ISSN: 2633-4828 Vol. 5 No.4, December, 2023

International Journal of Applied Engineering & Technology

COMPARISON AND ANALYSIS OF DATA AUGMENTATION METHODS FOR FIXED-LENGTH TIME SERIES DATA

Woosoon Jung[1], Jeong Tak Ryu[2], Kyuman Jeong[3] and Yoosoo Oh[4]

[1]Institute of Special Education & Rehabilitation Science, Daegu University, Gyeongsan-si, Republick of Korea

[2]Department of Electronic and Electrical Engineering, Daegu University, Gyeongsan-si, Republick of Korea

[3,4]School of AI, Daegu University, Gyeongsan-si, Republick of Korea

[1]quado.jung@gmail.com, [2]jryu@daegu.ac.kr, [3]kyuman.jeong@daegu.ac.kr and [4]yoosoo.oh@daegu.ac.kr

ABSTRACT

A fair way to evaluate designed ML models is to check the performance when using the same dataset. The dataset should have a sufficient number of samples and be well-validated. However, in cases where constructing a dataset from scratch is necessary, accurately evaluating ML models can be challenging due to data shortage. To overcome data shortage, data augmentation methods are employed. In this study, we propose methods for augmenting fixed-length time series data using KDE and semi-supervised learning techniques. In the KDE approach, we evaluate performance based on two types of kernels, while in the semi-supervised learning approach, we apply noise injection and Label Spreading methods. In both approaches, we adjust the augmentation ratio of the original data to track performance changes and determine the most effective method for the dataset at hand.

Keywords: Data Augmentation, Data Shortage, Time Series Data.

• INTRODUCTION

○ Motivation

Artificial Intelligence (AI) performance is absolutely influenced not only by the type and size of the AI model, but also by the dataset applied. Widely known datasets, from the small dataset MNIST to the large dataset ImageNet, have sufficient samples and are well-validated datasets. For this reason, many studies on vision-based AI models evaluate their performance using the corresponding dataset. If the application to which AI is being applied cannot use well-verified datasets such as MNIST [1], CIFAR [2], ImageNet [3], or COCO [4], data shortages or overfitting may occur. This is the case when using a custom dataset. Most reliable datasets are focused on the computer vision field, and it is difficult to build datasets in special-purpose systems such as wearable Human Activity Recognition (HAR) systems, behavioral data of the disabled, and sensor-based systems [5-11]. In this study, data augmentation techniques are applied to a custom dataset built to recognize finger movements. Measure performance changes by considering the data augmentation method, data augmentation rate, and size of the Machine Learning (ML) model, and select the most appropriate technique for the application.

○ Background

In a previous study that proposed energy-accuracy optimization, a wearable device was implemented and finger movements were recognized [12]. The output of the system consisting of a two-axis flex sensor and Micro Controller Unit (MCU) is the finger joint angle in the range of -180˚ to 180˚. The dataset is time series data in which the angles of the x- and y-axis finger joints change over time. The most suitable ML model in this study was Multi-Layer Perceptron (MLP) with a single hidden layer. Since the applied model is MLP, all samples in the dataset have n fixed sizes. The dataset used in this custom system was constructed directly due to cost and time limitations.

The simplest and most intuitive example of a data augmentation technique is to increase the number of images by rotating, cropping, or flipping them [13]. The techniques to be compared and analyzed in this study using time series data are the Kernel Density Estimation (KDE) method and the Label Spreading method with Gaussian noise added. First, the KDE method is a non-parametric estimation method that finds probability density using only observed data. In a previous study that performed data augmentation using the KDE method, it was

confirmed that the efficiency of data augmentation decreases as the number of original data increases [14]. Second, the Label Spreading method is one of the semi-supervised learning methods and is a method of labeling some unlabeled data points. In this study, unlabeled data samples are created by injecting noise into the original data, and then labeling is performed using the Label Spreading method. Finally, use the generated data sample to check performance changes.

