Unsupervised, Semi-supervised Interactive Force Estimations during pHRI via Generated Synthetic Force Myography Signals

U. Zakia¹, Arnab Barua², Xianta Jiang², Senior Member, IEEE, and C. Menon¹,³, Senior Member, IEEE
¹Menrva Research Group, Schools of Mechatronic systems Engineering and Engineering Science, Simon Fraser University, Metro Vancouver, BC, Canada
²Memorial University of Newfoundland, 230 Elizabeth Ave, St. John's, NL A1C 5S7, Canada
³Biomedical and Mobile Health Technology Laboratory, Department of Health Sciences and Technology, ETH Zurich, Switzerland

Corresponding author: C. Menon (email: cmenon@sfu.ca, phone: +1 778 782 9338; fax: +1 778 782 7514).

This research was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC), the Canadian Institutes of Health Research (CIHR) and the Canada Research Chair (CRC) program.

ABSTRACT — Recognizing applied hand forces using force myography (FMG) biosignals requires adequate training data to facilitate physical human-robot interactions (pHRI). But in practice, data is often scarce, and labels are usually unavailable or time consuming to generate. Synthesizing FMG biosignals can be a viable solution. Therefore, in this paper, we propose for the first time a dual-phased algorithm based on semi-supervised adversarial learning utilizing fewer labeled real FMG data with generated unlabeled synthetic FMG data. We conducted a pilot study to test this algorithm in estimating applied forces during interactions with a Kuka robot in 1D-X, Y, Z directions. Initially, an unsupervised FMG-based deep convolutional generative adversarial network (FMG-DCGAN) model was employed to generate real-like synthetic FMG data. A variety of transformation functions were used to observe domain randomization for increasing data variability and for representing authentic physiological, environmental changes. Cosine similarity score and generated-to-input-data ratio were used as decision criteria minimizing the reality gap between real and synthetic data and helped avoid risks associated with wrong predictions. Finally, the FMG-DCGAN model was pretrained to generate pseudo-labels for unlabeled real and synthetic data, further retrained using all labeled and pseudo-labeled data and was termed as the self-trained FMG-DCGAN model. Lastly, this model was evaluated on unseen real test data and achieved accuracies of 85%>R²>77% in force estimation compared to the corresponding supervised baseline model (89%>R²>78%). Therefore, the proposed method can be more practical for use in FMG-based HRI, rehabilitation, and prosthetic control for daily, repetitive usage even with few labeled data.

INDEX TERMS FMG signal, applied force estimation, unsupervised learning, generative adversarial networks, domain randomization, transformation functions, similarity score, semi-supervised learning, self-training, transfer learning,

I. INTRODUCTION

Force myography (FMG) is a non-invasive, wearable technology that can utilize force sensing resistors (FSRs) to detect resistance changes when pressure is applied. Signals captured by donning an FMG band around a limb on the upper or lower extremities can detect muscle contractions during daily activities via data-driven models [1, 2]. The FMG technique can be used in similar applications similar to the conventional surface electromyography (sEMG) technique that reads electrical current of the underlying muscles in action [3-6]. Measured electrical signals of underlying muscles using sEMG technique are faint and requires costly signal processing units and skin preparation for electrode placements [7]. Similarly, the FMG technique can be low cost with off-the-shelf FSRs, repeatable like sEMG, electrically robust, and requires minimal signal processing [1]. FMG technology has been studied in applications like sEMG such as gesture recognition,
prosthetic control, activities of daily life, rehabilitation, and HMI (human machine interaction) [4], [8-12].

Estimating isometric or dynamic hand force from wrist, forearm, and upper arm FMG signals were found promising [13]. Although a load cell or a force/torque (FT) sensor can read forces precisely, it is bulky, requires complex signal processing units and is difficult to move freely when used as wearable on a human body. On the other hand, an FMG band is easy to wear, allows better mobility when donned and thus human bio-signal can be included in control system design. Hence, FMG-based physical human-robot interactions (pHRI) was implemented in a recent study where human participants applied hand forces in dynamic motions during interactions with a linear robot using traditional machine learning (ML) techniques [14]. The study showed that intra-session models could estimate applied forces well (accuracies of 94%-R²>82%) where training and evaluation were carried out in the same session, had similar feature distributions, and were labeled properly. Recently, adapting domain adaptation, domain generalization and cross-domain generalization techniques enabled an SVR, and a convolutional neural network (CNN) model trained with labeled FMG data from multiple pHRI source domains to recognize applied hand forces in unknown pHRI target scenarios that the model never seen before [15-17]. Finetuning allowed required periodic calibrations to recognize any unseen, instantaneous signals. These studies relied on supervised learning technique where large pool of adequate, labeled training data were available.

However, in real-world pHRI applications, obtaining enough training data, having more participants, or labelling all data are not always possible. In the case of unlabeled data, unsupervised learning can be an option which does not need labeled data. It learns latent feature distributions and may generate labels based on the learning process. Also, semi-supervised or weakly supervised learning can reduce the dependencies on labeled training data where only a few labeled training datasets are available. In recent pHRI studies, there is a growing interest to use limited amounts of labeled data with large quantities of unlabeled data for realistic predictions [18-20]. Semi-supervised learning can be useful in such cases that utilizes few labeled training datasets to achieve similar performance like supervised learning with fully labeled data set. For scenarios where training data hard to collect or there is no previous data from other related sources are available, synthetic data could be a favorable alternative. The generative adversarial network (GAN), originally proposed by Ian Goodfellow [21] has been a proven technique in synthetic image generation, but not many bio signals are synthesized using this architecture. As a bio-signal, an FMG signal is transient, individual-specific, and each session’s data are affected by control factors such as sensor position shifts, limb motions, and postures during activities. There are serious concerns if a reality gap occurs when the generated synthetic data do not follow the real domain di is more challenging to generate synthetic FMG bio signals to control HRI because of higher risks associated with wrong predictions and hence, needs to be investigated.

A domain randomization approach can be adapted to mitigate the reality gap [22, 23]. This approach can make the simulated data as diverse as possible so that the real data would appear as another variation to the model [23, 24]. Data augmentation is a form of domain randomization that is applied on the collected data and performs certain refinements or transformations to render real-like environmental changes. Adapting carefully selected transformations that represent certain control factors affecting FMG signals can introduce wide variations in data and refine the synthetic data generation through the GAN making them more realistic. For real-like synthetic data generation, we need fake data close enough to real data while maintaining some individuality among the two datasets. In machine learning, Euclidean distance, Manhattan distance, Minkowski distance, and Cosine similarity scores are usually used for measuring similarities between two matrices. To compare real data with the generated synthetic data, implementing a metric is useful to quantify the similarity and to keep the reality gap at some minimum level.

