nature of the sensor, during online testing, the user is required to apply force for the six gestures constantly, otherwise, the picked up signals are hard to distinguish.

FIGURE 12. Online testing result on software.

CONCLUSION

In this study, we focused on sEMG data recognition and online testing controlling a prototype 3D printed prosthetic hand. Accuracy and good response time are two important factors. We compared the sEMG recognition performance of RNN, GRU, and, LSTM. Although there is a trade-off between the accuracy and window size, bigger window size does not always lead to better accuracy. An optimal window size can be explored. We also discussed that force is important during the steady-state of a gesture in order for the Myo armband sensor to detect the signal, otherwise, it is conceived as rest state. The placement variation of sensor also affects the data and our trained model. It is best to take data in many sessions to have robust data set. Also, during online testing which is close to real-world usage , the user has to apply force constantly. This causes fatigue and discomfort for the user.

In the future, approaches that can help improve the performance together with the optimization to the limitation of force problem will be further explored.

REFERENCES

1. A. Furui, S. Eto, K. Nakagaki, K. Shimada, G. Nakamura, A. Masuda, T. Chin, and T. Tsuji, "A myoelectric prosthetic hand with muscle synergy–based motion determination and impedance model–based biomimetic control," Science Robotics **4**, eaaw6339 (2019).
2. K. Nazarpour, *Control of Prosthetic Hands: Challenges and emerging avenues* (Institution of Engineering and Technology, 2020).
3. W. Li, P. Shi, and H. Yu, "Gesture recognition using surface electromyography and deep learning for prostheses hand: State-of-the-art, challenges, and future," Frontiers in Neuroscience **15** (2021).
4. U. C. Allard, F. Nougarou, C. L. Fall, P. Giguère, C. Gosselin, F. Laviolette, and B. Gosselin, "A convolutional neural network for robotic arm guidance using semg based frequency-features," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (IEEE, 2016) pp. 2464–2470.
5. J. Gibbard, "Open hand project," Bristol, UK, accessed July **20**, 2017 (2013).
6. X. Zha, L. Wehbe, R. J. Sclabassi, Z. Mace, Y. V. Liang, A. Yu, J. Leonardo, B. C. Cheng, T. A. Hillman, D. A. Chen, *et al.*, "A deep learning model for automated classification of intraoperative continuous emg," IEEE Transactions on Medical Robotics and Bionics **3**, 44–52 (2020).
7. S. Lobov, N. Krilova, I. Kastalskiy, V. Kazantsev, and V. A. Makarov, "Latent factors limiting the performance of semg-interfaces," Sensors **18**, 1122 (2018).
8. C. Pylatiuk, S. Schulz, and L. Döderlein, "Results of an internet survey of myoelectric prosthetic hand users," Prosthetics and orthotics international **31**, 362–370 (2007).

9. J. T. Belter and A. M. Dollar, “Performance characteristics of anthropomorphic prosthetic hands,” in *2011 IEEE International Conference on Rehabilitation Robotics* (IEEE, 2011) pp. 1–7.
10. Z. Zhang, C. He, and K. Yang, “A novel surface electromyographic signal-based hand gesture prediction using a recurrent neural network,” Sensors **20**, 3994 (2020).
11. M. Zanghieri, S. Benatti, A. Burrello, V. Kartsch, F. Conti, and L. Benini, “Robust real-time embedded emg recognition framework using temporal convolutional networks on a multicore iot processor,” IEEE transactions on biomedical circuits and systems **14**, 244–256 (2019).
12. S. Guo, M. Pang, B. Gao, H. Hirata, and H. Ishihara, “Comparison of semg-based feature extraction and motion classification methods for upper-limb movement,” sensors **15**, 9022–9038 (2015).
13. X. Xi, M. Tang, S. M. Miran, and Z. Luo, “Evaluation of feature extraction and recognition for activity monitoring and fall detection based on wearable semg sensors,” Sensors **17**, 1229 (2017).
14. J. Chung, C. Gulcehre, K. Cho, and Y. Bengio, “Empirical evaluation of gated recurrent neural networks on sequence modeling,” arXiv preprint arXiv:1412.3555 (2014).
15. S. Hochreiter and J. Schmidhuber, “Long short-term memory,” Neural computation **9**, 1735–1780 (1997).
16. S. Hochreiter, “The vanishing gradient problem during learning recurrent neural nets and problem solutions,” International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **6**, 107–116 (1998).
17. G. Langevin, “Inmoov,” URL http://www. inmoov. fr/project (2014).