Improving posture recognition among construction workers through data augmentation with generative adversarial network

J Zhao, E Obonyo and Q Yin

Architectural Engineering Department, The Pennsylvania State University, University Park, PA 16802, USA

qinyin@psu.edu

Abstract. Deep Neural Networks (DNN) models have shown high potential in recognizing workers' risky postures using data from wearable Inertial Measurement Units (IMUs). However, there is a data paucity challenge - DNN models require a large dataset with annotation for desirable performance. The research discussed in this paper proposes to address this problem through a data generation framework that leverages Generative Adversarial Network (GAN) to i) synthesize motion data, ii) augment training data, then iii) improve the recognition performance. Its potential was validated using naturalistic posture data of workers. Three GAN models were developed for data generation. A Train on Real and Test on Hybrid approach was used to quantitatively assess synthesized data and select sufficiently-trained GAN models. The performance of three commonly-used DNN models was compared after data augmentation. Results showed that the augmentation with GAN-synthesized data improved recognition accuracy by 1.2%-3% for varying postures. These findings suggest the feasibility of applying motion data augmentation with GAN models to advance automated construction safety monitoring.

1. Introduction

Machine Learning (ML)-enhanced posture recognition based on Wearable Sensor (WS)-acquired motion data can be used to reduce the risk of injury on the job site. Construction workers executing manual-intensive tasks are highly susceptible to Musculoskeletal Disorders (MSDs) due to overexposure to awkward postures [1]. Automated posture recognition from WS could help mitigate MSDs through early detection of risk exposure. Applying data-driven models with wearable Inertial Measurement Units (IMUs) has demonstrated promising results for posture recognition among workers in several studies leveraging both ML models [2-4] and Deep Neural Networks (DNN) models [5-7]. DNN models, in particular, are becoming increasingly popular in Time Series Classification [8]. Some studies also demonstrate that DNN models show better recognition performance than conventional ML models for recognizing construction activities [5, 7, 9]

The key to desirable DNN model performance is the availability of abundant labelled motion data [8, 10]. There is a prevailing data paucity challenge, which can lead to an overfitted DNN model with limited model generality. Obtaining labelled motion data have proven challenging because: 1) manual motion data annotation can be both inefficient and expensive in practice [10]; 2) data for emergent and unexpected events (e.g., accidentally fall) are especially hard to obtain [11], resulting in annotation scarcity and class imbalance, and; 3) sharing sensitive personal data when using WS may lead to privacy concerns and a reluctance among uses to share data [12].

The challenge of data paucity could be largely addressed through augmenting WS-based posture data. One promising augmentation approach is applying learning-based generative models [8], which generate realistic fake data through learning the latent distribution of real data. The principle of adversarial training has led to a massively popular generative modelling framework known as Generative Adversarial Network (GAN). GAN is a type of DNN integrating both generator and discriminator [13]. The generator uses the sampled noise data as input to produce fake data that are similar to the real data distribution as much as possible. The discriminator takes both the generated fake data from the generator and real data, where the goal is to determine whether the input data are real or fake. Both the generator and discriminator are trained together by playing the zero-sum game until they respectively converge. Since its introduction, GAN has led the way in generating high-quality output and breakthroughs in image generation [14].

GAN-based data generation has been successfully deployed in time-series data [1]. GAN-based sensory data generation focuses primally on augmenting sensory data (e.g., EEG, ECG, and IMUs) for improving patients' health monitoring. A review of closely related studies has been provided in Table 1. Deep Convolutional GAN (DCGAN) and Recurrent GAN (RGAN) are commonly used architectures for time-series data generation. DCGAN, designed for image generation, has demonstrated high potential in generating time-series data. Recurrent GAN [15] is designed specifically for generating time-series data. Comparing to regular GANs relying on full-connect layers alone, RGAN incorporates the recurrent layers and its variants, such as Long Short-Term Memory (LSTM) and Bi-directional LSTM, in generator and discriminator. RGAN frameworks have shown the ability to generate high-quality time-series data.

