BSDGAN: Balancing Sensor Data Generative Adversarial Networks for Human Activity Recognition

Yifan Hu
Department of Computer Science and Technology
Nanjing Tech University
Nanjing, China
Email: liquid0707@outlook.com

Abstract—The development of IoT technology has enabled various sensors to be integrated into mobile devices. Human Activity Recognition (HAR) based on sensor data has become an important part of the field of machine learning and ubiquitous computing. However, human activities are not evenly distributed because of the rarely performed activities. The dataset is extremely imbalanced. In this paper, we propose Balancing Sensor Data Generative Adversarial Networks (BSDGAN) to generate sensor data for rarely performed human activities. To address extreme imbalances in human activity dataset, an autoencoder is employed to initialize the training process of BSDGAN, which to ensure the data features of each activity can be learned. Add the generated activity data to the original data to generate a new dataset that increases the proportion of rarely performed human activities, which makes the dataset balanced. We deployed multiple human activity recognition models on two publicly available imbalanced human activity datasets, WISDM and UNIMIB. Experimental results show that the proposed BSDGAN can effectively capture the data features of real human activity sensor data, and generate realistic synthetic sensor data. Meanwhile, the balanced activity dataset is effective for the activity recognition model to improve the recognition accuracy.

Index Terms—Human activity recognition, generative adversarial networks, data augmentation

I. INTRODUCTION

With the popularization of smart mobile devices, the research on human activity recognition based on sensor data occupies an increasing proportion in the field of ubiquitous computing [1]. The miniaturization of sensors enables mobile devices to embed various sensors and collect various types of sensor data, including acceleration, gyroscope, magnetic field strength and pressure. The abundance of sensor data leads to Human Activity Recognition (HAR) based on sensor data widely employed in various fields of our daily life, such as health care [2], sleep quality analysis [3], and sports fitness [4]. Human activity recognition has great potential for development and research.

The collection and labeling of sensor data is the basis of the entire human activity recognition research. The performance of human activity recognition classifier is affected by an extreme imbalance in the amount of sensor data between activity classes [5]. However, most HAR researchers only focus on the accuracy of human activity recognition rather than the data quality of the human activity dataset. The vast majority of sensor data in human activity datasets comes from mobile smartphones and wearables [6]. Considering the purchase cost of smart devices and the labor cost of collecting and labeling sensor data, most human activity datasets have a highly imbalanced data amount between activity classes, the research of human activity recognition is facing the challenge of data amount imbalance.

In recent years, many data generation methods based on deep learning have been proposed. Among these, Generative Adversarial Networks (GAN) is the most effective and most interesting data generation method first proposed by Ian Goodfellow et al. [7]. In the field of computer vision, GAN has been proven that it can effectively generate many types of high-quality images, including human faces, animals, and landscapes [8]. Nevertheless, few researchers have successfully deployed independent GANs on human activity datasets to generate high-quality sensor data because of the huge data feature difference between different activity classes [9].

In this paper, we employed a unified independent generative adversarial network, BSDGAN, to improve the performance of human activity recognition models by oversampling human activity classes with small amounts of sensor data. The main contributions of this paper are summarized as follows:

1. We successfully deploy the proposed generative adversarial networks on imbalanced human activity datasets, and generate sensor data for specified human activities.

2. We added conditional constraints to the GAN framework and improved the loss function with a gradient penalty. An autoencoder is used to initialize the GAN training, providing the BSDGAN with knowledge of the data feature distribution of all human activities and contributing to stabilizing the training process.

3. We oversample the human activity classes with few data to balance the dataset, prove that the balanced human activity dataset can improve the performance of HAR models.

The rest of this paper is organized as follows. Section II reviews the related works regarding HAR and GAN. Section III presents the details of our proposed BSDGAN. Section IV
illustrates the performance of BSDGAN on imbalanced human activity datasets and presents the comparison of various HAR models before and after data balancing. Section V gives the conclusion of this paper and introduces our future work.

