Deep Regression Modeling for Imbalanced and Incomplete Time-Series Data

Murtadha D. Hssayeni and Behnaz Ghoraani, *Senior Member, IEEE*

***Abstract*—During the collection of time-series data, many reasons lead to imbalanced and incomplete datasets. Consequently, it becomes challenging to develop deep convolutional models without suffering from overfitting. Our objective in this paper was to investigate an emerging but rather underutilized framework of Conditional Generative Adversarial Networks (cGANs) for improving deep regression models for time-series data with an imbalanced and incomplete distribution. First, we investigated the potential of using a vanilla cGAN as a data imputation to improve the generalizability of the developed models to unseen data in such datasets. Next, we proposed a modified cGAN architecture with improved extrapolation and generalizability of the regression models. Our investigations used an imbalanced synthetic non-stationary dataset, a real-world dataset in Parkinson's disease (PD) application domain, and one publicly-available dataset for Negative Affect (NA) estimation. We found that vanilla cGAN failed to generate realistic time-series data due to severe mode collapse, limiting its application as a data imputation for imbalanced and incomplete data. Importantly, the proposed cGAN framework significantly improved extrapolation and generalizability for the prediction of regression scores with an average improvement of 56%, 34%, and 18%, respectively, in mean absolute error for the synthetic, PD, and NA datasets when compared with traditional Convolutional Neural Networks. The codes are publicly available on Github.**

***Index Terms*—Deep regression modeling, time-series data, generative adversarial networks, imbalanced and incomplete data, extrapolation.**

I. Introduction

IMBALANCED datasets commonly exist in real-world applications. This paper addresses challenges with such imbalanced datasets, specifically in the biomedical domain when estimating disease severity scores (i.e., regression scores). The data is usually collected from a small group of patients, resulting in a data representation with a geometric distribution [1], [2], [3], [4]. This distribution means long-tailed normal distribution with some scores or classes representing the distribution head and other rare ones representing the tail. Hence, the data collection leads to imbalanced and, in most cases, incomplete datasets as the recruited patients may not represent the entire range of the disease severity score. Deep-learning models suffer overfitting when trained on these constrained datasets that violate the parsimony principle [5]. Overfitted models have poor generalizability to testing samples with underrepresented or unseen regression scores [6]. Multiple approaches are developed to handle overfitting in classification problems with cross-sectional data such as images [7]. One approach is based on Generative Adversarial Networks (GANs) that have emerged in numerous applications [8], [9]. For instance, GANs have been used to learn and generate new samples as a data augmentation method. The generated samples were then used in the training process in addition to the real samples. Unlike cross-sectional data, the overfitting challenges of deep regression problems in imbalanced time-series signal data have been overlooked. This is while a practical solution to this problem is essential in health monitoring applications such as over-time monitoring of human health or disease severity [10].

Our primary focus in this paper is to improve deep learning performance when used for regression in applications with limited imbalanced datasets. Recursive partitioning with ensemble learning [6] and oversampling using data augmentation [11], [12] have been commonly considered for time-series modeling. However, these methods are challenged by the overfitting issue in time-series regression as their design does not consider challenges caused by imbalanced and incomplete datasets. In our preliminary work [13], we investigated a preliminary version of a Conditional Generative Adversarial Network (cGAN) as a regressor. The cGAN was evaluated only on a single dataset without investigating GAN mode collapse in time series.

In this work, we significantly expand the analysis and the applications of our preliminary work [13] in several aspects and establish the advantage of our proposed approach addressing the overfitting issue of deep regression models in imbalanced and incomplete time-series data. Our main contributions are threefold as follow:

- We investigate the mode collapse of cGAN caused by imbalanced regression data.
- We propose a novel formulation of cGAN to improve deep regression models' generalizability using time-series data.
- We evaluate the proposed cGAN for improving generalizability in a synthetic dataset and two real-world time-series datasets, compare its performance with a traditional Convolutional Neural Networks (CNN) and Long short-term

Manuscript received 4 January 2024; accepted 25 February 2024. Date of publication 19 March 2024; date of current version 23 November 2024. This work was supported by the National Science Foundation under Grant 1936586 and Grant 1942669. *(Corresponding author: Behnaz Ghoraani.)*

Murtadha D. Hssayeni is with the University of Technology, Baghdad 10066, Iraq.

Behnaz Ghoraani is with the Department of Electrical Engineering and Computer Science, Florida Atlantic University, Boca Raton, FL 33431 USA (e-mail: bghoraani@fau.edu).

Recommended for acceptance by Prof. T. Huang.

Digital Object Identifier 10.1109/TETCI.2024.3372435

memory (LSTM) model, and make the code publicly available.

Our contributions are novel as we propose the first single-stage regression model to tackle limited data while improving extrapolation to unseen ranges of scores. By a single stage, we mean using the generator as the regressor and not as a data generator for data imputation. The organization of the paper is as follows. Section III describes our proposed cGAN framework. In Section IV, we explain the implementation of the CNN-based for comparison purposes and our proposed cGAN regression model. We generated a synthetic dataset of non-stationary time-series signals with an imbalanced and incomplete representation for evaluation purposes as described in Section V. Section V also provides the details of the two real-world datasets used for our evaluation purposes. The results and conclusions are discussed in Sections VI and VII, respectively.

II. Literature Review

There are multiple approaches for learning from imbalanced data in classification problems, such as synthetic samples, data resampling, and cost-sensitive learning [14]. However, these solutions do not work directly for regression problems [15], [16], [17], [18]. The continuous scores in imbalanced regression problems imply both interpolation and extrapolation of the target scores, which is not the case in a set of categorical labels in classification problems. Zhu et al. propose a modification to the Synthetic Minority oversampling technique that maintains the ordinality in regression problems but does not deal with missing ranges of scores [15]. Similarly, Branco et al. adapt random oversampling and the addition of Gaussian noise to regression problems and propose a combination strategy of oversampling and undersampling based on target scores distribution [16]. Steininger et al. propose modifying the classification cost-sensitive methods by approximating the distribution of the imbalanced target scores and using it to weigh the loss during model training [17].

Instead of preprocessing the training data, Yang et al. propose Feature Distribution Smoothing (FDS) as a calibration layer that can be directly integrated with deep models [18]. They show a consistent improvement when adding FDS to the vanilla model. The previously mentioned approaches perform data resampling or optimize the extracted features. In this work, we propose a new cGAN framework that can be integrated with other methods to deal with limited regression data.

The oversampling strategy based on GANs has been beneficial in classification problems with cross-sectional and time-series data [12], [19], [20]. Its application has also shown some advantages in regression problems with cross-sectional data [21], [22]. cGAN as a variation of GAN [23] is used as an oversampling approach for imbalanced cross-sectional data in classification problems [12], [24]. This model takes class labels or regression scores as additional domain information to condition the generation process. This feature of cGAN makes it possible to generate data samples for minority classes or underrepresented regression scores [23]. However, vanilla cGAN suffers from mode collapse when the model is trained on limited data [25] and when the number of conditioning classes increases [26]. As a result, the application of cGAN in regression applications is expected to suffer significantly from mode collapse since there are unlimited classes in a regression problem. Also, the overfitting problem is worse for limited and imbalanced datasets, where the model overfits the majority classes only in the training data and has low sensitivity to the minority classes [7]. Therefore, the mode collapse issue is predicted to be even more problematic when the training data is imbalanced and incomplete. Hence, the cGAN-generated samples are predicted to have high similarities to each other due to severe mode collapse and, thus, not helpful in oversampling the data.

Recently, Aggarwal et al. proposed a new application of cGAN in cross-sectional data. Their proposed architecture did not use cGAN to produce simulated samples for data augmentation. Instead, they used the trained cGAN as the regression model [27]. Their method was robust when applied to noisy data and outperformed boosted-tree models. This architecture inspired our work to investigate an improved design of cGAN to address the underlying challenges specific to imbalanced time-series signals in regression problems.

III. Proposed Methodology

Our design consisted of a new cGAN framework to improve the estimation of regression scores from imbalanced and incomplete time-series signals. Two multi-layer models construct the cGAN, which are the generator (G) and discriminator (D) [23]. These models compete as adversaries, consisting of convolutional and dense layers. As seen from the vanilla cGAN architecture in Fig. 1(a), the generator is conditioned according to the class labels or regression scores in our case, so generator G learns to generate the closest fake signals to the real ones. The novel aspect of our framework is twofold. First, we condition the generator G on the real signals to generate the closest fake scores to the real ones (Fig. 1(b)). Next, we embed a CNN in the generator G to predict the regression scores from the signals (Fig. 1(c)).