Review in Table 1 shows that evaluating time-series data produced by the generative model is a difficult task. This is due to the vague definition of "realistic" and impractical visual assessment. Alternative quantitative assessment can be done by evaluating the "utility" of generated data. For example, if a classification model trained on simulated data from GANs achieves comparable performance on testing data to that obtained through training with real data, then generated data can be deemed to be of high quality [16, 17]. Such an approach was proposed as Train on Simulated Test on Real (TSTR) [15]. Its use under different evaluation scenarios has been considered effective. The Train on Hybrid and Test on Real (THTR) approach is widely adopted in data augmentation. When the original training datasets are combined with the data generated from trained GAN models, they can be used as the augmented Hybrid dataset for model training. The effectiveness of the augmentation can be done by comparing trained models between THTR and TRTR (Train on Real Test on Real).

Table 1. Review of related studies.

Study	Sensory Data	GAN Architecture	Signal Quality Evaluation	Augmentation Evaluation
[12]	IMUs	RGAN (LSTM)	Monitoring training loss	NA
[1]		DCGAN & RGAN (BiLSTM/LSTM)	Visual comparison [a]	THTR
[15]	Medical Data	RGAN (LSTM)	Quantitative comparison	TSTR
[14]	EEG	RGAN (LSTM)	Quantitative and visual	THTR
[20]	ECG & EEG	RGAN (LSTM)	Visual comparison	THTR
[21]		RGAN (LSTM)	Quantitative and visual	THTR
[16]	ECG	RGAN (LSTM)	Visual comparison	THTR
[22]		RGAN (Bi-LSTM)	Quantitative and visual	NA
[18]		DCGAN	Visual comparison	THTR

[a] Visually compared the generated data with real data.

[23]	DCGAN	NA	THTR
[17]	DCGAN	Quantitative (TSTR)	THTR

The research discussed in this paper is directed at addressing the data paucity challenge associated with the use of DNN-based posture recognition models through the outlined data augmentation approach. The authors propose to use a data generation framework that leverages GAN-based generative models to synthesize IMUs-based motion data, augment training data, and improve the recognition performance of DNN models. The proposed approach was validated using naturalistic posture data obtained from workers on construction jobsites. Three GAN architectures were developed for data generation. A Train on Real Test on Hybrid (TRTH) approach was proposed to quantitatively assess the quality of synthesized data. The feasibility of data augmentation with GAN was assessed by comparing the performances of commonly used DNN models before and after data augmentation. In the following section, the authors describe the methodology that was used to conduct the research. The results obtained from experiments are then discussed, followed by the conclusion and future works.

2. Research Methodology

2.1. Motion Data Collection and Pre-Processing

Seven construction workers are the subjects (S1-S7) in this study for motion data collection. Five IMUs sensors (Mbinet Lab Meta Motion C with units of three-axis accelerometer and gyroscope) were deployed on their hardhat (front), upper arm, chest center, right thigh, and right calf by sticking on the surface of cloth. Subjects performed their routine tasks for 20-30 minutes. Workers' postures were videotaped as ground-truth for labeling motion data. Nine commonly used postures were identified, including Bending (BT), Kneeling (KN), Squatting (SQ), Standing (ST), Walking (WK), Transitional Movement (TR), and Work Overhead (WO).

The collected motion data were down-sampled to 40 Hz and scaled to [-1, 1] using min-max normalization to address the unit difference across channels before being applied for training GAN models. Each data record was labeled with the video reference. This research used a 0.5-second window with 50% overlap for segmentation, resulting in 20 timestamps in a window. Each window was then annotated as the label of the majority data records it contained. Sensor output from five placements was combined, resulting in each window with 30 channels (five placements × two units/placement × three channels/unit). The collected motion data were pre-processed as 30,498 labeled windows with a dimension of 20 (data records) by 30 (channels).

2.2. Data Preparation for Experiment

Pre-processed windows were split into train (40%), real-augmentation (40%), and test (20%) datasets under stratified random sampling – a consistent ratio of posture classes was used across all the datasets. The train dataset was used to develop both recognition and generation models for postures. The real-augmentation dataset was combined with the train dataset to develop augmented posture recognition models. The latter was then used as the golden standard in the assessment of augmentation based on the use of generated data. All trained recognition models were evaluated on the test dataset to assess their performance before and after augmentation.

