International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

Research And Implementation of Data Augmentation Method for Lithium Battery Based on Reinforcement Learning

Feng Cao[1]
[1]Binary Graduate School, Ioi Business Park, NO. 1, 47100 Puchong, Selangor
Corresponding Email: fengcaoedu@163.com

Abstract: *With the rapid development of intelligent power systems and electric vehicle technology, high-performance lithium batteries have become a research hotspot. Monitoring the health status and predicting the remaining life of lithium batteries are crucial for ensuring the safe operation of batteries and the reliability of maintenance systems. However, the high cost of collecting lithium battery data and the relatively limited amount of data pose challenges to data-driven battery management systems (BMS). To address this issue, this study proposes a reinforcement learning-based data augmentation framework for lithium batteries, using Generative Adversarial Networks (GAN) to enhance the quality of lithium battery data. This paper employs Dynamic Time Warping (DTW) as the core algorithm to evaluate the similarity of synthetic data to real data in terms of time series, ensuring that the temporal characteristics of the augmented data are consistent with the original battery data. In this way, we guide the GAN generator to produce highly similar and diverse data, thereby expanding the training set and improving the accuracy of the battery status prediction model.*

This study first comprehensively analyzes the characteristics of data in the field of lithium battery applications, clarifying the importance of data augmentation technology in improving model performance. We designed and implemented a GAN framework based on reinforcement learning, where the discriminator not only evaluates the authenticity of the data but also supervises the temporal characteristics of the generator's output data. In the reinforcement learning environment, the generator continuously optimizes its strategy to pass the DTW evaluation metric, generating more accurate battery usage data. Experimental results show that compared to unenhanced data, the data generated by this framework exhibits higher accuracy and generalization ability in the battery performance prediction model.

Furthermore, this study also explores the potential applications of the proposed framework in other energy storage systems and similar sequential data processing tasks, demonstrating its broad applicability and flexibility. This research not only provides new research ideas for lithium battery data augmentation technology but also offers valuable technical support for the development of data-driven battery management systems.

Keywords: *Generative Adversarial Networks (GAN), Dynamic Time Warping (DTW), Reinforcement Learning, Lithium Battery Applications*

1. INTRODUCTION

1.1 Research Background

As a representative of modern battery technology, lithium batteries have demonstrated its great potential in areas such as smartphones, watches, electric vehicles, and energy storage. It's high energy density and good cycling characteristics make the preferred energy storage solution for many applications. However, despite their great technical and market success, Lithium batteries still face a series of challenges and problems. For instance, with the increasing demand for electronics and electric vehicles, the need for higher energy density and longer cycle life is becoming more and more urgent, and the energy density and cycle life of lithium batteries are gradually becoming a focus of attention. High energy density means that the equipment can have a longer working time, while long cycle life can reduce maintenance costs and extend the service life. This essay will discuss in detail the application of time series methods in lithium battery data enhancement. A variety of time series methods will be explored, such as interpolation techniques, dynamic regularization, noise addition methods, Generative Adversarial Networks (GANs) and Long Short-Term Memory Networks (LSTMs). For each method, the principles, applicable scenarios, and effectiveness in enhancing lithium battery data to reach the warning function will be explored.

Although various monitoring and management technologies have been developed to prevent thermal runaway, these technologies are often limited by insufficient data and the accuracy of the models. The behavior of lithium batteries depends on complex chemical and physical processes, which are influenced by many factors in practical applications, such as ambient temperature, battery aging, and charge-discharge cycle conditions. Therefore, it is particularly important to understand and predict the risk of thermal runaway based on data collected under real operating conditions.

1.2 Research questions

Against this background, this study focuses on the following core questions:

1. How can we effectively enhance the quality and diversity of lithium battery data using a GAN framework based on reinforcement learning, to address the issues of high data acquisition costs and limited data volume?
2. How can we use the Dynamic Time Warping (DTW) algorithm to evaluate the similarity between synthetic and real time-series data, ensuring that the enhanced data maintains the temporal characteristics of the original battery data?
3. In the process of lithium battery thermal runaway data augmentation, how can reinforcement learning be used to optimize the generator's strategy to produce more accurate and diverse battery usage data, thereby improving the accuracy and generalization ability of battery state prediction models?
4. How can we design and implement a lithium battery thermal runaway prediction and alarm system based on reinforcement learning, and verify its effectiveness and reliability in practical applications?This study will be conducted at both the theoretical and empirical levels. Data from Chinese A-share listed companies and questionnaire surveys will be used to construct regression models to reveal the interactive relationship between organisational culture, digital skills and business performance.

