Proceedings of the ASME 2025
International Design Engineering Technical Conferences and Computers and Information in Engineering Conference
IDETC/CIE2025
August 17-20, 2025, Anaheim, California

DETC2025-157108

UPPER EXTREMITY UNMEASURED MUSCLE ACTIVATION PREDICTION THROUGH RANDOM FOREST AND NEURAL NETWORK

James Yang
Department of Mechanical Engineering
Texas Tech University
Lubbock, Texas 79409, USA

Shadman Tahmid
Depart of Human Performance-Exercise Physiology
West Virginia University
Morgantown, West Virginia 26506, USA

Baivab Bhandari
Department of Mechanical Engineering
Texas Tech University
Lubbock, Texas 79409, USA

ABSTRACT

Patients with neuromuscular diseases often struggle to generate the muscle force required for daily activities and to maintain the necessary joint moments. Accurately measuring muscle force in these patients is both important and challenging. Electromyography (EMG) can be used to gather muscle activation data, which can then be converted into muscle force and joint torque values. While surface electrodes are effective for measuring superficial muscles, fine-wire electrodes are necessary for targeting deeper muscles. However, the use of fine-wire electrodes is invasive and requires skilled personnel and considerable preparation time. It's important to note that using EMG-driven modeling with only surface electrodes may lead to an underestimation of the net torque. In our previous work, an optimization-based method combining muscle synergy extrapolation and EMG-driven musculoskeletal modeling has been developed to predict unmeasured muscle activations. However, the optimization-based method could only predict maximum 3 unmeasured muscles. This paper aims to address the issue by investigating machine learning algorithms that can derive unmeasured muscle activation patterns using anthropometric and kinematic features. To train the models, muscle activation data from healthy subjects will be employed for two distinct networks: the random forest algorithm and the artificial neural network. These networks will then predict muscle activations for an unseen subject based on the input features provided. The results showed that the machine learning approaches are proven to be an alternative method.

Keywords: Muscle synergy extrapolation; Musculoskeletal modeling; muscle activation; machine learning; and rehabilitation

1. INTRODUCTION

The central nervous system (CNS) is a complex network that transmits neural commands to the upper and lower extremities. When activated by these neural signals, the musculotendon units that span specific joints generate the torque needed to move those joints. The coordinated movements of the joints in the upper and lower extremities contribute to overall human movement. However, brain injuries, such as strokes, can damage the CNS, preventing it from sending the essential commands required to generate muscle force. This loss of muscle strength diminishes torque production in the joints, making it difficult for patients to perform daily tasks. It is estimated that around one billion people worldwide suffer from neurological disorders [1]. Patients with neurological disorders achieve complete recovery of upper extremity functionality only 25% of the time [2]. A recovering patient needs rehabilitation therapies to strengthen injured muscles. Patients with neuromuscular diseases struggle to generate the necessary muscle force, which prevents them from maintaining the joint stability required to perform daily activities. Evaluating muscle force values is essential for understanding neuromuscular behavior during human movement. Although measuring muscle-tendon forces in vivo is quite challenging, researchers are actively developing computational methods to address this issue [2-7]. Recent research shows that motion capture systems can detect kinematic movements and provide essential feedback for specific rehabilitation tasks [3]. To monitor a patient's muscle recovery, it is important to measure both individual muscle forces and joint torques. Researchers utilize neuromusculoskeletal models to examine muscle-tendon functions and human movements; however, accurately determining joint torques has proven to be a challenge. Typically, joint torques are estimated from movement data using inverse dynamics, as discussed in [4]. The presence of multiple muscle-tendon units around a single joint result in redundancy, which complicates the calculation of muscle forces derived from joint torques. To address this redundancy, it's necessary to use an optimization approach that relies on an objective function to determine the individual muscle forces accurately [5]. Another way to estimate muscle forces is by using electromyography (EMG) signals. This technique measures the electrical activity generated by nerves when they stimulate the muscles. Surface EMGs specifically capture this neural activity from surface muscles during human movement [6]. Measured EMG signals can be used to compute joint torques, either without [7-11] or with [12-20] a musculoskeletal model. However, estimating joint torque using only EMG signals (without a musculoskeletal model) typically requires additional equipment, such as a dynamometer or force plates, during experimental data collection [7, 9]. Researchers used musculoskeletal models and experimentally derived EMG data to simulate movements of the upper [21] and lower extremity [13-16].

In our previous work, an optimization-based method combining muscle synergy extrapolation and EMG-driven modeling for one or multiple unmeasured muscles was reported [17-18]. However, when the total number of unmeasured muscles reaches to 4, the method could not be used to predict unmeasured muscle activations. This is the motivation to explore other alternative methods for this purpose.

Machine learning approaches have gained significant attention in biomechanical research, especially concerning the estimation of forces generated by skeletal muscles. Researchers have particularly focused on creating machine learning models that can approximate muscle tensions without needing to explicitly model the intricate physical behaviors associated with muscles. These models draw on various input data types, such as kinematic and kinetic measurements, EMG signals, and individual muscle weightings.

Numerous studies have been conducted to assess the performance of machine learning models in estimating muscle tensions. For example, Arjmand et al. [19] employed artificial neural networks (ANN) to forecast trunk muscle forces during static lifting activities. Cecchini et al. [20] used ANNs to estimate muscle forces based on kinematic data, particularly in a cycling scenario. Dao [21] applied a long short-term memory model as a recurrent deep neural network to predict forces in skeletal muscles. Rane et al. [22] trained a convolutional neural network (CNN) utilizing kinematic, kinetic, and EMG data from individuals engaged in walking tasks to estimate internal muscle forces. Furthermore, Gonzalez-Vargas et al. [23] created a prediction model that generated muscle-specific excitation patterns for locomotor scenarios involving healthy participants.

Research indicates that machine learning models can estimate unmeasured muscle forces. However, these models also face certain limitations. Data noise can influence their output, and they struggle to process data outside of the training set, leading to challenges in real-world applications. Traditionally, forward dynamics has been employed to estimate joint kinematics from muscle activations. Jiang et al. [24] utilized a CNN to predict shoulder movements based on EMG signals from limb muscles. Trigili et al. [25] employed a machine learning algorithm to recognize motion patterns from surface electrode muscle activations during reaching tasks that involve shoulder and elbow movements. Conversely, the reverse process of identifying muscle activations from kinetic data has been explored. Jonic et al. [26] applied neural network methods to predict muscle activations using hip and knee joint data from the lower limbs. Mohr et al. [27] utilized support vector machines (SVM) to classify gait muscle activation patterns based on knee injuries.

While the potential and promise of machine learning in biomechanical research are evident, it is worth noting that none of the existing studies have explicitly targeted rehabilitation tasks for the upper extremity, such as reaching activities. This research aims to address this gap by utilizing machine learning algorithms to estimate unmeasured muscle activations, specifically for the elbow and shoulder joints in the context of rehabilitation tasks.