2.3. Setup for Posture Recognition Models

LSTM-based models and their derivatives have demonstrated high performance in recognizing activities from construction workers [5-7]. The authors used three LSTM-based architectures to develop the posture recognition model - the basic LSTM model, Bi-directional LSTM model, and Convolutional LSTM model integrating convolutional layers for automated feature learning (see model detail in Figure 1.).

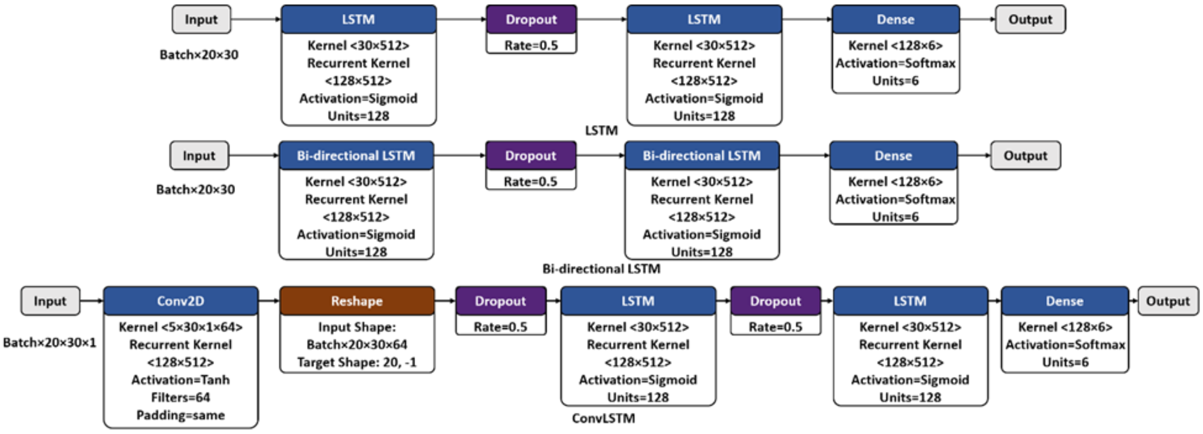


Figure 1. Comparison of model architecture.

The key difference between the three models is the DNN layers. The base LSTM model used two stacked LSTM layers. The Bi-directional LSTM model substituted the LSTM as Bi-directional LSTM layers. A convolutional layer was stacked on the base LSTM model to construct the Convolutional LSTM model. LSTM and output layers were kept the same across three implemented models to evaluate performance improvement from modified LSTM architectures.

The train dataset was further randomly split as 80% for training and 20% for validation, when training the recognition models[b]. In addition to Accuracy, Macro F1 Score was also used for evaluating classification performance, given the imbalanced dataset. The recognition model with the highest Macro F1 Score after all training epochs was saved and evaluated on the test dataset. Each recognition model was trained and assessed for five rounds with different training-validation splitting. The average recognition performance on the same test dataset was compared to evaluate the model performance.

2.4. Setup for the Posture Data Generation Models

Both DCGAN and RGAN architectures were further adapted into three generative models, namely DCGAN model, 1DRGAN model, and 2DRGAN model, as shown in Figure 2. The ConvLSTM model was used as the discriminator in the three GAN architectures because it achieved higher recognition performance among the ones that were evaluated (see Figure 3.). The deployment of the ConvLSTM-based discriminator also helped stabilize the GAN training more than the LSTM discriminator in empirical tests that were performed. The different performance of the generators for the three models is discussed further in subsequent paragraphs.