II. RELATED WORK

A. Human Activity Recognition

Human activity recognition is divided into visual image-based activity recognition and sensor data-based activity recognition. Compared with human activity recognition methods based on visual images, sensor-based methods have the advantages of small data amount, low cost, and strong embeddability, occupying a major position in the research and application of human activity recognition. With the popularity of smartphones, human activity recognition based on smartphones has been applied in many research fields. Some researchers employed traditional machine learning methods to recognize human activities, such as K-Nearest Neighbors [10], Decision Tree [11] and Random Forest [12].

Deep learning has been used for human activity recognition by researchers due to its rapid development in recent years.

Traditional machine learning methods always require hand-crafted data features. In order to achieve desired recognition results, researchers have devoted a lot of effort to researching and designing effective features to improve HAR performance [13]. In contrast, deep learning methods can automatically extract features from sensor data [14]. C.A. Ronaoo et al. [15] used a CNN with two hidden layers to analyze accelerometer data and gyroscope data, achieved activity classification accuracy of over 90% on the University of California's Open Activity Dataset (UCI DATASET) [16]. Y. Chen et al. [17] used a Long Short-Term Memory neural network (LSTM) to analyze the sensor data. R. Mutegeki et al. [18] combined the CNN with LSTM, and the recognition accuracy on the UCI dataset reached 92%.

Unfortunately, HAR research is completely dependent on the data quality of human activity datasets. Due to the inconsistencies in the difficulty of collecting different human activity data, the data amount of some activities in the human activity dataset is always small, resulting in the human activity dataset being extremely imbalanced. In this paper, we adopt GANs framework to generate sensor data and balance the human activity dataset with the generated activity data.

B. Generative Adversarial Networks

Generative Adversarial Networks (GAN) has developed rapidly in recent years. Inspired by the original GAN, researchers have proposed many variants of GAN, WGAN [19] and its improved version WGAN-GP [20] are proposed to solve the problem of mode collapse and vanishing gradients of the original GAN during training. Mirza M et al. [21] proposed CGAN to bring label information into the training process of GAN, so that GAN can generate data for specific classes. Radford et al. [22] proposed DCGAN, they use deep convolutional neural networks to replace the MLP in the original GAN and replace the fully connected layer with a global pooling layer to reduce the amount of computation.

Based on CGAN and DCGAN, Odena A et al. [23] proposed ACGAN. When the discriminator of ACGAN discriminates the real data, it also classifies the real data. The results of the classification are feedback to the generator to improve the data quality. The researchers of BAGAN [24] point out that when ACGAN works on unbalanced and small datasets, it cannot effectively generate data for the minority classes. BAGAN improves the loss function of ACGAN to solve the self-contradiction between the two loss functions of ACGAN.

M. Alzantot et al. [25] proposed SenseGEN to generate activity data, which is the first application of GANs framework on human activity datasets. However, the discriminator and generator model of SenseGEN are trained separately, the generator model cannot learn from the feedback of the discriminator model, resulting in poor quality of the generated activity data. Alharbi et al. [26] employed WGAN to generate activity sensor data. Considering the activity data is time-sequence data, they adopt RNN based model to build the generator and discriminator model, which makes their GAN structure difficult to train. Razvan Pascanu et al. [27] showed that the neural network architecture based on the RNN model has unstable convergence during training. The generator and discriminator model of the proposed BSDGAN in this paper is built by 1D-transposed CNN and 1D-CNN. In the following sections, we will demonstrate that GANs built by pure convolutions can also generate data for different human activities.

