논문 22-47-03-11 The Journal of Korean Institute of Communications and Information Sciences '22-03 Vol.47 No.03
https://doi.org/10.7840/kics.2022.47.3.492

Synthetic Data Generation Using GAN for RUL Prediction of Supercapacitors

Miracle Udurume[◆], Chigozie Uzochukwu Udeogu[*], Angela C. Caliwag[*], Wansu Lim[°]

ABSTRACT

The remaining useful life (RUL) prediction of supercapacitors is an important part of supercapacitors management system. To accurately predict the RUL of supercapacitor, a large amount of capacity data is required which can be difficult to acquire due to privacy restrictions and limited access. Previous works have employed the use of deep learning models to synthetically generate data. However, a prerequisite ensuring the success of these models depends on their ability to preserve the temporal dynamics of the data. This paper presents a generative adversarial network (GAN) for synthetic data generation and a long short-term memory (LSTM) network for accurate RUL prediction. Firstly, the GAN model is employed for synthetic data generation and LSTM for RUL prediction. We show that the GAN model is capable of preserving the temporal dynamics of the original data and also prove that the generated data can be used to accurately carry out RUL prediction. Our proposed GAN model was able to achieve an accuracy of 85% after 500 epochs. The performance of the generated data set with the LSTM model achieved an RMSE of 0.29. The overall results show that synthetic data can be used to achieve excellent performance for RUL prediction.

Key Words : Deep learning models, generative adversarial network, supercapacitors, synthetic data generation, remaining useful life

Ⅰ. Introduction

Supercapacitors are known for their high power density, wide temperature range as well as long service life[3]. Supercapacitors are applied in a wide range of fields such as microgrid[4], renewable energy[5], hybrid vehicles[6]. Due to the wide use of supercapacitors, the remaining useful life (RUL) will directly affect the reliability and safety of the entire system or machine[7,8]. RUL is the length of time a machine or system is likely to operate before it requires repair or replacement. RUL helps to predict the current health status of a system and allows scheduled maintenance and optimization of operation efficiency[9]. There are two ways by which RUL prediction can be achieved mainly the model-based method and data-driven method. The model-based method combines the use of different models and filtering methods to achieve prediction. However, due to the complex nature of supercapacitors, the model-driven approach is complex and difficult to implement. In contrast, the data-driven method attempts to derive the degradation process of a system from measured data using machine learning

※ This work was supported by the Ministry of SMEs and Start-ups(S3010704), and by the National Research Foundation of Korea (2021R1I1A3056900)

◆ First Author : Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, 학생회원

° Corresponding Author : Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering, wansu.lim@kumoh.ac.kr, 정회원

* Kumoh National Institute of Technology, Department of Aeronautics, Mechanical and Electronic Convergence Engineering

논문번호 : 202110-264-C-RE, Received September 30, 2021; Revised December 14, 2021; Accepted December 24, 2021

techniques[10]. This method is used to carry out RUL predictions based on historical data measured from systems. Hence, the prediction accuracy of data-driven methods depends not only on the quality but the quantity of the historical data which is generally difficult to obtain for use in real cases[11]. A solution to employ is the use of synthetic data generation (SDG) which is the process of generating data artificially to preserve privacy in cases where data are limited due to privacy requirements. The generated synthetic data can be used as training data for machine learning algorithms.

In this paper, we present a deep learning model for SDG. Specifically a Generative adversarial Networks (GANs) is used to generate super capacitor data. Unlike other methods, GAN is capable of preserving the temporal dynamics of the data. That is, it is capable of generating a data set with a certain trend. The main contributions of this paper are as follows: firstly, we generate the capacity data from the original supercapacitor data set. Secondly, we compare the results between the original data set and the generated data set. Lastly, we use the generated data set to perform RUL prediction of supercapacitors using an LSTM model.

The remainder of this paper is organized as follows. The related works are presented in Section II. Section III introduces the method employed in detail. Section IV describes the experimental results while Section V gives the conclusion of the paper.

