Unreal Engine-based Data Augmentation to Improve Real-world Human Activity Recognition with Wearable Devices

Xingyu Zhou[1,2], Keisuke Mizutani[1] and Kento Tokuyama[1,*]

[1]Digital Transformation Unit, Chugai Pharmaceutical Co., Ltd., Tokyo, Japan
[2]Nagoya University, Nagoya, Japan

Abstract

Human activity recognition (HAR) has garnered significant attention owing to its potential applications in fields such as healthcare, automated driving, and personalized user interfaces. The use of wearable devices for collecting real-time motion data has spurred ongoing research in activity detection and prediction. However, the development of robust HAR models faces a critical challenge: the difficulty in acquiring diverse and comprehensive datasets of human activities. Ethical considerations and practical limitations often impede the collection of real-world motion data, particularly for sensitive or uncommon activities. To address this challenge, we introduce a novel system—the Unreal Data Generator—that leverages the capabilities of Unreal Engine 5. This tool facilitates the synthesis of time series data for a wide range of activities and scenarios by simulating data collection from multiple wearable device locations using virtual human models. Experiments demonstrated that augmenting real-world datasets with synthetic data generated by the Unreal Data Generator significantly improves the performance of deep learning-based HAR models. Specifically, by training our deep convolutional neural network model with both real-world and synthetic data, the accuracy in the WISDM benchmark test improved from 0.784 to 0.823. This approach offers a promising solution for improving the robustness and generalizability of HAR models by providing access to a rich and diverse range of training data.

Keywords

convolutional neural network, human activity recognition, synthetic data, Unreal Engine

1. Introduction

Human activity recognition (HAR) is a rapidly evolving field that enables machines to automatically recognize and interpret human actions and movements. HAR systems have numerous applications in healthcare [1, 2], manufacturing [3], autonomous vehicles [4], and sports analysis [5]. These systems utilize various sensor technologies, including wearable devices, smartphones, and radar sensors, to collect continuous signal data related to human activity. Sophisticated algorithms, primarily machine learning models such as deep learning approaches, are employed to classify this data into meaningful action categories.

Recent advancements in wearable device technology have significantly accelerated progress in HAR [6]. These sensor devices can be easily worn, enabling convenient and efficient data acquisition. To capture motion and orientation in three-dimensional space, these devices are typically equipped with 3-axis accelerometers and gyroscopes. In addition to accelerometers and gyroscopes, wearable devices can measure physiological signals such as heart rate and skin temperature. Their small size and long battery life enable the collection of large datasets over extended monitoring periods. The availability of readily accessible and diverse data has substantially enhanced the performance of deep learning models in recognizing human activities [7].

MiGA@IJCAI25: International IJCAI Workshop on 3rd Human Behavior Analysis for Emotion Understanding, August 29, 2025, Guangzhou, China.

*Corresponding author.
zhouxingyu4590@gmail.com (X. Zhou); mizutani.keisuke41@chugai-pharm.co.jp (K. Mizutani); tokuyama.kento26@chugai-pharm.co.jp (K. Tokuyama)
0009-0009-3553-644X (K. Tokuyama)

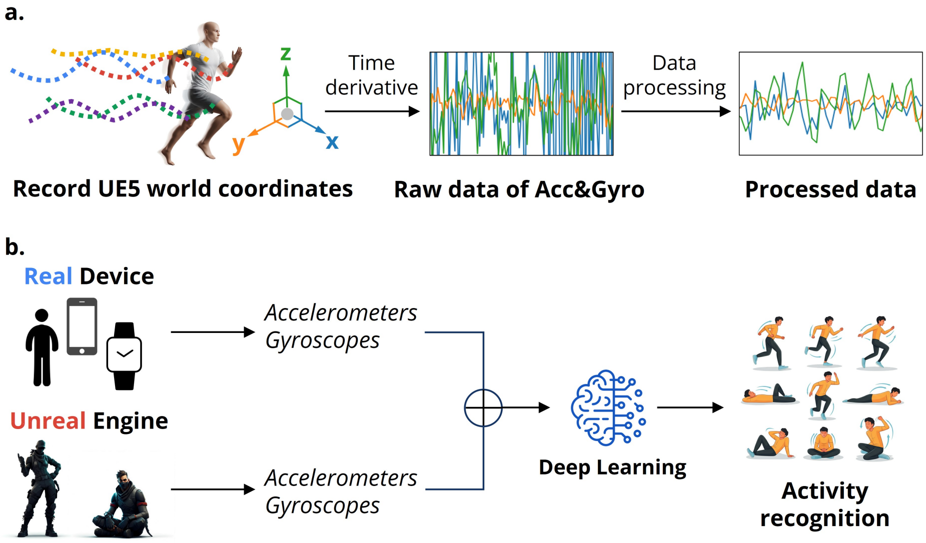

Figure 1: Overview of the proposed data augmentation framework using Unreal Engine. (a) The pipeline for generating synthetic sensor data, from recording world coordinates in UE5 to creating processed accelerometer and gyroscope signals. (b) The concept of augmenting real-world data with the generated synthetic data to train a deep learning model for Human Activity Recognition (HAR).