In this study, we propose a two-phased “unsupervised, self-trained FMG-based deep convolutional GAN (unsupervised, self-trained FMG-DCGAN)” algorithm. An unsupervised adversarial learning approach was used for understanding the latent feature distributions followed by semi-supervised learning to evaluate intra-session test data. For the investigation, pHRI application between one participant and a Kuka LBR IIWA 14 robot in 1D (X, Y and Z dimensions) was considered, as shown in Fig. 1. In its initial phase, an unsupervised FMG-based deep convolutional generative adversarial network (unsupervised FMG-DCGAN) model was implemented where both the generator and the discriminator were convolutional neural networks (CNNs). A set of transformation functions for data augmentation was used to implement domain randomization adding variabilities in the synthetic data. In our study, we selected cosine similarity score to find the similarities between the real and fake data because it is useful in determining how similar the two data are irrespective of their size [25]. Employing cosine similarity as a benchmark score, each transformation function was investigated in generating real-like synthetic data and a GAN model was saved for the next phase. In the final phase, each GAN model generated with different transformation functions was evaluated. The discriminator of each GAN with weights from initial phase was pretrained with a few real labeled training data and later was used to generate pseudo-labels for both unlabeled real training data and unlabeled synthetic data. The model was further trained with all real training (labeled and pseudo-labeled) and generated synthetic data (pseudo-labeled) and was called the self-trained FMG-DCGAN model. Finally, this model was evaluated on test data in estimating applied
instantaneous forces during interactions. Using the cosine similarity score and generated synthetic data volume as decision criteria, the best model with the optimal transformation function was identified. Major contributions in this study were:

1. investigating real-like synthetic FMG data generation using domain randomization via data augmentation, and obtaining the appropriate technique in this case,
2. investigating unsupervised and semi-supervised learning techniques where training data were scarce or large volumes of data were not available, and
3. implementing these techniques in challenging FMG-based pHRI using one FMG forearm band.

The rest of this article is organized as follows. Section II describes the proposed method, experimental setup and material, and the protocol followed in data collection. Performance evaluations are stated in Section III and discussed in Section IV. The article concludes in Section V.

II. MATERIALS AND METHODS

A. PHRI SETUP FOR DATA COLLECTION

For pHRI, a participant interacted with a 6-DoF Kuka robot (Kuka LBR iIWA 14 R820, KUKA Robotics, Augsburg, Germany) in 1D space, as shown in Fig. 2 (a). The robot was mounted securely on top of a table. As a customized end-effector, a cylindrical gripper was attached to the flange of the robot via a custom-made adapter. A 6-axis force/torque sensor (NI DAQ 6259, National Instruments, Austin, TX, US) was placed between the cylindrical gripper and the adapter to read applied grasp forces in motions as the true label generator. The gripper was kept oriented at \([0, \pi, 0]\) for an easy grasp, as shown in Fig. 2 (f). A custom-made 16-channel forearm FMG band using 16 FSRs (TPE 502C, Tangio Printed Electronics, Vancouver, Canada) was used for capturing muscle contractions during interactive activities (NI DAQ 6341, National Instruments, Austin, TX, US), as shown in Fig. 1 (g) [1]. An external desktop PC (Intel Core i7 processor and Nvidia GTX-1080 GPU) was used to connect all devices with the Kuka Sunrise controller. Force control was implemented for compliant collaboration between the robot and the applied external force so that the robot could move in space proportional to the applied forces and dynamic motions. Matlab (Matlab, Mathworks, Natick, MA, USA) scripts were written with Kuka Sunrise Toolbox [26] and V-REP robot simulator to control the robot.

B. PROTOCOL

One healthy adult (P1) participated in this study with his written consent as approved by the Office of Research Ethics, Simon Fraser University, British Columbia, Canada. A 16 channel FMG band was donned on the participant’s dominant forearm and was not removed during a session. For pHRI with the Kuka robot, P1 stood steadily in position in front of the robot, grasped the cylindrical gripper and applied interactive forces in simple dynamic 1D motions, as shown in Fig. 1a. Separate sessions were conducted to interact in 1D ‘X’, ‘Y’ and ‘Z’ dimensions (Fig. 2.b-d). In 1D-X, dynamic motion
during applied force was front and back and vice versa (displacement in X: 0.4m); in 1D-Y, motion was from right to left and vice versa (displacement in Y: 0.55m); while in 1D-Z, motion was moving the gripper up and down and vice versa (displacement in Z: 0.45m). These motions required arm flexion, extension, abduction, and adduction, were continued for a certain time (approximately 90s of interactions without fatigue) while termed hereinafter as a 'repetition'. Dynamic motions during the interactions were confined in a 6-axis rectangular plane set through Matlab script [Fig. 2(e)].

C. DATASET
For each interaction session in 1D-X, Y, Z directions, 2 repetitions of labeled sample data, $D_C \in \{X_{C1}, Y_{C1}\}$ U $D_{C2} \in \{X_{C2}, Y_{C2}\}$ were collected for training (source distribution $D_C$) and 1 repetition of labeled test data ($D_T \in \{X_{T1}, Y_{T1}\}$) for evaluation purpose (target distribution $D_T$). All distributions of $D_C$ and $D_T$ had feature spaces of $X \in \mathbb{R}^{N \times M}$ where N = number of samples and M = 16 FMG channels. Approximately, a total of 4,000×16 FMG samples were collected with corresponding labels, while 3,200×16 FMG samples were used for training and 600×16 FMG samples were used for testing. The source data were used for training the GAN and to calibrate the FMG signals for both generating the synthetic data and training the proposed model.

D. UNSUPERVISED, SEMI-SUPERVISED SELF-TRAINED FMG-DCGAN ALGORITHM
The proposed algorithm was developed in two consecutive phases where the final phase was dependent on the outcome of its initial phase, as shown in Fig. 3. An unsupervised FMG-DCGAN architecture was introduced in phase I for real-like synthetic data generation. In phase II, a self-trained FMG-DCGAN model was investigated for estimating instantaneous applied forces.

1) Phase I: Generating synthetic FMG data
The unsupervised FMG-DCGAN architecture had a generator (model G) and a discriminator (model D) where convolutional and convolutional-transpose layers were implemented respectively, similar to the DCGAN architecture [27]. The FMG-DCGAN model was proposed for synthetic data ($D_{SJ} \in \{S_{SJ}\}$) generation from unlabeled real FMG data ($D_C \in \{X_{C1}, X_{C2}\}$) of 3200×16 samples. A
few techniques were adapted in this initial phase to reduce the reality gap between the real and the synthetic data as much as possible and is discussed below.