- **DATA AUGMENTATION METHOD**

  - **KDE Method**

The most representative example of a density estimation method is a histogram, which visualizes bins and the amount of data within each bin. The kernel density estimation technique estimates the distribution (density) of the original variable from the distribution of the observed data, thus well preserving the characteristics of the original data [15, 16]. Rather than assuming that the relationship between data samples follows a specific distribution, the distribution is estimated using actual observed data. The definition of the kernel function used in KDE is that the integral value is 1 (Equation 1), and it is symmetrical around the origin and is a non-negative function (Equation 2).

$$\int_{-\infty}^{\infty} K(u)\,du = 1 \quad (1)$$

$$K(u) = K(-u), K(u) \geq 0, \forall\, u \quad (2)$$

In the equation 3, K is the kernel, the i-th data sample of the random variable x is x_i, and h is a parameter called bandwidth. Bandwidth makes the shape of the kernel sharp or smooth.

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh} K\left(\frac{x - x_i}{h}\right) \quad (3)$$

Figure 1 shows Tophat and Gaussian kernel. The bandwidth of the Gaussian kernel is large, and the bandwidth of Tophat is small, showing a gentle and sharp shape, respectively.

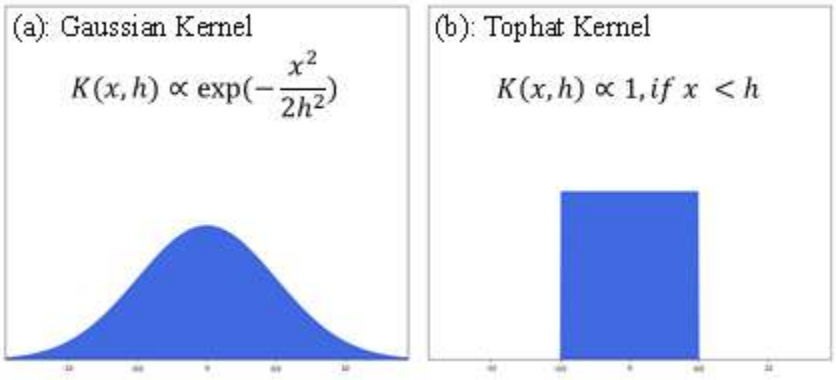

Fig. 1. Kernel shape according to bandwidth (h)

  - **Label Spreading Method**

Semi-supervised learning is a method that combines the characteristics of supervised learning and unsupervised learning. Semi-supervised learning is used when only some data is labeled and the rest is unlabeled. The main parameter of the Label Spreading algorithm is the kernel, and Radial Basis Function (RBF) and K-Nearest-Neighbors (KNN) kernels are mainly used. Both kernels are functions that measure similarity between data samples, and determine similarity through distance in dimensional space. In this study, the KNN algorithm was adopted, and the number of neighbors is determined through repeated experiments.

- **EXPERIMENTAL SETUP**

In this chapter, data augmentation techniques for fixed-length time series data are applied, and performance changes according to the parameters of each technique are confirmed. The optimal algorithm explored with the same dataset in [12] was MLP. The optimal scenario explored considering energy consumption in the system was MLP processing 28 samples per given time. Since MLP requires a fixed-length input, the output of the flex sensor that changes over time was preprocessed and arranged in order by a set number. The original dataset constructed in the same manner as above has 17 classes defined, and there are 300 samples for each class. Here, 250 samples per class are used as a test set and are not exposed during the learning and data augmentation process. The remaining 50 samples per class are used for both augmentation and learning processes. Previous research [14] showed that increasing the amount of data to learn does not necessarily have a positive effect on performance. If the original dataset has not been sufficiently verified or does not reflect the characteristics of the original dataset due to excessive data augmentation, data augmentation may actually cause performance reduction. For this reason, the rate of data augmentation is also explored in the experiment. Figure 2 is an example of over-augmented data samples. Augmenting data at an augmentation rate that is too high can break down the boundaries between classes.