III. BSDGAN

We dub our generative adversarial networks framework as Balancing Sensor Data GAN (BSDGAN). BSDGAN is a complete generative adversarial network framework, consisting of a generator model and a discriminator model. Human activity datasets are usually extremely imbalanced, traditional GANs tend to generate sensor data for activity class with a large number of data samples, rather than the minority class data we need. We adopt an autoencoder to initialize the training of BSDGAN to deal with it so that BSDGAN can learn the data features of all activity classes. The architecture of BSDGAN is shown in Fig. 1. Random noise and label information are simultaneously fed into the generator model to generate fake sensor data. Subsequently, the fake sensor data with label information and the real sensor data are both fed into the discriminator, the discrimination results will be employed to improve the parameters of the generator model.

A. Autoencoder Initialization

Autoencoders can converge towards good solutions easily [28]. We apply an autoencoder to initialize the GAN, letting it close to a good solution at the initial stage, away from mode collapse. Furthermore, the encoder part of the autoencoder is adopted to infer the distribution of latent vectors for different human activity classes. All sensor data in the human activity dataset will be used to train the autoencoder, and the training process is performed by minimizing mean absolute error.
B. Design of Generator Model

The goal of the generator model is to learn the data feature distribution of real sensor data $p(x|c_i)$ in specific activity class $c_i$, then generate high-quality synthetic data $G(z|c_i)$. The generator model of BSDGAN consists of an embedding layer and a pre-trained decoder, as shown in Fig. 2. where $x$-axis represents the data sample number, and $y$-axis represents the acceleration value. Blue and green solid lines represent the $x$ and $y$-axis acceleration value, and the orange solid line represents the $z$-axis acceleration value. Random noise $z$ and label information $c_i$ are fed into the embedding layer to generate labeled noise. The pre-trained decoder shares the weight parameters and network architecture with the generator model. The labeled noise can be converted to the sensor data $G(z|c_i)$ in specific activity class by 1D-transposed convolutional neural network layer in the decoder. The weight parameters of decoder in generator can be updated during the process of adversarial training.

D. Improved Loss Function

The training process of GAN is the process of the discriminator against the generator. The goal of generator $G$ is to generate data that fools the discriminator, and the goal of discriminator $D$ is to distinguish real data from generated data. Therefore, the loss function of original GAN is defined as:

$$\min_G \max_D V(D, G) = \mathbb{E}_{x_r \sim X_r}[logD(x_r)] + \mathbb{E}_{x_g \sim X_g}[log(1 - D(x_g))]$$  \hspace{1cm} (1)$$

where $x_r$ is the real sensor data, $X_r$ is the real data distribution, $x_g = G(z)$ is the generated sensor data, and $X_g$ is the generated data distribution.

The original GAN cannot generate data for specified classes. Conditional GAN (CGAN) adds constraints on the basis of the original GAN, enabling it to generate synthetic data of
specified classes. The loss function of CGAN is defined as follows:
\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim X_r}[\log D(x|y)] + \mathbb{E}_{x \sim X_g}[\log(1 - D(x|y))] \tag{2}
\]
where \( y \) is the conditional constraint. It is combined with input noise \( z \) to form a joint hidden layer representation. The discriminator discriminates between \( x_g \) and \( x_r \) based on the condition \( y \).