2. MATERIALS AND METHODS

2.1 Experimental Data Collection

The experimental protocol received approval from the Internal Review Board (IRB) at Texas Tech University. The motion capture system employed in the study consisted of eight Eagle-4 cameras, each equipped with 10 retroreflective markers placed on the arm during various tasks (Motion Analysis Corporation, California, USA), as illustrated in Figure 1. Each camera has a resolution of four megapixels and can achieve a maximum frame rate of 500 frames per second. Motion capture was conducted at a frequency of 100 Hz using Cortex software (Motion Analysis Corporation, CA, USA). The motion capture area measured 3.04 × 3.04 square meters, with cameras strategically positioned around it at approximately 2.74 meters in height, as shown in Figure 1(b). The markers were placed on the anatomical landmarks of the upper extremity to minimize movement, following the protocols established in [50], as depicted in Figure 1(a).

The analog EMG signals were captured using the Trigno Wireless Biofeedback System (manufactured by DELSYS Incorporated, Massachusetts, USA), which included 16 surface EMG sensors. A total of 6 EMG sensors were utilized to assess the electrical activity of the muscles: three sensors were positioned on the elbow muscles (biceps brachii, long head of the triceps, and lateral head of the triceps), while the other three were placed on the shoulder muscles (anterior deltoid, clavicular part of pectoralis major, and thoracic portion of latissimus dorsi).

EMG signals were recorded at a frequency of 1299 Hz simultaneously with the motion capture software, Cortex.

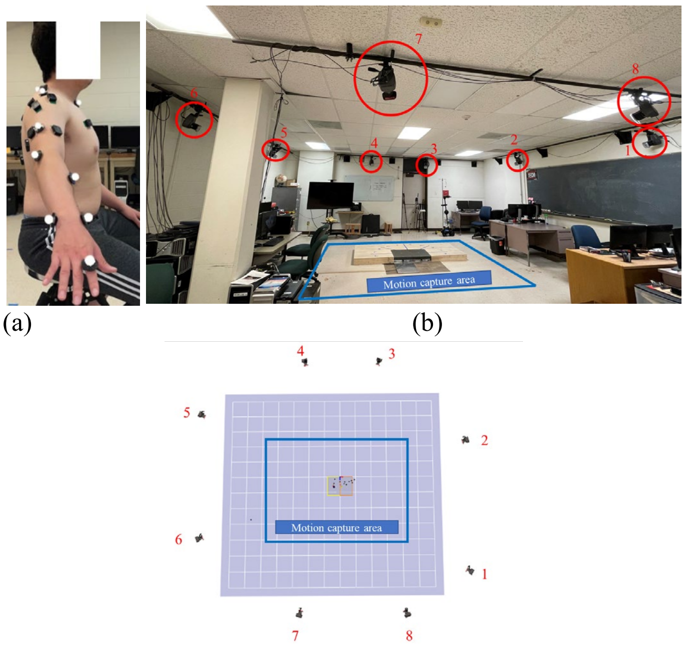


Figure 1: Lab setting: (a) Placement of motion capture markers and EMG for the participant, (b) motion capture setup featuring an 8-camera system, and (c) arrangement of the cameras.

The placement of the EMG electrodes and the preparation of the skin were conducted following the protocols set forth in [28-29]. To minimize interference from external noise, any excess hair on the skin was trimmed. An alcohol swab was used to clean off surface oils and other impurities.

A total of fourteen healthy male participants were initially recruited for the study. However, only ten of the participants (mass 83.37 +/- 13.84 kg, height 179.81 +/- 8.38 cm) were included in the postprocessing stage due to issues such as incomplete motion capture data or defective EMG recordings. The remaining four participants were excluded from the analysis to ensure data quality and reliability.

Prior to data collection, the participants signed informed consent form and subject's anthropometric measurements such as standing height and body mass were recorded for each participant before for scaling purposes. Additionally, a static standing T-pose trial was conducted for scaling purposes in OpenSim. Each subject was then instructed to perform reaching tasks. A schematic of the task is shown in Figure 2. The reaching movement was designed to simulate a common rehabilitation exercise where participants were instructed to reach for a small object placed on a table while seated. The object was placed on a table such a way that the subject had to extend the elbow fully.

To ensure data reliability and consistency, each participant performed the task three times. This approach provided multiple observations and reduced the influence of random variations. A five-minute break between trials ensured minimal effect of muscle fatigue.

To prepare the motion capture data for analysis, the markers were first labeled in Cortex (Figure 3(a)) [30]. During labeling, a marker set was established containing 10 markers. The markers were identified manually by visually inspecting the initial captured frame and identifying the markers based on their location and appearance using the graphical user interface (GUI) of Cortex. Then the markers were tracked throughout the trial to ensure that all markers are visible. Once tracked, the labeled marker positions were filtered using a 6 Hz cutoff frequency to remove sudden marker movements.

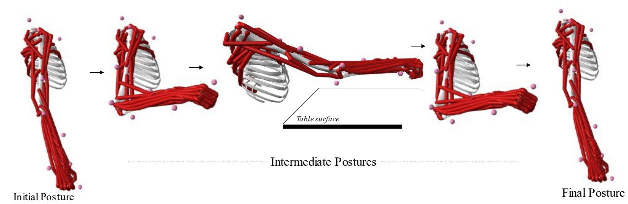


Figure 2: Snapshots of different postures for the reaching task

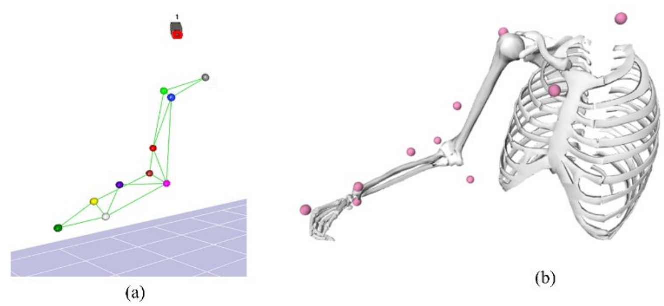


Figure 3: Marker positions for motion capture in (a) Cortex software, and (b) in the OpenSim upper extremity model [14]

During EMG signal recording, electrodes were placed on the skin to capture the electrical activity of the underlying muscles. However, due to the movement of the skin and electrodes, unwanted noise could be introduced into the recorded signals. Therefore, EMG processing was carried out to remove this unwanted noise and enhance the quality of the data. A custom MATLAB script was used to filter the raw EMG. Firstly, a high-pass filter with a cutoff frequency of 50 Hz was applied to eliminate any low-frequency noise. Next, the data were rectified to ensure only positive values were retained. Subsequently, a low-pass filter with a cutoff frequency of 6 Hz, implemented using a fourth-order Butterworth filter, was used to further smooth the data [28]. This filtering process aided in reducing noise and isolating the desired muscle activation patterns.

So far, the processed EMG data are called muscle excitations. Next step will be to transfer muscle excitations to activations. Obtained EMG excitations go through first order activation dynamics where muscle excitation e is converted to neural activation u with a first order differential equation (Eq. 1).

This equation is solved in discrete time using backward finite difference approximation. Eq. (2) describes conversion of neural activation (u) to muscle activation (a) [8].

$$\frac{du(t)}{dt} = (c_1 e(t) + c_2)(e(t) - u(t)) \dots\dots\dots\dots\dots\dots\dots (1)$$

$$a(t) = (1 - c_3)u(t) + c_3\left[\frac{g_1}{g_2(u(t) + g_3)^{g_4} + g_5} + 1\right] \dots (2)$$

where $e(t)$ represents excitation and $u(t)$ as neural activation. The constants c_1 and c_2 are defined in Eqs. (3) and (4). The value of constant c_3 is approximate to be 0.1 [8]. Coefficients g_1 to g_5 are taken as constants with values -7.62, 29.28, 0.88, 17.22 and 4.11, respectively [8]. Here, τ_{act} and τ_{deact} are activation and deactivation time delay parameters [8].

$$c_1 = \frac{1}{\tau_{act}} - \frac{1}{\tau_{deact}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (3)$$

$$c_2 = \frac{1}{\tau_{deact}} \dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots\dots (4)$$

Normalization was performed using the maximum activation value observed for each muscle. Figure 4 shows an example EMG data throughout various stages of the data processing pipeline including muscle activation dynamics.