Despite advancements fueled by readily available data, significant challenges persist in developing robust HAR systems. Acquiring sufficient data for accurate predictions, particularly for infrequent yet critical activities such as falls and syncope, remains a major obstacle. Conducting large-scale studies with diverse participants further complicates data collection. Furthermore, ensuring data quality and mitigating sensor noise are crucial considerations. Future research must focus on addressing these limitations while improving the generalizability of HAR systems to unseen scenarios and diverse populations.

To address these data acquisition challenges, we developed Unreal Data Generator, the architecture of which is illustrated in Figure 1. Our methodology comprises two main stages. First, as detailed in Figure 1a, we leverage Unreal Engine, a real-time 3D creation tool, to simulate human movements. The world coordinates are recorded and then converted into raw sensor signals, which are subsequently refined through a dedicated processing pipeline. Second, as depicted in Figure 1b, this processed synthetic data is used to augment real-world datasets. By simulating a wide range of activities and scenarios under controlled parameters, this approach effectively overcomes the limitations and ethical concerns of collecting large-scale real-world data, especially for rare events. The primary objective is to leverage synthetic data generated within Unreal Engine to enhance the accuracy and robustness of deep learning-based HAR models by augmenting real-world datasets and improving performance on human activity recognition tasks.

2. Related Work

Numerous studies have been conducted on HAR using multi-channel time series data from sensors such as accelerometers and gyroscopes, obtained from wearable devices and smartphones [8]. In recent years, there has been extensive research on the application of machine learning techniques for activity classification models in HAR. For model improvements, approaches can be classified into two types: model-centric improvement and data-centric improvement.

In model-centric improvement, HAR using neural networks has been actively studied. For example, convolutional neural networks (CNNs) have been employed to effectively capture local patterns in sensor data [9], while long short-term memory (LSTM) networks have demonstrated their ability to model the temporal dynamics of human activities [10]. Furthermore, U-Net type models, originally developed for image segmentation, have also been explored for their potential in HAR tasks involving sequential data [11].

While these deep learning models have shown promising results and continue to be actively developed, they often require large amounts of training data to achieve optimal performance. This requirement for extensive datasets poses a significant challenge for the practical implementation of HAR systems. Therefore, a data-centric approach, focusing on improving data quality and exploring effective data augmentation techniques, becomes crucial for building robust and generalizable HAR models with limited resources. Kim et al. [12] demonstrated that data augmentation with oversampling solved the data imbalance problem for medical applications. Some research has applied generative adversarial networks (GANs) to generate synthetic data for HAR tasks [13, 14, 15]. For example, Lupión et al.[15] were the first to propose a conditional Wasserstein GAN (cWGAN) for generating accelerometer signals, demonstrating its superiority over standard conditional GANs for data augmentation in HAR. While some recent work has focused on improving GAN architectures directly, Zhang et al. (2025) took a different approach by proposing a differentiable framework that automatically learns to select and combine traditional, prior-driven handcrafted operations with a generative model, aiming to leverage the strengths of both methodologies [16].

The application of GANs has shown promise in generating synthetic data for HAR tasks. However, a potential limitation of relying solely on GANs lies in ensuring the diversity and comprehensiveness of the generated data. Simulation-based approaches offer a complementary strategy by allowing for the creation of varied datasets through the manipulation of virtual environments and parameters. For instance, Waqar et al. [17] demonstrated that synthetic 3D animation data can effectively enhance radar-based HAR predictions. Cauli and Recupero [18] demonstrated that synthetic data augmentation with Unity, which is a popular game engine, improved video action recognition. Their study showed that 3D animation data can be leveraged to construct diverse motion datasets, providing a promising strategy to address the data scarcity challenges associated with human subject experiments. Various studies have explored the use of game engines for generating synthetic data in HAR, including image [19], video data [20], and skeletal poses [21]. However, to the best of our knowledge, no studies have explored the application of game engines to generate synthetic data from accelerometers and gyroscopes for improving HAR systems with wearable devices. Here, we first demonstrate the novel application of a game engine, specifically using Unreal Engine 5 (UE5), to generate synthetic time series data from accelerometers and gyroscopes for HAR systems using wearable devices. Furthermore, our study explores a methodology for integrating this game engine-generated data into real-world datasets, addressing the data scarcity challenges associated with human subject experiments. This integration method contributes to an improved approach for generating comprehensive and diverse data.