- **Domain Randomization via data augmentation**
  A variety of transformation functions $t(x)$s were generated that could potentially mimic situations where sensor position shifts, unused sensors or sensors that capture low signals could occur during the data collection process when an FMG band was donned on upper extremities. These are shown in Fig. 4 and discussed below.

  a. **Temporal cutout**, $t(x_{el})$: a random contiguous section (consecutive any 3 channels) of the time-series signal (every 50ms of cutout window) was replaced with zeros.

  b. **Sensor dropout**, $t(x_{sd})$: any random three FMG channels were set to zeros in every 10ms input of the time-series signal.

  c. **Spatial shift**, $t(x_{ss})$: 16 FMG channels data were right-shifted by 1 for every 10ms input of the time-series signal.

  d. **Noise**, $t(x_{n})$: random zero-mean gaussian noise (3200, 100) was generated and used as input.

  e. **Signal mixing**, $t(x_{sm})$: random noise $t(x_{n})$ was used in conjunction with $t(x_{el}), t(x_{sd}),$ and $t(x_{ss})$ to produce mixed signals $t(x_{sm-elic}), t(x_{sm-end}), t(x_{sm-ess})$, respectively.

The transformation functions took real data of image shape of $(1,16,1)$ as input, transformed the signal, and finally reshaped the transformed data as $T$ $(1,100,1)$ for input to model G. Padding with mean values of the corresponding 16 features was used for reshaping the transformed data. The other input was the noise signal $(3200, 100)$, which was also reshaped as $N$ $(1,100,1)$ before feeding into model G.

- **Cosine similarity score $\sigma(s)$**
  Cosine similarity score measures the cosine of the angle between two n-dimensional vectors $(D_c$ and $S_D)$ projected in a multi-dimensional space. Its range is between 0 to 1. It is defined as:

  $$\sigma(s) = \cos(D_c, S_D) = \langle D_c \cdot S_D \rangle / \left( \| D_c \| \| S_D \| \right) \quad (1)$$

  where $D_c, S_D$ is the dot product and $\| D_c \|^2 \| S_D \|$ is the cross product of two vectors $D_c$ and $S_D$. Higher values indicate more similarities between the vectors.

  - **The proposed FMG-DCGAN architecture**
    In this architecture, model G was engaged in generating fake FMG signals while model D was employed to learn the discriminative feature distributions of both real and fake signals and classify them accordingly. For model G, inputs were either noise signals $(N_S \in \{X_{S_1}\})$ or transformed FMG signals $(T_S \in \{X_{T_1}\}$ with shapes of $(1, 100, 1)$, or aggregated noise and transformed signals, as discussed in Section II.C.1. Model D received real FMG signals $(D_c \in \{X_{C_1}, X_{C_2}\})$ and
fake FMG signals/synthetic data \(D_{\text{SI}} \in \{S_{\text{SI}}\} \) generated by model G, where both inputs to model D were shapes of \((1,16,1)\). Only those generated fake signals were considered as synthetic data for the next phase that obtained higher cosine similarity scores \(\sigma(s)\) along with greater generated-to-input-data ratio \(\hat{\delta}\).

- **Model G architecture**
  Three successive convolutional 2D transposed (conv2DTranspose) layers [no. of filters: 128, 64, 1 and filter size: \((1,5)\), \((1,10)\), and \((1,5)\)] were implemented with strides of \((1,1)\), \((1,4)\), and \((1,2)\) and the ‘same’ padding. Each conv2DTranspose layer was followed by a batch normalization layer and a leaky Relu layer. For the final output of fake generated signal \(S_{\text{D}}\) of a shape of \((1,16,1)\), the tanh activation function was used. This architecture was used in 1D-X, Y and Z dimensions.

- **Model D architecture**
  Two convolutional (conv) blocks were implemented sequentially where each block had a conv2D layer followed by a leaky ReLU and a dropout layer with a rate of 0.3 to reduce overfitting. Convolutional 2D (conv2D) layers were implemented with strides of \((1,2)\) and the ‘same’ padding. The number of filters used in the conv blocks for 1D-X dimension was 32, 16, 1, while it was 128, 64, 1 in 1D-Y and Z dimensions with the same filter sizes of \((1,5)\), \((1,10)\), and \((1,5)\) in each dimension. The convolutional blocks were followed by three dense layers of 20, 10 and 1 neurons for 1D-X while only one dense layer of 1 neuron was used for 1D-Y, and Z. A sigmoid function was used to classify the real input and the generated signals.

- **GAN Loss & Optimization**
  Real data and fake data were labeled as 0 and 1 respectively to calculate losses of model D and model G using binary cross entropy (BCE) loss. The Adam optimizer with a learning rate (LR) of 1E-04 and Beta1, Beta2 = 0.9, 0.999 was used for both G and D models during training. An initial training was performed with model D. For training the FMG-DCGAN in generating real-like synthetic data, output from model D was expected to be 1 for real data and 0 for fake data. Therefore, the total loss for model D was calculated as the sum of the loss from the real data used in training model D and the loss from the synthetic data generated by model G. On the contrary, to maximize model G’s performance in convincing model D that the generated data were real, \(G_{\text{Loss}}\) was calculated and back propagated to improve G. Therefore, \(G_{\text{Loss}}\) had similar values with flipped labels. \(D_{\text{Loss}}\) and \(G_{\text{Loss}}\) were set up such that:

\[
D_{\text{Loss}} = \log(D(x)) + \log(1 - D(G(t(x))))
\]

\[
G_{\text{Loss}} = -\log \left(D(G(t(x)))\right)
\]

where, \(D(x)\) was the output from model D on an instance of training dataset \(x\) at time \(i\), \(G(t(x))\) was the generated data, \(D(G(t(x)))\) was the model D’s output on the generated data at instant \(i\), and \(t(x)\) was the transformed signal used as input to the Model G. The \(D_{\text{Loss}}\) and the \(G_{\text{Loss}}\) were back propagated to improve model D in discriminating better between real and fake signals and to improve model G in generating better quality signal that could fool model D, respectively. Both models were trained until they reached convergence, and diversified data were generated where fake data were as good
as real data by preventing mode collapse (1.5 > G_{loss}, 0.8 and 0.7 > D_{loss}, 0.5).

ii. Selecting optimal $t(s)$ via $\sigma(s)$ and $\delta$

Cosine similarity score was implemented to compare the real data with the generated fake data from model G. Each fake data was compared with all real data, $\sigma(s)$ was calculated, and the mean value of these scores was generated. Only those synthetic data were saved as future training data that obtained a score of $\sigma(s)$ > 0.8, i.e., similarity between fake data and real data would be at least 80%. Data was normalized for score comparison between the real data and the synthetic data, where each synthetic data was compared with all 3200 x 16 real samples, $D_C$. Transformation function that could generate more real-like synthetic data ($\sigma(s)$ > 0.8 with maximum $\delta$) was used to optimize the model performance. Each transformation function was implemented separately, synthetic data was generated, performance was evaluated, and a model was saved with its weights for the next phase.