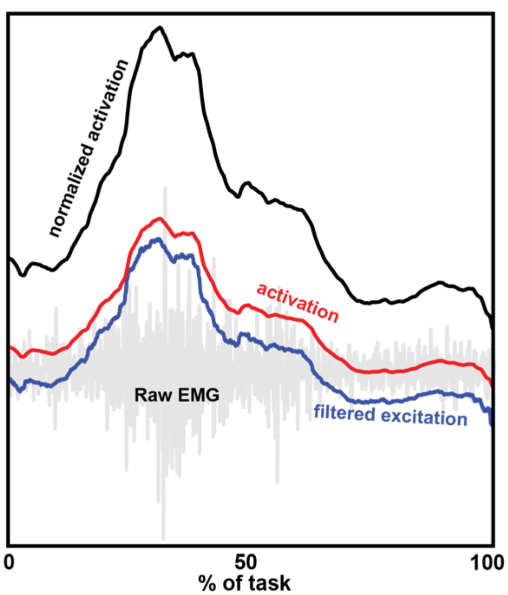


Figure 4: EMG signals of biceps long muscle for Task 1

2.2 Machine Learning Method

Filtered and normalized muscle activations from ten subjects are fed to the machine learning algorithms, random forest (RF) and ANN as an array. Inverse kinematics tool in OpenSim is used to calculate joint kinematics by tracking the marker positions [12]. Along with subjects' height and weight, joint kinematics from five degrees of freedom (DoFs) of the right upper extremity are set as input features for the prediction.

The available EMG data are split into three sets: training, validation, and test sets. For both RF and ANN, a 70-20-10 ratio is used to train the networks, where 70% of the data is allocated for training, 20% for validation, and 10% for testing. This splitting ensures that the model is trained on a large portion of the data, validated on a separate set to tune hyperparameters, and finally tested on unseen data to assess its generalization performance. Once the training is done, the networks predict muscle activations for all the muscles of the 10th subject.

2.2.1 Random Forest

The first method to train the model is RF. This method proposed by Breiman et al. [31] is based on the theory of decision trees. But instead of evaluating one decision tree, the algorithm combines multiple random decision trees to reach a conclusion based on majority.

In RF, at each node, a subset of predictors is randomly chosen, and the best predictor within that subset is used to split the node. This random selection and splitting process enhances the performance of RF, surpassing other commonly used classifiers such as discriminant analysis, SVM, and neural networks. Moreover, RF exhibits resistance to overfitting, a common issue in machine learning models, which further contributes to its appeal and effectiveness [32]. RF is also capable of handling high-dimensional data efficiently. While processing high-dimensional datasets, other methods like radial basis function and SVM can become computationally heavy. In contrast, RF automatically selects the most relevant features, reducing computational time and improving overall efficiency. Furthermore, RF demonstrates robustness against noise and outliers, making it a reliable choice in practical scenarios where data imperfections are common.

Even though RF solves classification problems, regression-based calculation is also possible to make continuous output. As EMG activations are in temporal domain, the algorithm treats each time point as an independent step and makes sure to keep continuity between each step. The model is trained using MATLAB's '*TreeBagger*' function, where the number of trees is set to 100 with a minimum leaf size of 10.

2.2.2 Artificial Neural Network

In addition to RF, this article also evaluates the performance of ANN to train the model and predict muscle activation. The ANN model is trained using Levenbrg-Marquardt backpropagation algorithm [33-34] which minimizes the sum of squared errors between the activation predicted by the network and experimental EMG activations. This algorithm makes an initial guess of activation based on provided features for a single time point. The algorithm then minimizes the squared error between this initial prediction and original EMG activation and updates the prediction [35]. The computation progresses until set convergence criteria are met. Training the ANN required a specified number of hidden units (15 in this case) with 10 neurons for each layer. The number of hidden units determines the complexity and capacity of the network to learn and represent patterns in the data.

The neural network is trained using MATLAB's *'train'* function. While training, the network adjusts its weights and biases and checks for the accuracy of the prediction. The network evaluates its performance using the validation set at each training epoch. Once the training is complete, the network makes prediction of muscle activation of the unseen 10th subject.

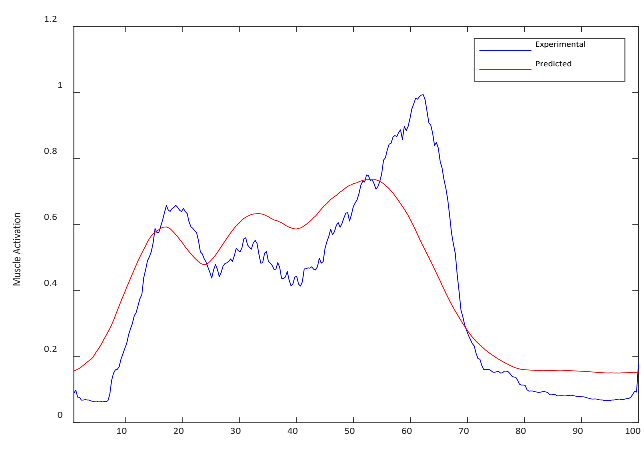
As shown in Table 1, four models are trained based on different combinations of features and algorithms. For Models 1 and 2, the training features are only height and weight. To incorporate the subject's strategy to perform a task, joint kinematics from 5 DoFs of the model are also added to the input feature set for Models 3 and 4. For training algorithms, Models 1 and 3 are trained with RF while Models 2 and 4 are trained with ANN.

Table 1. Four different models used to train the networks

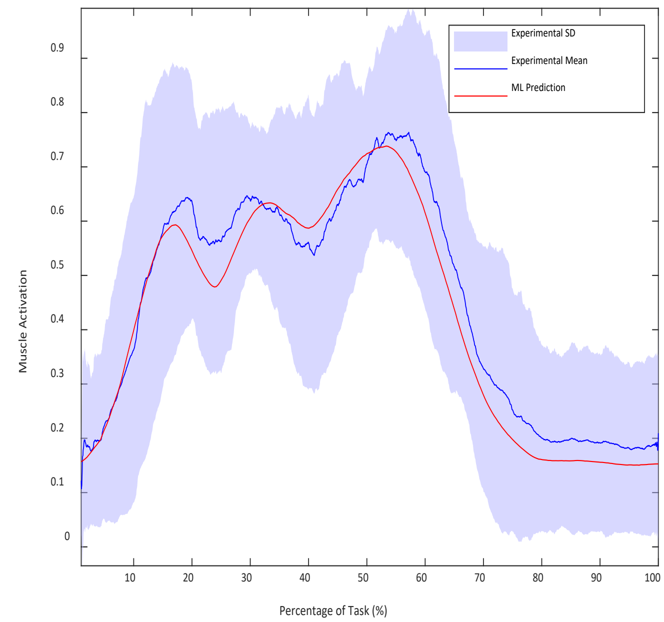
	Features				Training algorithm	
	Total number of features	Subject height	Subject weight	Joint kinematics from 5 DoFs of the model	RF	ANN
Model 1	2	✓	✓		✓	
Model 2	2	✓	✓			✓
Model 3	7	✓	✓	✓	✓	
Model 4	7	✓	✓	✓		✓

3. RESULTS AND DISCUSSION

Figures 5 to 12 show predicted muscle activations for two upper extremity muscles using different machine learning models. Even though the models determine muscle activations of all target muscles that are used for training, for this evaluation, results have been shown for one muscle chosen from the right elbow joint (biceps) or one from the right shoulder joint (deltoid), i.e., each time only one unmeasured muscle can be predicted. Comparisons are made of all four models. For each case, the predicted muscle activations are first compared with experimental EMG activation of the target muscle. Later it is compared with the total spectrum of muscle activation for all 10 subjects.