3. Methods

3.1. Development of Unreal Data Generator

UE5.4 was selected as the standard engine for this study. The motion-matching template was downloaded, and all necessary plugins and configurations were set up following the official documentation [22]. To enhance character movement, the advanced locomotion system (ALS) was integrated into the motion-matching template. We utilized the standard default and female mannequins provided by Unreal Engine 5. Furthermore, we created three additional mannequin variations by applying animation layers, resulting in a total of five switchable mannequins. Walking animations from Mixamo [23] were modified by adjusting the leg weights (<0.2) within each layer to prevent animation errors. Additionally, Foot IK was implemented to improve realism by dynamically adjusting foot-ground interactions. Movement speed control was incorporated, allowing users to adjust walking and running speeds using the numeric keys 1–4.

Because UE5's default mannequin offers only basic actions—running, walking, standing, and jumping—custom animations need to be added for specific actions, such as climbing stairs and sitting. To implement the stair animations, a stair detection capsule and a Boolean variable were first added to

Table 1
Dataset Profile.

	Unreal data	WISDM	DSADS
Device	Unreal Engine 5.4	Cell phone	MTx 3-DOF orientation trackers
Measured parameters	World coordinates and rotation angles	Accelerometer	Accelerometer, gyroscope
Number of positions	6	1	5
Position	Front pants leg pocket, torso, arms, and legs	Front pants leg pocket	Torso and arms, legs
Sampling rate (Hz)	50	20	25
Subjects	5	36	8

indicate when a character was "on stairs." Then, stairs were configured as a separate collision layer that overlapped only with the character's stair capsule. Logic was implemented so that the Boolean variable switched to "true" when the stair capsule overlapped with the stair collision. Stair-specific animations (e.g., "running upstairs" and "ascending stairs") from Mixamo were integrated into the motion-matching system. A condition in the animation player triggered the appropriate animations when the Boolean variable was true. The sitting action was implemented using a montage system. Fifteen types of sitting animations were downloaded from Mixamo and used to create different montage files, with data captured and updated iteratively.

The data collection system was developed by placing sockets on designated bone locations and using the Break Vector and Break Rotator math functions to obtain world coordinates and rotation angles, respectively. In the Unreal Data Generator, the sampling rate, measurement time, and measurement points can be freely configured. For this study, time series data were collected at a sampling rate of 50 Hz for approximately 300 s for each activity across different animations (default mannequins, preset female mannequins, and three custom animation layers). Details of the dataset are presented in Table 1.

3.2. Data Processing

The Unreal data generated by UE5.4 contain time series data of unreal world coordinates (x, y, z) and rotation information as roll, pitch, and yaw $(\theta_x, \theta_y, \theta_z)$. The relationships between the coordinates (r), rotation angles (θ), velocity (v), acceleration (a), and angular velocity (ω) are defined by the following equations (1–4):

$$r_t = (x_t, y_t, z_t), \theta_t = (\theta_{xt}, \theta_{yt}, \theta_{zt}) \tag{1}$$

$$v_t = \frac{dr}{dt} \approx \frac{r_{t+\Delta t} - r_t}{\Delta t} \tag{2}$$

$$a_t = \frac{dv}{dt} \approx \frac{v_{t+\Delta t} - v_t}{\Delta t} \tag{3}$$

$$\omega_t = \frac{d\theta}{dt} \approx \frac{\theta_{t+\Delta t} - \theta_t}{\Delta t} \tag{4}$$

Since the acceleration data measured by wearable devices in the real world are affected by gravity and changes in device posture, we analyzed the effects of gravity on real-world data and applied gravity correction to the Unreal data as follows:

- Calculation of average acceleration per axis for each motion.
- Determination of gravity compensation values: For each axis, the average acceleration was compared to the gravitational acceleration (9.81 m/s²).
- If the absolute difference between the average acceleration and 9.81 (or -9.81 for downward acceleration) was within a threshold of 2.0 m/s², then 9.81 or -9.81 was used as the gravity compensation value for that axis. It was assumed that the sensor was primarily aligned with gravity.

- Handle outliers: If the average acceleration on any axis fell outside the threshold of ± 2.0 m/s^2 from 9.81 or -9.81, the median acceleration for that axis was used as the gravity compensation value instead. This approach is more robust for handling outliers and noisy data.
- Calculation of the rotation matrices: Euler-ZYX rotation matrices were computed from the rotation angle data at each time step of the Unreal data.
- Apply rotation and gravity compensation: The gravity compensation vector (obtained in steps 2 and 3) was multiplied by the rotation matrix (calculated in step 4) for each time step. This process transformed the gravity compensation vector from the sensor coordinate system into the world coordinate system, accounting for the sensor's orientation.
- The resulting compensated gravity vector was added to the raw acceleration readings on each axis to account for the real-world effects of gravity.

Finally, a Butterworth low-pass filter [24] was applied to the processed data to remove noise.