This design ensures that the embedded CNN in G learns the temporal patterns within a time-series segment (w) associated with different regression scores y without using the scores directly to train G. Instead, y beside w is input to D to distinguish fake scores from real scores, $G(z|w)$ and y, respectively. G is conditioned on real signals w to map noise z from the latent space p_z to like-real scores. As a result, G will learn the data distribution instead of memorizing the regression scores; thus, the model generalizes better to unseen data. The generator, G, from the trained cGAN network, represents the regression model to estimate the scores of new time-series segments. Our proposed architecture will train G as the final regression model. This is while the existing data imputation methods balance the training data by generating synthetic samples using methods such as GAN or cGAN and then training a separate CNN model [12].

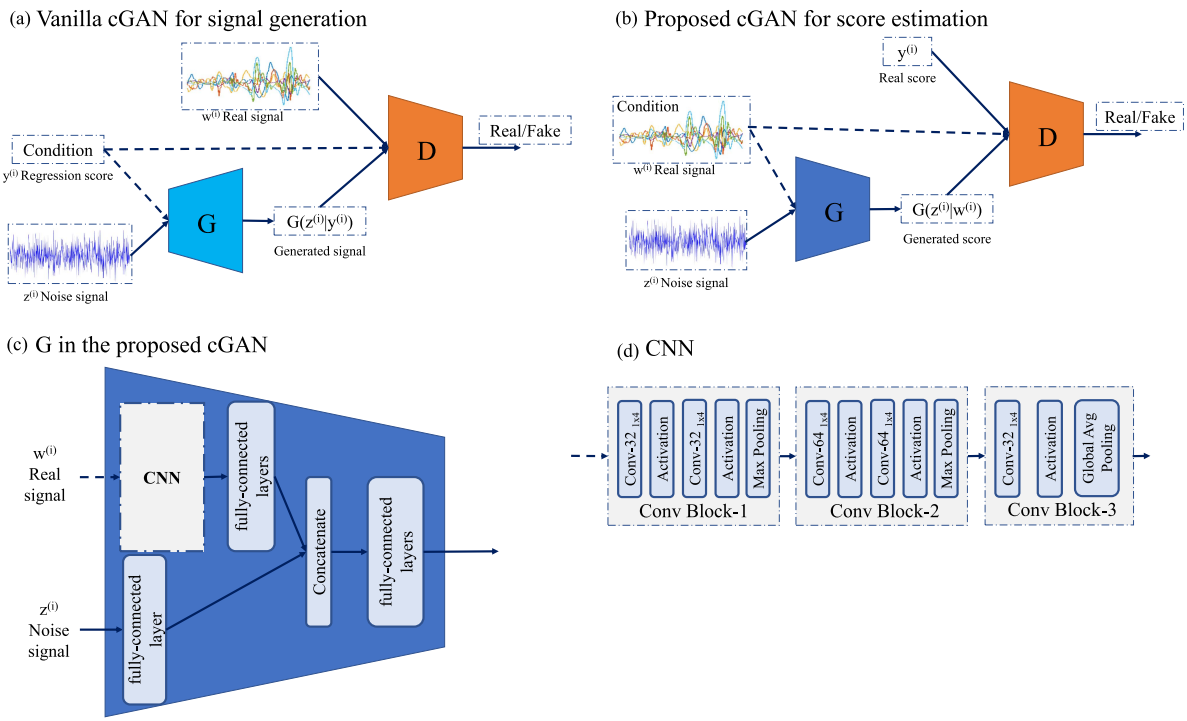

Fig. 1. (a) Design of vanilla cGAN. (b) The modified cGAN as proposed in this study. (c) The design of the generator G in our proposed cGAN. (d) The architecture of the embedded CNN in the generator G.

IV. Methods

Our comparative work investigated CNN to train a deep regression model and a vanilla cGAN for signal generation. In this section, first, we describe the CNN and vanilla cGAN models, followed by our proposed modified cGAN framework as a regression model.

A. Convolutional Neural Networks

A 1-dimensional (1D) CNN model was developed to estimate the regression scores for the segmented time-series signal. The CNN model consists of multiple convolutional blocks, as shown in Fig. 1(d) followed by a global average pooling layer and a block of two fully-connected layers. The depth of the network increases as we go to blocks in higher levels. For instance, the first block consists of two layers of 32 convolutional filters of width 4, and the second one consists of two layers of 64. Each convolutional layer is followed by a ReLU activation layer that captures the non-linear patterns. Each block is followed by a max-pooling layer. The last block has one convolutional layer followed by the global average pooling layer that summarizes the extracted features. There are 128 nodes in the first fully-connected layer, followed by a 0.5-rate dropout layer. One node was assigned in the output layer to output the estimated score. Constructing wider CNN networks is performed by repeating Conv Block-2 to increase the number of convolutional layers.

B. Vanilla cGAN

The ability of a vanilla cGAN to generate simulated time-series signals was investigated. We specifically focused on the performance and mode collapse when the training data was limited and imbalanced. The vanilla cGAN architecture is displayed shown in Fig. 1(a). The generator is conditioned on real scores and maps noise from a latent space to generated signals as the discriminator distinguishes the fake vs. real signals given the condition score. The generator architecture takes two inputs: score as a condition and noise. The score input undergoes a dense layer with 128 units, ReLU activation, and dropout, while the noise input follows a similar dense layer configuration. The outputs from both paths are concatenated, and the resulting vector passes through a dense layer and a reshape layer. Subsequently, a series of convolutional transpose layers, each accompanied by batch normalization and ReLU activation, upsample the data by using a stride of 2. The final convolutional transpose layer produces the generator's output, which is linearly activated and has the same window length of real signals.

We attempted to reduce mode collapse by implementing the minibatch discriminator as in Salimans et al. [28]. Using the similarity between the samples of a given batch, the discriminator prevents mode collapse. The process is as follows. A custom layer measures the similarity by multiplying the extracted-feature vectors by a weight tensor. It then finds the distance between the produced matrix's rows as the similarity vector and concatenates it with the extracted feature vector. The fully connected layer uses this final vector to detect fake vs. real samples.

Equation (1) describes a min-max game used to train the cGAN model. With stable training, generator G learns a model distribution p_g by minimizing $\log(1 - D(G(z|y)))$. This distribution approximates the real data distribution p_{data} and is used to draw like-real signals. We maximize $log(D(w|y))$ and

$\log(1 - D(G(z|y)))$ to train the discriminator D to learn to detect the fake signals. The game between D and G converges to an equilibrium at a saddle point.

$$\min_G \max_D V(D,G) = E_{w\sim p_{data}(w)}\left[\log D(w|y)\right] + E_{z\sim p_z(z)}\left[\log\left(1 - D\left(G\left(z|y\right)\right)\right)\right] \tag{1}$$

C. Proposed Modified cGAN

Fig. 1(b) illustrates the architecture of the proposed cGAN for regression score estimation [23]. Like the vanilla cGAN, the cGAN is trained using a min-max game shown in (2). However, the major difference is that the cGAN is conditioned on the signal w instead of the regression score. The generator G is shown in Fig. 1(c). It minimizes $\log(1 - D(G(z|w)))$ so that the generator learns to generate a model distribution p_g similar to p_{data}, the real data distribution. It maximizes $log(D(y|w))$ and $\log(1 - D(G(z|w)))$ so that the discriminator D learns to detect the fake vs. real scores. This process is repeated until it reaches a saddle point with an equilibrium between G and D. It is worth mentioning that in our framework, G is conditioned on real signals. This change prevents the generator from learning the data distribution and improves generalizability. The real and fake loss train G indirectly by their backpropagation through the discriminator D. Such design is expected to attenuate overfitting as it ensures that G does not see the real scores.

$$\min_G \max_D V(D,G) = E_{y\sim p_{data}(y)}[\log D(y|w)] + E_{z\sim p_z(z)}[\log(1 - D(G(z|w)))] \tag{2}$$

The convolutional blocks in G follow the CNN architecture with ReLU activation layers. Leaky ReLU layers are used in D as the best practice in designing GANs. We followed five steps to train the cGAN for k iterations:

- Select n time-series segments $\left\{w^{(1)}, ..., w^{(n)}\right\}$ with their scores $\left\{y^{(1)}, ..., y^{(n)}\right\}$ from the data distribution p_{data} to create a mini-batch.
- Use the mini-batch to update θ_d, the discriminator weights:

$$\nabla_{\theta_d} \frac{1}{n}\sum_{i=1}^{n} \log D\left(y^{(i)}|w^{(i)}\right) \tag{3}$$

- Construct a mini-batch of n noise vectors $\left\{z^{(1)}, ..., z^{(n)}\right\}\}$ sampled from a uniform distribution.
- Use the mini-batch of fake samples $G(z|w)$ to update θ_d, the discriminator weights:

$$\nabla_{\theta_d} \frac{1}{n}\sum_{i=1}^{n} \log\left(1 - D\left(G\left(z^{(i)}|w^{(i)}\right)\right)\right) \tag{4}$$

- Use the previously created mini-batch to update θ_g (the generator weights) with fixed θ_d:

$$\nabla_{\theta_g} \frac{1}{n}\sum_{i=1}^{n} \log D\left(G\left(z^{(i)}|w^{(i)}\right)\right) \tag{5}$$

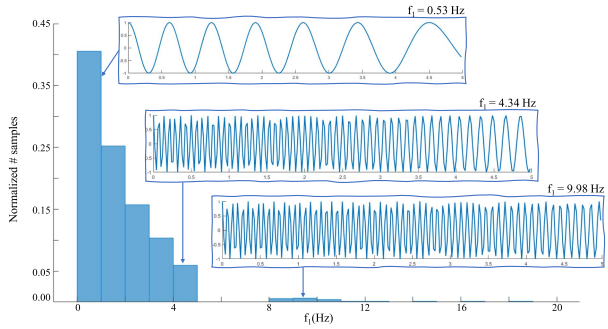

Fig. 2. Geometric distribution of the regression score (i.e., the chirp's final frequency f_1) in the synthetic training set is shown. We displayed three synthetic samples with regression scores of 0.53, 4.33, and 9.98. The testing set distribution (not shown here) was uniform between 0 and 20.

V. Datasets

A. Synthetic Dataset

We generated a synthetic time-series signal dataset in our investigation. The signals were non-stationary in time and frequency. The distribution of the generated signals was selected to be like-real geometric. We removed some ranges of the regression score to generate an incomplete distribution. The generated non-stationary signals were chirp signals commonly seen in different real-world applications. Estimating the parameters of these chirp signals has been a major objective [29]. We generated a quadratic-phase chirp signal according to (6).

$$x(t) = \cos\left(2\pi t\left(\frac{f_1 - f_0}{T}t + f_0\right)\right) \tag{6}$$

where f_0 is the initial frequency with a final frequency f_1 at time T. Five-second chirps $\left(w^{(i)} \in \mathbb{R}^{250\times 1}\right)$ were generated with $fs = 50$ Hz by randomly selecting f_0 from a uniform distribution between 0–20 Hz. f_1 was selected between 0 and 20 Hz from a geometric or uniform distribution depending on whether the data was used as the training or testing set, respectively.

The chirp signals were considered inputs to the deep models, with f_1 frequency being the ground-truth regression score. One training and one testing set were generated, each with 1000 samples. For the training set, the regression score f_1 was selected from a geometric distribution shown in Fig. 2. The regression scores of the testing set had a uniform distribution, as shown in Fig. 8(a). We intentionally enforced a missing range between 5 Hz and 8 Hz in the regression score distribution of the training set. In contrast, the testing set represented the entire range equally. Such training and testing sets were deliberately selected to investigate the generalizability of the trained models to unseen or minority samples. The codes for the synthetic dataset generations are publicly available on GitHub [30].

B. Parkinson's Disease Dataset

Most PD patients experience abnormal involuntary, dyskinetic movements at some point during the disease. These dyskinesias are troublesome and must be managed by adjusting the dose and/or frequency of PD medication(s). However, effective

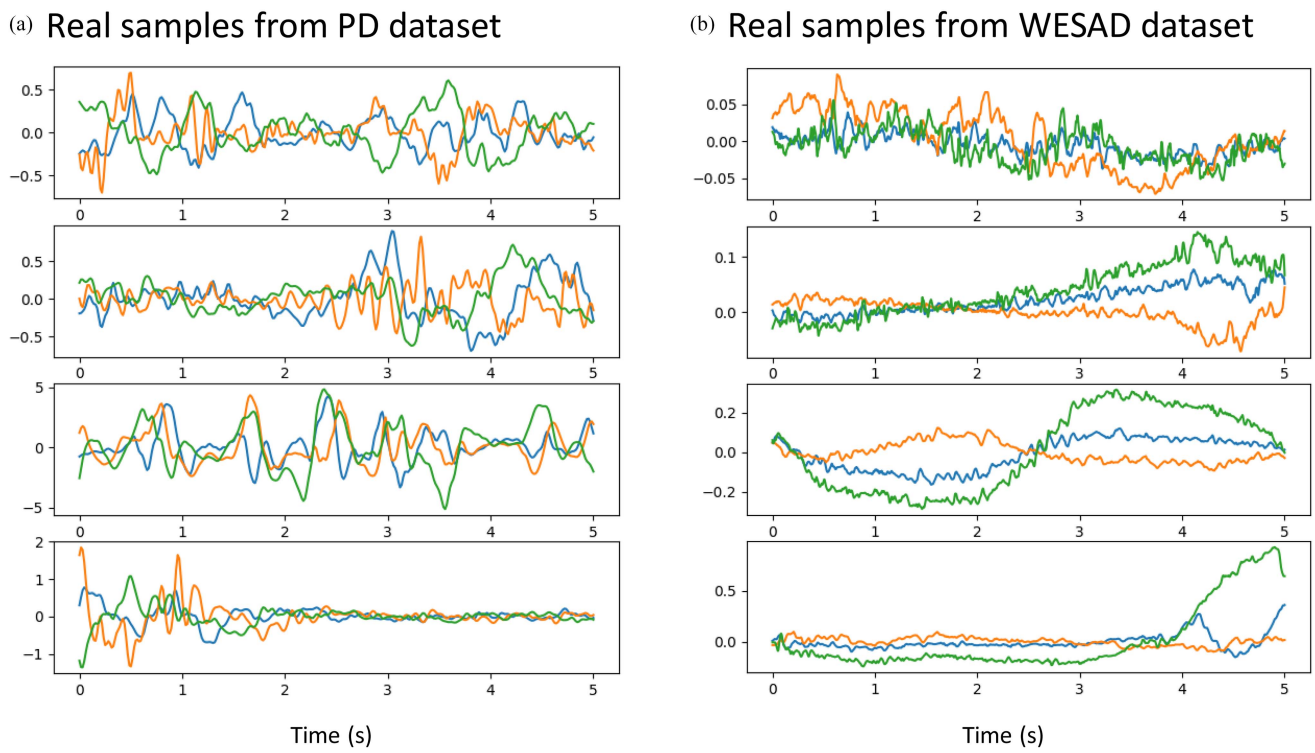

Fig. 3. Real segments of data from the PD (a) and WESAD (b) datasets that show the three axes of the accelerometer.

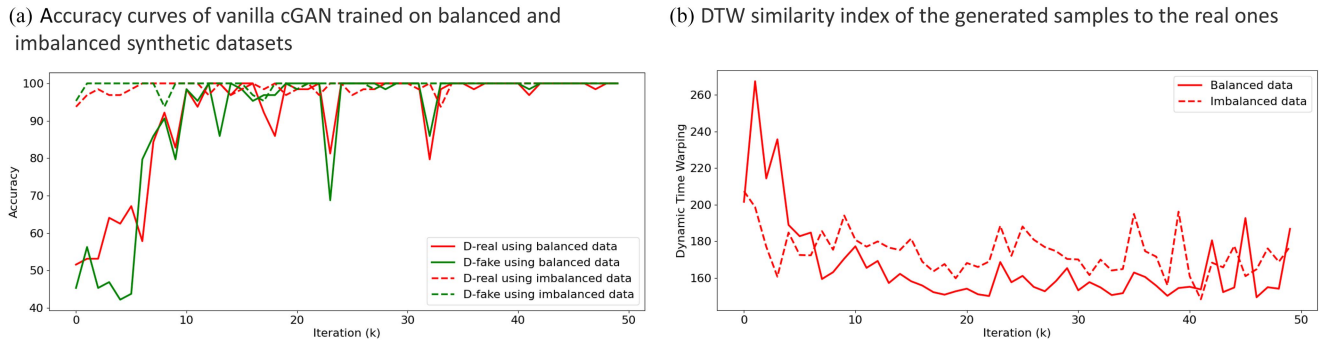

Fig. 4. (a) Accuracy and (b) dynamic time warping curves of the balanced and imbalanced synthetic datasets are displayed. Training on imbalanced data leads to overfitting right after the first epoch when the accuracy of discriminator D in detecting real and fake samples crossed 90%.