Ⅱ. Related Works

The role of SDG is rapidly increasing since machine learning algorithms are trained with an incredibly large amount of data which can usually be very difficult to obtain or generate. SDG can be generated using two methods: mathematical models and deep learning.

2.1 Supercapacitors

Supercapacitors is a type of capacitor that has a capacitance value extremely higher than the conventional electrolyte capacitors. It consists of positive and negative electrodes (current collectors), a separator, and the electrodes. The separator is a membrane that insulates the electrodes and guarantees only the mobility of ions rather than the electric connection between them[12]. The cations and the anions of the electrolyte solution are respectively attracted by the negative and positive electrodes forming thin double layers without the need to charge transfer[13]. Supercapacitors have a demonstrated robustness in applications such as hybrid vehicles, wind turbine pitch systems, automotive, power grid, energy and utilities, aerospace, military radio communication devices and medical fields[14] as shown in Fig. 1.

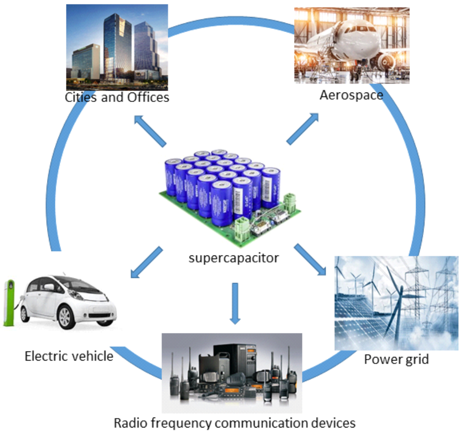

Fig. 1. Application of Super Capacitors.

2.2 Mathematical models

Mathematical models involve generating data through the use of probabilistic models such as the Gaussian mixture model, Autoregressive moving average, Monte Carlo method, and Markov chain. These models use statistical analysis and computational algorithms to better understand the data and generate random samples. In [15], the authors proposed a statistical model combining Fourier series and autoregressive moving average (ARMA) to synthetically generate weather conditions such as wind speed and grid demand data. Probabilistic analysis of hybrid system configuration was carried out based on the synthetic data to understand component ramping requirements. In [16], the Markov chain method was used to generate operational data for subsequent use in monitoring, predicting, controlling battery pack state of health based on capacity variations. Here the electric vehicle data set was used to generate transition probability matrices using the concept of Markov chain propagation. In [17], the application of the Gaussian mixture model is used as a method for generating independent and identically distributed samples of data which was originally obtained from immersive virtual environments to solve the issue of the small sample size associated with the immersive virtual environment experiments. The Gaussian mixture model was trained by implementing an expectation-maximization algorithm and the K-mean algorithm was used to initialize its parameters. Despite the advantages that can be obtained from the use of mathematical models there are still some disadvantages to them which include the difficulty and complexity in building such models also they require the use of expert engineers to carry out such experiments in other to achieve the desired results.

2.3 Deep learning model

Deep learning models are created using neural networks. They contain the inner layer, hidden layer, output layer. An example of deep learning models for SDG as stated earlier are Variational autoencoders (VAEs) and GAN. VAE are examples of generative models that use variational bayesian inference to approximate the probability density. They work with a pair of entities called encoder and decoder. In [18], a VAE-based synthetic generation method is proposed for the imbalanced learning problem. The VAE was trained to sample values of the latent variables which are likely to produce original data after which new samples were generated from the conditional distribution of the data fed to the latent variable. In [19], research was conducted to test the data construct effect of the VAE by different latent variables. The VAE is known to generate data that looks exactly like the original data. The experimental results showed that the generated data always has a single style when the latent variable number is small. This can be considered as a disadvantage especially when working with time-series data. To generate time-series data, the generative model must be able to preserve temporal dynamics such that new sequences respect the original relationships between variables across time. GAN has been known for its promising result with time-series data. A GAN approach is presented for generating high-fidelity synthetic data based on the batteries parameter. The result showed that the generated data could also be used to increase training data for the state of charge estimation algorithms. In [20], the authors proposed a novel data-driven approach to synthetic data generation by utilizing deep generative adversarial networks (GAN) to learn the conditional probability distribution of essential features in the real dataset and generate samples based on the learned distribution. In this paper, we apply GAN to generate synthetic data and use said generated data to perform RUL prediction.