1.3 Research contributions

With the rapid development of artificial intelligence technology, reinforcement learning, as an important learning method, has attracted widespread attention in the field of lithium-ion batteries. Among them, the data augmentation method for lithium-ion batteries based on reinforcement learning is an important way to solve the performance optimization and safety stability of lithium-ion batteries. This paper starts from the research background and significance, and conducts an in-depth discussion and analysis of the data augmentation method for lithium-ion batteries based on reinforcement learning, aiming to provide an effective solution for the performance improvement, safety, and stability of lithium-ion batteries. For students, this design is both challenging and practical, which can cultivate students' scientific research ability and the ability to solve practical problems, and also has higher requirements for students' professional knowledge.

Task requirements: 1. Study the basic knowledge of lithium-ion batteries, including their structure, working principles, and performance evaluation, etc., to provide a theoretical basis for subsequent research. 2. Discuss the basic concepts and common algorithms of reinforcement learning, such as Markov decision processes, value functions, and policies, etc., to provide a basic knowledge for subsequent research. 3. Analyze the performance optimization issues of lithium-ion batteries, such as improving energy density and extending battery life, and explore the possibility of improving the performance of lithium-ion batteries through reinforcement learning methods. 4. Study the data augmentation method for lithium-ion batteries based on reinforcement learning, improve the generalization ability of the model by increasing the diversity of training data, and design a data augmentation algorithm suitable for lithium-ion batteries. 5. Conduct experiments and analyze the results to verify the effectiveness and superiority of the proposed data augmentation method for lithium-ion batteries based on reinforcement learning. 6. Discuss the direction and possibilities of future research, such as further optimizing the algorithm, and exploring more data augmentation methods for lithium-ion batteries to improve the performance and service life of lithium-ion batteries.

2. LITERATURE REVIEW

2.1 Reinforcement Learning Algorithm

Reinforcement learning is a type of experiential learning where an agent continuously explores and exploits its environment, making decisions based on feedback in the form of rewards. Developed from theories such as animal learning, stochastic approximation, and optimal control, reinforcement learning is an unsupervised online learning technique. It focuses on learning the mapping from environmental states to actions, enabling the agent to adopt an optimal policy based on maximizing reward values. The agent perceives state information from the environment, searches for effective strategies (i.e., which strategies yield the most efficient learning), and selects optimal actions. These actions lead to state transitions and result in delayed rewards. The evaluation function is then updated based on these rewards. After completing one learning iteration, the agent enters the next round of training, repeating this cycle iteratively until the entire learning condition is satisfied, at which point the learning process terminates. Reinforcement learning obtains optimal strategies through interaction with the environment, which is particularly effective in dealing with the complex charging and discharging processes of lithium batteries. Fu W [1] et al., in response to the problems of insufficient intrusion detection data and slow updates of mainstream detection methods, proposed a data generation method for intrusion detection based on generative adversarial networks. First, the overall data is digitized and normalized to maintain data integrity. Then, the ACGAN model is used to learn the hidden features of the data and generate new data. Finally, the similarity and effectiveness of the generated data are evaluated from multiple perspectives. Experimental results show that the generated data has features similar to the original data and can be used to enhance the original dataset to meet the needs of intrusion detection systems. In addition, Peng J [2] et al. proposed a new data-driven method using feature enhancement and adaptive optimization. First, the features of battery aging are extracted online, and then feature enhancement techniques, including box-box transformation and time window processing, are used to fully exploit the potential of the features. On this basis, a RUL prediction model is established using gradient boosting decision trees, and the model parameters are adaptively optimized using particle swarm optimization technology. This method has been applied to actual lithium-ion battery degradation data.

2.2 Time Series Models

A time series is a sequence of data points arranged in temporal order. Typically, a time series is a sequence obtained at equally spaced (uniform) time intervals, making it a discrete type of data. Time series are widely used in numerous research fields, such as financial forecasting, weather prediction, earthquake prediction, and industrial production.

2.2.1 Characteristics of Uniform Sequences

(1)Stationarity

Uniform sequences may exhibit stable behavior, where the changes of data points over time or space are relatively uniform and stable, without obvious trends or seasonality.

(2)Linear or Approximately Linear Trends

In some cases, uniform sequences may show linear or approximately linear trends, meaning that the data points are distributed along a straight line or an approximate straight line.

(3)Uniform Distribution

For some uniform sequences, data points may be distributed uniformly over time or space, without obvious concentration or clustering tendencies.

2.2.2 Preprocessing of Time Series

(1)Stationarity Test

The stationarity test is a statistical method used to determine whether a time series is stationary. Stationarity refers to the property of a time series where its statistical characteristics do not change over time, maintaining constant mean, variance, and autocorrelation structure. Stationarity is a prerequisite for many time series analysis methods. Figure 1 illustrates a stationary sequence.