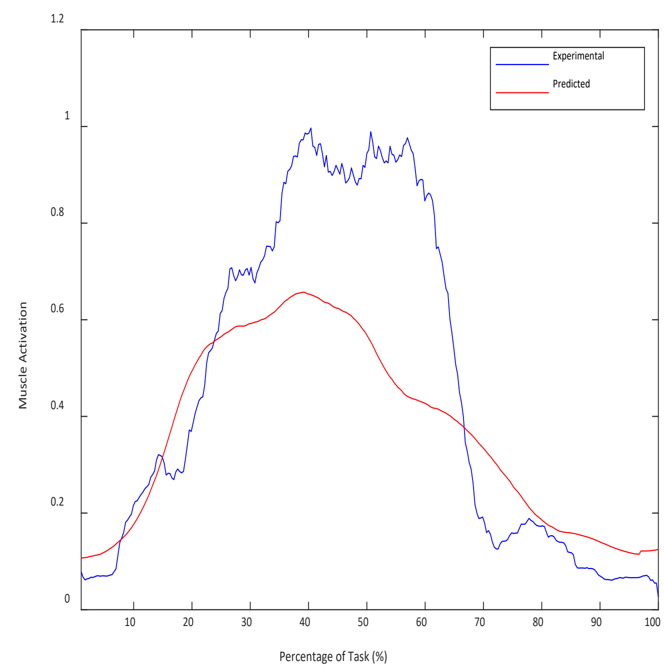


(a)

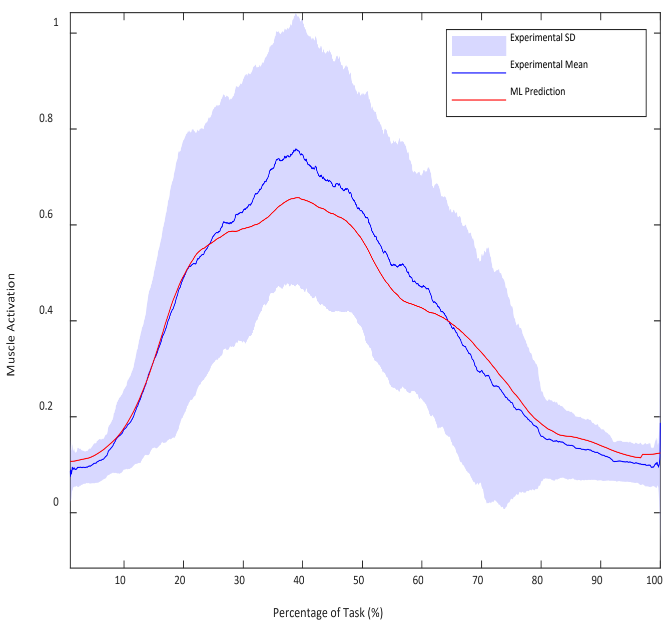


(b)

Figure 5: Model 1: Muscle activation profile predicted using RF with 2 features for biceps: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire biceps dataset of all 10 subjects.

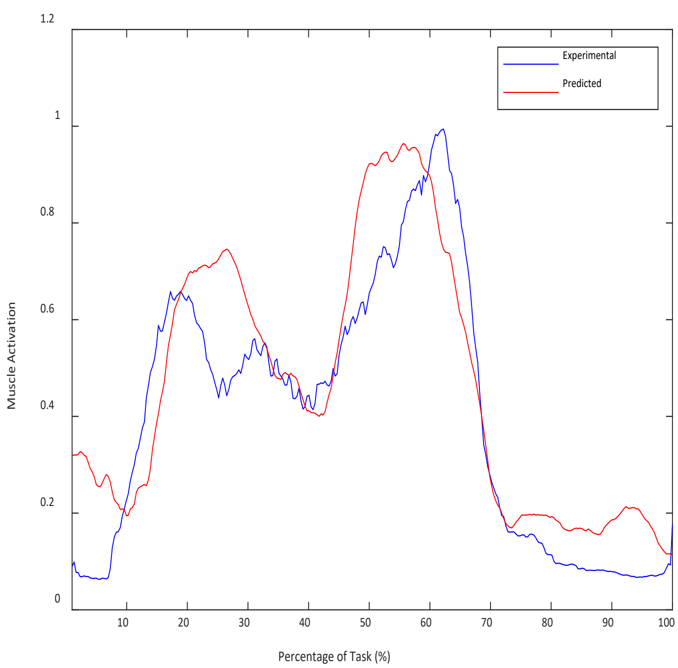


(a)

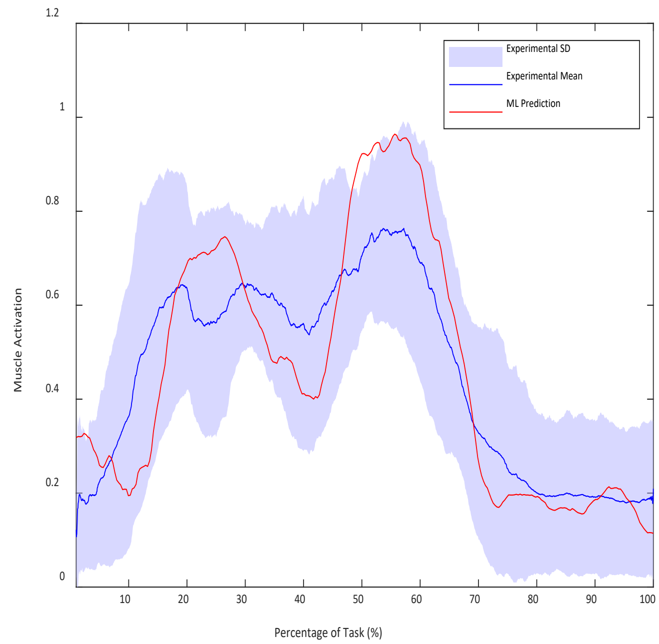


(b)

Figure 6: Model 1: Muscle activation profile predicted using RF with 2 features for deltoid: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire deltoid dataset of all 10 subjects.