Ⅲ. Methodology

In this section, we present the GAN model used for synthetic data generation as well as the process used for RUL prediction as shown in Fig. 2. The GAN model is used to synthetically generate new data set after which the generated data is used to perform RUL prediction.. The proposed method for the data generation consists of two steps the model training, and the sample generation step. In each step, we carry out different processes in other to generate synthetic data. In the model training, the proposed GAN model is built and trained to learn the input data while in the sample generation step the data is generated and evaluated to see if the model was able to learn the data set.

3.1 Dataset

In this paper, the dataset used consists of the values of supercapacitor parameters such as current, power, voltage and resistance measured every 0.1 second. The data is obtained from a charge-discharge cycle test of an actual 21000F supercapacitor. The supercapacitor has a rated working voltage of 4.2V, absolute maximum voltage

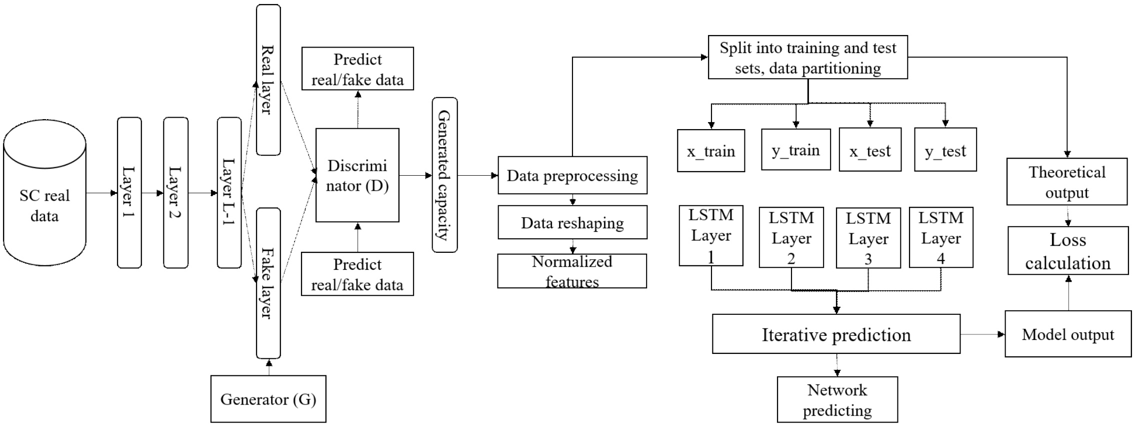

Fig. 2. System overview.

of 4.25V, absolute maximum current of 20A, operating temperature range of -20°C-55°C, maximum working temperature of up to 60°C, and an internal resistance greater than 1.50mΩ. The total number of real samples used consists of data from 1750 charge-discharge cycles. Among 1750 cycles, 1400 cycles are used for training while the rest are used for testing.

3.2 Data Generation

We define an Unconditional GAN model as shown in Fig. 3 which is used to generate synthetic datasets. GAN consists of two adversarial models: a generative model G that captures the data distribution, and a discriminative model D that estimates the probability that a sample came from the training data rather than the generative model. Both G and D could be a non-linear mapping function. To learn a generator distribution p_g over data x, the generator builds a mapping function from prior noise distribution $p_{z(Z)}$ to data space as $G(Z)$ and the discriminator $D(x)$, outputs a single scalar representing the probability that x came from training data rather than p_g. G and D are both trained simultaneously while we adjust parameters for G to minimize $\log(1-D(G(z)))$ and adjust parameters for D to minimize $\log D(x)$, as if they are following the two-player min-max game with value function:

$$\min_G \max_D (D,G) = E_{x\sim P_{g(x)}}[\log D(x)] + E_{z\sim P_{z(x)}}[\log(1-D(G(z))] \tag{1}$$