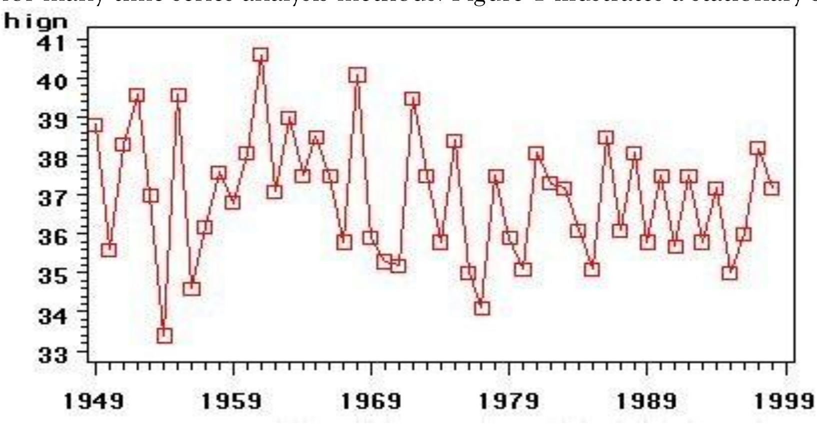


Figure 1

There are two types of stationarity: weak stationarity (Weak Stationarity) and strict stationarity (Strict Stationarity). The main difference between them is that strict stationarity requires the joint distribution of the sequence to be constant over time, while weak stationarity only requires the mean, variance, and autocorrelation structure to be constant over time, without the need for a constant joint distribution[3].

In practical time series analysis, the requirement for weak stationarity is more common and easier to satisfy, and many time series models are based on the assumption of weak stationarity.

(2)Methods for Stationarity Test

Common methods include plotting time series graphs, autocorrelation graphs, and partial autocorrelation graphs to observe the presence of trends, seasonality, etc. Visualization techniques are used to make preliminary judgments about stationarity. In this case, we will use time series graphs for the stationarity test[4].

2.3 **DTW Model**

Dynamic Time Warping (DTW) is an efficient technique specialized in evaluating the similarity of two time series, even if these series do not correspond exactly in time or speed. This method has a wide range of applications in fields such as speech recognition, data mining and bioinformatics. DTW is particularly important when analyzing data related to thermal runaway of lithium batteries, especially in dealing with the similarity of time series and their alignment. DTW model uses dynamic programming to construct a cost matrix, record the distances of all data points between two sequences, and find a path with the lowest cost, which demonstrates the optimal alignment between the two sequences. Qiu Lianpeng [26] proposed a noise robust dynamic time regularization algorithm (NoiseDTW) for the problem that the noise present in the sequences tends to lead to local overstretching and compression
during time series matching. The algorithm first introduces additional noise in the original signal, which suppresses the problem of aligning one point to multiple points that exists in sequence alignment. Then, the effect of the randomness of the noise on the time series similarity measure is reduced by finding an optimal matching path among multiple possible matching paths between two time series.

The noise in sequences can easily lead to excessive stretching and compression in local areas during time series matching, proposed a noise-robust Dynamic Time Warping algorithm (NoiseDTW). The algorithm first introduces additional noise into the original signal to suppress the problem of one point aligning with multiple points in sequence alignment. Then, by finding the optimal matching path among multiple possible matching paths between the two time series, it reduces the impact of noise randomness on the similarity measurement of time series[5].

2.4 GAN model

The training process of Generative Adversarial Networks (GAN) is an iterative process. The generator strives to produce more realistic data to enhance its ability to deceive the discriminator, while the discriminator works to improve its accuracy in distinguishing between real and fake data. This adversarial process continues until the quality of the data generated by the generator is so high that the discriminator can no longer distinguish between real and fake data, thereby achieving the training objective of the model. As shown in Figure 2, the GAN architecture consists of two main components: the generator G and the discriminator D.

The generator G receives random noise from a known probability distribution and generates a sample that is similar in distribution to the training data. Through adjustments during the training process, the distribution of the generated samples is made to approximate the distribution of the real data. The discriminator is used to distinguish between generated and real data and outputs a value between 0 and 1, indicating the probability that the sample is real data. These two networks are trained through alternating optimization, with the generator producing samples similar in distribution to the training data and the discriminator judging the authenticity of the samples and updating the network parameters. This continues until the generated samples can no longer be distinguished by the discriminator, thus achieving the goal of the generator producing high-quality samples. The specific implementation of the GAN is illustrated in Figure 2.