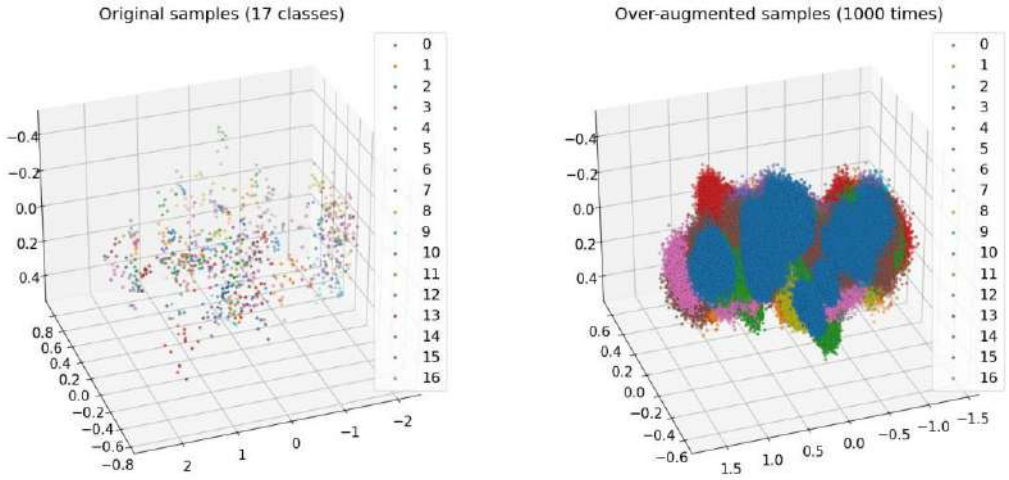

Fig. 2: An example of over-augmented data samples (1000 times)

If the number of training data samples increases, a larger ML model may be needed. If the data shortage environment is overcome, sufficient learning is possible and can be applied to a model with more parameters. Overall, in this study, performance evaluation is conducted according to the type of kernel, data augmentation rate, and size of the MLP model for each method. Both KDE and Label Spreading methods were evaluated using Scikit-learn's API [17]. To check the number of MLP parameters required for each augmentation factor, an experiment is conducted by changing the number of nodes in the hidden layer of the MLP to 50, 100, 150, and 200.

- **Data Augmentation using KDE Method**

The kernel types, which are the main parameters of KDE used in the experiment, were Tophat and Gaussian. Figure 3 shows the data augmentation results according to the type of each kernel. As can be intuitively seen, the Tophat kernel has a narrow bandwidth, so the range in which new data samples are created is narrow, and the Gaussian has a wide bandwidth, so new samples are created in a spread form from the original data.

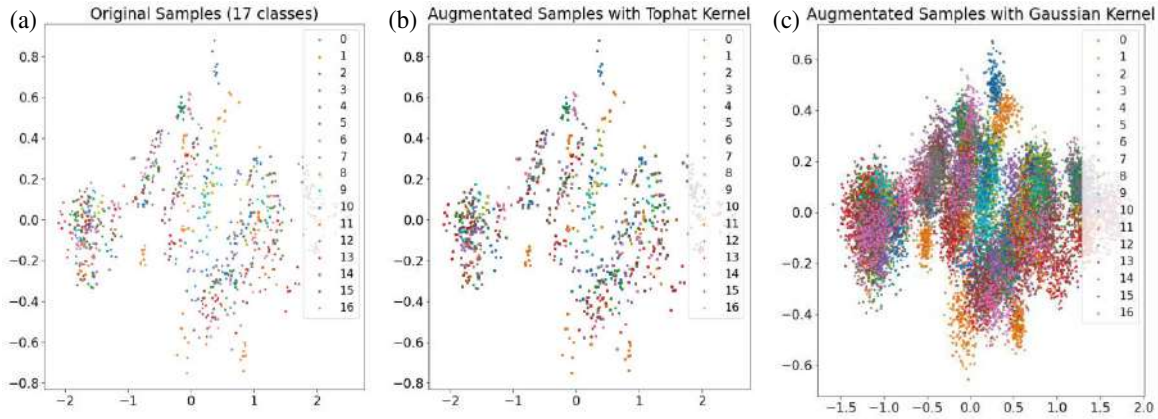

Fig. 3: Original dataset and dataset augmented by KDE method
(a): Original dataset, (b): Tophat kernel, (c): Gaussian kernel.