medication adjustments require the treating clinician to receive a detailed report on how the dyskinesia scores of a patient vary during a typical day. To address this need, this paper developed a deep regression model to estimate dyskinesia severity scores from movement data collected using a wearable on the wrist and one on the ankle of PD patients. Our study included 15 PD patients (6 female, 9 male) with an average age of 58±10 years and an average disease duration of 10±4 years. The full details of the subjects are shown in Table I. The patients' dyskinesia was assessed using a modified Abnormal Involuntary Movement Scale (mAIMS). A rating ranging from 0 to 4, indicating the absence of dyskinesia to severe dyskinesia, is assigned by a neurologist to each of the four limbs, head/neck, trunk, and the overall body. The cumulative mAIMS score is derived by adding up these individual sub-scores, yielding a total score that falls within the range of 0 to 28. The mAIMS represents the regression score in the PD dataset. The average mAIMS in the dataset was to 8.6±3.7. The study protocol [31], [32] was approved by the institutional review boards of the University of Rochester. All participants in the study signed the informed consent form.

Two 3-axial inertial sensors from Great Lakes NeuroTechnologies Inc., Cleveland, OH, were placed on the participants, one on the most affected wrist and one on the most affected ankle. The sampling frequency of the sensors was $f_s = 64$ Hz. The participants were asked to perform a set of daily living actives at every hour for 3-4 rounds and over a four-hour duration. This setup resulted in a total of 58 rounds, with each round being, on average 13.7±1.6 minutes in duration. At each round, a movement disorder neurologist rated the participants' dyskinesia severity using the mAIMS score. mAIMS ranges from 0

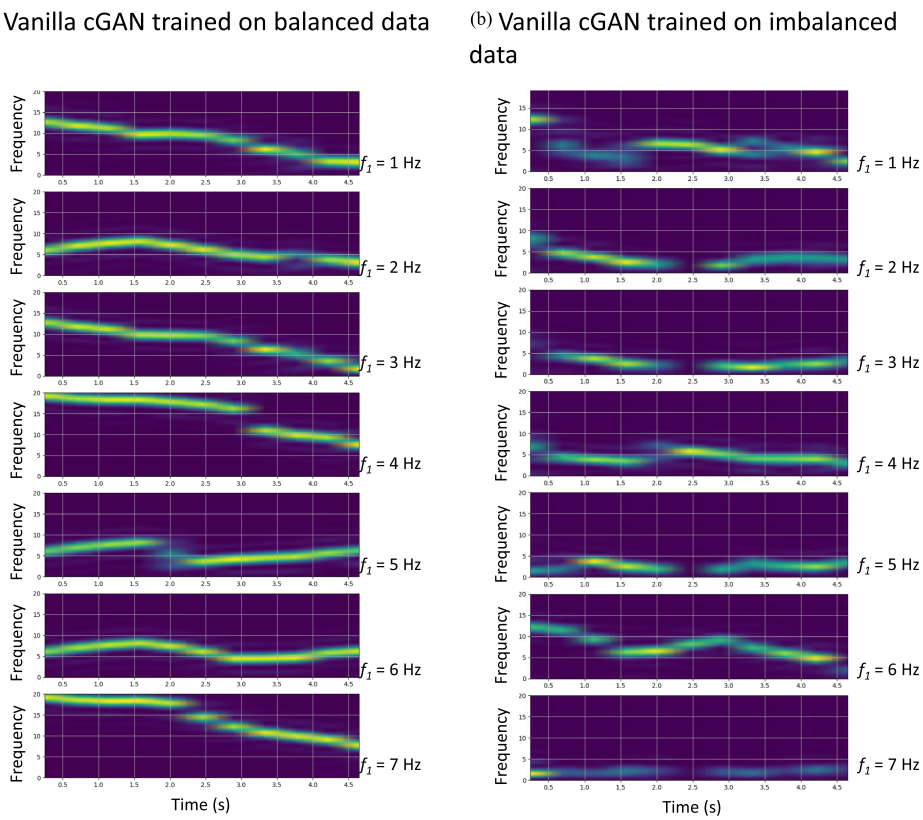

Fig. 5. Some generated samples' spectrograms are shown. Panel A is with cGAN from the balanced synthetic dataset, and Panel B is from the imbalanced one. Panel B shows mode collapse when changing the latent vector or the condition f_1. In B, the cGAN generates samples with unmatched final frequency to the condition f_1.

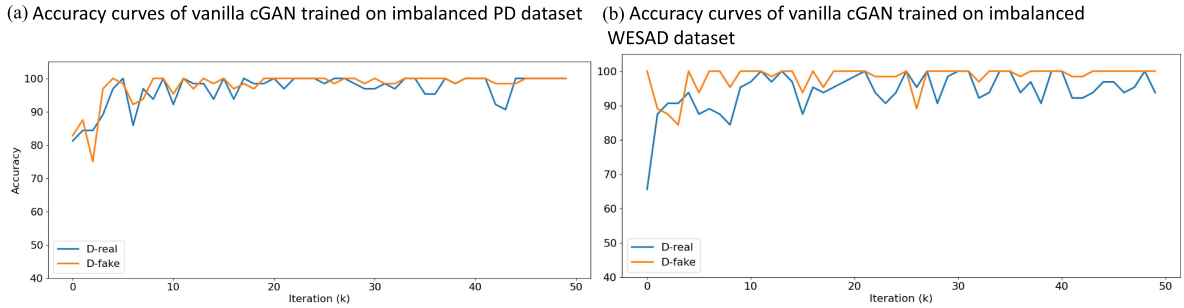

Fig. 6. (a) Vanilla cGAN accuracy curves when trained on the PD dataset. (b) The vanilla cGAN accuracy curves when trained on the WESAD dataset. The discriminator D started overfitting real and fake samples quickly with a detection accuracy $> 90\%$ that caused cGAN training to fail.

for no dyskinesia and 28 for severe dyskinesia. The data were segmented into windows with a 5-second duration and no overlaps. This process resulted in 2,280 data samples with a duration of 320 samples and a dimension of 6, indicating the number of axes in two sensors ($w^{(i)} \in \mathbb{R}^{320\times 6}$). Real segments from the three axes of the accelerometer are shown in Fig. 3(a). Finally, the mAIMS score of each round was used as the ground-truth regression score of all the 5-second time-series data samples segmented from that recording round.

C. WESAD Dataset

We used WESAD [33], a publicly available dataset with multimodal physiological data recorded during different affective states. The data were recorded from the chest using a wearable sensor device to measure respiration, electrocardiogram, electromyography, electrodermal activity, skin temperature, and acceleration at 700 Hz sampling frequency. Fifteen participants (3 female, 12 male) with an average age of (27.5±2.4 years) participated in the data collection. The participants underwent three conditions: baseline, stress, and amusement, followed by guided mediation. The participants completed the self-report Positive and Negative Affect Schedule (PANAS) after being exposed to each condition [34].