To make the training process of GANs framework more stable, WGAN improves the loss function of GAN. GAN’s loss function based on Jensen-Shannon divergence, which makes the training process difficult. WGAN adopts Wasserstein Distance instead of JS divergence to measure the difference between generated data and real data. Wasserstein Distance is defined as:
\[
W(X_r, X_g) = \inf_{\gamma \in \Pi(X_r, X_g)} \mathbb{E}_{(x, y) \sim \gamma} [D(x|y)] \tag{3}
\]
where \( \Pi(X_r, X_g) \) represents the all joint distributions between \( X_r \) and \( X_g \). However, wasserstein distance is difficult to deal with in practice, researchers employed Kantorovich-Rubinstein duality as an alternative:
\[
W(X_r, X_g) = \sup_{\|D\|_{L} \leq 1} (\mathbb{E}_{x \sim X_r}[D(x)] - \mathbb{E}_{x \sim X_g}[D(x)]) \tag{4}
\]
where \( \|D\|_{L} \leq 1 \) means discriminator \( D \) follows the 1-Lipschitz function. So the objective of WGAN is obtained as:
\[
W(X_r, X_g) = \mathbb{E}_{x \sim X_r}[D(x)] - \mathbb{E}_{x \sim X_g}[D(x)] \tag{5}
\]
WGAN directly adopts weight clipping when dealing with 1-Lipschitz constraints, which results in the parameters of the discriminator focusing on the maximum and minimum values, easily leading to gradient vanishing and gradient explosion. WGAN-GP employed gradient penalty to improve the loss function of the discriminator model. The loss function of WGAN-GP is defined as:
\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim X_g}[D(x|y)] - \mathbb{E}_{x \sim X_r}[D(x)] - \lambda \mathbb{E}_{\hat{x} \sim X}[\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1]^2] \tag{6}
\]
where \( \hat{x} = \alpha x_r + (1 - \alpha)x_g \), \( \alpha \) is in the range 0 to 1. \( \lambda \) is a hyperparameter of the penalty extent.

In this paper, we combined the improvements of WGAN-GP on the original GAN with the conditional constraints of CGAN. The loss function of BSDGAN is defined as follows:
\[
\min_G \max_D V(D, G) = \mathbb{E}_{x \sim X_r,y \sim Y_r}[\log D(x|y)] + \mathbb{E}_{x \sim X_g,y \sim Y_g}[\log(1 - D(G(x|y)|y))] + \mathbb{E}_{x \sim X_r,y \sim Y_{wrong},y \sim Y_g}[\log(1 - D(x|y_{wrong}))] - \lambda \mathbb{E}_{\hat{x} \sim X,y \sim Y}[\|\nabla_{\hat{x}} D(\hat{x})\|_2 - 1]^2] \tag{7}
\]
where \( y_r \) is the real label and \( Y_r \) is the set of all real data labels. In an imbalanced dataset, ground truth labels randomly sampled from the dataset are still imbalanced. Therefore, we refer to BAGAN and randomly select a label \( y_g \) for each generated data from the balanced label set \( Y_g \). To enhance the learning of class information from the real dataset, we add an additional cross-entropy loss for misclassified cases, and the misclassified data label is set to \( y_{wrong} \).

### E. Pseudocode of Algorithm

The pseudocode of the dataset balance using BSDGAN is shown in Algorithm 1. During the balancing process, BSDGAN takes the data amount of the largest class as the balancing criterion and oversamples each human activity class. When the amount of data for all activity classes is greater than the balance standard, the balancing process ends, this means that the dataset reaches a balanced state.

#### Algorithm 1 The process of human activity dataset balance

**Input:** random noise \( z \); real dataset \( x_{real} \);

**Output:** the balanced dataset \( x_b \)

1: Split the real dataset \( x_{real} \) into real sensor data \( x_r \) and real data label \( y_r \);
2: Import the trained generator model \( G \) and discriminator model \( D \);
3: Calculate the amount of data for each class, set the maximum value to \( N_{c}^{max} \);
4: \textbf{for} \( i=1 \) to \textit{len(classes)} \textbf{do}
5: \hspace{0.5cm} Set the data volume of \( i^{th} \) class to \( N_{c}^{i} \);
6: \hspace{0.5cm} \textbf{while} \( N_{c}^{i} < N_{c}^{max} \) \textbf{do}
7: \hspace{1cm} Generate the fake data \( x_{g}^{i} \) for the specified class \( c_i \) with \( G(z; c_i) \);
8: \hspace{1cm} Employ discriminator model \( D \) to verify the output of generator as \( c_g \);
9: \hspace{1cm} \textbf{if} \( c_i = c_g \) \textbf{then}
10: \hspace{1.5cm} Add the generated data \( x_{g}^{i} \) to the fake dataset \( x_g \);
12: \hspace{0.5cm} \textbf{end if}
13: \hspace{0.5cm} \textbf{end while}
14: \hspace{0.5cm} \textbf{end for}
15: \hspace{0.5cm} Combine the generated dataset \( x_g \) with the real dataset \( x_r \) to get the balanced dataset \( x_b \);
16: \textbf{return} \( x_b \)