- **Data Augmentation using Label Spreading Method**

Before applying the Label Spreading Method, apply Gaussian noise. The range of Gaussian noise was set to within 1% of the range of the raw data. Raw data has values ranging from -180˚ to 180˚, so the noise range is applied within 3.6˚. Additionally, the raw data range after noise injection is processed so that it does not exceed the actual sensor output range of -180˚~180˚. Figure 4 shows the process of injecting noise into the original data and Label Spreading.

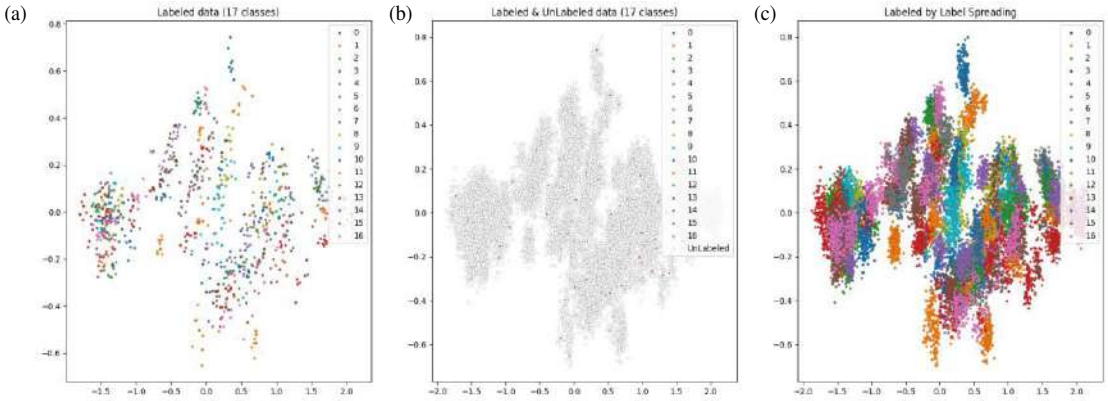

Fig. 4: Dataset augmented by label spreading method
(a): Original dataset, (b): Noise injection, (c): Label Spreading applied.

Figure 5 (a), (b), (c), and (d) show the experimental results when the number of nodes in the hidden layer of the MLP is 50, 100, 150, and 200, respectively. Overall, as the augmentation factor increases, KDE and Label Spreading using the Tophat kernel show a trend of increasing performance according to the augmentation factor. On the other hand, in KDE with Gaussian kernel, performance tends to decrease as the augmentation factor increases. It can be inferred that the wide bandwidth of the Gaussian kernel is not suitable for the dataset. The highest overall performance is the Label Spreading method with an accuracy of 92.67%, and the augmentation factor is 40 (figure 5(b)). The case with the highest increase in accuracy is Label Spreading (augmentation factor: 50) in figure 5(c). The performance of the MLP learned with the original dataset was 78.43%, but with data augmentation, the performance was achieved at 92.20%, showing a performance improvement of about 13.77%.