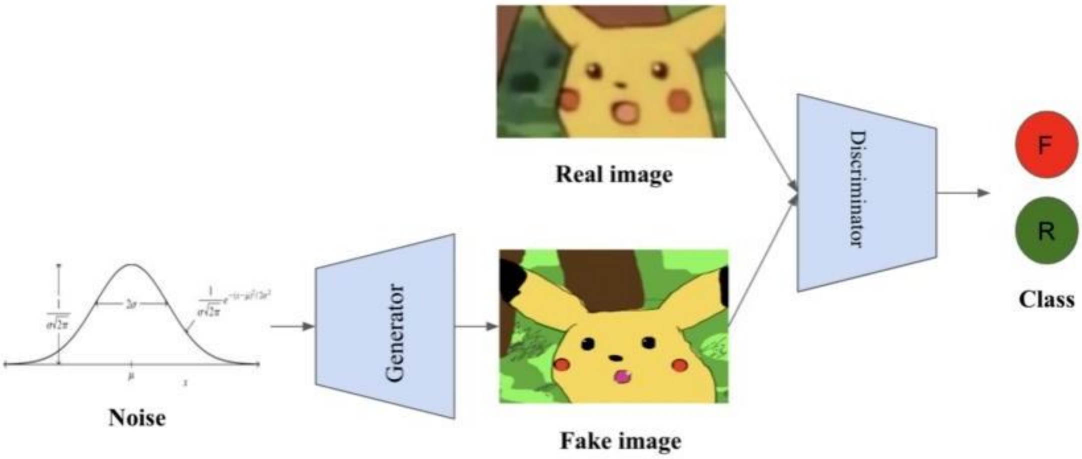


Figure 2

2.5 Research hypothesis

Hypothesis 1：Data enhancement methods based on Generative Adversarial Networks (GANs) can significantly improve the generalization ability of Li-ion battery performance prediction models to more accurately predict the health status and remaining life of batteries.

Hypothesis 2：Dynamic Time Warping (DTW) algorithm can effectively evaluate the similarity between the augmented data and the real data in the time series, ensure that the generated data is highly consistent with the original data in terms of time alignment and feature distribution, and thus enhance the robustness of the model.

2.5 **Summary**

1. The research on data augmentation methods for lithium-ion batteries holds significant theoretical and practical value in the current field of energy storage. By integrating techniques such as deep learning, Generative Adversarial Networks (GAN), and reinforcement learning, researchers are committed to overcoming the challenges faced in acquiring battery test data and enhancing the generalization ability and practical applicability of models.
2. In practical applications, data augmentation methods based on reinforcement learning provide a solid foundation for the development of battery technology. In fields such as electric vehicles and renewable energy storage, the performance and lifespan of batteries directly affect the reliability and economic viability of the entire system.

3. Introduction to Related Technologies

3.1 Normal Noise

Normal noise is the most common and simple method of data augmentation. For the numerical data in this paper, cutting and rotation are not suitable for data augmentation of this type of data. Gaussian noise (or normal noise) is a common type of random noise that is universally present in natural and artificial systems. Its implementation principle is based on the statistical characteristics of the normal distribution. The normal distribution is a natural expression of real-world data because it simulates the sum of many independent random variables, and according to the Central Limit Theorem, the sum of these independent variables tends to form a normal distribution.

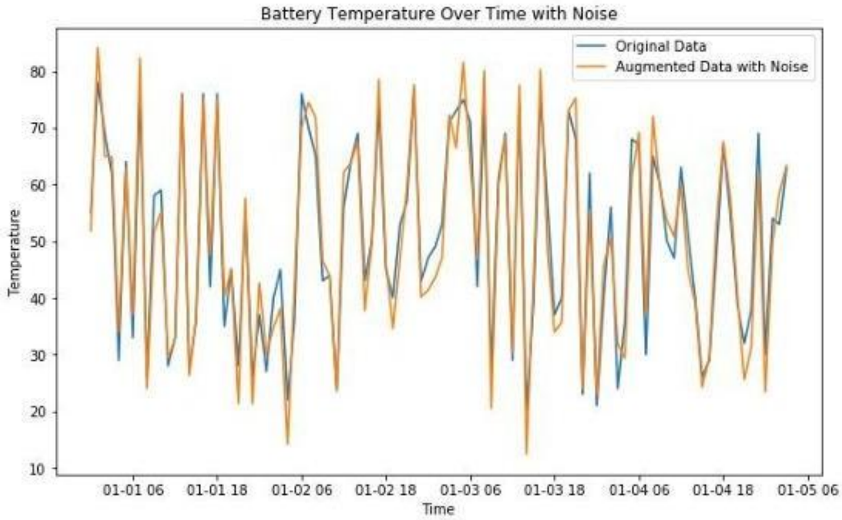
The method of adding noise to the original data is generally used to strengthen data with higher purity. However, through visual analysis of the original data, the temperature change curves and data distribution are relatively smooth. Data augmentation can be performed by adding thermal noise that follows a normal distribution to the thermal runaway data. For example, adding random noise that conforms to the Gaussian (normal) distribution to the original data can model real experimental scene transformations and improve the robustness of the model.

The implementation principle of this algorithm is to satisfy

$X_{noisy} = x + \epsilon$,

where

x is the original data, ϵ is the noise that follows a Gaussian distribution $N(0,\sigma^2)$, and σ is the standard deviation of the noise, which can be adjusted as needed.