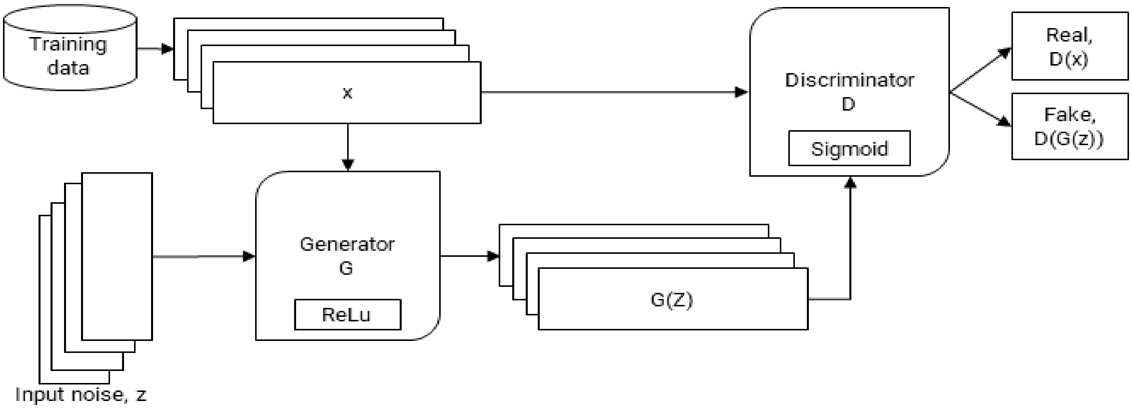

Fig. 3. Unconditional GAN.

In the generator net, a prior z with dimensionality of 100 was drawn from a uniform distribution. Both z and x are mapped to hidden layers with Rectified Linear unit (ReLu) activation before being mapped to the second combined ReLu layer. We then use a sigmoid unit layer as output for generating the capacitance samples. The model was trained using stochastic gradient decent with mini-batch size of 128 and initial learning rate of 0.01 which was decreased down to 0.001. Also momentum was used with initial value of 0.5. A dropout with probability of 0.5 was applied to both the generator and the discriminator and best estimate of log-likelihood on the validation set was used as stopping point. We further explain the processes used for data generation as shown in Fig. 4.

Step 1. Model training: The 21000F supercapaitor dataset was loaded with the aim of generating more capacity data as an indication that synthetic data can be used in cases of limited data for accurate prediction of supercapacitor. We define a class GAN with Adam optimizer as well as generator noise. We stack and train the generator to fool the discriminator. Next, we train the model by epoch number and create a path to save, load, and return the generated data set. The model consists of hyperparameters such as batch normalization, activation functions, and dense layers. Two different types of activation functions was used for both the generator and the discriminator. The relu activation function was used for the generator while the sigmoid function was for the discriminator. The sigmoid function is used for the discriminator as it takes real values as input and outputs them in the range of 0 to 1 hence, the larger the input the closer the output value will be. We use the Adam optimizer with tuned hyperparameters which include a learning rate of 0.001. We choose the learning rate to be as small as possible and a momentum term beta function of 0.5 to avoid training oscillation and instability of the model. We train the model using 500 epochs and generate new data which is then saved as a CSV file.

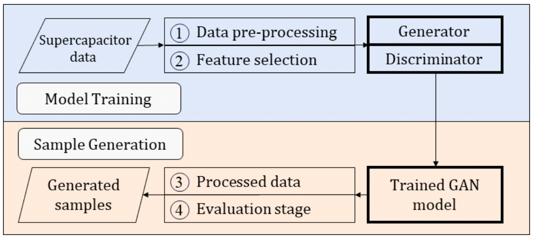

Fig. 4. Methodology for training of GAN model.

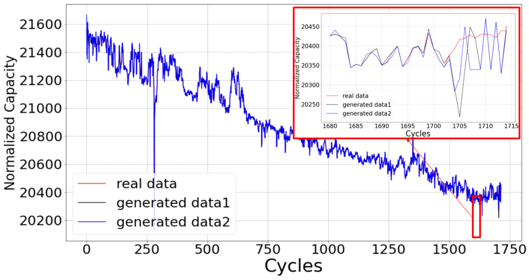

Fig. 5. Plot of real and generated data set.