### IV. EXPERIMENTATION

#### A. Datasets

We use two public human activity datasets, WISDM [29] and Unimib-SHAR [30] to train and evaluate our BSDGAN. WISDM contains 6 types of human activity classes with a total of 1,098,207 rows of data (54,901 human activity instances) collected by 36 users. Each row of data contains three-axis acceleration values and timestamp information. The Unimib-SHAR (ADL) dataset contains 9 daily living activities with a total of 7,579 instances of human activity data, each activity instance contains 151 rows of raw data. Each row of data contains three-axis acceleration data, timestamp, and raw signal amplitude, collected by 30 volunteers aged between 18
and 60. These two human activity datasets both suffer from severe data imbalance, the data distributions of the two datasets are shown in Fig. 4.

![Sensor Data Distribution of WISDM (Left) and Unimib-SHAR (Right).](image)

**B. Training Process**

In this section, the framework of all experiments is Tensorflow 2.5.0 based on Python. We use Adam algorithm as the optimizer for generator model and discriminator model. The learning rate is 0.002, beta1 and beta2 are set to 0.5 and 0.9. The size of batch is 128, default latent vector is 100 dimensions. We train 100 epochs on two human activity datasets, each epoch takes 88s on WISDM and 15s on Unimib-SHAR.

During the training process of BSDGAN on the WISDM dataset and UNIMIB dataset, the loss of generator and discriminator is shown in Fig. 5. Benefit from the pre-trained encoder and decoder, BSDGAN can quickly achieve the Nash equilibrium of the generator and discriminator on both datasets. This result demonstrates that BSDGAN can learn the data distribution of real sensor data on imbalanced datasets and generate high-quality synthetic data.

**C. Quality Assessment of Generated Data**

Frechet Inception Distance [31] Score (FID) is employed to evaluate the quality of the generated activity data, it is usually employed to calculate the distance between the data feature distribution of the generated data and real data, which is defined as:

$$
FID(x_r, x_g) = \|\mu_r - \mu_g\|^2 + \text{Tr}(\Sigma_r + \Sigma_g - 2(\Sigma_r \Sigma_g)^{\frac{1}{2}}) \tag{8}
$$

where $\mu_r$ is the mean of the real data features, $\mu_g$ is the mean of the generated data features, $\Sigma_r$ is the covariance matrix of the real data features, $\Sigma_g$ is the covariance matrix of the generated data features.

All FID scores are calculated from the real data of the validation set and the data generated by the generator model. The lower the FID score of the generated data, the higher the data quality. In this paper, we define the FID score between the real data of the training set and the data of the validation set as the best score, the FID score between the data reconstructed by the autoencoder and the data of the validation set as the worst result. We deployed CGAN, ACGAN, BAGAN, and the proposed BSDGAN on the above activity datasets, evaluating the data quality of generated data by these GAN frameworks. On WISDM, the comparison results of the generated data quality for each activity class are shown in Table I, and the comparison results of the generated data quality on Unimib-SHAR-ADL are shown in Table II.

We visualize the generated activity data on two activity datasets, as shown in Fig. 6 and Fig. 7. The first row in figures are the real sensor data visualization, and the remaining two rows are the generated sensor data visualization. The data visualization of different human activities is obviously different, the visual view of sitting and standing is relatively flat, while the view of other human activities, such as Jogging and Walking, fluctuates greatly. Besides, there are differences between the generated data for each human activity, which conforms to the data features of real human activities.