The proposed DCGAN generator was based on a 200-element vector of Gaussian random numbers as input as suggested in related studies [18, 19]. The resulting dense layer converted the input vector into a 1D representation of motion data. Output from the dense layer was then reshaped into a 3D tensor, with dimensions set to 5 (length) by 15 (width) by 128 (layers). Two consecutive Conv2DTranspose layers were applied for upsampling, which produced the tensor with the dimension of 20 by 30 by 64. A Conv2D layer with a single kernel transformed the 3D tensor into a 20 by 30 2D output. The hyperbolic tangent (tanh) activation function was used to ensure that the values of output are within the range of [-1, 1], which was the same as constructed windows of motion data.

[b] Recognition models were trained with a batch size of 300 for 300 epochs for minimizing the Categorical Cross Entropy as loss function.

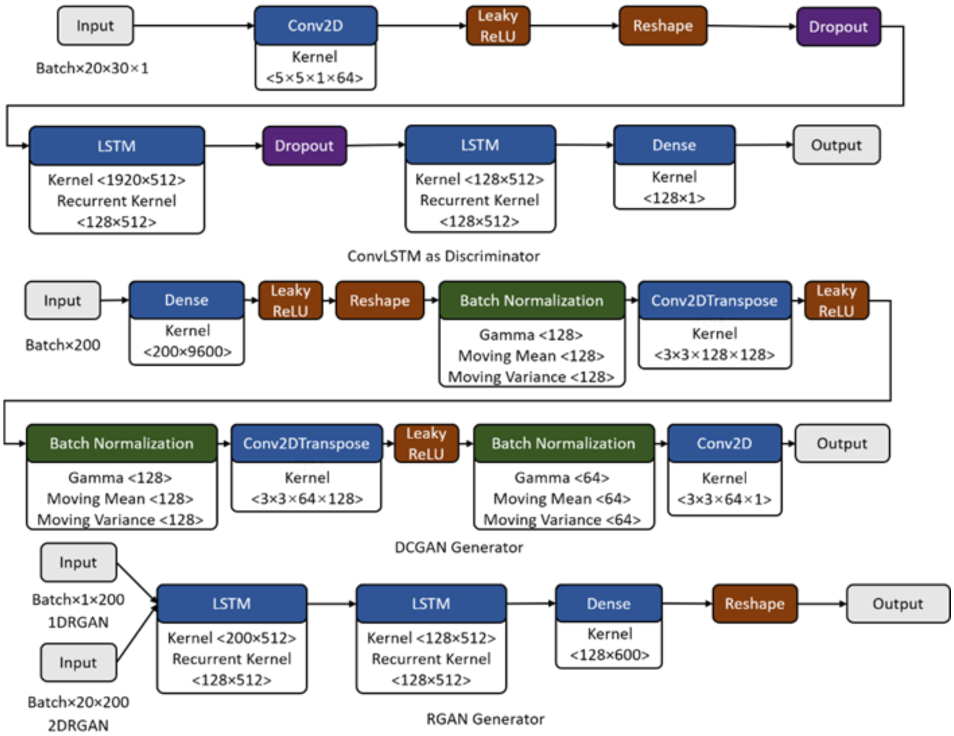


Figure 2. Comparison of GAN architectures

The generator in 1DRGAN used the same input as that in DCGAN. The RGAN model setup described in Figure 2. followed the RGAN designed for bio-signal generation in [14, 15]. Two stacked LSTM layers were used to convert the input vector into temporal patterns. The dense layer converted the output from LSTM layers as the 1D representation. The tanh activation function was also used to bound the output value within [-1, 1]. Lastly, the 1 by 600 vector was reshaped into a 20 by 30 as a generated window. The 2DRGAN model was the same as 1DRGAN except that a 2D matrix (20 by 200) of Gaussian random numbers is input, as performed in [15] to improve the generation of time-series data.

The authors also compared the GAN-based models with other learning-based generative models. Autoencoders (AE) and variational autoencoders (VAE), as used in Abdelfattah et al. [14]'s work, were deployed for benchmarking. Output from the trained decoders in AE and VAE were used as generated data – it had the same label as the input window.

Table 2. Setup for training generative models.