The PANAS questionnaire was used to score positive affect and negative affect (NA). NA scores represented a geometric distribution (Fig. 10(a)), which makes it challenging to develop regression models to estimate the NA score. Hence, we used the

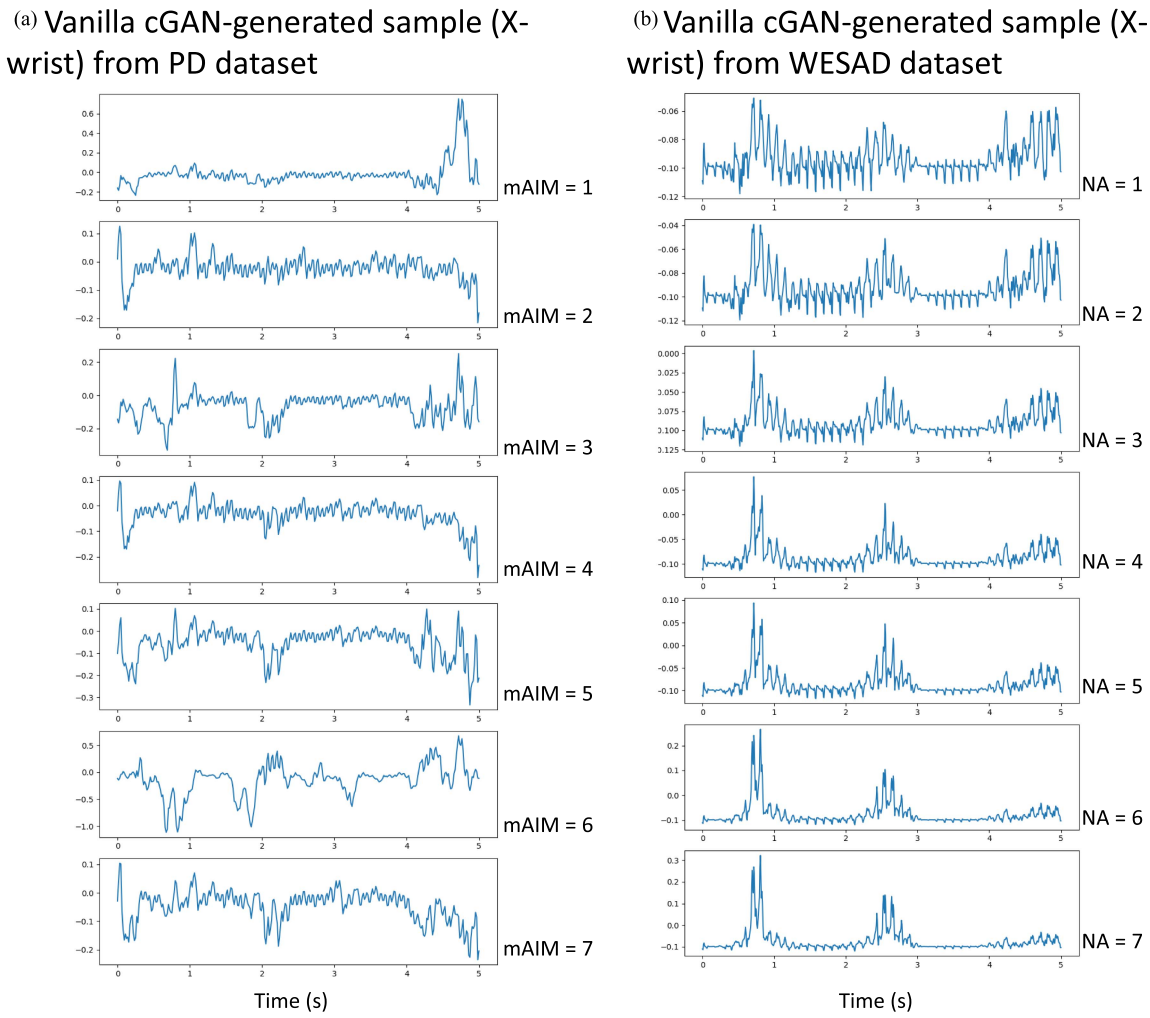

Fig. 7. Trained vanilla cGAN was used to generate X-axis data samples using the PD (a) and WESAD (b) datasets. The model suffered from mode collapse thus failed to generate diverse samples mainly due to the discriminator's overfitting and the imbalanced dataset during training.

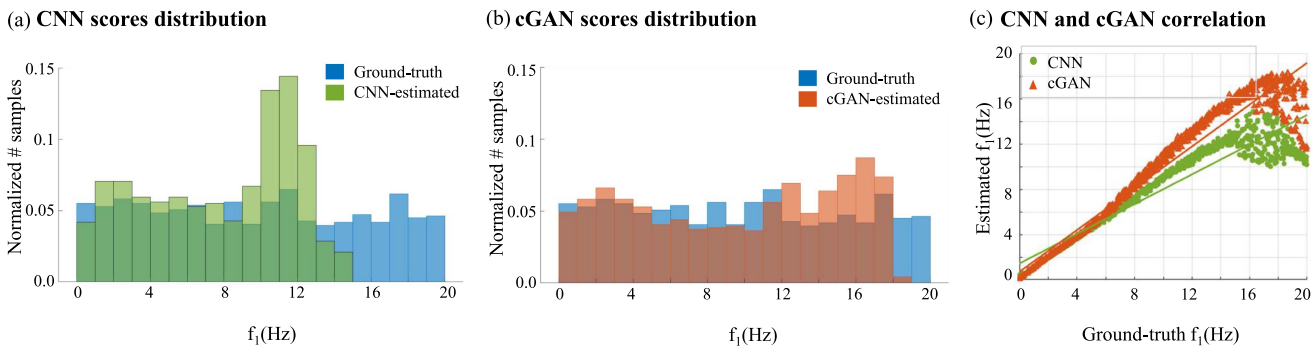

Fig. 8. Results from the imbalanced synthetic dataset: Distribution of the ground-truth vs. estimated score using (a) CNN and (b) proposed cGAN models. (c) The correlation between the ground truth and estimated scores is shown using the proposed cGAN and CNN models. The distribution of the estimated scores using the proposed cGAN was closer to uniform distribution and the ground-truth scores compared to the CNN-estimated scores.

WESAD dataset for our evaluation purposes in this study and NA as the regression score. Within PANAS, 10 items specifically gauge negative affects as distinct dimensions, including feelings of distress, annoyance, guilt, and fear. Each of these items is assigned a score ranging from 1 to 5. Consequently, The NA score ranges between 10 for calmness and 50 for subjective distress. In Hssayeni et al., for a single modality model, we showed that a vanilla CNN model on the accelerometer data resulted in the highest correlation for NA estimation [35]. Hence, in this work, we will use the acceleration signals to estimate the

TABLE I
DETAILS OF PARKINSON'S DISEASE SUBJECTS

Subject#	Age	Disease Duration	Average mAIMS
1	42	9	15.75
2	50	6	8.75
3	59	9	4.5
4	54	10	10.25
5	59	13	6
6	49	12	7.75
7	72	16	13.67
8	73	17	11.75
9	44	8	7.75
10	77	9	7.25
11	59	3.5	4.25
12	68	3.5	8.75
13	51	4	2.25
14	58	14	7
15	65	15	13.5

NA scores as our test setup. Our prepossessing involved applying a 0.1-64 Hz bandpass filter on the acceleration signals. The data were downsampled by two and segmented into windows with a duration of one minute (714 data samples) and a dimension of 3, indicating the number of axes ($w^{(i)} \in \mathbb{R}^{21k\times 3}$). Real acceleration segments are shown in Fig. 3(b).

VI. RESULTS AND DISCUSSION

We first investigated the signal generation ability of a vanilla cGAN and any training issues. Next, we evaluated the performance of our proposed cGAN and compared it with CNN. We made the codes for training and testing our modified cGAN architecture proposed available to the public on Github [30].

A. Vanilla cGAN for Data imputation

First, we trained a vanilla cGAN model on the balanced synthetic dataset with uniformly distributed scores to ensure it can generate expected signals. Second, we replaced the training data with the imbalanced synthetic dataset with geometrically distributed scores, PD dataset, and WESAD dataset to investigate the performance of the vanilla cGAN for signal generation and any issues with mode collapse.

The balanced and imbalanced synthetic datasets each had 1,000 data samples to ensure they only differed on the score distribution. Using Dynamic Time Warping (DTW) metric, we assessed the quality of the generated signals. The DTW metric has commonly been used in the literature by the community to evaluate the quality of the GAN-generated time series data [36], [37]. DTW measures similarity between two time-series signals with lower DTW scores indicating a higher similarity of the generated signals to the real ones.

The average DTW scores of a batch of the generated and a batch of real signal samples were calculated after every 1,000 training iterations. The accuracy and DTW curves for the synthetic datasets were shown in Fig. 4(a) and (b), respectively. One observation was that the vanilla cGAN did not overfit the balanced data until 10,000 iterations, while it overfitted the imbalanced data after 1,000 iterations with the discriminator accuracy reaching above 90%. Another observation was that the DTW score was consistently higher for the cGAN trained on the imbalanced data, which means a worse similarity to the real data. Fig. 4 also shows that the discriminator D of both balanced and imbalanced data overfitted the training data because the training data is limited.