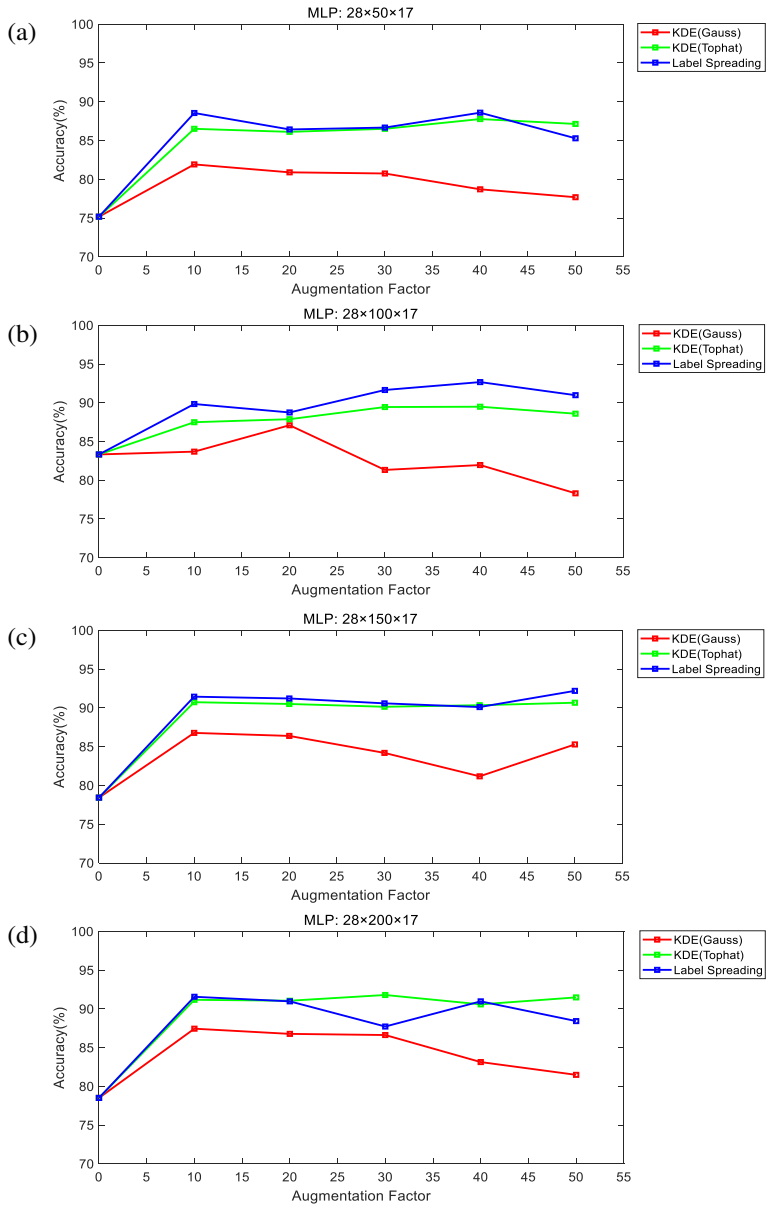

Fig. 5. Performance change graph according to augmentation factor and size of MLP. Number of nodes in hidden layer (a): 50, (b): 100, (c): 150, (d): 200.

- **CONCLUSIONS**

In this study, augmentation was applied to a dataset consisting of fixed-length time series data using the KDE method and the Label Spreading method. As a result, an accuracy of up to 92.67% and an accuracy increase of up to 13.77% were achieved. Even though the test set for evaluation was not exposed during the training and data augmentation process, accuracy improvement was achieved. As a result, the Label Spreading method injecting Gaussian noise showed the best performance. In the KDE method using the Tophat kernel, a linear increase in accuracy was confirmed.

ACKNOWLEDGMENTS.

This work was supported by the Ministry of Education of the Republic of Korea and the National Research Foundation of Korea (NRF-2022S1A5C2A07091326).