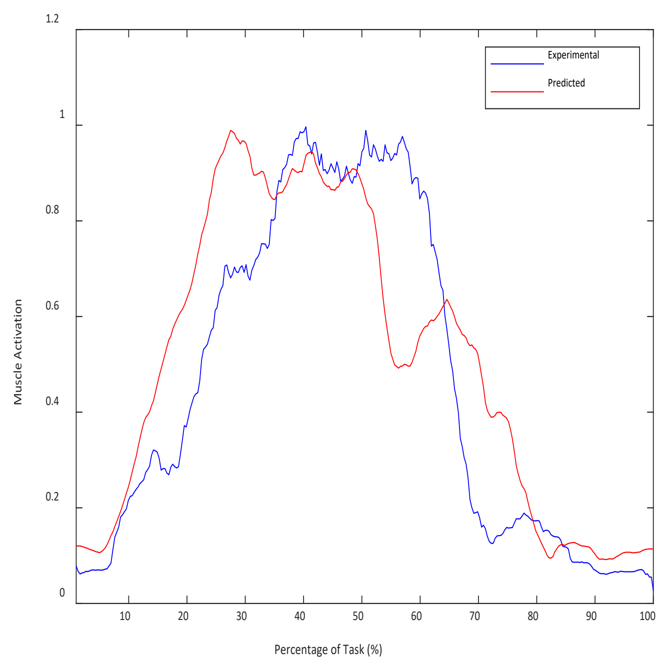


(a)

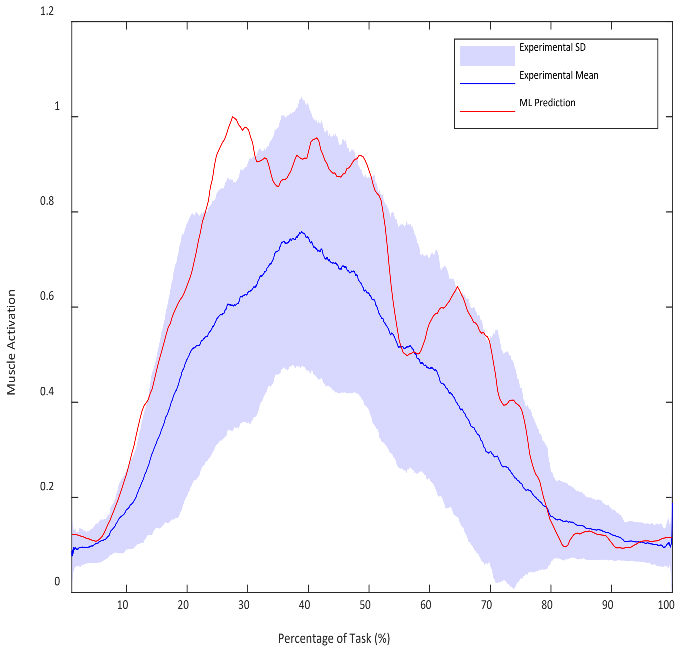


(b)

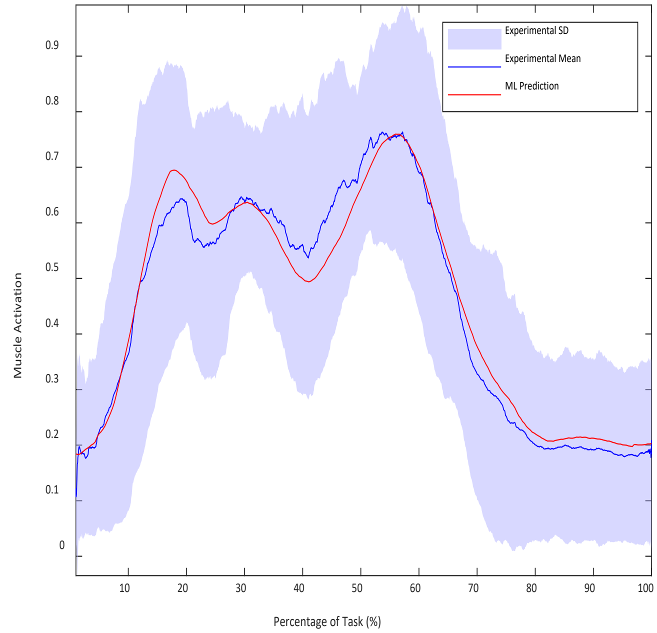
Figure 7: Model 2: Muscle activation profile prediction using ANN with 2 features for biceps: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire biceps dataset of all 10 subjects.



(a)



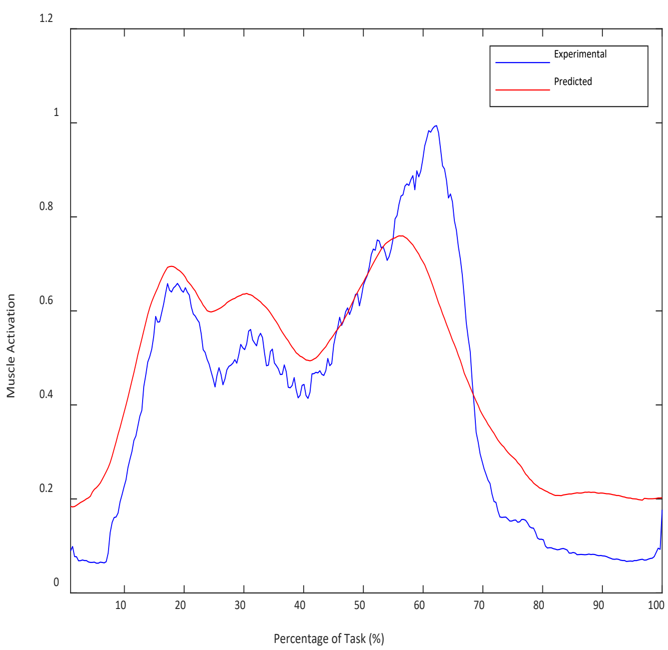
(b)



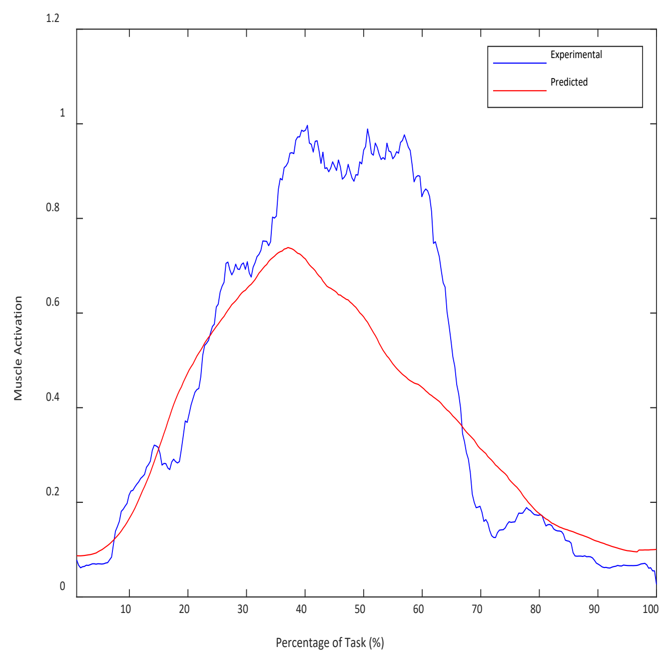
(b)

Figure 8: Model 2: Muscle activation profile prediction using ANN with 2 features for deltoid: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire deltoid dataset of all 10 subjects