As shown in Figure 3 , an example of augmenting battery temperature data using normal noise (Gaussian noise) is presented, illustrating the trends of the original and augmented data.

xnoisy x ò

Figure 3

3.1.1 Oxford_Battery_Degradation_Dataset_1 Dataset

This dataset, created by researchers at the University of Oxford and released in 2015, contains measurement data on battery degradation. It includes data from 8 lithium-ion batteries undergoing repeated charge-discharge cycles. The cycling protocol involves simulating the usage scenario of power batteries in urban driving conditions at a constant ambient temperature of 40 °C using the ARTEMIS urban driving schedule. The batteries are subjected to repeated 2C constant-current discharge and recharging. After every 100 cycles, the batteries are charged and discharged at a 1C rate to test their characteristics, including voltage, current, temperature, and cycle count. This process is repeated until the end of the battery's life. Throughout the cycling process, the battery's voltage, current, and surface temperature are recorded at 1-second intervals. The battery parameters used are shown in Table 1:

Table 1 The parameters of datasets

Parameter Name	Value
Nominal Capacity	0.74Ah
Charging Cut-off Voltage	4.2V
Cut-off Capacity	0.43Ah
Negative Electrode Material	Graphite
Positive Electrode Material	N/A

The dataset primarily records the battery's voltage, heater usage, and temperature data at various test points during the testing process. The data is timestamped, allowing for the tracking of changes in the

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

battery's condition over time. Specific columns include voltage, heater power (U_heater), temperature at the middle heating point (T1_middle heatpoint - T), temperature at the top position terminal (T2_top pos. term. - T), temperature below the middle heating point (T3_middle under heatpoint - T), and ambient temperature (T4_ambient - T). These data reflect the performance metrics of the battery under standard operating conditions

As shown in Figure 4, this dataset provides a detailed monitoring of lithium batteries. By recording voltage and multiple temperature measurement points, it can be used to analyze the behavior and performance of batteries under various operating conditions. Such data is crucial for optimizing battery design, improving battery management systems, and enhancing battery safety.

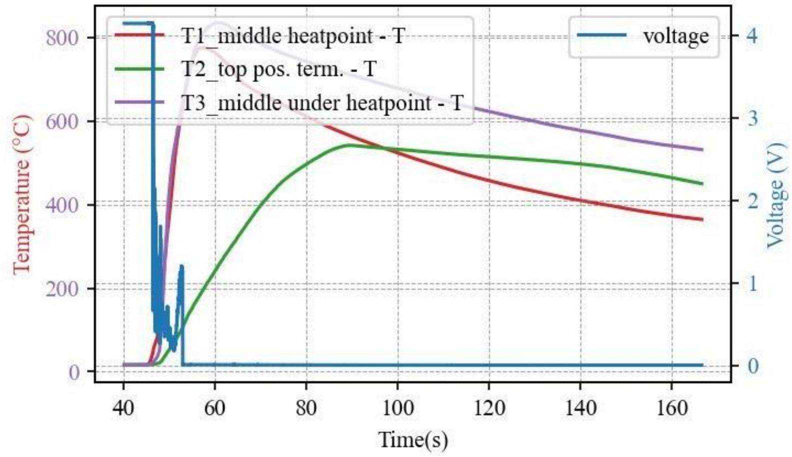


Figure 4

3.1.2 Data Preprocessing

When reading the data, it is necessary to handle missing values. Therefore, the fillna function is used to fill in the gaps. When running the two test sets, data1 and data2, the data needs to be normalized. Data normalization is a common step in data preprocessing, aimed at adjusting the range of numerical data to a fixed interval, such as [0,1]. The advantage of doing this is that it reduces the impact of scale differences among features in the data, making the algorithm more stable and faster during training. It also helps to improve the convergence speed and accuracy of the model. The initial data is normalized using the following formula, which helps the model train better and avoids interference from the robustness of the data during model training.

$$X_{norm} = \frac{X - X_{min}}{X_{max} - X_{min}}$$

In the formula,

X_{norm} represents the normalized data,

X_{min} is the minimum value in the data,

X_{max} is the maximum value in the data, and

X is the original data.

Since this paper is based on lithium battery data for experiments, the data related to thermal runaway (TR) is labeled accordingly. In the original dataset, thermal runaway is categorized into three levels: no thermal runaway, mild thermal runaway, and severe thermal runaway. To facilitate the subsequent addition and improvement of the alarm system, it is necessary to standardize these classifications. The following code snippet is used to unify the labels in the training set:

```
# Define the mapping dictionary
label_mapping = {'Severe TR': 2, 'Mild TR': 1, '0': 0}
# Use the map function to convert text labels to numerical labels df['Severe TR'] = df['Severe TR'].map(label_mapping)
```

After the initial preprocessing of the training set, the time-series data is smoothed using convolution.