The generator with the lowest DTW measure was saved. The saved generator generated chirps with ending frequencies (f_1) between 1–20 Hz. The spectrograms of some generated samples are displayed in Fig. 5(a) and (b) using the balanced and imbalanced training dataset, respectively. The cGAN trained on the balance data generated chirp signals with the expected ending frequencies. The initial frequencies also covered a wide range, indicating that the model learned the random pattern of the starting frequency. However, the cGAN trained on the imbalanced data faced mode collapse as clearly shown in Fig. 5(b) for set frequencies of f_1 as 2–5 Hz. Moreover, the imbalanced dataset's generator could not capture the patterns of the final frequency. Hence, we conclude that the imbalanced training data was another reason for mode collapse besides limited data and the increase in the number of classes.

We repeated this process by training the vanilla cGAN on the two real-world datasets. Fig. 6 provides the training accuracy curves. The discriminator started overfitting early, leading to mode collapse, as seen from the sample-generated signals in Fig. 7. These generated samples are not useful for data imputation and could even degrade the newly trained model's performance. Therefore, the vanilla cGAN fails to improve the generalizability of regression models in incomplete time-series data applications.

B. Proposed cGAN Model vs. CNN for Regression Score Estimation

The investigation of a vanilla cGAN revealed its limitations as a data imputation method when dealing with imbalanced and incomplete training data. Here, we use our proposed modified cGAN architecture for estimating regression scores without performing any data imputation and compare its performance with a CNN model. We applied each method to the imbalanced synthetic and real-world datasets.

For the synthetic dataset, the models were trained using the imbalanced synthetic training set and evaluated using the balanced testing set. For the PD and WESAD datasets, the models were trained and tested using leave-one-participant-out cross-validation to ensure no data would leak from the training to the testing set. This is important as the intra-window similarity within each participant is high, and if we randomly shuffle the windows to construct a held-out set for testing, we cannot assess the models' generalizability on unseen data. We used 20% of the training data as our validation set to optimize the model's hyperparameters. We applied the trained model to the held-out test data and compared the estimated regression scores from the model to the ground-truth scores. Pearson correlation, r, and mean absolute error, MAE, were used to measure performance.

Keras library with TensorFlow backend was used to implement the CNN, and cGAN models in Python [38]. We set cGAN latent space to 100 and trained it using $k = 35 \times 10^3$ iterations

TABLE II
TESTING PERFORMANCE OF THE PROPOSED CGAN MODEL AND CNN ON THREE IMBALANCED AND INCOMPLETE DATASETS

Method	Imbalanced synthetic dataset (the ending frequency of chirp signals)		PD dataset (dyskinesia scores from sensor data)		WESAD dataset (negative affect from sensor data)	
	correlation	mean absolute error	r	MAE	r	MAE
CNN model	0.95	1.93	0.85	2.68	0.56	3.30
Proposed cGAN	**0.97**	**0.84**	**0.88**	**1.77**	**0.73**	**2.71**

The bold values highlight the top-performing methods across different datasets scenarios.

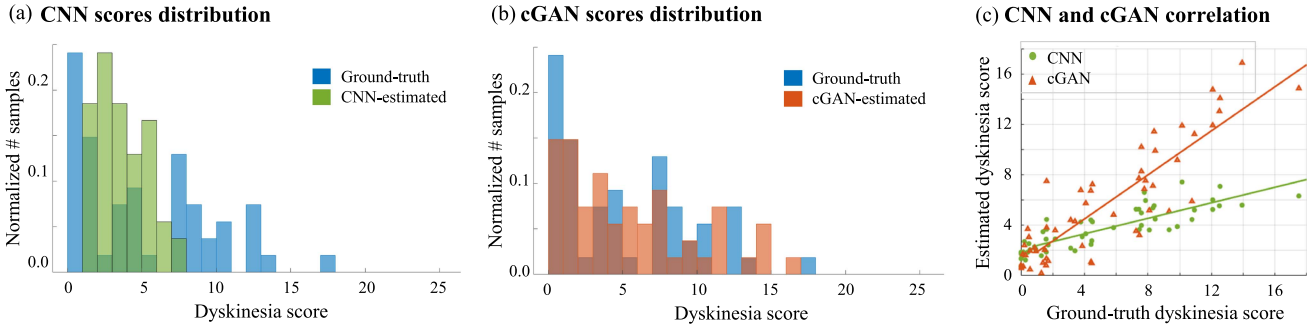

Fig. 9. Results from the PD dataset: Distribution of the ground-truth vs. estimated scores using (a) CNN and (b) proposed cGAN models. (c) The correlation between the ground truth and estimated scores is shown using the proposed cGAN and CNN models. The estimated scores using the proposed cGAN showed a geometric distribution similar to the ground-truth scores compared to the CNN-estimated scores, which had a much narrower range.

with mini-batches of size 32. A grid search was performed to select the depth of the CNNs and cGAN with the highest performance on the validation data. We increased the models' depth by up to four times, repeating Conv Block-2. Next, we applied the selected model to the held-out test data. During testing, we applied G ten times on each data segment using a different noise vector and used the average of the ten estimated scores as the final score following the work of [27].

First, we evaluated the performance of the proposed cGAN architecture and CNN model for estimating the regression score on the imbalanced synthetic dataset as shown in Fig. 8. The distribution of the estimated score from each model is shown over the distribution of the ground-truth scores provided in Fig. 8(a) and (b), respectively, for the CNN and proposed cGAN models. The correlation performance of the models is illustrated in Fig. 8(c). The following are our observations. The proposed cGAN resulted in a correlation of 0.97 and 0.84 MAE, outperforming the CNN model with a 0.95 correlation and 1.93 MAE. The CNN model was challenged to estimate unseen regression scores and did not estimate any regression scores beyond 15 Hz, with most of the samples being under 12 Hz. However, the proposed cGAN had improved performance and could estimate regression scores close to 20 Hz. CNN and the proposed cGAN could interpolate the missing scores of 5–8 Hz. However, the proposed cGAN outperformed CNN by extrapolating to unseen regression scores $>$ 15 Hz. The estimated regression scores' distribution from our proposed cGAN was closest to the ground-truth distribution (Fig. 8(b)).

Next, we applied the two models on the PD dataset to estimate dyskinesia scores and the WESAD dataset to estimate NA scores. Similarly, using each model, we showed the estimated regression scores' distribution over the ground truth distribution in each dataset. Fig. 9 provides the results for the PD dataset, and Fig. 10 for the NA estimation in the WESAD dataset. Consistent with what we observed in the case of the synthetic dataset, the proposed cGAN was able to estimate samples with a distribution closer to the ground-truth distribution than the CNN model. Interestingly, the CNN model could not correctly estimate dyskinesia scores for scores greater than 8, while the proposed cGAN estimated scores up to 17. The proposed cGAN model on the PD datasets resulted in a correlation of 0.88 and 1.77 MAE with the ground-truth scores, better than the CNN model with a 0.85 correlation and MAE of 2.68. Similarly, the proposed cGAN outperformed CNN for NA estimation with $r = 0.73$ and 2.71 MAE compared to $r = 0.56$ and 3.30 MAE.

In all cases, the proposed cGAN offered a significant improvement over CNN with a 56%, 34%, and 18% improvement in MAE, respectively, for the synthetic, PD, and WESAD datasets (see Table II). The average MAE of the estimated scores over every five score intervals is illustrated in Fig. 11. As indicated by the increasing gap between the MAE of the proposed cGAN and CNN models for the higher scores (i.e., minority scores), we can conclude that the improvement of our model vs. CNN was even more evident in data ranges with minority scores.