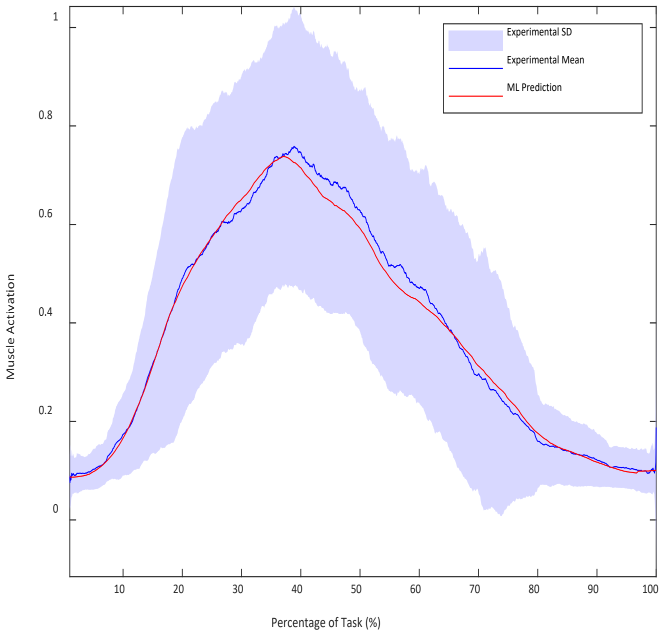
Figure 9: Model 3: Muscle activation profile prediction using RF with kinematics inputs for biceps: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire biceps dataset of all 10 subjects.



(a)

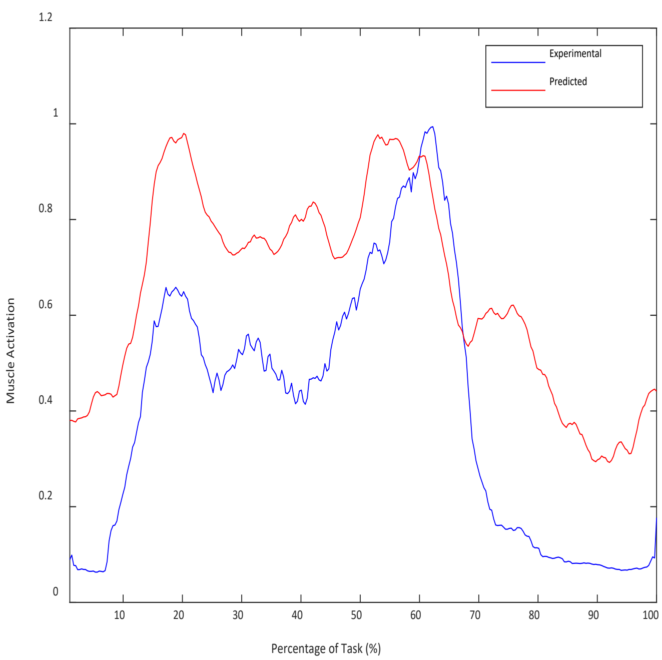


(a)

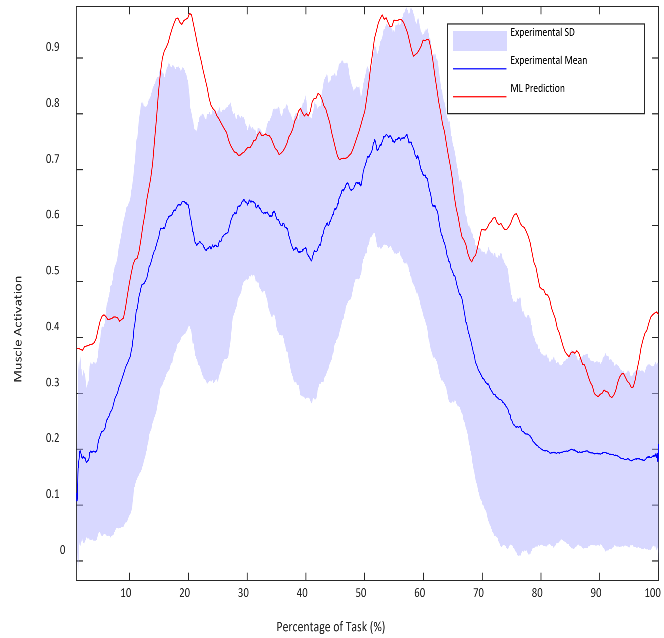


(b)

Figure 10: Model 3: Muscle activation profile prediction using RF with kinematics inputs for deltoid: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire deltoid dataset of all 10 subjects

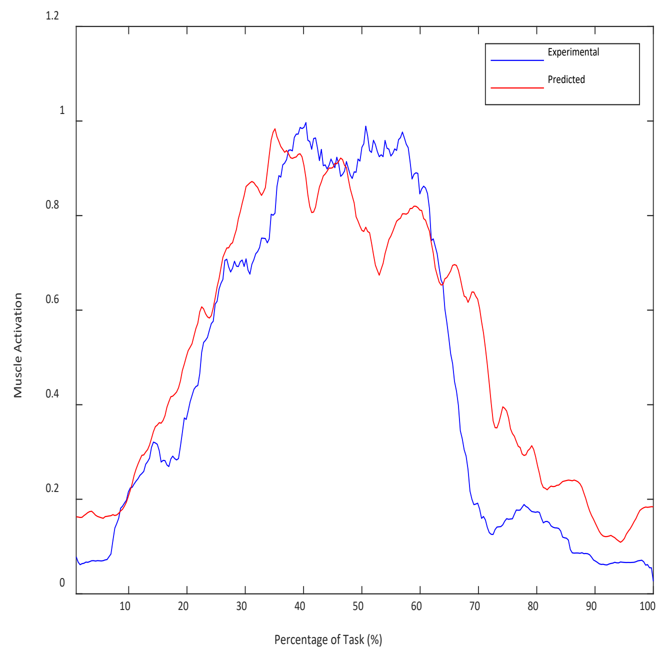


(a)

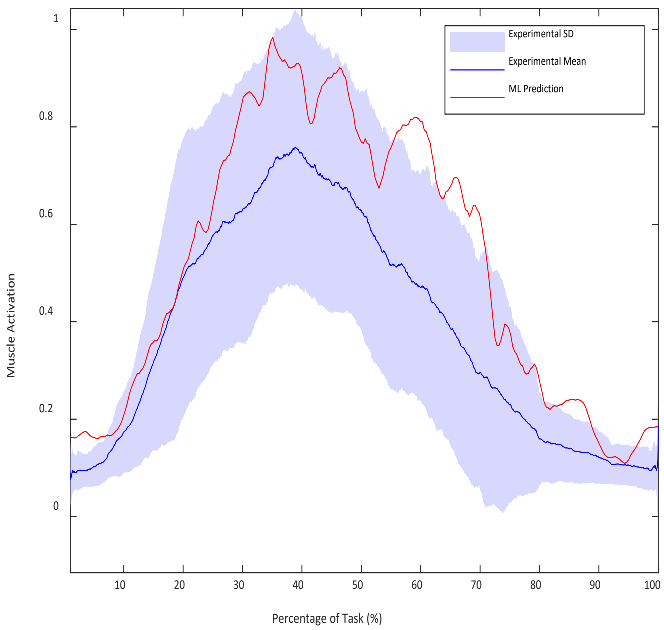


(b)

Figure 11: Model 4: Muscle activation profile prediction using ANN with kinematics inputs for biceps: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire biceps dataset of all 10 subjects.