Generative Model	Epoch	Batch Size[c]	Loss Function[d]
DCGAN, 1DRGAN, 2DRGAN	1000	300	Minimax Loss derived from Binary Cross-Entropy
AE		30	Pixel-level Binary Cross-Entropy
VAE			Pixel-level Binary Cross-Entropy+ KL Divergence

The detailed model training setup for implemented generative models is summarized in Table 2. The generative models were trained for each posture because they were unconditional (they had no control of the label for generated data).

[c] Half batch of generated fake data and randomly selected real data were feed into the discriminators of GAN-based models to calculate the Binary Cross-Entropy.

[d] For GAN-based model, it represents the loss function for discriminator.

2.5. *Evaluating Generative Models*

2.5.1. Quality assessment of generated motion data. The TRTH approach was used to assess the generated motion data quantitatively. The real data of a certain posture class in the test dataset were replaced by data generated from a trained generative model. If a trained recognition model maintained relatively high performance (measured by accuracy) when recognizing the replaced posture in the Hybrid dataset, the generated data were deemed to be realistic with close resemblance to the real data. The trained posture recognition model was used directly under the TRTH approach, which further reduced the computational efforts of model re-training when evaluating the generative models. The TRTH evaluation was conducted after every 100 training epochs for each generative model. The ConvLSTM model was selected as the classifier because it showed the highest performance among the tested recognition models. The TRTH evaluation process was repeated five times with different generated datasets. The average performance was computed to reduce the evaluation bias.

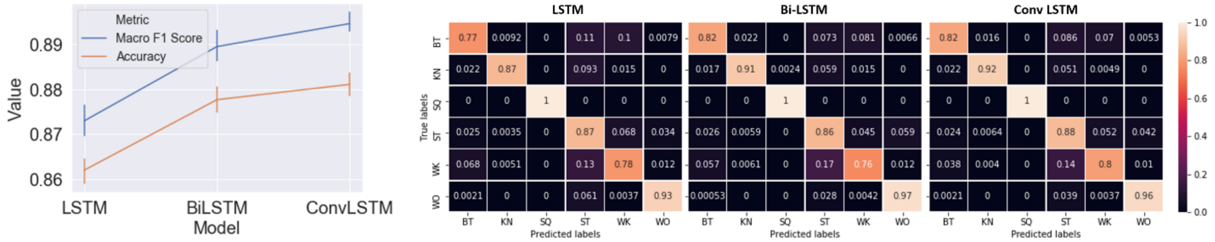
2.5.2. Evaluation of data augmentation. The entire train dataset was randomly split into two halves with stratification. The posture recognition model (trained on Train_Half_1 dataset) was evaluated using the test dataset, which was used as the baseline performance without augmentation. Motion data of a certain posture were then extracted from the Train_Half_2 subset and combined with Train_Half_1 dataset for re-training the recognition model. The trained model was evaluated against the test dataset as a baseline after augmentation with real data. Next, motion data for a given posture were generated from the trained generative models and used to augment the Train_Half_1 dataset. The recognition models were subsequently re-trained using augmented Train_Half_1 (include both generated and original Train_Half_1 datasets) and evaluated against the test dataset. The results represented the performance after augmentation with generated data. The ConvLSTM model was used for posture recognition as it showed the highest performance among the three tested recognition models. The ConvLSTM model demonstrated high recognition performance (with accuracy over 0.9) for the posture KN, SQ, and WO, as shown in Figure 3.-d. The authors selected the postures BT, ST, and WK for augmentation. The generative model showed the highest performance under the TRTH evaluation approach was identified from Figure 4. for each posture. The Macro F1 Score on the test dataset was compared before and after data augmentation. The score represents the effectiveness of data augmentation. The evaluation process was repeated five iterations using different generated datasets to reduce evaluation bias.

3. Findings and Discussion

This study is aimed at addressing the data paucity challenge through a data generation framework that leverages GAN to i) synthesize motion data, ii) augment training data, and iii) improve the recognition performance. The following sections discuss the results from the evaluation of different recognition models and their performance after data augmentation.