3.3. Benchmark Real Datasets

As real benchmark datasets, we selected two types: WISDM [25] and DSADS [26] (Table 1). The WISDM dataset included six types of motions—walking, jogging, upstairs, downstairs, sitting, and standing—recorded using a cell phone placed in the subject's front pant leg pocket. The DSADS dataset included 19 activities, in addition to those in the WISDM dataset, recorded using MTx 3-DOF orientation trackers positioned on five body locations: the torso, right arm, left arm, right leg, and left leg.

3.4. Activity Recognition Model Development

The learning task was defined as a six-activity classification problem using 10 s windowed data sampled at 20 Hz (i.e., a window size of 200). For HAR model development, we implemented a one-dimensional convolutional neural network (1D-CNN) based on a previous study [2]. We selected this architecture because our primary goal is to validate our novel synthetic data augmentation method using a foundational baseline model. The 1D-CNN is a standard backbone for HAR using multi-channel time-series data [27, 28], making it a suitable choice for this purpose. The model comprises two convolutional blocks and one adaptive average pooling layer. Each convolutional block includes a one-dimensional convolutional layer and a normalization layer. A single fully connected layer is used as the classifier. To evaluate the prediction performance of the developed models, we divided the real data by subject and conducted k-fold cross-person validation, with K set to 5 for the WISDM dataset and 4 for the DSADS dataset. We investigated the benefits of incorporating synthetic data into the training process under two scenarios: one using only real-world data and the other using a combined dataset of real-world data and Unreal Engine-generated synthetic data.

4. Results

4.1. Development of Unreal Data Generator

We developed Unreal Data Generator for synthetic data generation to improve the HAR system with accelerometers and gyroscopes. By utilizing UE5 as the standard engine, diverse motion data can be generated across various scenarios through intuitive key operations, much like controlling a game character. For instance, the data collected while walking on a paved road differs from that collected while walking on a bumpy road. Movement speed is also adjustable, allowing it to run straight and fast or to keep circling around the area. The Motion Matching in UE5 enables smooth animation transitions, thus offering natural and realistic motion responses to complex user operation inputs. Additionally, the sensor position for data collection and the sampling rate can be specified arbitrarily. This level of customizability is particularly useful for optimizing sensor positions on wearable devices and for other related applications.