![The generated sensor data on WISDM.](image)
TABLE I
Comparison of FID for generated data by different GANs on WISDM.

| Activity   | Autoencoder | CGAN | ACGAN | BAGAN | BSDGAN | Real Data |
|------------|-------------|------|-------|-------|--------|-----------|
| Jogging    | 298.11      | 240.21 | 197.66 | 165.35 | 119.62 | 119.62    |
| Walking    | 129.28      | 103.89 | 85.45  | 73.28  | 68.14  | 60.15     |
| UpStairs   | 150.82      | 138.77 | 90.98  | 96.19  | 72.89  | 42.31     |
| DownStairs | 97.37       | 93.75  | 86.69  | 78.01  | 60.72  | 51.28     |
| Sitting    | 247.50      | 214.87 | 152.97 | 103.29 | 96.18  | 66.34     |
| Standing   | 151.06      | 134.94 | 88.56  | 83.64  | 54.58  | 28.84     |

TABLE II
Comparison of FID for generated data by different GANs on Unimib-SHAR.

| Activity       | Autoencoder | CGAN | ACGAN | BAGAN | BSDGAN | Real Data |
|----------------|-------------|------|-------|-------|--------|-----------|
| StandingUpFs   | 148.65      | 105.68 | 75.65 | 63.81 | 48.27  | 28.81     |
| SittingDown    | 113.51      | 96.23  | 89.05  | 85.21 | 61.14  | 57.31     |
| GoingUpS       | 131.51      | 106.39 | 83.21  | 74.98 | 64.82  | 48.32     |
| Running        | 147.92      | 132.59 | 103.45 | 83.32 | 78.16  | 52.75     |
| GoingDownS     | 131.51      | 106.39 | 83.21  | 74.98 | 64.82  | 48.32     |
| Jumping        | 91.43       | 84.87  | 73.18  | 68.40 | 63.85  | 52.53     |
| SittingDown    | 92.61       | 85.64  | 76.35  | 68.52 | 41.73  | 38.74     |

TABLE III
Classification accuracy on WISDM before balance.

| Activity   | KNN   | RF    | DT    | CNN  | LSTM | CNNS-LSTM |
|------------|-------|-------|-------|------|------|-----------|
| Jogging    | 0.9208 | 0.9756 | 0.8378 | 0.9936 | 0.9777 | 0.9988 |
| Walking    | 0.9669 | 0.9906 | 0.7819 | 0.9826 | 0.9885 | 0.9932 |
| UpStairs   | 0.1899 | 0.3922 | 0.3270 | 0.7589 | 0.9345 | 0.9777 |
| DownStairs | 0.0960 | 0.1269 | 0.3182 | 0.7566 | 0.9216 | 0.9489 |
| Sitting    | 0.9861 | 0.9653 | 0.9722 | 0.9844 | 0.9723 | 0.9896 |
| Standing   | 0.9885 | 0.9358 | 0.9128 | 0.9702 | 0.9771 | 0.9702 |
| Accuracy   | 0.7849 | 0.8333 | 0.7199 | 0.9393 | 0.9714 | 0.9878 |

TABLE IV
Classification accuracy on WISDM after balance.

| Activity   | KNN   | RF    | DT    | CNN  | LSTM | CNNS-LSTM |
|------------|-------|-------|-------|------|------|-----------|
| Jogging    | 0.9310 | 0.9714 | 0.9924 | 0.9927 | 0.9959 | 0.9994 |
| Walking    | 0.9687 | 0.9899 | 0.9866 | 0.9817 | 0.9951 | 0.9969 |
| UpStairs   | 0.7739 | 0.7835 | 0.7828 | 0.9315 | 0.9678 | 0.9637 |
| DownStairs | 0.7736 | 0.7527 | 0.7897 | 0.9138 | 0.9669 | 0.9782 |
| Sitting    | 0.9970 | 0.9931 | 0.9953 | 0.9653 | 0.9826 | 0.9940 |
| Standing   | 0.9783 | 0.9909 | 0.9776 | 0.9748 | 0.9885 | 1.0 |
| Accuracy   | 0.9039 | 0.9138 | 0.9241 | 0.9600 | 0.9887 | 0.9891 |