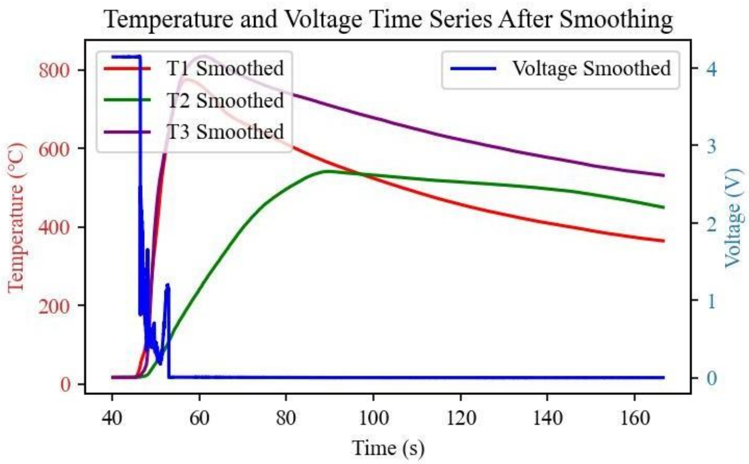


Figure 5 shows the result of smoothing the time-series data from Figure 4 using convolution.
Figure 5

3.2 Data Augmentation Model for Time Series

The objective of this study is to develop a deep learning-based time-series data generation model, utilizing the architecture of Generative Adversarial Networks (GAN) in conjunction with Gated Recurrent Units (GRU), a variant of Recurrent Neural Networks (RNN).

In terms of model architecture, the study constructs four main components: the generator, discriminator, embedder, and recovery model. Both the generator and discriminator employ GRU layers, which are particularly suited for processing and generating sequential data. The generator aims to produce realistic time-series data, with its output layer using the sigmoid activation function to ensure that the output values range between 0 and 1, matching the normalized data range. The discriminator's task is to distinguish whether the input data is real or generated by the generator, thereby guiding the generator to produce more realistic data. The embedder
extracts key features from the time-series data, while the recovery model attempts to reconstruct the original data from these features.

During training, each model component is compiled using the Adam optimizer, with the update formula as follows:

Adam :
t 1t
v_t

$\sqrt{\quad}$

where represents the model parameters, is the learning rate, m_t and v_t are

the first and second moment estimates respectively, and is a small constant added
for numerical stability. This optimizer is suitable for handling large-scale data and parameters as it can adaptively adjust the learning rate for each parameter.

In the training of the generator and discriminator, the standard GAN training strategy is adopted. The generator attempts to deceive the discriminator, making it unable to distinguish between real and generated data, while the discriminator strives to make accurate judgments. Finally, the generated time-series data is inverse-normalized back to the original numerical range and saved in DataFrame format for subsequent analysis. This study not only demonstrates the application potential of deep learning in time-series data generation but also provides an effective tool for further data augmentation and simulation.

3.2.1 Basic Loss Functions

The discriminator's objective is to accurately distinguish between real and generated data. Therefore, its loss function consists of two parts: one for processing real data samples and the other for processing generated fake data samples. When the discriminator processes real data, its goal is to identify these samples as real (with a label of 1), and when processing generated data, it should identify them as fake

International Journal of Environmental Sciences
ISSN: 2229-7359
Vol. 11 No. 6, 2025
https://theaspd.com/index.php

(with a label of 0). Thus, the discriminator's loss function can be expressed as:

$$D_{loss} = -\frac{1}{2}\left(E_{x \sim p_{data}}[\log(D(x))] + E_{z \sim p_z(z)}[\log(1 - D(G(z)))]\right)$$

Here, D(x) represents the discriminator's output for a real data sample x, G(z) represents the fake data sample generated by the generator from noise z, and D(G(z)) is the discriminator's output for these fake samples. E denotes the expectation operation, taken over the real and generated data respectively.

The generator aims to produce data that is as realistic as possible, in order to deceive the discriminator into mistaking it for real data. Therefore, the generator's loss function is based on the discriminator's evaluation of the generated data, with the goal of making these evaluations as close to 1 as possible. The generator's loss function can be defined as:

$$G_{loss} = -E_{z \sim p_z(z)}[\log(D(G(z)))]$$

Here, the generator G attempts to maximize the probability that the discriminator predicts the generated data as real, D(G(z)). Thus, the generator's task is to maximize the likelihood of the discriminator misclassifying fake data as real.

This design of the loss functions creates an "adversarial" relationship between the discriminator and the generator during training. The discriminator tries to distinguish between real and fake data, while the generator tries to deceive the discriminator and prevent it from making the correct judgment. This mechanism is the core of GAN, helping the model gradually improve the quality of the generated data during training.

3.2.2 Logarithmic Loss Function

loss='binary_crossentropy' is a commonly used loss function in machine learning for binary classification tasks, also known as log loss. This loss function measures the difference between the predicted probabilities output by the model and the actual labels, guiding the model's parameter updates during training to make its predictions more accurate. For binary classification problems, where the labels y can only take the values 0 or 1, and the predicted probability is p (usually the model's output after being transformed by an activation function like sigmoid), the formula for binary cross-entropy is:

$$L(y, p) = -(y\log(p) + (1 - y)\log(1 - p))$$

The calculation can be divided into two parts based on the value of the actual label y.
If the actual label y is 1, the loss is $-\log(p)$, meaning that if the predicted probability P is close to 0, the loss will be very large. If the actual label y is 0, the loss is $-\log(1-p)$ meaning that if the predicted probability P is close to 1, the loss will

also be very large.