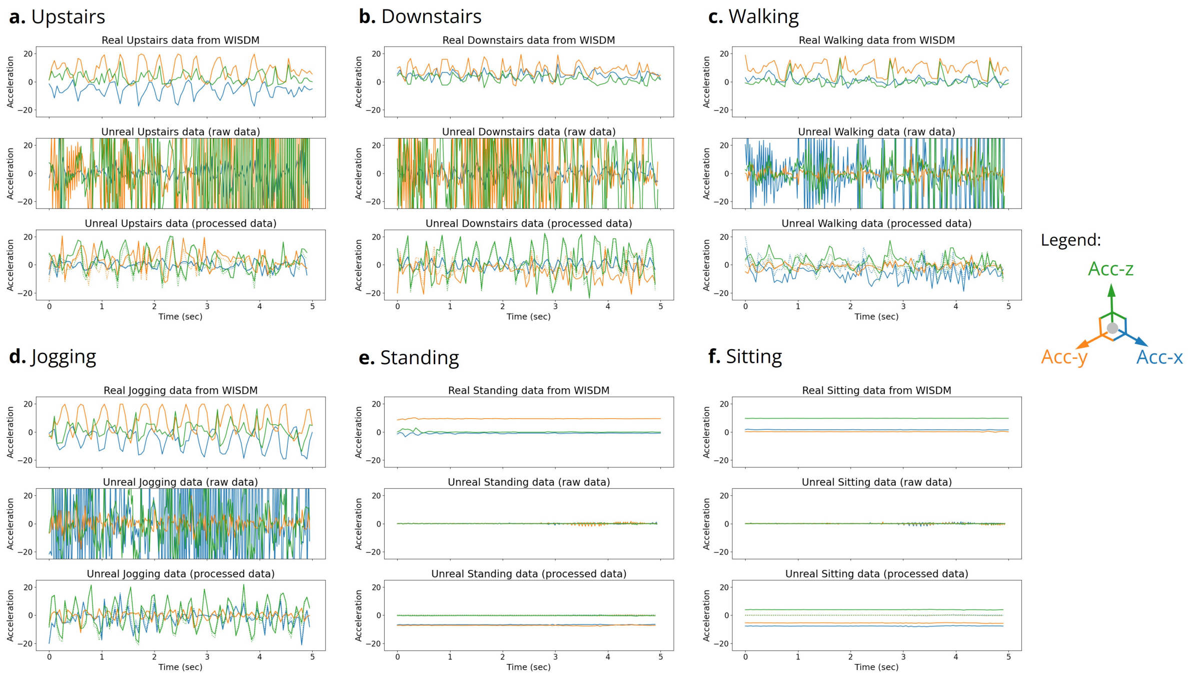

Figure 2: Comparison between real and unreal datasets. This figure compares real-world acceleration data from the WISDM dataset with synthetic data generated using Unreal Engine at 50 Hz, both before and after processing. Each column represents different activity: (a) upstairs, (b) downstairs, (c) walking, (d) jogging, (e) standing, and (f) sitting. Within each subplot, the top row displays the real acceleration data, the middle row shows the raw synthetic data from Unreal Engine, and the bottom row presents the processed Unreal Engine data. In all plots, blue represents acceleration along the x-axis, orange represents the y-axis, and green represents the z-axis. The x-axis of each subplot represents time in seconds. The dotted and solid lines in the bottom-row plots indicate the data processed with only the low-pass filter and with both the filter and gravity correction, respectively.

4.2. Unreal Data Generation for HAR

We developed an unreal data generation system using the real-time 3D creation tool UE5. To extract unreal data, we utilized the real-world coordinate system in UE5 and recorded six activities—walking, jogging, upstairs, downstairs, sitting, and standing—using five types of mannequins sampled at 50 Hz. Activity animations were obtained from the Fab website (Unreal Engine marketplace). The values for accelerometers and gyroscopes were calculated based on the time differences in the coordinate dynamics for each record. We recorded each motion for 5 minutes, resulting in a total of 150 minutes of motion capture data (5 mannequin types $\times$ 6 motion types).

Figure 2 compares the real (top subplot) and unreal (middle and bottom subplots) datasets. As shown in Figure 2, the raw synthetic data generated by Unreal Engine exhibited significant noise and deviated considerably from the real-world data. This discrepancy arises primarily because the synthetic acceleration is derived by numerically differentiating position data twice, a method highly susceptible to amplifying noise. In contrast, real-world accelerometers measure acceleration directly as a physical force and often include hardware-level filters, resulting in inherently smoother signals.

To address this, we applied a two-step processing method. First, a Butterworth low-pass filter was used to remove the high-frequency noise inherent in the differentiation process. Second, to account for the influence of gravity present in real-world measurements, we implemented the gravity correction detailed in Section 3.2. The processed unreal data were significantly closer to the measured real-world values (see bottom subplots in Figure 2).

Table 2
Human activity recognition on cross-person validation. Boldface highlights the best performance, and underline highlights the second-best performance. Evaluation metrics: Micro-accuracy (Accuracy), Macro-AUC (AUC), Macro-F1 score (F1), Macro-precision (Precision), Macro-recall (Recall). Checkmark (✓) indicates the component was used.

Dataset	Unreal data	Lowpass filter	Gravity correction	Accuracy	AUC	F1	Precision	Recall
WISDM (1 sensor)	✓	✓	✓	**0.8255**	**0.9498**	**0.7877**	**0.7808**	**0.7966**
	✓	✓		<u>0.8038</u>	0.9431	0.7550	0.7430	0.7687
	✓			0.7898	<u>0.9432</u>	0.7506	0.7326	<u>0.7731</u>
				0.7841	0.9367	<u>0.7581</u>	<u>0.7590</u>	0.7608
DSADS (5 sensors)	✓	✓	✓	**0.9838**	<u>0.9987</u>	**0.9838**	**0.9847**	**0.9838**
	✓	✓		0.9495	0.9944	0.9496	0.9527	0.9495
	✓			0.9386	0.9944	0.9382	0.9427	0.9386
				<u>0.9753</u>	**0.9992**	<u>0.9756</u>	<u>0.9778</u>	<u>0.9753</u>

4.3. Improved Cross-person Human Activity Recognition with Unreal Data

To validate the improvement in HAR accuracy for different datasets, we trained separate 1D-CNN based activity recognition models for WISDM and DSADS datasets, using Unreal Engine-generated synthetic data to augment each. Model performance was evaluated using k-fold cross-person validation in terms of accuracy, precision, recall, F1-score, and AUC. We compared the performance of models trained solely on real-world data with those trained on augmented datasets incorporating synthetic data. The results are summarized in Table 2.

On the WISDM dataset, the baseline model trained solely on real-world data achieved an accuracy of 78.41%. Augmenting the dataset with Unreal data generated by UE5 improved HAR performance. Specifically, applying a low-pass filter and gravity correction to the Unreal data further enhanced performance, resulting in an accuracy of 82.55%. Similarly, on the DSADS dataset, the addition of processed unreal data also improved HAR performance, whereas raw Unreal data negatively impacted model accuracy. These findings highlight the importance of incorporating low-pass filtering and gravity correction during data preprocessing when developing HAR models using Unreal Engine-generated data.