3.1. *Comparison of Recognition Models*

Results in Figure 3. have shown that the base LSTM model achieved a relatively high performance (Macro F1-0.873) when it was used to recognize six postures across seven different individuals. Both the modified BiLSTM and ConvLSTM further improved the recognition performance by an average of 1.95% and 2.52%, respectively. A close examination of the confusion matrix shows that the ConvLSTM model has improved LSTM model's performance in recognizing each posture.



LSTM	BT	KN	SQ	ST	WK	WO
BT	0.77	0.0092	0	0.11	0.1	0.0079
KN	0.022	0.87	0	0.093	0.015	0
SQ	0	0	1	0	0	0
ST	0.025	0.0035	0	0.87	0.068	0.034
WK	0.068	0.0051	0	0.13	0.78	0.012
WO	0.0021	0	0	0.061	0.0037	0.93

Bi-LSTM	BT	KN	SQ	ST	WK	WO
BT	0.82	0.022	0	0.073	0.081	0.0066
KN	0.017	0.91	0.0024	0.059	0.015	0
SQ	0	0	1	0	0	0
ST	0.026	0.0059	0	0.86	0.045	0.059
WK	0.057	0.0061	0	0.17	0.76	0.012
WO	0.00053	0	0	0.028	0.0042	0.97

Conv LSTM	BT	KN	SQ	ST	WK	WO
BT	0.82	0.016	0	0.086	0.07	0.0053
KN	0.022	0.92	0	0.051	0.0049	0
SQ	0	0	1	0	0	0
ST	0.024	0.0064	0	0.88	0.052	0.042
WK	0.038	0.004	0	0.14	0.8	0.01
WO	0.0021	0	0	0.039	0.0037	0.96

Figure 3. Performance comparison of recognition models.

3.2. Evaluate the Quality of Generated Data

The evaluation of generative models during the training process is provided in Figure 4. The 1DRGAN tended to have a relatively stable performance improvement during the beginning of the training process before the quality of generated data deteriorated due to overtraining (e.g., the generated BT posture data from 1DRGAN after training over 500 epochs). The recognition performance on the Hybrid test dataset was comparable to the Real test dataset, which suggests that the quality of generated data is relatively high for “tricking” a well-trained classifier.

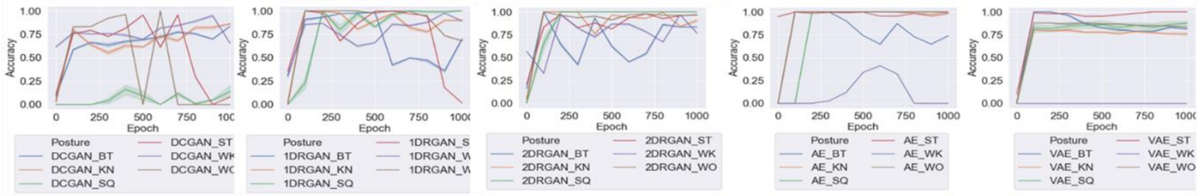


Figure 4. Evaluation of generated data

The 2DRAGN also generated high-quality motion data as 1DRGAN, while it showed instability over the training process, e.g., when being trained for posture BT. The DCGAN model required more training epochs to generate data of a similar quality to real data compared to the RGAN models on posture BT, ST, WK, and WO. The DCGAN model failed to generate the posture SQ. This is evident by that DCGAN showed low performance on Hybrid test data when generating SQ and no sign of performance improvement.

Both AE and VAE models showed stable performance improvement during the training process. It also took less than 200 training epochs for the two models to learn how to generate high-quality motion data. However, both models failed to generate high-quality motion data for WK - the recognition performance on Hybrid dataset was not comparable to that on Real dataset.

3.3. Evaluation of Data Augmentation

Results in Figure 5. show augmenting the training dataset with real motion data improved the model recognition performance for the three postures by a range of 4.5%-6.3%. The increased performance suggests that the data augmentation can effectively improve the performance of the DNN-based recognition model. Using the GAN-generated data for augmentation also improved the recognition for BT (by 3.0% using 1DRGAN) and WK (by 1.3% using DCGAN), respectively. These results suggest the feasibility of applying GAN-based models to augment limited training datasets. The highest model performance after augmentation for each of the tested postures was achieved by GAN-based models. The quality of GAN-generated motion data appeared to have outperformed those generated using the benchmark AE and VAE models.