3.2.3 Data Distribution Before and After Augmentation

4. Figures 6 to 9 visualize the results of data augmentation using noise addition, LSTM, and GAN, respectively. To select the most suitable method and train the model, the distributions of the data before and after augmentation are analyzed to determine the best approach for augmenting the lithium battery dataset.

5. Figure 6 illustrates the distribution of voltage (Vo), where the blue histogram of the original data is close to normal distribution and is mainly concentrated around 4.2, while the enhanced data shows a red histogram with a wider distribution, especially in the middle region where the frequency is lower. The difference in distribution between the two suggests that the data enhancement process may have altered the concentration trend and dispersion of the data. Additionally this experiment also depicts the distribution of Max Temp, where the blue histogram of the raw data shows two peaks, around 200 and 1000, displaying a bimodal distribution characteristic; in contrast, the red histogram of the enhanced data has a distinct peak in the range of 600 to 800 and a reduced frequency in the lower and higher temperature intervals, which reflects the fact that the data enhancement may have introduce new properties or changes in distribution.

6. As shown in Fig. 7, the distributions of Vo (voltage) features and Max Temp features are plotted with data enhancement by DTW (Dynamic Time Warping) method. The raw data is shown as a purple histogram and follows the blue kernel density estimation curve, which is mainly concentrated near the peak at 4.2, showing a clear single-peak normal distribution characteristic. In the temperature distribution plot, the distribution of the raw data is represented by a blue histogram and curve, showing the characteristics of a bimodal distribution, with the main peaks occurring near the temperature points of 200 and 1000. While the enhanced data is represented in red color with its relatively wide distribution, especially in the middle temperature interval with lower frequency, revealing the influence of the data enhancement process on the shape of the distribution and the location of the peaks.

7. As shown in Fig. 8, in the distribution of the raw voltage (Vo) data and the maximum temperature (Max Temp), indicated by the blue histogram and the corresponding kernel density estimation, the data are mainly concentrated in the high-frequency region near the peak of 4.2 V, which may represent the standard voltage value of the battery when it is fully charged. The GAN-enhanced voltage data, on the other hand, represented by the red histogram, shows a much wider distribution, especially in the region of intermediate voltage values with significantly lower frequencies. The raw temperature data shows a bimodal distribution of blue histograms with high frequency peaks located in the interval around 200 to 1000 degrees, which may reflect the two extreme temperature states of battery operation under specific conditions. The augmented data, represented as a red histogram, shows a distinct peak interval between 600 and 800 degrees with reduced frequencies at lower and higher temperatures, suggesting that the augmented data may have increased the simulation of scenarios at moderate temperatures while decreasing the occurrence of extreme temperature values. This change in distribution may help the model learn the behavior of the battery over this temperature range, leading to more accurate predictions for different operating temperature conditions.

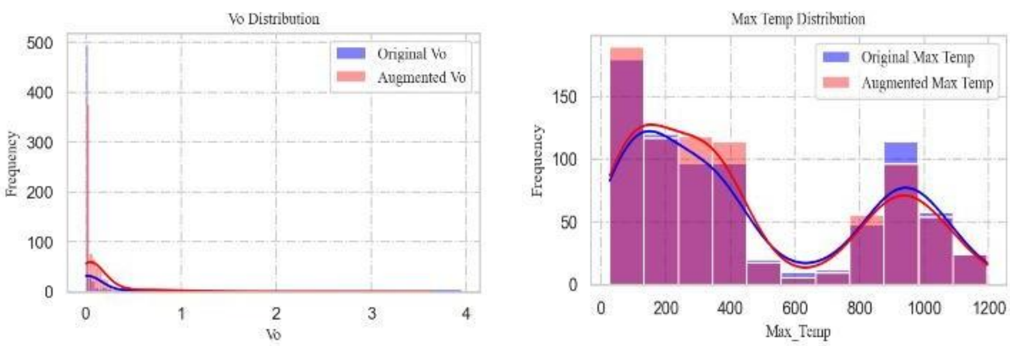
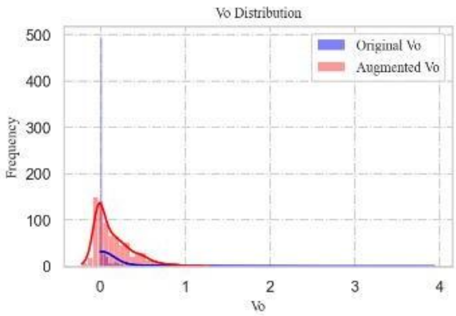