Step 2. Sample generation: For the generation process step we build the model using 2 input layers and 5 dense layers. We start with a seed size of 17 and a model step once the data has been generated we test the model again using a different seed size to check the variation in both data generated. We define a model step in the evaluation stage to check the samples generated by the GAN model. In training the GAN model, the original data is loaded to the model to see how well the model ia able to learn the dataset. Using data from 1750 cycles, 2 different data with 1750 cycles are also generated. In viewing all cycles, the difference between the original and generated dataset is not observed. Hence, a zoomed-in version from cycles 1630 to 1750 is shown in the subplot in Fig. 5. As shown in the figure real data represents the original data set while generated data1 and generated data2 represent the two generated data sets. Generated data1 and generated data2 are obtained using seed values of 17 and 20 respectively both generated data sets are trained using 500 epochs with a mini-batch size of 128. For the second generated dataset, we tune the model hyperparameters a bit by using a learning rate of 0.001 with a momentum term beta function of 0.6.

3.3 RUL prediction

The concept of RUL is utilized to predict the lifespan of a system to minimize catastrophic failures in both the manufacturing and service sectors. In this paper, after our data set has been generated we then perform RUL prediction on the generated data set. We implement the RUL prediction using a long-short term memory (LSTM) network. LSTM is an artificial recurrent neural network used in the field of deep learning and unlike standard feed-forward neural networks, LSTM has feedback connections that can be used to process not just single data points but also an entire sequence of data. LSTM unit composes of a cell, an input gate, an output gate, and a forget gate. A diagram of a simple LSTM network is shown in Fig. 6. The LSTM model is used for the remaining useful life prediction of the supercapacitor. The aim of predicting RUL using the generated dataset is to show that in cases of limited dataset, synthetic data can also be applied for research purposes and can still perform and produce accurate result as the original dataset would.

In this paper, the input variable used for the LSTM model was the original dataset as well as the generated dataset. Both datasets are used in order to obtain the validity of the GAN model. The input data is prepared using a 2D feature NumPy array with random integers. The data set is split in the ratio of 80/20, 80 been used for training while 20 for testing. The data is preprocessed using feature scaling to scale the data to be valued between 0 and 1. Scaling data before feeding it into the neural network is a good practice as it helps for optimal performance that is for noise removal and normalization. Next, we predict values taking N=100 (where N=100 is the number of training dataset used) with a sliding window of I=10 and y_train will contain values of I+1 values which we want to predict after which, we convert the x_train and y_train into NumPy array values and reshape into a 3D array which is accepted by the LSTM model. After reshaping, we built the architecture by making an object of the sequential model. We add the LSTM layer with parameters (units: the dimension of output space, input shape: the shape of the training set, return_sequences: True or false which determines whether to return the last output in the output sequence or the full sequence. We add 4 of the LSTM layers each with a dropout layer of value 0.2). The final layer is the output layer which is a fully connected dense layer (units = 1 as we are predicting only one value that is I+1). The dense layer operates on the input layers and returns the output and every neuron at the previous layer is connected to the neurons in the next layer hence it is called a fully connected dense layer. We compile the model using ‘adam optimizer’ (which is a learning rate optimization used while training of deep neural network models) and error is calculated by loss function ‘mean squared error’ (as it is a regression problem we use a mean squared error loss function) using the formula in (2) below.

$$R_{\mathrm{MSE}} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(d)^2} \tag{2}$$

where d represents the capacity and N is the number of cycles.

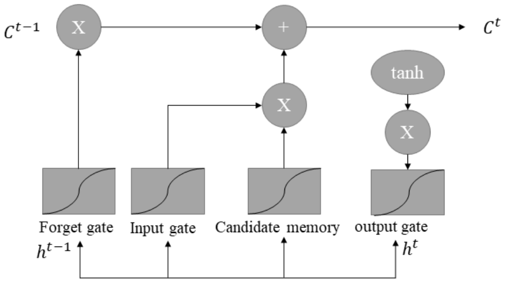

Fig. 6. Working principle of an LSTM network.