2) Phase II: Self-trained FMG-DCGAN model

In this phase, calibration/target training data (few labeled real data, and a few unlabeled real data) and a large amount of generated real-like synthetic data (unlabeled) were used. This phase began by pretraining model D from phase I. A small amount of labeled source data ($D_{S1} \subset D_C$) and a large volume of unlabeled data (mainly synthetic data ($D_S$)) and few real data ($D_{R1} \subset D_C$) were used in self-training. Finally, the model was evaluated on target test data, $D_T$. For finetuning, a fully connected layer was used to replace the last layer of model D for regressing force estimations. Several steps were followed in generating a self-trained FMG-DCGAN model using an Adam optimizer with learning rate, LR = 1E-04 and epoch, E = 500 to calculate mean squared error (MSE) losses, as discussed below.

i. Pretraining Model D via Transfer Learning

- In step 1, the saved model D from phase I was finetuned with a few labeled ‘target training data’ or ‘calibration data’ ($D_{C1} \in \{X_{C1}, Y_{TC1}\}$). This helped the model to adapt learning real target distributions and converge quickly and was termed hereafter as the ‘pretrained FMG-DCGAN’ model.

- The pretrained FMG-DCGAN model was used in step 2 to generate pseudo labels $Y_{PC}$ for the unlabeled real data, $D_1 \in \{X_{C2}\}$ and pseudo labels $Y_{PS}$ for unlabeled synthetic data, $D_2 \in \{S_D\}$.

- In step 3, the pretrained model was further trained with all true labeled and pseudo labeled real and synthetic data ($D_{C1} \in \{X_{C1}, Y_{TC1}\}$ U $D_{S1} \in \{X_{C2}, Y_{PC}\}$ U $D_{S2} \in \{S_D, Y_{PS}\}$) and was termed as the self-trained FMG-DCGAN model.

ii. Estimating Interactive Force with Self-Trained FMG-DCGAN model

The self-trained FMG-DCGAN model was used to predict interactive force on target test data ($D_T \in \{X_T, Y_{T1}\}$) of instantaneous FMG signals in every 10 ms window size.

At an instant time, $t$, target test input FMG signals $\{X_T\}$ were used for validation purpose. Hence, for target task $T_1$ ($\{Y_{T1}, f(\cdot)\}$, a predictive function $f(\cdot)$ was used to estimate interaction forces in 1D such that, $f: X_T \rightarrow F'_t$.

\[
f_x(c) = F'_t = y, (X_T, f, \phi)
\]

(4)

The model attempted to find the best parameter space $\phi$ from the proposed parameter set $\gamma$ which was determined by computing the loss function using the true force label space $Y_{T1}$:

\[
\phi = L(F'_t - F_{xt}) = \arg \min_{\phi} \sum_{t=1}^{T} (F'_t - Y_{T1})^2
\]

(5)

3) Evaluating Force estimations

For performance evaluation of force estimation, root mean square error (RMSE) was used to calculate the average squared difference between the estimated and the real value. RMSE for a single observation was:

\[
RMSE = \frac{1}{w} \sum_{t=1}^{T} (F'_t - Y_{T1})^2
\]

(6)

where $w$ was the number of responses, $Y_{T1}$ was the target force labels, and $F'_t$ was the predicted forces at an instant $t$.

RMSE was calculated as the fraction of RMSE to the observed range of the measured data:

\[
RMSE = \frac{RMSE}{Y_{Tmax} - Y_{Tmin}}
\]

(7)

Co-efficient of determination ($R^2$) was used to evaluate model performance in force estimation to determine the dependencies of the dependent variable $Y$ on the independent variables $X$. It indicated how well the regression model performed in fitting test data such that:

\[
R^2 = 1 - \frac{SS_{res}}{SS_{tot}}
\]

(8)

where, $SS_{res}$ was the residuals sum of squares that measured variance in residual data not explained by the model and $SS_{tot}$ was the sum of totals that measured variations present in data observed by the model. $R^2$ values varied between 0 and 1.

III. RESULTS

The performance of the proposed algorithm was evaluated using the mean cosine similarity score ($\bar{\sigma}(s)$) and the generated-to-input-data ratio ($\delta$) in phase I, while $R^2$, RMSE and NRMSE were used in phase II (using Python 3.7 and Tensorflow with GTX1060). For hPIR in 1D motions in X, Y and Z dimension, separate FMG-DCGAN model was implemented to generate synthetic data and evaluated on corresponding target distributions. Each FGM-DCGAN model (both Model D and G) with unique $t(x)$ were saved as h5 files with their weights for future use in phase II. To compare the proposed model performance, a baseline CNN model using model D architecture employing supervised learning was generated using the same calibration/target training data, DC ($\{X_{C1}, Y_{TC1}\}$ U $\{X_{C2}, Y_{TC2}\}$) with true labels (3200 x 17 samples) and was evaluated on same target test data, $D_T$ (600 x 16 samples). Performance evaluations of 1D-X, Y and Z are reported in Table I, II and III, respectively. Approximately $E < 33$ epochs with a runtime of $t(E) < 0.35$.
seconds were required to generate synthetic data in each dimension.

A. pHRi in 1D-X

In phase I, transformation functions, \( t(x_c), t(x_d), t(x_0), \) and \( t(x_s) \) generated unlabeled synthetic data \( S_0 \) in the ranges of 2300×16 to 3100×16 samples with \( \sigma(s) > 88\% \). In the case of signal mixing, \( t(x_m-n_c), t(x_m-n_0), \) and \( t(x_m-n_s) \) generated synthetic data in the range of 4600×16 to 6300×16 samples.