The confusion matrix for HAR classification in the WISDM benchmark is shown in Figure 3. The accuracy of the HAR model improved significantly for motions involving movement when Unreal data were included. Although classification accuracy for differentiating between the static postures of sitting and standing did not show improvement, the accuracy in dynamic/static motion classification demonstrated noticeable gains. Furthermore, the confusion matrix revealed improved classification accuracy for similar motions, such as walking/jogging and climbing upstairs/downstairs.

5. Discussion

Unreal Data Generator facilitates the acquisition of acceleration and gyroscopic sensor data across a wide range of motions and scenarios. Typically, 3D animations in game engines operate by repeatedly playing back pre-recorded animation data, which results in identical sensor data being generated from the same animation. However, the ALS in UE5 employs a dynamic procedural animation system that adapts to the environment and player inputs in real time. This approach blends animations and adjusts them based on factors such as terrain, speed, and direction, creating more fluid and realistic character movements. Compared with traditional animation techniques in game engines, these advanced methods provide a more immersive and responsive character movement system. In this study, we developed a more flexible movement animation by integrating ALS, motion matching, and Foot IK techniques.

As mentioned previously, generating natural and diverse movements using default mannequins can

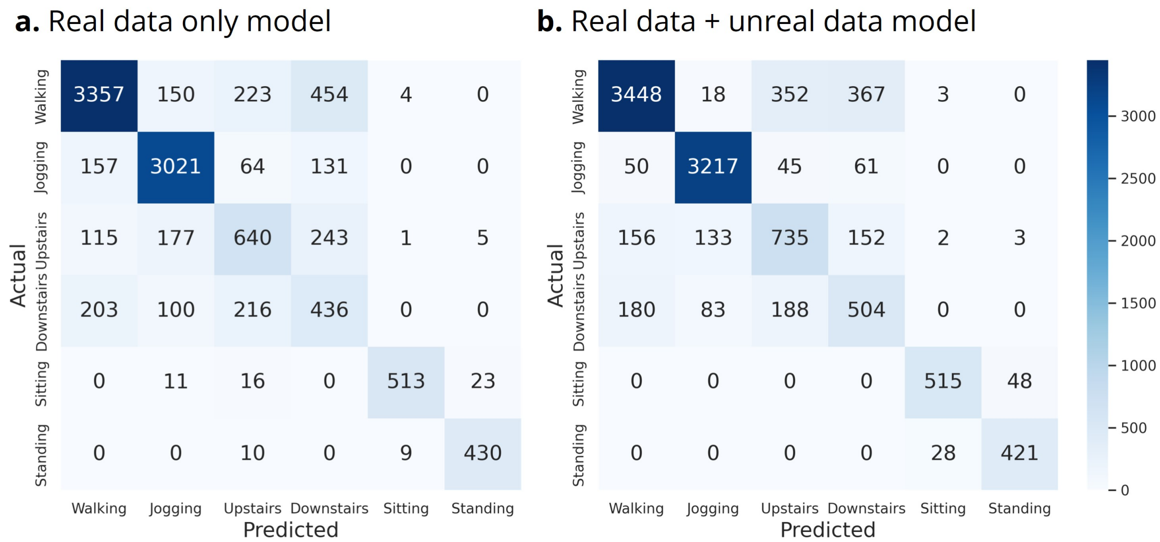

Figure 3: Confusion matrix of WISDM activity classification (a) result of modeling with only real data and (b) that of Unreal data augmentation modeling. X-axis indicates predicted labels, and y-axis indicates actual labels.

be easily achieved. However, generating data for specific custom actions requires the addition of new animations. For instance, in this study, a sitting animation montage was downloaded from FAB to capture sitting motion data. The FAB database currently offers a wide variety of motions, providing a diverse range of animations that can be utilized. When specific motion data are not available in the FAB database, new animations must be created. This process can be streamlined by leveraging software that generates motions from videos (DeepMotion [29]) or using generative AI tools specialized in video generation (Runway Gen-3 Alpha [30]).

The primary differences between Unreal Engine-generated data and real-world sensor data lie in the nature of the measurements. The real data generator provides world-space coordinates, dynamics, and angular rotation, whereas real-world wearable devices measure acceleration, angular velocity, and other digital biomarkers, such as heart rate, using their own algorithms. Notably, while real-world sensors measure acceleration as the force applied to the device—affected by gravity—acceleration derived from world coordinate dynamics in Unreal Engine does not account for gravity. This distinction is crucial when comparing or integrating data from these two sources.

Our results demonstrate the profound impact of correcting for this gravitational effect. In the real world, the gravitational force registered by a 3-axis accelerometer changes based on the sensor's orientation. Our method of applying a gravity vector transformed by the sensor's rotation matrix successfully embeds this orientation-dependent component into the synthetic data. The effectiveness of this is most apparent in static activities like 'Standing' and 'Sitting' (Figure 2e, 2f). For these poses, the raw synthetic acceleration is nearly zero, as position coordinates are constant. In contrast, a real sensor registers non-zero values corresponding to the constant pull of gravity. Our gravity correction replicates this real-world phenomenon, transforming the unrealistic near-zero signals into realistic static offsets, thus creating much more faithful training data.