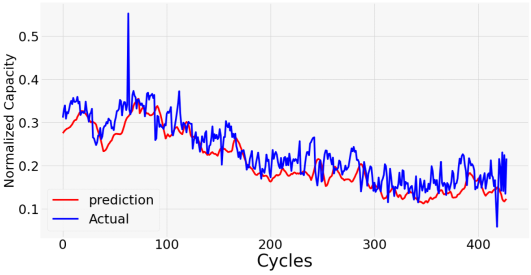

Fig. 7. RUL prediction of generated data set.

We then fit the model on 30 epoch (epochs are the number of times we pass the data into the neural network) and a batch size of 50 (we pass the data in batches segmenting the data into smaller parts so as for the network to process the data in parts). Furthermore, we create test data similar to train data to convert to NumPy array and reshape the array to a 3D shape. Lastly, we make predictions and calculate the 'root mean squared error' (the smaller the root mean square error score the better the model has performed). The result of the RUL prediction is seen in Fig. 7.

Ⅳ. Results and Discussion

This section provides information about the dataset used, the results obtained, and the inference that can be made based on the result. The GAN architecture was evaluated on the datset obtained from a charge-discharge cycle of an actual 21000F supercapacitor. The total number of real samples used consists of data from the 1750 charge-discharge cycles. Among 1750 cycles, 1400 cycles have been used for training while the rest are used for testing. This data set was chosen to test the behavior of the GAN model on decreasing trends of data. A GAN model has been employed to generate the synthetic data. The result of the original data and generated data plot is shown in Fig. 5. Table 1 summarizes the model parameters used for the GAN model training which was generated with a training accuracy of 85%. The result for the obtained validity of the GAN model is shown in table 2. We compared the RMSE and MAPE result of the average gotten from the generated datasets to that of the original dataset. The root mean square error (RMSE) reflects the stability of the model while the mean absolute percent error (MAPE) not only considers the error between the predicted value and the truth value but also considers the proportional relationship between them. Table 2 shows the RMSE and MAE performance error gap between the average datasets and the original data is close which proves that the GAN model was able to learn the original dataset well. For the RUL prediction, the epoch number of the LSTM is set to 250, the hidden unit containing 50 units and a dropout of 0.2 was used. Fig. 6 shows the RUL prediction result. The original data obtained an RMSE value of 0.030 while that of the generated data was 0.029.

Table 1. Parameters for GAN model training.

Parameter	Value
Number of epoch	1000
Batch size	128
Learning rate	0.001
optimizer	Adam
Beta function	0.5

Table 2. Performance evaluation result.

Dataset	RMSE	MAPE
Average of generated dataset	0.029	0.025
Original dataset	0.030	0.027

Ⅴ. Conclusion

Data generation has become a vital tool for researchers and data scientists. To carry out algorithms and research purposes a large amount of data is needed. Generation of data is expensive and time-consuming. This paper utilizes a Generative adversarial network-based approach to generate synthetic data. The technique was employed on the capacity column of the supercapacitor data. The result shows that the GAN model can generate synthetic data while preserving the temporal dynamics of the data. The synthetic data generated can be used in cases of limited datasets and also presents an intuitive technique for increasing training data for RUL prediction. Rul prediction was carried out on generated dataset and results show the ability to predict RUL using data generated by the GAN model.

References

[1] K. Choi, et al., "Implementation of battery management circuit to improve flight time for unmanned aerial vehicles," *J. KICS*, vol. 44,

no. 4, pp. 285-288, Feb. 2020.

[2] K. Choi, et al., “Implementation of RF energy harvesting circuit for wireless charging of iot sensor nodes,” *J. KICS*, vol. 44, no. 4, pp. 755-758, Apr. 2019.

[3] S. Liu, et al., “Review on reliability of supercapacitors in energy storage applications,” *Appl. Energy*, vol. 278, no. 11, pp. 54-36, Aug. 2020.

[4] T. Ma, et al., “Development of hybrid battery-supercapacitor energy storage for remote area renewable energy systems,” *Appl. Energy*, vol. 10, no. 10, pp. 58-63, Dec. 2014.