(a)



(b)

Figure 12: Model 4: Muscle activation profile prediction using ANN with kinematics inputs for deltoid: (a) Comparison with experimental EMG from the 10th subject, and (b) Comparison with the entire deltoid dataset of all 10 subjects.

Tables 2 and 3 summarize statistical accuracy of the prediction. Table 3 shows the Pearson's correlation coefficients for muscle activation prediction using different machine learning models.

Table 2 RMSE between muscle activation predicted by machine learning algorithms and experimentally collected EMG activation of the 10th subject.

	Biceps	Deltoid
Model 1	0.1453	0.2122
Model 2	0.1344	0.1866
Model 3	0.1310	0.1922
Model 4	0.2849	0.1451

Table 3 Pearson's correlation coefficient between muscle activation predicted by machine learning algorithms and experimentally collected EMG activation of the 10th subject

	Biceps	Deltoid
Model 1	0.8586	0.8985
Model 2	0.9000	0.8559
Model 3	0.9299	0.9073
Model 4	0.8790	0.9239

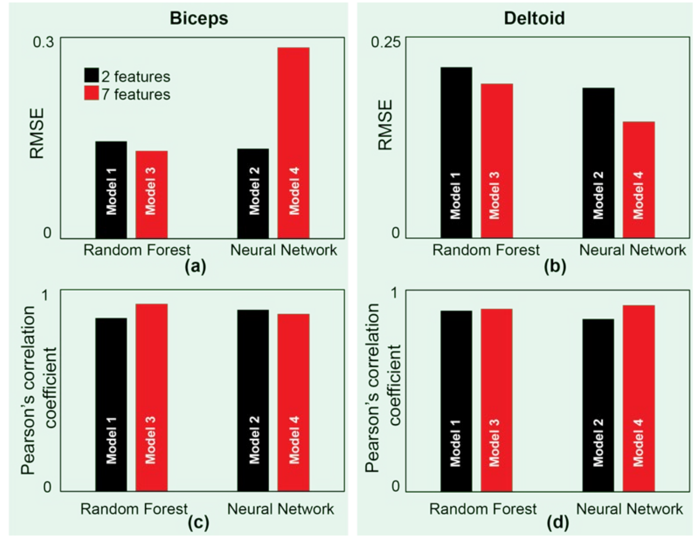


Figure 13: Statistical comparison of different machine learning models in terms of RMSE and Pearson's correlation coefficient. (a) RMSE of biceps, (b) RMSE of deltoid, (c) Pearson's correlation coefficient of biceps, and (d) Pearson's correlation coefficient of deltoid

When incorporating joint kinematics from 5 DoFs in the model besides height and weight, the correlation coefficients improve for both algorithms. The RF achieved a correlation coefficient of 0.9299, and the ANN achieved a correlation coefficient of 0.8790. In terms of RMSE, the best accuracy is achieved for Model 3, where network is trained with RF and input features are height, weight, and joint kinematics.

Similar to biceps muscle, the correlation coefficients are higher for deltoid when using the combined set of input features compared to using only height and weight. When considering height and weight, the RF achieves a correlation coefficient of 0.8985, and the ANN algorithm achieved a correlation coefficient of 0.8559. These coefficients indicate a moderately strong positive linear relationship between height and weight and deltoid muscle activation. When incorporating joint kinematics from 5 DoFs in the model along with height and weight, the correlation coefficients further improve for both algorithms. The RF achieved a correlation coefficient of 0.9073, and the ANN achieved a correlation coefficient of 0.9239.

Even though all the methods with machine learning achieved over 80% accuracy, there are some limitations researchers should be concerned about when applying machine learning algorithms to predict muscle activations. The dataset used here to train the networks consists of only male subjects, which means the trained network will not be able to make prediction for females. Additionally, to generalize population, patient specific data need to be added to the training dataset. It is recommended to add more diverse data sets with patient specific population to potentially enhance the network performance.

This pilot study only shows prediction based on a few features. Height and weight are not the only features that can be enough to predict muscle activation. Therefore, joint kinematics are added for making predictions. But still, it is important to figure out which features are important to make the prediction. A single task can be done with different strategies, for example, for the reaching task it is noticed during data collection, some subjects start moving the shoulder before bending elbow and vice versa. The variation in the strategy brings variation in the activation prediction. Maybe incorporation of the strategy as an additional input feature will improve networks' prediction capability. Therefore, future research should figure out the list of important features to train neural networks.

The complexities in both RF and ANN could contribute to the deviation of the results. For RF networks, the number of trees is an important parameter that was tuned to get better prediction. The final number that is set is 100 for the number of trees. However, with additional or a totally new dataset, this parameter may need to be tuned again. For ANN, important hyperparameters include the number of hidden layers and number of neurons. A sensitivity analysis regarding the number of hidden layers and neurons is a crucial step. A higher number of hidden layers can provide better estimation, but too many can lead to overfitting to the current dataset, meaning the model may perform well on the training dataset but fails to predict new unseen data. It occurs more with small dataset; therefore, machine learning algorithms require more data to overcome overfitting and provide more efficient prediction.

A drawback to the use of machine learning algorithms is that they require higher computational power compared to musculoskeletal modeling. The simulations in this article were carried out with High Performance Computing Center (HPCC) at Texas Tech University, which has a total core of 16,812 in the Quanah server. Among them 72 cores were assigned to this calculation. RF worked a lot faster than ANN. Each simulation using RF took around 5 to 10 minutes whereas the ANN required 6 to 8 hours per computation.

Another drawback of using joint kinematics and anthropometric data to predict unmeasured muscle activations is that they do not directly account for external loads or muscle strength. Future models could be enhanced by incorporating task-specific load parameters and subject-specific strength measures, such as maximum voluntary contractions using instruments like dynamometer, which would lead to more physiologically informed predictions.

Two different machine learning algorithms are used for this study: RF and ANN. But other algorithms may need to be explored. For example, like RF, SVM is another classifier algorithm that can be used. CNN is another machine learning algorithm that is being consistently used for image processing. The EMG dataset here can also be represented like two-dimensional image data, as muscle activation in one axis and time data in another. Future research can be carried out to check if CNN will be able to comprehend the EMG data and predict muscle activation.

4. CONCLUSION

This paper explored a new approach to predict muscle activation by implementing different machine learning algorithms. In this case, muscle activation data obtained for Task 3 for all 10 subjects were used to train two different machine learning algorithms: RF and ANN. By varying the number of features and model specific hyper parameters, the machine learning approaches proved to be more efficient than the musculoskeletal model-based approach. The machine learning approaches were generally used for upper extremity here, but they can be implemented for lower extremity as well as other human motions. For example, muscle activations obtained from spine muscles can also be used to train the networks to predict activations based on different postures. This will help evaluate muscle forces around the spine and quantify lower back pain. Other machine learning algorithms need to be evaluated. Like RF, SVM is another classifier algorithm that can be used. CNN is another machine learning algorithm consistently being used for image processing. The EMG dataset here can also be represented similarly to two-dimensional image data, as its muscle activation in one axis and time in the other axis. Future research can be carried out to check if CNN or LSTM and other machine learning approaches can comprehend the EMG data and predict muscle activation.