D. Usability of Synthetic Data

The ultimate goal of BSDGAN is to balance human activity datasets with generated data and improve the classification accuracy of human activity recognition models. The generation of these data is performed automatically by Algorithm 1 to ensure that the activity dataset is balanced. We adopted K-Nearest Neighbors (KNN), Random Forest (RF), Decision Tree (DT), three traditional machine learning methods and Convolutional Neural Network (CNN), Long Short-Term Memory (LSTM), CNN-LSTM three classical deep learning methods as classifiers for human activity recognition to evalu-
The performance of the above six human activity recognition classifiers before and after the balance on the WISDM dataset is shown in Table III and Table IV, and the performance on the Unimib-SHAR dataset before and after the balance is shown in Table V and Table VI. The classification accuracy of traditional machine learning methods has been greatly improved after the dataset is balanced. For example, the accuracy of Decision Tree on WISDM has increased from 71.99% to 92.41%, and the accuracy of Random Forest on Unimib-SHAR has increased from 59.70% to 85.00%.

The deep learning methods also solved the problem of poor recognition accuracy for minority activity classes. For example, the accuracy of CNN for upstairs in WISDM has increased from 75.89% to 93.15%, and the accuracy of LSTM for LyingDownFS on Unimib-SHAR has increased from 59.65% to 95.14%. The above experimental results show that our BSDGAN can effectively generate high-quality human activity data, and the human activity dataset balanced with the generated data can obviously improve the accuracy of human activity recognition classifiers.

V. CONCLUSION AND FUTUREWORK

In this paper, we use a generative adversarial network framework to generate human activity sensor data, and the generated sensor data are adopted to balance the human activity dataset. We employ an autoencoder with an intermediate embedding model to give the GAN framework prior knowledge for all activity classes directly, helps stabilize the GAN training process. Besides, we add conditional constraints to enable the GAN framework generate activity data for target human activity classes. Our experimental results on two public datasets show that the performance of HAR classifiers are all significantly improved after the dataset balanced, solves the problem that the recognition accuracy of deep learning method-based HAR classifiers is poor for the minority activity classes.

REFERENCES

[1] C. Jobanputra, J. Bavishi, and N. Doshi, “Human activity recognition: A survey,” Procedia Computer Science, vol. 155, pp. 698–703, 2019.
[2] Y. Wang, S. Cang, and H. Yu, “A survey on wearable sensor modality centred human activity recognition in health care,” Expert Systems with Applications, vol. 137, pp. 167–190, 2019.
[3] A. Sathyanarayana, F. Ofli, L. Fernandez-Luque, J. Srivastava, A. Elmagarmid, T. Arora, and S. Taheri, “Robust automated human activity recognition and its application to sleep research,” in 2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW). IEEE, 2016, pp. 495–502.
[4] J. W. Lockhart, T. Pulickal, and G. M. Weiss, “Applications of mobile activity recognition,” in Proceedings of the 2012 ACM Conference on Ubiquitous Computing, 2012, pp. 1054–1058.
[5] Y. Chen and C. Shen, “Performance analysis of smartphone-sensor behavior for human activity recognition,” IEEE Access, vol. 5, pp. 3095–3110, 2017.
[6] K. Chen, D. Zhang, L. Yao, B. Guo, Z. Yu, and Y. Liu, “Deep learning for sensor-based human activity recognition: Overview, challenges, and opportunities,” ACM Computing Surveys (CSUR), vol. 54, no. 4, pp. 1–40, 2021.
[7] I. Goodfellow, “Nips 2016 tutorial: Generative adversarial networks,” arXiv preprint arXiv:1701.00160, 2016.
[8] J. Bao, D. Chen, F. Wen, H. Li, and G. Hua, “Cvae-gan: fine-grained image generation through asymmetric training,” inProceedings of the IEEE international conference on computer vision, 2017, pp. 2745–2754.