These specific corrections for gravity and signal noise are part of a broader challenge in simulation-based research known as the "Sim-to-Real Gap"—the discrepancy between simulated and real-world data [31]. Our study demonstrates that it is essential to mitigate this gap through a dedicated data processing pipeline to improve the data's fidelity. We addressed this by implementing a low-pass filter for noise and using Euler rotation matrices for gravity correction. While this direct signal processing approach proved effective, future research could explore alternative methods to bridge this gap. For instance, incorporating domain adaptation techniques or leveraging Generative Adversarial Networks (GANs) could offer ways to learn the transformation from simulated to realistic data automatically [32, 17].

6. Limitations

While this study demonstrates the potential of using Unreal Engine for data augmentation, it has several limitations. First, the scope of our evaluation was limited. The experiments were conducted on two public datasets, covering only six basic daily activities such as walking, running, and sitting, and used a single backbone architecture (1D-CNN). Furthermore, our study focused on validating a simulation-based approach and did not include a direct comparative analysis against other categories of data augmentation, such as model-based techniques like GANs. Second, our methodology for bridging the Sim-to-Real Gap was confined to a direct signal processing pipeline (i.e., low-pass filtering and gravity correction). The exploration of other advanced methods, such as automated domain adaptation, was not covered in this work.

Acknowledgments

We thank Epic Games Japan for the complimentary use of Unreal Engine for this research.

Declaration on Generative AI

During the preparation of this work, the authors used Gemini in order to: draft content, Grammar and spelling check, Paraphrase and reword, Text translation, Improve writing style, and Literature review generation. After using this tool, the authors reviewed and edited the content as needed and took full responsibility for the content of the publication.

References

[1] D. Umbricht, W. Cheng, F. Lipsmeier, A. Bamdadian, M. Lindemann, Deep learning-based human activity recognition for continuous activity and gesture monitoring for schizophrenia patients with negative symptoms, Frontiers in Psychiatry 11 (2020) 574375. doi:10.3389/fpsyt.2020.574375.

[2] S. Momo, K. Kosuke, H. Takaya, K. Kijun, T. Kento, Improved generalized performance of hemodynamics scenarios prediction with digital biomarkers by Conv1D approach, in: IEEE SMC, Proceedings. 2023, 2023.

[3] N. Yoshimura, T. Maekawa, T. Hara, A. Wada, Y. Namioka, Acceleration-based activity recognition of repetitive works with lightweight ordered-work segmentation network, Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies 6 (2022) 1–39.

[4] M. Tammvee, G. Anbarjafari, Human activity recognition-based path planning for autonomous vehicles, Signal Image Video Process. 15 (2021) 809–816. doi:10.1007/s11760-020-01800-6.

[5] T. Tuncer, F. Ertam, S. Dogan, A. Subasi, An automated daily sports activities and gender recognition method based on novel multikernel local diamond pattern using sensor signals, IEEE Trans. Instrum. Meas. 69 (2020) 9441–9448. doi:10.1109/TIM.2020.3003395.

[6] S. Zhang, Y. Li, S. Zhang, F. Shahabi, S. Xia, Y. Deng, N. Alshurafa, Deep learning in human activity recognition with wearable sensors: a review on advances, Sensors (Basel) 22 (2022) 1476. doi:10.3390/s22041476.

[7] M. Yi, W. Lee, S. Hwang, A human activity recognition method based on lightweight feature extraction combined with pruned and quantized cnn for wearable device, IEEE Trans. Con. Electron. 69 (2023) 657–670. doi:10.1109/TCE.2023.3266506.

[8] E. Ramanujam, T. Perumal, S. Padmavathi, Human activity recognition with smartphone and wearable sensors using deep learning techniques: a review, IEEE Sens. J. 21 (2021) 13029–13040. doi:10.1109/JSEN.2021.3069927.

[9] J. Yang, M. Nguyen, P. San, X. Li, S. Krishnaswamy, Deep convolutional neural networks on multichannel time series for human activity recognition, in: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI' 15), AAAI Press, 2015, pp. 3995–4001.

[10] K. Xia, J. Huang, H. Wang, Lstm-cnn architecture for human activity recognition, IEEE Access 8 (2020) 56855–56866. doi:10.1109/ACCESS.2020.2982225.

[11] Y. Zhang, Z. Zhang, Y. Zhang, J. Bao, Y. Zhang, H. Deng, Human activity recognition based on motion sensor using U-Net, IEEE Access 7 (2019) 75213–75226. doi:10.1109/ACCESS.2019.2920969.