It is, however, essential to note the recognition performance for the posture ST deteriorated after augmenting with data from generative models. The decreased performance might be attributed to that the learned distribution of posture ST from generative models was not resembled to the real distribution. The low data quality of generated ST data becomes noise to the real data. In addition, none of the tested GAN-based models appeared to outperform the others in all the tested postures. The nature of motion data that can be generated from a certain type of GAN model will be investigated further in subsequent efforts.

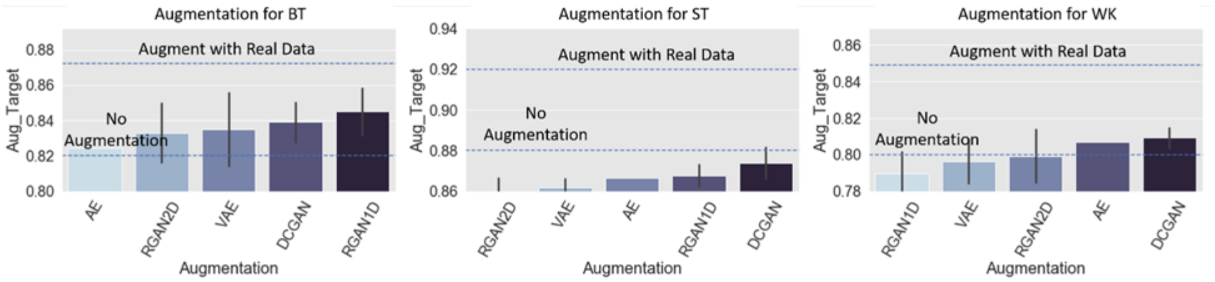


Figure 5. Evaluation of data augmentation.

4. Conclusions and Future Research

The lack of a large, annotated dataset limits the application of Deep Neural Networks (DNN)-based recognition models. The authors propose to address this data paucity challenge through a Generative Adversarial Network (GAN)-based data augmentation framework. The effective use of this data augmentation approach was explored using real workers' posture data. Initial results have demonstrated that i) Convolutional Long Short-Term Memory (LSTM) model achieved a higher recognition performance among the three tested LSTM-based recognition models; ii) the proposed Train on Real Test on Hybrid approach is appropriate for evaluating the quality of generated data and identifying a sufficiently trained Generative Adversarial Network model for data augmentation; iii) the performance of recognition model (Convolutional LSTM) improved by 1.3% and 3.0% for two of the three tested postures. However, the recognition for the standing posture was not improved after augmentation, which might be attributed to the low-quality of generated data. These results suggest that GAN-generated motion data could be effectively used to augment limited datasets thus improving the performance of DNN-based recognition models.

It is, however, important to note that a sufficiently trained GAN model can still generate low-quality posture data. This could deteriorate the model's recognition performance with respect to accuracy. The quality of GAN-generated data may be further improved through implementing the conditional GAN models to control the type of postures generated; and investigating the appropriate GAN architectures for varying postures. These will be explored in subsequent efforts.

References

[1] Wang D, Dai F and Ning X 2015 Risk Assessment of Work-Related Musculoskeletal Disorders in Construction: State-of-the-Art Review. *Journal of construction engineering and management,* **141(6)**, pp. 04015008

[2] Chen J, Qiu J and Ahn C 2017 Construction worker's awkward posture recognition through supervised motion tensor decomposition. *Automation in Construction,* **77**, pp. 67-81

[3] Ryu J, Seo J, Jebelli H and Lee S 2018 Automated Action Recognition Using an Accelerometer-Embedded Wristband-Type Activity Tracker. *Journal of construction engineering and management,* **145(1)**, pp. 04018114

[4] Yang Z, Yuan Y, Zhang M, Zhao X and Tian B 2019 Assessment of Construction Workers' Labor Intensity Based on Wearable Smartphone System. *Journal of construction engineering and management,* **145(7)**, pp. 04019039