[9] R. A. Hamad, L. Yang, W. L. Woo, and B. Wei, “Joint learning of temporal models to handle imbalanced data for human activity recognition,” Applied Sciences, vol. 10, no. 15, p. 5293, 2020.

[10] P. Paul and T. George, “An effective approach for human activity recognition on smartphone,” in 2015 IEEE International Conference on Engineering and Technology (ICETECH). IEEE, 2015, pp. 1–3.

[11] L. Fan, Z. Wang, and H. Wang, “Human activity recognition model based on decision tree,” in 2013 International Conference on Advanced Cloud and Big Data. IEEE, 2013, pp. 64–68.

[12] Z. Feng, L. Mo, and M. Li, “A random forest-based ensemble method for activity recognition,” in 2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC). IEEE, 2015, pp. 5074–5077.

[13] H. H. Ali, H. M. Mofthah, and A. A. Youssif, “Depth-based human activity recognition: A comparative perspective study on feature extraction,” Future Computing and Informatics Journal, vol. 3, no. 1, pp. 51–67, 2018.

[14] J. Wang, Y. Chen, S. Hao, X. Peng, and L. Hu, “Deep learning for sensor-based activity recognition: A survey,” Pattern Recognition Letters, vol. 119, pp. 3–11, 2019.

[15] M. Arjovsky, S.-B. Cho, “Evaluation of deep convolutional neural network architectures for human activity recognition with smartphone sensors,” 2015.

[16] D. Anguita, A. Ghio, L. Oneto, X. Parra Perez, and J. L. Reyes Ortiz, “A public domain dataset for human activity recognition using smartphones,” in Proceedings of the 21th international European symposium on artificial neural networks, computational intelligence and machine learning, 2013, pp. 437–442.

[17] Y. Chen, K. Zhong, J. Zhang, Q. Sun, X. Zhao et al., “Lstm networks for mobile human activity recognition,” in Proceedings of the 2016 International Conference on Artificial Intelligence: Technologies and Applications, Bangkok, Thailand, 2016, pp. 24–25.

[18] R. Mutegeki and D. S. Han, “A cnn-lstm approach to human activity recognition,” in 2020 International Conference on Artificial Intelligence in Information and Communication (ICAIC). IEEE, 2020, pp. 362–366.

[19] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in International conference on machine learning. PMLR, 2017, pp. 214–223.

[20] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” Advances in neural information processing systems, vol. 30, 2017.

[21] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” arXiv preprint arXiv:1411.1784, 2014.

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

[23] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in International conference on machine learning. PMLR, 2017, pp. 2642–2651.

[24] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in International conference on machine learning. PMLR, 2017, pp. 214–223.

[25] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” Advances in neural information processing systems, vol. 30, 2017.

[26] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” arXiv preprint arXiv:1411.1784, 2014.

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

[28] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in International conference on machine learning. PMLR, 2017, pp. 2642–2651.

[29] G. Mariani, F. Scheidegger, R. Istrate, C. Bekas, and C. Malossi, “Bagan: Data augmentation with balancing gan,” arXiv preprint arXiv:1803.09655, 2018.

[30] M. Arjovsky, S. Chintala, and L. Bottou, “Wasserstein generative adversarial networks,” in International conference on machine learning. PMLR, 2017, pp. 214–223.

[31] I. Gulrajani, F. Ahmed, M. Arjovsky, V. Dumoulin, and A. C. Courville, “Improved training of wasserstein gans,” Advances in neural information processing systems, vol. 30, 2017.

[32] M. Mirza and S. Osindero, “Conditional generative adversarial nets,” arXiv preprint arXiv:1411.1784, 2014.

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

[34] A. Odena, C. Olah, and J. Shlens, “Conditional image synthesis with auxiliary classifier gans,” in International conference on machine learning. PMLR, 2017, pp. 2642–2651.