For comparison purposes, it is interesting to mention that in a prior work [39], a bidirectional LSTM was able to estimate dyskinesia scores of $r = 0.87$ correlation and MAE of 1.74, which was comparable to the performance of the proposed cGAN; however, similar to CNN, the bidirectional LSTM was not able to extrapolate beyond dyskinesia score of 10. Fig. 12 shows the

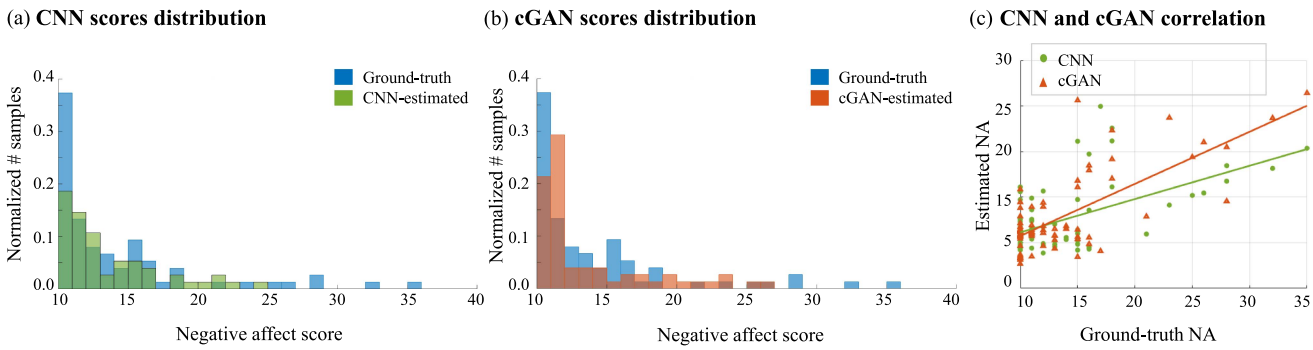

Fig. 10. Results from the WESAD dataset: Distribution of the ground-truth vs. estimated scores using (a) CNN and (b) proposed cGAN models. (c) The correlation between the ground truth and estimated scores is shown using the proposed cGAN and CNN models. The distribution of estimated scores using both models had a closer geometric distribution to the ground-truth scores, but the estimations using the proposed cGAN were more accurate, as seen in (c).

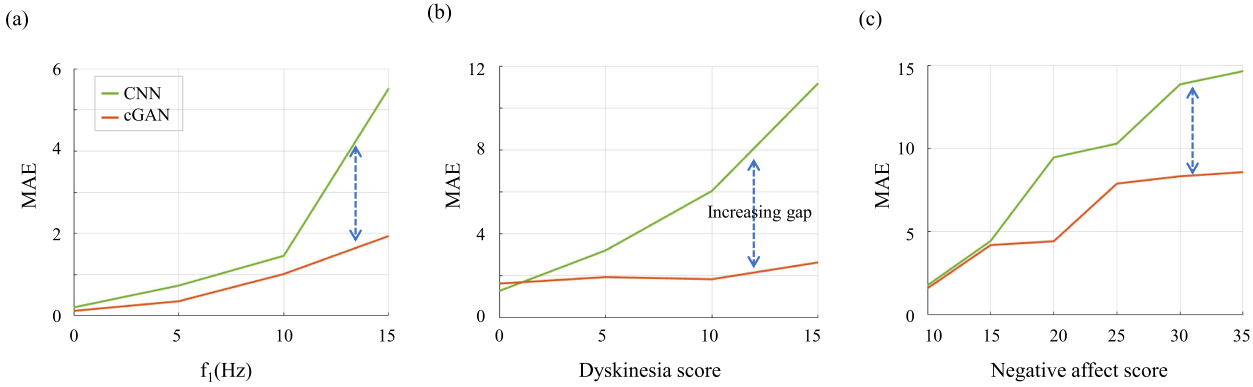

Fig. 11. At every five score interval, the MAE for the proposed cGAN (red) and CNN (green) models was calculated and displayed for estimation of the (a) f_1 frequency in the synthetic dataset, (b) dyskinesia score in the PD dataset, and (c) NA scores in the WESAD dataset. Since the score distribution is geometric, the slow decrease in the proposed cGAN's performance compared to the CNN model's performance for the higher score samples indicates its improved generalizability to minority data.

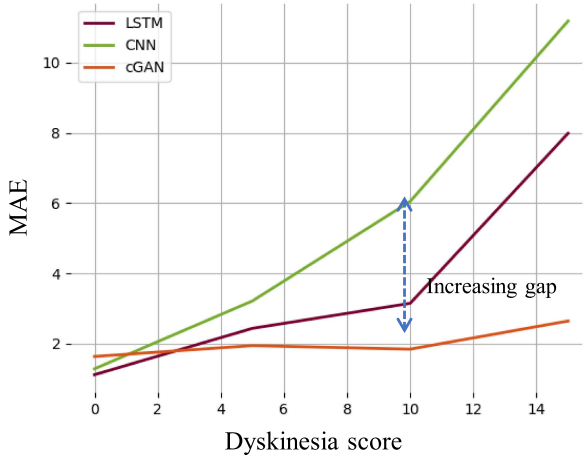

Fig. 12. MAE for the proposed cGAN (red), CNN (green) models, and LSTM (brown) for multiple ranges of scores.

significant increase in MAE when estimating higher dyskinesia scores in comparison to CNN and cGAN. The performance of the LSTM model was better than CNN but significantly lower than cGAN. The LSTM was evaluated using the same PD dataset used in our current work and the same test setting.

Our results demonstrate the improvement we gained using the proposed cGAN framework in terms of generalizability to unseen data samples. Our findings about the improved performance of the modified cGAN architecture were consistent with the observations on noisy cross-sectional data as reported in [27]. Below, we summarize our main observations:

- Vanilla cGAN may lead to early mode collapse when trained using imbalance time-series data. As indicated by visual inspections, the generated samples from the minority scores may not be consistent with the time-series patterns related to the scores. As a result, vanilla cGNA-generated samples may not improve generalizability over unseen ranges and could hurt the regression models' performance.
- The proposed cGAN architecture was able to train a regression model using incomplete and imbalanced time-series data, as evidenced by the improved generalizability to unseen data compared to CNN.
- The proposed method learned the real distribution of the regression scores better than CNN could better interpolate its estimation to the missing ranges during the training process.
- There is still room for improvement. Our future work involves extending the proposed framework by imposing the range of the expected regression scores further to enhance the extrapolation ability to the unseen regression range.

VII. Conclusion

This paper proposed a novel solution for addressing deep regression models' extrapolation and generalizability challenges in imbalanced and incomplete time-series data. We demonstrated that vanilla cGAN suffers from severe mode collapse and cannot be used as a data imputation method in regression applications with time-series data when the available data is limited or imbalanced. Our next contribution was the development of a new formulation of cGAN to address this limitation and improve the generalizability of deep regression models. Our framework conditioned the cGAN on the raw signals instead of the regression scores and embedded a CNN model in the generator to learn the data distribution and patterns associated with different regression scores. This method trained the generator indirectly through the discriminator to attenuate overfitting. The application of the proposed cGAN on three imbalanced, incomplete datasets illustrated the ability of this new approach to learn the distribution of different data types and accurately estimate minority and unseen scores. The model was compared to a CNN model, which in all the datasets, exhibited lower extrapolation and generalizability abilities to unseen samples, further indicating the effectiveness of our proposed deep regression model for imbalanced and incomplete time-series data.

References

[1] Y. Zhang, B. Kang, B. Hooi, S. Yan, and J. Feng, "Deep long-tailed learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 9, pp. 10795–10816, Sep. 2023.

[2] K. D Angeli et al., "Class imbalance in out-of-distribution datasets: Improving the robustness of the TextCNN for the classification of rare cancer types," *J. Biomed. Inform.*, vol. 125, 2022, Art. no. 103957.

[3] G. Holste et al., "Long-tailed classification of thorax diseases on chest X-ray: A new benchmark study," in *Proc. MICCAI Workshop Data Augmentation, Labelling, Imperfections*, 2022, pp. 22–32.

[4] W. Park, I. Park, S. Kim, and J. Ryu, "Robust asymmetric loss for multi-label long-tailed learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2023, pp. 2703–2712.

[5] D. M. Hawkins, "The problem of overfitting," *J. Chem. Inf. Comput.*, vol. 44, no. 1, pp. 1–12, 2004.

[6] T. Alam, C. F. Ahmed, S. A. Zahin, M. A. H. Khan, and M. T. Islam, "An effective recursive technique for multi-class classification and regression for imbalanced data," *IEEE Access*, vol. 7, pp. 127615–127630, 2019.

[7] J. M. Johnson and T. M. Khoshgoftaar, "Survey on deep learning with class imbalance," *J. Big Data*, vol. 6, no. 1, pp. 1–54, 2019.

[8] I. Goodfellow et al., "Generative adversarial nets," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2014, pp. 2672–2680.

[9] H. Alqahtani, M. Kavakli-Thorne, and G. Kumar, "Applications of generative adversarial networks (GANs): An updated review," *Arch. Comput. Methods Eng.*, vol. 28, no. 2, pp. 525–552, 2021.