ACKNOWLEDGEMENTS

This study was partially supported by the National Science Foundation (Award #: 2014278). The authors acknowledge the HPCC at Texas Tech University for providing computational resources that have contributed to the research results reported within this paper. URL: http://www.hpcc.ttu.edu.

REFERENCES

[1] Lanzoni, Daniel, et al. "Design of customized virtual reality serious games for the cognitive rehabilitation of retrograde amnesia after brain stroke." Journal of Comp. and Information Science in Engineering 22.3 (2022).

[2] Anderson, Frank C., and Marcus G. Pandy. "Static and dynamic optimization solutions for gait are practically equivalent." Journal of Biomech. 34.2 (2001): 153-161.

[3] Lloyd, D. G., and T. F. Besier. "An EMG-driven musculoskeletal model for estimation of the human knee joint moments across varied tasks." Journal of Biomechanics 36 (2003): 765-776.

[4] Thelen, Darryl G., Frank C. Anderson, and Scott L. Delp. "Generating dynamic simulations of movement using computed muscle control." Journal of Biomechanics 36.3 (2003): 321-328.

[5] Buchanan, Thomas S., et al. "Estimation of muscle forces and joint moments using a forward-inverse dynamics model." Medicine and Science in Sports and Exercise 37.11 (2005): 1911.

[6] Shao, Qi, et al. "An EMG-driven model to estimate muscle forces and joint moments in stroke patients." Computers in Biology and Medicine 39.12 (2009): 1083-1088.

[7] Zonnino, Andrea, and Fabrizio Sergi. "Model-based estimation of individual muscle force based on measurements of muscle activity in forearm muscles

during isometric tasks." IEEE Transactions on Biomedical Engineering 67.1 (2019): 134-145.
[8] Meyer, Andrew J., Carolynn Patten, and Benjamin J. Fregly. "Lower extremity EMG-driven modeling of walking with automated adjustment of musculoskeletal geometry." PloS One 12.7 (2017): e0179698.
[9] Kian, Azadeh, et al. "Static optimization underestimates antagonist muscle activity at the glenohumeral joint: A musculoskeletal modeling study." Journal of Biomechanics 97 (2019): 109348.
[10] Heintz, Sofia, and Elena M. Gutierrez-Farewik. "Static optimization of muscle forces during gait in comparison to EMG-to-force processing approach." Gait & Posture 26.2 (2007): 279-288.
[11] Seth, Ajay, et al. "OpenSim: simulating musculoskeletal dynamics and neuromuscular control to study human and animal movement." PLoS Computational Biology 14.7 (2018): e1006223.
[12] Saul, Katherine R., et al. "Benchmarking of dynamic simulation predictions in two software platforms using an upper limb musculoskeletal model." Computer Methods in Biomechanics and Biomedical Engineering 18.13 (2015): 1445-1458.
[13] Doorenbosch, Caroline AM, and Jaap Harlaar. "A clinically applicable EMG–force model to quantify active stabilization of the knee after a lesion of the anterior cruciate ligament." Clinical Biomech. 18.2 (2003): 142-149.
[14] Kellis, Eleftherios, and V. Baltzopoulos. "The effects of antagonist moment on the resultant knee joint moment during isokinetic testing of the knee extensors." European Journal of Applied Physiology and Occupational Physiology 76.3 (1997): 253-259.
[15] Amarantini, David, and Luc Martin. "A method to combine numerical optimization and EMG data for the estimation of joint moments under dynamic conditions." Journal of Biomechanics 37.9 (2004): 1393-1404.
[16] Liu, M. M., Herzog, W., and Savelberg, H. HCM. "Dynamic muscle force predictions from EMG: an artificial neural network approach." Journal of Electromyography and Kinesiology 9.6 (1999): 391-400.
[17] Tahmid, S., Font Llagunes, J. M., and Yang, J., Upper Extremity Muscle Activation Pattern Prediction through Synergy Extrapolation and Electromyography-Driven Modeling, *ASME Journal of Biomechanical Engineering*, 146(1):011005 (10 pages), 2024.
[18] Tahmid, S., and Yang, J., “Simultaneous prediction of multiple unmeasured muscle activations through muscle synergy analysis.” *ASME Journal of Biomechanical Engineering*, 147(3): 031002 (9 pages).
[19] Arjmand, Navid., et al. "Relative performances of artificial neural network and regression mapping tools in evaluation of spinal loads and muscle forces during static lifting." Journal of Biomechanics 46.8 (2013): 1454-1462.
[20] Cecchini, Giulio, et al. "Neural networks for muscle forces prediction in Cycling." Algorithms 7.4 (2014): 621-634.
[21] Dao, Tien Tuan. "From deep learning to transfer learning for the prediction of skeletal muscle forces." Medical & Biological Eng. & Computing 57 (2019): 1049-1058.
[22] Rane, Lance, et al. "Deep learning for musculoskeletal force prediction." Annals of Biomedical Engineering 47 (2019): 778-789.
[23] Gonzalez-Vargas, Jose, et al. "A predictive model of muscle excitations based on muscle modularity for a large repertoire of human locomotion conditions." Frontiers in Computational Neuroscience 9 (2015): 114.
[24] Jiang, Y., et al. "Shoulder muscle activation pattern recognition based on sEMG and machine learning algorithms." Computer Methods and Programs in Biomedicine 197 (2020): 105721.
[25] Trigili, Emilio, et al. "Detection of movement onset using EMG signals for upper-limb exoskeletons in reaching tasks." Journal of Neuroengineering and Rehabilitation 16 (2019): 1-16.
[26] Jonic, S., and Popovic, D. "Machine learning for prediction of muscle activations for a rule-based controller." Proc. of the 19th Annual International Conference of the IEEE Engineering in Medicine and Biology Society. Magnificent Milestones and Emerging Opportunities in Medical Engineering (Cat. No. 97CH36136). Vol. 4. IEEE, 1997.
[27] Mohr, M., et al. "Classification of gait muscle *activation patterns according to knee injury history using a* support vector machine approach." Human Movement Science 66 (2019): 335-346.
[28] Konrad, P. "The ABC of EMG: A practical introduction to kinesiological electromyography." (2005): 30-5.
[29] Stegeman, D., and Hermens, H. "Standards for surface electromyography: The European project surface EMG for non-invasive assessment of muscles (SENIAM)." Enschede: Roessingh Research and Development 10 (2007): 8-12.
[30] Olney, S. J., and Winter, D. A. "Predictions of knee and ankle moments of force in walking from EMG and kinematic data." Journal of Biomech. 18.1 (1985): 9-20.
[31] Breiman, Leo. "Random Forests." Machine Learning 45 (2001): 5-32.
[32] Liaw, A., and Wiener, M. "Classification and regression by randomForest." R News 2.3 (2002): 18-22.
[33] Levenberg, K. "A method for the solution of certain non-linear problems in least squares." Quarterly of Applied Mathematics 2.2 (1944): 164-168.
[34] Marquardt, D.W. "An algorithm for least-squares estimation of nonlinear parameters." Journal of the Society for Industrial and Applied Mathematics 11.2 (1963): 431-441.
[35] Wilamowski, B. M., and Yu, H. "Improved computation for Levenberg–Marquardt training." IEEE Transactions on Neural Networks 21.6 (2010): 930-937.