### TABLE I  Model Performance in Force estimations during pHRi in 1D-X

| pHRi       | \( T(x) \) | Input for GAN training (unlabeled) | Generated synthetic FGM data, \( S_0 \times 16 \) | Generated-to-input-data ratio, \( \delta \) | Cosine Similarity Score, \( \sigma(s) \) | Calibration/Target training data | Target test data | \( R^2 \) | NRMSE | RMSE |
|------------|------------|-----------------------------------|-----------------------------------------------|---------------------------------------------|-------------------------------------|-------------------------------|----------------|--------|-------|------|
| 1D-X       | \( t(x_0) \) | Temporal cut out                   | Unlabeled real FGM data \((D_0 \in \{X_c, X_c, X_0\})\) | 2331×16 samples                             | 0.73                                 | 91.76\%                        | Labeled real data \((D_0 \in \{X_c, Y_{cs}\})\) \(+\) pseudo-labeled real data \((D_0 \in \{X_0, Y_{cs}\})\) | 0.1479          | 0.1702 | 0.1657 |
|            | \( t(x_d) \) | Sensor dropout                     |                                              | 2915×16 samples                             | 0.91                                 | 88.13\%                        |                                              | 0.1481          | 0.1590 | 0.1637 |
|            | \( t(x_0) \) | Spatial shift                      |                                              | 3196×16 samples                             | 0.99                                 | 90.76\%                        |                                              | 0.1509          | 0.1647 | 0.1681 |
| Noise      | \( t(x_0) \) |                                              |                                              | 3118×16 samples                             | 0.97                                 | 92.09\%                        |                                              | 0.1637          | 0.3121 | 3.41N  |
| Noise with | \( t(x_0) \) | Temporal cut out                   |                                              |                                              | 0.93                                 | 90.76\%                        |                                              | 0.1702          | 3.47N  |
|            | \( t(x_d) \) | Sensor dropout                     |                                              |                                              | 0.93                                 | 90.76\%                        |                                              | 0.1702          | 3.47N  |
|            | \( t(x_s) \) | Spatial shift                      |                                              |                                              | 0.73                                 | 92.18\%                        |                                              | 0.1702          | 3.47N  |
| Noise      | \( t(x_s) \) |                                              |                                              |                                              | 0.99                                 | 88.50\%                        |                                              | 0.1702          | 3.47N  |

Baseline CNN model (supervised learning with model D arch.)

81.72\% 1392 2.89N

### TABLE II  Model Performance in Force estimations during pHRi in 1D-Y

| pHRi       | \( T(x) \) | Input for GAN training (unlabeled) | Generated Synthetic FGM data, \( S_0 \times 16 \) | Generated-to-input-data ratio, \( \delta \) | Cosine Similarity Score, \( \sigma(s) \) | Calibration/Target training data | Target test data | \( R^2 \) | NRMSE | RMSE |
|------------|------------|-----------------------------------|-----------------------------------------------|---------------------------------------------|-------------------------------------|-------------------------------|----------------|--------|-------|------|
| 1D-Y       | \( t(x_0) \) | Temporal cut out                   | Unlabeled real FGM data \((D_0 \in \{X_c, X_c, X_0\})\) | 3129×16 samples                             | 0.977                                | 85.68\%                        | Labeled real data \((D_0 \in \{X_c, Y_{cs}\})\) \(+\) pseudo-labeled real data \((D_0 \in \{X_0, Y_{cs}\})\) \(+\) pseudo-synthetic data \((D_0 \in \{S_0, Y_{cs}\})\) | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_d) \) | Sensor dropout                     |                                              | 2358×16 samples                             | 0.736                                | 84.52\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_0) \) | Spatial shift                      |                                              | 2388×16 samples                             | 0.750                                | 85.63\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
| Noise      | \( t(x_0) \) |                                              |                                              | 3121×16 samples                             | 0.970                                | 83.98\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
| Noise with | \( t(x_0) \) | Temporal cut out                   |                                              |                                              | 0.935                                | 84.93\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_d) \) | Sensor dropout                     |                                              |                                              | 0.712                                | 81.59\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_s) \) | Spatial shift                      |                                              |                                              | 0.712                                | 81.59\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
| Noise      | \( t(x_s) \) |                                              |                                              |                                              | 0.988                                | 84.78\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |

Baseline CNN model (supervised learning with model D arch.)

78.155\% 0.1476 5.55N

### TABLE III  Model Performance in Force estimations during pHRi in 1D-Z

| pHRi       | \( T(x) \) | GAN tr. FGM Data (unlabeled) | Generated Synthetic FGM data, \( S_0 \times 16 \) | Generated-to-input-data ratio, \( \delta \) | Cosine Similarity Score, \( \sigma(s) \) | Calibration/Target training data | Target Test Data | \( R^2 \) | NRMSE | RMSE |
|------------|------------|-------------------------------|-----------------------------------------------|---------------------------------------------|-------------------------------------|-------------------------------|----------------|--------|-------|------|
| 1D-Z       | \( t(x_0) \) | Temporal cut out                   | Unlabeled real FGM data \((D_0 \in \{X_c, X_c, X_0\})\) | 1541×16 samples                             | 0.480                                | 83.42\%                        | Labeled real data \((D_0 \in \{X_c, Y_{cs}\})\) \(+\) pseudo-labeled real data \((D_0 \in \{X_0, Y_{cs}\})\) \(+\) pseudo-synthetic data \((D_0 \in \{S_0, Y_{cs}\})\) | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_d) \) | Sensor dropout                     |                                              | 1763×16 samples                             | 0.550                                | 81.53\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_0) \) | Spatial shift                      |                                              | 2704×16 samples                             | 0.845                                | 82.46\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
| Noise      | \( t(x_0) \) |                                              |                                              | 1705×16 samples                             | 0.532                                | 84.10\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
| Noise with | \( t(x_0) \) | Temporal cut out                   |                                              |                                              | 0.897                                | 85.70\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_d) \) | Sensor dropout                     |                                              |                                              | 0.759                                | 84.76\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |
|            | \( t(x_s) \) | Spatial shift                      |                                              |                                              | 0.958                                | 83.95\%                        |                                              | 0.1481          | 0.1702 | 0.1657 |

Baseline CNN model (supervised learning with model D arch.)

88.17\% 0.1061 5.73N

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see https://creativecommons.org/licenses/by/4.0/
with $\sigma(s) > 89\%$. In phase II, for each transformation functions, a separate self-trained FMG-DCGAN model was pretrained and evaluated. In all cases, it achieved impressive performances in force estimations ($R^2 > 73\%$, NRMSE < 0.175 and RMSE < 3.5N). The baseline model using supervised learning had $R^2 > 81\%$, NRMSE < 0.14 and RMSE < 3N. These results are reported in Table I.