REFERENCES

1. LeCun, Y., Cortes, C., & Burges, C.J.C. (n.d.). The MNIST Database of Handwritten Digits. http://yann.lecun.com/exdb/mnist/
2. Krizhevsky, A., Hinton, G. (n.d.). Learning Multiple Layers of Features from Tiny Images. Technical Report. https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf
3. Deng, J., Dong, W., Socher, R., Li, L.-J., Li, K., & Fei-Fei, L. (2009). ImageNet: A Large-Scale Hierarchical Image Database. In 2009 IEEE Conference on Computer Vision and Pattern Recognition (pp. 248-255). IEEE. https://doi.org/10.1109/CVPR.2009.5206848
4. Lin, T.-Y., Maire, M., Belongie, S., Hays, J., Perona, P., Ramanan, D., ... & Zitnick, C. L. (2014). Microsoft COCO: Common Objects in Context. In European Conference on Computer Vision (pp. 740-755). Springer, Cham. https://doi.org/10.1007/978-3-319-10602-1_48
5. Thiry, P., Houry, M., Philippe, L., Nocent, O., Buisseret, F., Dierick, F., Slama, R., Bertucci, W., Thévenon, A., & Simoneau-Buessinger, E. (2022). Machine Learning Identifies Chronic Low Back Pain Patients from an Instrumented Trunk Bending and Return Test. Sensors, 22(13), 5027. https://doi.org/10.3390/s22135027
6. Jantawong, P., Hnoohom, N., Jitpattanakul, A., & Mekruksavanich, S. (2021). A Lightweight Deep Learning Network for Sensor-based Human Activity Recognition using IMU sensors of a Low-Power Wearable Device. In 2021 25th International Computer Science and Engineering Conference (ICSEC) (pp. 459-463). Chiang Rai, Thailand. https://doi.org/10.1109/ICSEC53205.2021.9684631
7. Mekruksavanich, S., Jantawong, P., & Jitpattanakul, A. (2023). Deep Learning Approaches for HAR of Daily Living Activities Using IMU Sensors in Smart Glasses. In 2023 Joint International Conference on Digital Arts, Media and Technology with ECTI Northern Section Conference on Electrical, Electronics, Computer and Telecommunications Engineering (ECTI DAMT & NCON) (pp. 474-478). Phuket, Thailand. https://doi.org/10.1109/ECTIDAMTNCON57770.2023.10139685
8. B. R., Dayal, H. S., & Sankpal, K. (2019). Emotion Classification Using Single-Channel EEG. In 2019 International Conference on Computing, Power and Communication Technologies (GUCON) (pp. 360-366). New Delhi, India.
9. Niu, X., Wang, Z., & Pan, Z. (2019). Extreme Learning Machine-Based Deep Model for Human Activity Recognition with Wearable Sensors. Computing in Science & Engineering, 21(5), 16-25. https://doi.org/10.1109/MCSE.2018.110145933
10. Islam, S. M. M., & Talukder, K. H. (2023). Exploratory Analysis of Smartphone Sensor Data for Human Activity Recognition. IEEE Access, 11, 99481-99498. https://doi.org/10.1109/ACCESS.2023.3314651
11. Oguntala, G. A., et al. (2019). SmartWall: Novel RFID-Enabled Ambient Human Activity Recognition Using Machine Learning for Unobtrusive Health Monitoring. IEEE Access, 7, 68022-68033. https://doi.org/10.1109/ACCESS.2019.2917125
12. Jung, W., & Lee, H.G. (2022). Energy–Accuracy Aware Finger Gesture Recognition for Wearable IoT Devices. Sensors, 22(13), 4801. https://doi.org/10.3390/s22134801
13. Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. Journal of Big Data, 6, 60. https://doi.org/10.1186/s40537-019-0197-0
14. Jung, W., & Lee, H.G. (2022). Data Augmentation using a Kernel Density Estimation for Motion Recognition Applications. Journal of Korea Society of Industrial Information Systems, 27(4), 19-27.

15. Parzen, E. (1962). On Estimation of a Probability Density Function and Mode. The Annals of Mathematical Statistics, 33(3), 1065-1076. https://doi.org/10.1214/aoms/1177704472

16. Izenman, A. J. (1991). Recent Developments in Nonparametric Density Estimation. Journal of the American Statistical Association, 86(413), 205-224. https://doi.org/10.1080/01621459.1991.10475021

17. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine Learning in Python. Journal of Machine Learning Research, 12, 2825-2830. Retrieved from http://jmlr.org/papers/v12/pedregosa11a.html