[5] M. Aszczur, et al., “An optimisation and sizing of photovoltaic system with supercapacitor for improving self-consumption,” *Appl. Energy*, vol. 3, no. 6, pp. 26-19, Aug. 2020.

[6] D. Eroldi, et al., “Sizing for fuel cell/supercapacitor hybrid vehicles based on stochastic driving cycles,” *Appl. Energy*, vol. 183, no. 6, pp. 45-50, Sep. 2016.

[7] J. F. Sun, et al., “Recent progresses in high energy density all pseudocapacitive-electrode-materials-based asymmetric supercapacitors,” *J. Mater. Chem., A*, vol. 5, no. 8 pp. 9443-9464, Aug. 2017.

[8] W.-S. Si, et al., “Remaining useful life estimation—A review on the statistical data driven approaches,” *Eur. J. Oper. Res.*, vol. 213, no. 1, pp. 1-14, Aug. 2011.

[9] C. Hu, et al., “A new remaining useful life estimation method for equipment subjected to intervention of imperfect maintenance activities,” *Chin. J. Aeronaut.*, vol. 31, no. 3, pp. 514-528, Mar. 2018.

[10] J. Liu, et al., “A multi-step predictor with a variable input pattern for system state forecasting,” *Mech. Syst. Signal Process.*, vol. 23, no. 3, pp. 1586-1599, Jul. 2009.

[11] Y. Roh, et al., “A survey on data collection for machine learning: A big data - ai integration perspective,” *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 4 pp. 1328-1347, Apr. 2021.

[12] Z. S. Iro, et al., “A brief review on electrode materials for supercapacitor,” *Int. J. Electrochem. Sci.*, vol. 11, no. 12, pp. 10628-10643, Dec. 2016.

[13] K. Vulrilehto, et al., “Supercapacitors-basics and applications,” *IEEE Trans. Computat. Soc. Syst.*, vol. 23, Jun. 2014.

[14] R. Gu, et al., “A novel battery/Ultracapacitor hybrid energy storage system analysis based on physics-based lithium-ion battery modelling,” in *Proc. ITEC*, pp. 1-6, May 2015.

[15] J. Chen, et al., “Probabilistic analysis of hybrid energy systems using synthetic renewable and load data,” in *Proc. ACC*, pp. 4723-4728, Bournemouth Int. Centre Labour, England, Sep. 2017

[16] M. Pyne, et al., “Generation of synthetic battery data with capacity variation,” in *Proc. CCTA*, pp. 476-480, Aug. 2019.

[17] AU. Chokwitthaya, et al., “Applying the Gaussian mixture model to generate large synthetic data from a small data set,” *IEEE Trans. Emerg. Topics Comput.* vol. 7, no. 3, pp. 34-38, Sep. 2019.

[18] Z. Wan, et al., “Variational autoencoder based synthetic data generation for imbalanced learning,” *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 10, no. 11, pp. 1-7, Dec. 2017.

[19] M. Lakshminarayanan, et al., “Generating high-fidelity synthetic battery parameter data: solving sparse dataset challenges,” in *Proc. IJSER*, pp. 41-47, Aug. 2021.

[20] C. Zhang, et al., “Generative adversarial network for synthetic time series data generation in smart grids,” in *Proc. ICC*, pp. 1-6, Pittsburgh, Pennsylvania, Aug. 2018.

미 라 클 (Miracle Udurume)
2021년 3월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어
[ORCDI:0000-0003-4401-3316]

치 고 지 (Chigozie Uzochukwu Udeogu)
2021년 3월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어
[ORCDI:0000-0002-3199-9107]

안 젤 라 (Angela C. Caliwag)
2019년 10월~현재 : 금오공대, 항공기계융합전공
<관심분야> 임베디드 시스템, 지능형 제어
[ORCID:0000-0002-0279-935X]

임 완 수 (Wansu Lim)
2014년 9월~현재 : 금오공대 전자공학부 부교수
<관심분야> 임베디드 시스템, 지능형 제어
[ORCID:0000-0003-2533-3496]