**B. pHRI in 1D-Y**

In phase I, transformation functions, $t(x_0)$, $t(x_{sd})$ $t(x_{ntc})$, and $t(x_0)$ generated unlabeled synthetic data SD in the ranges of $2300 \times 10$ to $4200 \times 16$ samples with a similarity score of $\sigma(s) > 83\%$. Signal mixing transformation functions, $t(x_{sm-ntc}), t(x_{sm-nss})$ and $t(x_{sm-nss})$ generated synthetic data in the range of $5900 \times 16$ to $6300 \times 16$ samples with $\sigma(s) > 84\%$. In phase II, separate self-trained FMG-DCGAN model was pretrained and evaluated for each $t(x)$. The proposed model achieved good performances in force estimations ($R^2 > 75\%$, NRMSE < 0.16 and RMSE < 6.0N) while the baseline model had $R^2 > 78\%$, NRMSE < 0.15 and RMSE < 5.6N, as reported in Table II.

**C. pHRI in 1D-Z**

Transformation functions, $t(x_{tc}), t(x_{sd})$ $t(x_{ss})$, and $t(xn)$ generated unlabeled synthetic data SD in the ranges of $1500 \times 16$ to $2600 \times 16$ samples with $\sigma(s) > 81\%$ in phase I. For signal mixing, $t(x_{sm-ntc}), t(x_{sm-nsd})$ and $t(x_{sm-nss})$ generated synthetic data in the range of $4800 \times 16$ to $6100 \times 16$ samples with $\sigma(s) > 83\%$. In phase II, for each transformation functions, the self-trained FMG-DCGAN model achieved impressive performances in force estimations ($R^2 > 73\%, \text{NRMSE} < 0.175 \text{and RMSE} < 3.5N$). The reported results for the baseline model were $R^2 > 88\%, \text{NRMSE} < 0.11 \text{and RMSE} < 5.8N$, as shown in Table III.

**IV. DISCUSSIONS**

Force myography (FMG) is a non-invasive bio signal that can capture muscle contractions and can be used in applications such as gesture recognition, prosthetic control, rehabilitation protocols for individuals with mobility issues, helping people perform activities of daily life, and enhancing the safety of human-robot interactions (HRI). In current practices of FMG-based applications, machine learning models are trained by obtaining enough training data, including data from enough participants and/or labeling all data. These steps are laborious and not always possible. Therefore, data scarcity motivated this study for synthetic FMG data generation via generative adversarial network (GAN) to alleviate the need to collect a large amount of training data for the application.

However, generating appropriate bio signals is a challenging task. There are risks associated with wrong predictions when the generated synthetic data do not follow the real domain due to a reality gap. In literature, few research are conducted on synthetic data generations such as electrocardiogram (ECG), electroencephalogram (EEG), electromyography (EMG) bio signals [28-30] in user recognition, motor functions or cardiac anomalies detection, and grasping action recognition. These studies implemented data augmentation, attention-based transformation, generative pretrained transformer (GPT) model using unsupervised, semi-supervised or self-supervised learning approaches. Therefore, this study conducted FMG bio-signal generation specially in the challenging high-dimensional pHRI environment for the first time and bridged the gap in literature. The ability to generate synthetic FMG bio-signals could be beneficial by decreasing the need to collect large amounts of data in the real-world applications.

The domain randomization approach described in this study helped mitigate the reality gap by making the simulated data as diverse as possible. Adapting carefully selected transformations that represented authentic environmental changes generated more realistic synthetic data. The GAN loss bounds were selected through trial-and-error basis along with the cosine similarity score threshold so that the generator could produce adequate real-like synthetic data for future training. These ranges ensured that the model did not suffer mode collapse such that the fake data from the generator were similar but different from the real data. The trade-off between the quality and quantity of the generated data were established via the loss boundaries and the score threshold. Through the tradeoff between quality and quantity, we ensured the generation of a considerable number of synthetic samples as well as maintaining sufficient variability among them. Implementing semi-supervised learning overcame dependencies on labeled training data where only few labeled real data with a large volume of unlabeled data was used to self-train the GAN model. Also, implementing self-training with fewer labeled data showed the feasibility of the model in practical scenarios. Hence, the proposed algorithm demonstrated viability in recognizing human applied force with very few labeled real FMG data in practical pHRI applications. Therefore, it could be a viable solution for FMG-based HRI, rehabilitation, and prosthetic control where humans interact with machines for daily activities on a regular basis.

In this study, collecting training data (only a few were labeled) and evaluating test data with the proposed algorithm were conducted in the same session to simulate and compare to a situation where data collection in various sessions is costly, difficult, or impossible. Hence, the training data and test data had similar feature distributions, similar tasks, and the same sensor positions on the targeted limb. Therefore, domain randomization was implemented so that variations could be introduced such that it might become another session dataset for the model. During the adversarial training, real data that were transformed for generating synthetic data, were also used in pretraining the semi-supervised model. Due to the signal mix in phase I, investigated transformation functions modified and randomized the real distribution such that the model could learn better the latent features. Only during phase II, was the model introduced to the real feature distributions for pretraining. Therefore, implementing these functions helped reduce overfitting too.

In the literature, CNN based GAN models are more
The proposed model was able to learn interaction patterns from each single instance of spatiotemporal data and estimated applied forces quickly so that the interactions with the robot would appear instantaneous to the participant. As a pilot study, its performance with the time-series FMG data in 1D-X, Y and Z dimensions were impressive. Therefore, we focused our investigation on the FMG-DCGAN model.