[5] Kim K and Cho Y K 2020 Effective inertial sensor quantity and locations on a body for deep learning-based worker's motion recognition. *Automation in Construction,* **113**, pp. 103126

[6] Lee H, Yang K, Kim N and Ahn C R 2020 Detecting excessive load-carrying tasks using a deep learning network with a Gramian Angular Field. *Automation in Construction,* **120**, pp. 103390

[7] Zhao J and Obonyo E 2020 Convolutional long short-term memory model for recognizing construction workers' postures from wearable inertial measurement units. *Advanced Engineering Informatics,* **46**, pp. 101177

[8] Iwana B K and Uchida S 2020 An Empirical Survey of Data Augmentation for Time Series Classification with Neural Networks. *arXiv preprint arXiv:2007.15951*

[9] Slaton T, Hernandez C and Akhavian R 2020 Construction activity recognition with convolutional recurrent networks. *Automation in Construction,* **113**, pp. 103138

[10] Wang J, Chen Y, Gu Y, Xiao Y and Pan H 2018 SensoryGANs: an effective generative adversarial framework for sensor-based human activity recognition. in *2018 International Joint Conference on Neural Networks (IJCNN)*: IEEE. pp. 1-8

[11] Chen K, Zhang D, Yao L, Guo B, Yu Z and Liu Y 2020 Deep learning for sensor-based human activity recognition: overview, challenges and opportunities. *arXiv preprint arXiv:2001.07416*

[12] Alzantot M, Chakraborty S and Srivastava M 2017 Sensegen: A deep learning architecture for synthetic sensor data generation. in *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*: IEEE. pp. 188-193

[13] Goodfellow I, Pouget-Abadie J, Mirza M, Xu B, Warde-Farley D, Ozair S, Courville A and Bengio Y 2014 Generative adversarial nets. in *Advances in neural information processing systems*. pp. 2672-2680

[14] Abdelfattah S M, Abdelrahman G M and Wang M 2018 Augmenting the size of EEG datasets using generative adversarial networks. in *2018 International Joint Conference on Neural Networks (IJCNN)*: IEEE. pp. 1-6

[15] Esteban C, Hyland S L and Rätsch G 2017 Real-valued (medical) time series generation with recurrent conditional gans. *arXiv preprint arXiv:1706.02633*

[16] Nikolaidis K, Kristiansen S, Goebel V, Plagemann T, Liestøl K and Kankanhalli M 2019 Augmenting physiological time series data: A case study for sleep apnea detection. in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*: Springer. pp. 376-399

[17] Ramponi G, Protopapas P, Brambilla M and Janssen R 2018 T-cgan: Conditional generative adversarial network for data augmentation in noisy time series with irregular sampling. *arXiv preprint arXiv:1811.08295*

[18] Hatamian F N, Ravikumar N, Vesal S, Kemeth F P, Struck M and Maier A 2020 The Effect of Data Augmentation on Classification of Atrial Fibrillation in Short Single-Lead ECG Signals Using Deep Neural Networks. in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*: IEEE. pp. 1264-1268

[19] Radford A, Metz L and Chintala S 2015 Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*

[20] Haradal S, Hayashi H and Uchida S 2018 Biosignal data augmentation based on generative adversarial networks. in *2018 40th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*: IEEE. pp. 368-371

[21] Harada S, Hayashi H and Uchida S 2019 Biosignal Generation and Latent Variable Analysis with Recurrent Generative Adversarial Networks. *IEEE Access,* **7**, pp. 144292-144302

[22] Zhu F, Ye F, Fu Y, Liu Q and Shen B 2019 Electrocardiogram generation with a bidirectional LSTM-CNN generative adversarial network. *Scientific reports,* **9(1)**, pp. 1-11

[23] Chen G, Zhu Y, Hong Z and Yang Z 2019 EmotionalGAN: Generating ECG to Enhance Emotion State Classification. in *Proceedings of the 2019 International Conference on Artificial Intelligence and Computer Science*. pp. 309-313