[10] H. F. Nweke, Y. W. Teh, M. A. Al-Garadi, and U. R. Alo, "Deep learning algorithms for human activity recognition using mobile and wearable sensor networks: State of the art and research challenges," *Expert Syst. Appl.*, vol. 105, pp. 233–261, 2018.

[11] T. T. Um et al., "Data augmentation of wearable sensor data for Parkinson's disease monitoring using convolutional neural networks," in *Proc. ACM Int. Conf. Multimodal Interaction*, 2017, pp. 216–220.

[12] Q. Wen, L. Sun, X. Song, J. Gao, X. Wang, and H. Xu, "Time series data augmentation for deep learning: A survey," 2020, *arXiv:2002.12478*.

[13] M. D. Hssayeni, J. Jimenez-Shahed, and B. Ghoraani, "Dyskinesia estimation of imbalanced data using a deep-learning model," in *Proc. IEEE 44th Annu. Int. Conf. Eng. Med. Biol. Soc.*, 2022, pp. 3195–3198.

[14] G. Haixiang, L. Yijing, J. Shang, G. Mingyun, H. Yuanyue, and G. Bing, "Learning from class-imbalanced data: Review of methods and applications," *Expert Syst. Appl.*, vol. 73, pp. 220–239, 2017.

[15] T. Zhu, Y. Lin, Y. Liu, W. Zhang, and J. Zhang, "Minority oversampling for imbalanced ordinal regression," *Knowl.-Based Syst.*, vol. 166, pp. 140–155, 2019.

[16] P. Branco, L. Torgo, and R. P. Ribeiro, "Pre-processing approaches for imbalanced distributions in regression," *Neurocomputing*, vol. 343, pp. 76–99, 2019.

[17] M. Steininger, K. Kobs, P. Davidson, A. Krause, and A. Hotho, "Density-based weighting for imbalanced regression," *Mach. Learn.*, vol. 110, pp. 2187–2211, 2021.

[18] Y. Yang, K. Zha, Y. Chen, H. Wang, and D. Katabi, "Delving into deep imbalanced regression," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 11842–11851.

[19] R. R. Sarra, A. M. Dinar, M. A. Mohammed, M. K. A. Ghani, and M. A. Albahar, "A robust framework for data generative and heart disease prediction based on efficient deep learning models," *Diagnostics*, vol. 12, no. 12, 2022, Art. no. 2899.

[20] O. I. Khalaf, A. SR, S. Dhanasekaran, and G. M. Abdulsahib, "A decision science approach using hybrid EEG feature extraction and GAN-based emotion classification," *Adv. Decis. Sci.*, vol. 27, no. 1, pp. 172–191, 2023.

[21] X. Ning, L. Yac, X. Wang, B. Benatallah, M. Dong, and S. Zhang, "Rating prediction via generative convolutional neural networks based regression," *Pattern Recognit. Lett.*, vol. 132, pp. 12–20, 2020.

[22] O. Janeh, G. Bruder, F. Steinicke, A. Gulberti, and M. Poetter-Nerger, "Analyses of gait parameters of younger and older adults during (non-) isometric virtual walking," *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 10, pp. 2663–2674, Oct. 2018.

[23] M. Mirza and S. Osindero, "Conditional generative adversarial nets," 2014, *arXiv:1411.1784*.

[24] G. Douzas and F. Bacao, "Effective data generation for imbalanced learning using conditional generative adversarial networks," *Expert Syst. Appl.*, vol. 91, pp. 464–471, 2018.

[25] T. Karras, M. Aittala, J. Hellsten, S. Laine, J. Lehtinen, and T. Aila, "Training generative adversarial networks with limited data," in *Proc. Adv. Neural Inf. Process. Syst.*, 2020, pp. 12104–12114.

[26] M. Shahbazi, M. Danelljan, D. P. Paudel, and L. Van Gool, "Collapse by conditioning: Training class-conditional GANs with limited data," 2022, *arXiv:2201.06578*.

[27] K. Aggarwal, M. Kirchmeyer, P. Yadav, S. S. Keerthi, and P. Gallinari, "Conditional generative adversarial networks for regression," *Cs Stat.*, vol. 133, no. 10, pp. 142–146, 2019.

[28] T. Salimans, I. Goodfellow, W. Zaremba, V. Cheung, A. Radford, and X. Chen, "Improved techniques for training GANs," in *Proc. Annu. Conf. Neural Inf. Process. Syst.*, 2016, pp. 2234–2242.

[29] X. Meng, A. Jakobsson, X. Li, and Y. Lei, "Estimation of chirp signals with time-varying amplitudes," *Signal Process.*, vol. 147, pp. 1–10, 2018.

[30] M. Hssayeni, "Imbalanced time-series data regression using conditional generative adversarial networks," 2022. [Online]. Available: https://github.com/Murtadha44/cGAN_vs_CNN_for_time_series_regression

[31] T. O. Mera, M. A. Burack, and J. P. Giuffrida, "Objective motion sensor assessment highly correlated with scores of global Levodopa-induced dyskinesia in Parkinson's disease," *J. Parkinsons Dis.*, vol. 3, no. 3, 2013, Art. no. 399.

[32] C. L. Pulliam, M. A. Burack, D. A. Heldman, J. P. Giuffrida, and T. O. Mera, "Motion sensor dyskinesia assessment during activities of daily living," *J. Parkinsons Dis.*, vol. 4, no. 4, pp. 609–615, 2014.

[33] P. Schmidt, A. Reiss, R. Duerichen, C. Marberger, and K. V. Laerhoven, "Introducing wesad, a multimodal dataset for wearable stress and affect detection," in *Proc. ACM Int. Conf. Multimodal Interaction*, 2018, pp. 400–408.

[34] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of positive and negative affect: The PANAS scales," *J. Pers. social Psychol.*, vol. 54, no. 6, 1988, Art. no. 1063.

[35] M. D. Hssayeni and B. Ghoraani, "Multi-modal physiological data fusion for affect estimation using deep learning," *IEEE Access*, vol. 9, pp. 21642–21652, 2021.

[36] A. M. Delaney, E. Brophy, and T. E. Ward, "Synthesis of realistic ECG using generative adversarial networks," 2019, *arXiv:1909.09150*.

[37] R. A. Zanini and E. L. Colombini, "Parkinson's disease EMG data augmentation and simulation with DCGANs and style transfer," *Sensors*, vol. 20, no. 9, 2020, Art. no. 2605.

[38] M. Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015. [Online]. Available: http://tensorflow.org/

[39] M. D. Hssayeni, J. Jimenez-Shahed, M. A. Burack, and B. Ghoraani, "Dyskinesia estimation during activities of daily living using wearable motion sensors and deep recurrent networks," *Sci. Rep.*, vol. 11, no. 1, pp. 1–12, 2021.

Murtadha D. Hssayeni was born in Babil, Iraq, in 1990. He received the B.Sc. degree in computer engineering from the University of Technology-Baghdad, Baghdad, Iraq, in 2012, the M.Sc. degree in in computer engineering from the Rochester Institute of Technology (RIT), Rochester, NY, USA, in 2017, and the Ph.D. degree in computer engineering from Florida Atlantic University (FAU), FL, USA, in 2017. He is currently an Assistant Professor with the University of Technology-Baghdad. He became a Postdoctoral Fellow. During his studies, he worked on multiple biomedical signal and image analysis laboratories, and published several research papers in the field, making his code and collected data publicly available. His research interests include biomedical signal and image processing, tensor decomposition, and machine learning, especially deep learning. His work covers various areas, including continuous estimation of Parkinson's Disease symptoms, activity recognition, detection of cardiac dysfunction, and intracranial hemorrhage segmentation.

Behnaz Ghoraani (Senior Member, IEEE) received the Electrical and Computer Engineering Ph.D. degree from Ryerson University, Toronto, ON, Canada, in 2010. She is currently an Associate Professor with the Department of Computer and Electrical Engineering, Florida Atlantic University (FAU), Boca Raton, FL, USA. Before joining FAU, she was an Assistant Professor with the Department of Biomedical Engineering, Rochester Institute of Technology, Rochester, NY, USA, from 2012 to 2016. From 2010 to 2012, she was a Postdoc with the College of Medicine, University of Toronto, Toronto. Her research interests include generating clinically relevant engineering solutions to tackle significant bottlenecks in data analytics with an emphasis on computer-aided clinical decision-making, long-term and continuous health monitoring, remote and personalized therapeutic management, non-stationary and multidimensional signal analysis, adaptive signal feature extraction, and traditional and deep-learning machine learning.