Observations showed that the FMG-DCGAN model generated more synthetic data when transformation functions \( t(x_{\text{sm}}) \) based on signal mixing used. Specifically, signal mixing of noise with spatial shift \( t(x_{\text{sm}}) \) generated a large volume of synthetic data (6400x16 > 5700x16 samples) in each 1D interactions of X, Y and Z dimensions with greater generated-to-input-data ratio (99% > \( \delta \) > 89%) and mean similarity score in the range of 90% > \( \sigma(s) \) > 84%. The volume of such synthetic data was almost double the calibration data collected during real-time interactions. A transformation function that used noise mixing with temporal cut out \( t(x_{\text{ntc}}) \) also obtained impressive results (6000x16 > 5700x16 samples, 94% > \( \delta \) > 89%, and 91% > \( \sigma(s) \) > 84%). Therefore, Transformation function \( t(x_{\text{sm}}) \) ranked highest, while \( t(x_{\text{ntc}}) \) ranked second in all pHRI scenarios. A statistical t-test was conducted to compare samples for the mean generated by the two functions \( t(x_{\text{sm}}) \) and \( t(x_{\text{ntc}}) \). The selected transformation function \( t(x_{\text{sm}}) \) was found to be statistically significant (\( t = 4.75, 0.005 \)) at a 95% confidence level. Hence, it was obviously the better choice to optimize the model in generating maximum outcome. Fig. 5 (a, b) shows the performance comparison of the proposed model for each transformation function with the baseline model in terms of the accuracies and error (\( R^2 \) and RMSE). The baseline model was trained with labeled 3200x16 samples of real data while the self-trained FMG-DCGAN was trained with >9000x16 samples (1600x16 real labeled data aggregated with ~6200x16 pseudo-labeled synthetic data and 1600x16 real pseudo-labeled data). Therefore, the proposed model was trained with labeled data and pseudo-labeled data that had a ratio of 1:4. Both the self-trained FMG-DCGAN and the baseline model were evaluated on the same target test data. For 1D-X, Y and Z, the baseline model achieved force estimation accuracies in \( R^2 \) of 82%, 78% and 88%, respectively. On the contrary, the unsupervised, semi-supervised self-trained model with \( t(x_{\text{sm}}) \) accomplished impressive accuracies in 1D-X, Y, and Z as \( R^2 \) of 77.32%, 77.19% and 84.27%, respectively, as shown in Fig. 5 (c).

V. CONCLUSION

In practice, obtaining enough training data, having more participants, or labelling all data are not always possible. Domain adaptation and generalization could work when a large volume of training data is available which is not always feasible to collect. In common industrial pHRI tasks, a quickly trained model with fewer labeled or unlabeled data is preferred. In such cases, synthetic data generation could be a solution to accumulate adequate training datasets. Also,

**FIGURE 5.** Performance evaluation of unsupervised, self-trained FMG-DCGAN model with different transformation functions compared to the supervised baseline model: a) plot of \( R^2 \) in 1D-X, Y, Z, b) RMSE in 1D-X, Y, Z (pink dot: unsupervised self-trained FMG-DCGAN model, yellow dot: supervised baseline model), and c) boxplot of self-trained FMG-DCGAN model with \( t(x_{\text{sm}}) \) in 1D-X, Y, Z compared to supervised baseline model in 1D-X, Y, Z (\( R^2 \) & RMSE).

This article has been accepted for publication in IEEE Access. This is the author’s version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/ACCESS.2022.3187115
unsupervised, or semi-supervised (weakly supervised) learning can overcome the dependencies on labeled training data where only few labeled training data is available. The proposed unsupervised, self-trained FMG-DCGAN model with certain transformation functions could estimate interactive forces in 1D with impressive performances (85%>R2=77%) compared to the baseline model (89%>R2=78%). Therefore, unsupervised GAN combined with self-trained technique can improve recognizing human applied forces in practical FMG-based pHRI applications in 1D. Also, generating real-like FMG bio-signals could be beneficial addressing data crisis in real-world scenarios. We believe these findings contribute to the development of a more discreet and practical wearable device for a variety of applications.

Due to time-constraints, this algorithm was evaluated during pHRI between only one participant and the Kuka robot in 1D-X, Y, Z directions. In this preliminary study, our focus was to investigate the possibilities of using generated synthetic FMG data in training a self-trained FMG-DCGAN model to address labeled data scarcity. However, in future studies, involving more participants and interacting in 3D with other deep learning approaches such as LSTM, RNN or GRU based GAN architectures can be further investigated. Also, implementing domain generalization with the proposed unsupervised, self-trained FMG-DCGAN model could be studied in future.

ACKNOWLEDGMENT

The Principal Investigator, Carlo Menon, and members of his research team have a vested interest in commercializing the technology tested in this study, if it is proven to be successful and may benefit financially from its potential commercialization. The data are readily available upon request.