[12] Y.-W. Kim, K. Joa, H.-Y. Jeong, S. Lee, Wearable imu-based human activity recognition algorithm for clinical balance assessment using 1d-cnn and gru ensemble model, Sensors (Basel) 21 (2021) 7628. doi:10.3390/s21227628.

[13] M. Chan, M. Noor, A unified generative model using generative adversarial network for activity recognition, J. Ambient Intell. Hum. Comput. 12 (2021) 8119–8128. doi:10.1007/s12652-020-02548-0.

[14] X. Li, J. Luo, R. Younes, Activitygan: generative adversarial networks for data augmentation in sensor-based human activity recognition, in: Adjunct Proceedings of the 2020 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the 2020 ACM International Symposium on Wearable Computers, 2020, pp. 249–254.

[15] M. Lupión, F. Cruciani, I. Cleland, C. Nugent, P. M. Ortigosa, Data augmentation for human activity recognition with generative adversarial networks, IEEE Journal of Biomedical and Health Informatics 28 (2024) 2350–2361. doi:10.1109/JBHI.2024.3364910.

[16] Y. Zhang, Q. Gao, R. Hu, Q. Ding, B. Li, Y. Guo, Differentiable prior-driven data augmentation for sensor-based human activity recognition, IEEE Transactions on Computational Social Systems (2025) 1–13. doi:10.1109/TCSS.2025.3565414.

[17] S. Waqar, M. Pätzold, A simulation-based framework for the design of human activity recognition systems using radar sensors, IEEE Internet Things J. 11 (2024) 14494–14507. doi:10.1109/JIOT.2023.3344179.

[18] N. Cauli, D. Reforgiato Recupero, Synthetic data augmentation for video action classification using Unity, IEEE Access 12 (2024) 156172–156183. doi:10.1109/ACCESS.2024.3485199.

[19] N. Pai, P. Chen, P.-Y. Chen, Z. Wang, Home fitness and rehabilitation support system implemented by combining deep images and machine learning using unity game engine, Sens. Mater. 34 (2022) 1971–1990. doi:10.18494/SAM3734.

[20] T. Kim, M. Peven, W. Qiu, A. Yuille, G. Hager, Synthesizing attributes with unreal engine for fine-grained activity analysis, in: 2019 IEEE Winter Applications of Computer Vision Workshops (WACVW), IEEE, New York, 2019, pp. 35–37. doi:10.1109/WACVW.2019.00013.

[21] D. Ludl, T. Gulde, C. Curio, Enhancing data-driven algorithms for human pose estimation and action recognition through simulation, IEEE Trans. Intell. Transp. Syst. 21 (2020) 3990–3999. doi:10.1109/TITS.2020.2988504.

[22] Epic Games, Inc., Motion matching in unreal engine, 2025. URL: https://dev.epicgames.com/documentation/en-us/unreal-engine/motion-matching-in-unreal-engine.

[23] Mixamo Inc., Mixamo animation library website, 2025. URL: https://www.mixamo.com.

[24] S. Butterworth, On the theory of filter amplifiers, Wirel. Eng. 7 (1930) 536–641.

[25] J. Kwapisz, G. Weiss, S. Moore, Activity recognition using cell phone accelerometers, ACM SIGKDD Explor. Newsl. 12 (2011) 74–82. doi:10.1145/1964897.1964918.

[26] B. Barshan, M. Yüksek, Recognizing daily and sports activities in two open source machine learning environments using body-worn sensor units, Comput. J. 57 (2014) 1649–1667. doi:10.1093/comjnl/bxt075.

[27] F. J. Ordóñez, D. Roggen, Deep convolutional and lstm recurrent neural networks for multimodal wearable activity recognition, Sensors 16 (2016). doi:10.3390/s16010115.

[28] S. Gupta, Deep learning based human activity recognition (har) using wearable sensor data, International Journal of Information Management Data Insights 1 (2021) 100046. doi:10.1016/j.jjimei.2021.100046.

[29] Deep Motion Inc., Deep motion, 2025. URL: https://www.deepmotion.com/.
[30] Runway AI Inc., Introducing gen-3 alpha: A new frontier for video generation, 2025. URL: https://runwayml.com/research/introducing-gen-3-alpha.
[31] E. Salvato, G. Fenu, E. Medvet, F. A. Pellegrino, Crossing the reality gap: A survey on sim-to-real transferability of robot controllers in reinforcement learning, IEEE Access 9 (2021) 153171–153187. doi:10.1109/ACCESS.2021.3126658.
[32] A. Akbari, R. Jafari, Transferring activity recognition models for new wearable sensors with deep generative domain adaptation, in: Proceedings of the 18th International Conference on Information Processing in Sensor Networks, ACM, New York, USA, 2019, pp. 85–96. doi:10.1145/3302506.3310391.