REFERENCES

[1] Z. G. Xiao and C. Menon, “Towards the development of a wearable feedback system for monitoring the activities of the upper-extremities,” J. Neuroeng. Rehabil., vol. 11, no. 2, pp. 1–13, Jan. 2014.
[2] A. Radmard, E. Scheme, and K. Englehart, “High-density force myography: A possible alternative for upper-limb prosthetic control,” J. of Rehab. R. & D., vol. 53, no. 4, 2016.
[3] Z. G. Xiao, and C. Menon, “Performance of forearm FMG and sEMG for estimating elbow, forearm and wrist positions,” Journal of Bionic Engineering, vol. 14, no. 2, pp. 284-295, 2017.
[4] X. Jiang et.al, “Exploration of force myography and surface electromyography in hand gesture classification,” Medical engineering & physics, vol. 41, pp. 63-73, 2017.
[5] M. Connan et al., “Assessment of a wearable force-and electromyography device and comparison of the related signals for myococontrol,” Frontiers in neurorobotics, vol. 10, pp. 17, 2016.
[6] A. Belyea, K. Englehart, and E. Scheme, “FMG Versus EMG: A comparison of usability for real-time pattern recognition based control,” IEEE Transactions on Biomedical Engineering, vol. 66, no. 11, pp. 3098-3104, 2019.
[7] C. Dissinghorst-Klug, T. Schmitz-Rode, and G. Rau, “Surface electromyography and muscle force: Limits in sEMG-force relationship and new approaches for applications,” Clinical biomechanics, vol. 24, no. 3, pp. 225-235, 2009.
[8] E. Fujiwara, Y. T. Wu, C. K. Suzuki, D. T. G. de Andrade, A. R. Neto and E. Rohmer, “Optical fiber force myography sensor for applications in prosthetic hand control,” 2018 IEEE 15th International Workshop on Advanced Motion Control (AMC), 2018, pp. 342-347, doi: 10.1109/AMC.2019.8371115.
[9] E. Cho, R. Chen, L.-K. Merhi, Z. Xiao, B. Pousset, and C. Menon, “Force myography to control robotic upper extremity prostheses: a feasibility study,” Frontiers in bioengineering and biotechnology, vol. 4, pp. 18, 2016.
[10] X. Jiang et.al, “Force exertion affects grasp classification using force myography,” IEEE Transactions on Human-Machine Systems, vol. 48, no. 2, pp. 219-226, 2017.
[11] K. H. Chu, X. Jiang, and C. Menon, “Wearable step counting using a force myography-based ankle strap,” Journal of rehabilitation and assistive technologies engineering, vol. 4: 2055668317746307, 2017, doi:10.1177/2055668317746307.
[12] A. Kadkhodayan, X. Jiang, and C. Menon, “Continuous prediction of finger movements using force myography,” Journal of medical and Biological Engineering, vol. 36, no. 4, pp. 594-604, 2016.
[13] M. Sakr, et.al, “Estimation of user-applied isoformic force/torque using upper extremity force myography,” Frontiers in Robotics and AI, vol. 6, no.120, Nov. 2019, doi: 10.3389/frobt.2019.00120.
[14] U. Zakia, and C. Menon, “Estimating exerted hand force via force myography to interact with a biaxial stage in real-time by learning human intentions: a preliminary investigation,” Sensors, vol. 20, no. 7, Apr. 2020, doi: 10.3390/s20072104.
[15] U. Zakia, and C. Menon, “Toward Long-Term FMG Model-Based Estimation of Applied Hand Force in Dynamic Motion During Human-Robot Interactions,” IEEE Transactions on Human-Machine Systems, vol. 51, no. 4, pp. 310-323, 2021.
[16] U. Zakia, and C. Menon, “Force Myography-Based Human Robot Interactions via Deep Domain Adaptation and Generalization,” Sensors, vol. 22, no. 1, pp. 211, 2022.
[17] U. Zakia and C. Menon, “Human-Robot Collaboration in 3D via Force Myography Based Interactive Force Estimations Using Cross-Domain Generalization,” in IEEE Access, vol. 10, pp. 35835-35845, April 2022, doi:10.1109/ACCESS.2022.32164103.
[18] J. E. Van Engelen, and H. H. Hoos, “A survey on semi-supervised learning,” Machine Learning, vol. 109, no. 2, pp. 373-440, 2020.
[19] I. Triguero, S. Garcia, and F. Herrera, “Self-labeled techniques for semi-supervised learning: taxonomy, software and empirical study,” Knowledge and Information systems, vol. 42, no. 2, pp. 245-284, 2015.
[20] X. Li et.al, “Learning to self-train for semi-supervised few-shot classification,” Advances in Neural Information Processing Systems, vol. 32, pp. 10276-10286, 2019.
[21] I. Goodfellow, et al., A. Courville, and Y. Bengio, “Generative adversarial nets,” Advances in neural information processing systems, vol. 3, pp. 2672-2680, 2014, doi: 10.3156/sjost.29.5_177.2.
[22] S. Höfer et al., “Sim2Real in Robotics and Automation: Applications and Challenges,” in IEEE Transactions on Automation Science and Engineering, vol. 18, no. 2, pp. 398-400, April 2021, doi: 10.1109/TASE.2021.3064065.
[23] J. Tremblay et al., “Training Deep Networks with Synthetic Data: Bridging the Reality Gap by Domain Randomization,” 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2018, pp 1082-10828, doi: 10.1109/CVPRW.2018.00143.
[24] J. Tobin et al., W. Zaremba and P. Abbeel, "Domain randomization for transferring deep neural networks from simulation to the real world," IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2017, pp. 23-30, doi:10.1109/IROS.2017.8202133.
[25] X. Peipii, L.Zhang, and F. Li, "Learning similarity with cosine similarity ensemble." Information Sciences, vol. 307, 2015, pp. 39-52.
[26] M. Safaee and P. Neto, "KUKA Sunrise Toolbox: Interfacing Collaborative Robots With MATLAB," IEEE Robotics & Automation Magazine, vol. 26, no. 1, pp. 91-96, March 2019, doi: 10.1109/MRA.2018.287776.
[27] A. Radford, L. Metz, and S. Chintala, “Unsupervised representation learning with deep convolutional generative adversarial networks,” arXiv preprint arXiv:1511.06434, 2015.
[28] MG. Kim, and SB Pan, "A Study on User Recognition Using the Generated Synthetic Electrocardiogram Signal," Sensors, vol. 21, no.
[29] J. Y. Cheng et al., "Subject-aware contrastive learning for biosignals," arXiv preprint arXiv:2007.04871, 2020.

[30] J. J. Bird et al., "Synthetic Biological Signals Machine-generated by GPT-2 improve the Classification of EEG and EMG through Data Augmentation," IEEE Robotics and Automation Letters, vol. 6, no. 2, pp. 3498-3504, 2021.

Umme Zakia obtained BSc in Electronics & Computer Science in 2001 and MSc in Computer Science & Engineering in 2007 from Jahangirnagar University and North South University, Bangladesh. She is currently pursuing PhD degree in Engineering Science at Simon Fraser University, BC, Canada and working as a research assistant at Menrva Research Group. Her research interests are machine learning, human activity recognition, robotics, HRI, communication networks and VLSI. Umme has published several articles in peer-reviewed journals and conferences.

Arnab Barua was born in Chittagong, Bangladesh in 1995. He pursued his B.Sc. degree in Computer Science from International Islamic University Chittagong in 2018 and currently pursuing M.Sc. in Computer Science from Memorial University of Newfoundland. He has been working as a Research Assistant in Xianta Jiang’s lab since 2020. He has contributed to research associated with human activity and is an author of multiple publications related to human activity. His research interests include improvement of Human Activity Recognition and Gait Analysis using divergent applications of Machine Learning.

Xianta Jiang received his Ph.D. from the Computer Science of the Simon Fraser University (2015) and Zhejiang University (Dual degree), where his thesis research focused on mental workload assessment using eye-tracking technology. He then completed a post-doc fellowship at the MENYA research group, Engineering Science, Simon Fraser University, working on the area of detecting human activity by processing muscle activities data with a machine learning algorithm. He latterly worked as a senior research associate in the Surgical Simulation Research Lab (SSRL) in the Department of Surgery at the University of Alberta, studying team cognition using Augmented Reality (AR) and eye-tracking technologies. Dr. Jiang is taking the assistant professor position in the CS of MUN starting in June 2019. His research interests include Intelligent Human-Machine Interaction, Ubiquitous Computing in Healthcare, Human Behavior Recognition, Bio-Signal Processing, Eye-Tracking, and Pupil Diameter, Human Factors.

Carlo Menon received a “Laurea” degree and a PhD degree from the University of Padua (Italy) respectively in 2001 and 2005. He was a visiting graduate student at Carnegie Mellon University (USA) in 2004. He took a Research Fellow position at the European Space Agency (The Netherlands) in 2005 and 2006. He subsequently became Professor at Simon Fraser University (Canada) where he founded the Menrva Research Group, held a Tier 1 Canada Research Chair and received a number of Scholar Awards including both the Canadian Institutes of Health Research (CIHR) and the Michael Smith Foundation for Health Research awards. He joined ETH Zurich in 2021 where he currently leads the Biomedical and Mobile Health Technology laboratory. Carlo has published over 300 articles in journals and conferences.