Figure 6



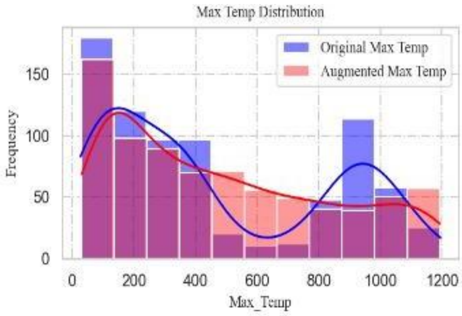
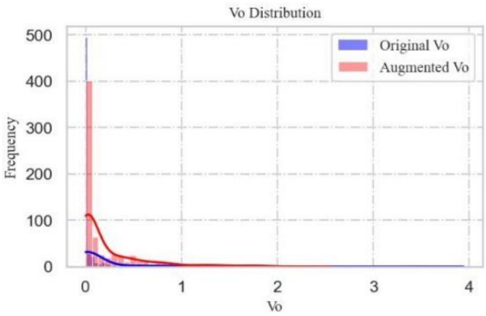
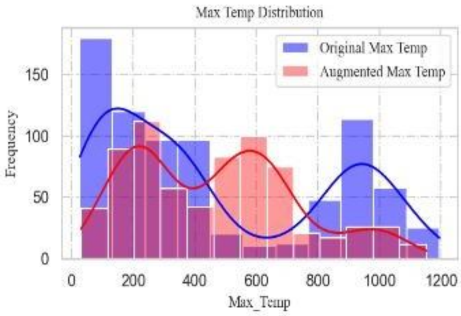


Figure 7

Figure 8





4 Empirical Analysis

This chapter will delve into the critical training parameters that have a decisive impact on the performance of the lithium battery thermal runaway prediction model. Two distinct parameter training methods will be thoroughly introduced: parameter training based on Dynamic Time Warping (DTW) and parameter training based on Generative Adversarial Networks (GAN). These two methods optimize model parameters from different perspectives, providing targeted training strategies for the characteristics of time-series data and non-time-series data respectively.

4.1 Experimental Environment

1. TensorFlow (Version Requirement >=1.15.0): TensorFlow is an open-source software library for numerical computation, particularly suited for large-scale machine learning. Its core feature is the use of data flow graphs to organize computations, making the process efficient and scalable. TensorFlow supports multiple platforms and can run efficiently on both CPUs and GPUs. In the field of deep learning, TensorFlow provides robust support, including a variety of pre-built deep learning models and training techniques.
2. Scikit-learn (Version Requirement >=0.21.3): Scikit-learn is an open-source machine learning library for Python. It supports a variety of machine learning algorithms, including classification, regression, clustering, and dimensionality reduction techniques. Built on Numpy and Scipy, Scikit-learn provides a set of simple and effective tools for data mining and analysis.

4.2 Training Parameters

4.2.1 Parameters Based on LSTM and Noise Addition Models

In the LSTM augmentation model, the look_back parameter, which determines the size of the window for time-series prediction, is set to 3. This means that each prediction is based on the data from the previous three time points. A larger look_back value provides more historical information but may also lead to the model handling more complex relationships, requiring more computational resources. In the LSTM layer, the number of units (50 and 30) affects the model's complexity and its ability to capture long-term dependencies in the data. Generally, the more units there are, the more complex patterns the model can learn, but this also increases training time and may lead to overfitting if too many units are used. Finally, the learning rate of the Adam optimizer is set to 0.01, a commonly used value that balances speed and convergence stability. A higher learning rate may cause instability during training, while a lower one may slow down the training process.

In the noise addition model, random fluctuations, spike disturbances, and time-dependent noise are introduced to simulate the instability that might be encountered in real-world applications. Random fluctuations, generated based on a normal distribution, have their amplitude controlled by the scale parameter. In this code, a smaller standard deviation is used for temperature data, while a different standard deviation setting is applied to voltage data to reflect the noise levels of different types of data. Spikes are simulated by inserting large positive or negative values (+5 or -5) at random time points to represent sudden peak disturbances, which may indicate anomalies in the data collection process or equipment malfunctions. Finally, time-dependent noise, generated based on a sine wave function with a fixed amplitude of 0.1, is used to simulate periodic changes, with the period adjusted by modifying the parameters of the sine function.

4.2.2 Parameters and Training Based on GAN

In the application of Generative Adversarial Networks (GAN), this paper first focuses on the network architectures of the generator and discriminator, ensuring that they can effectively collaborate to generate high-quality data. The generator is typically constructed as a deconvolutional neural network, starting from a densely connected layer and gradually expanding into the desired data shape and size through multiple layers of transposed convolutions. This architecture allows the generator to form complex data structures, such as images or sequence data, starting from a simple noise vector. Meanwhile, the discriminator adopts the form of a traditional convolutional neural network, tasked with determining the authenticity of the input data. Starting from the raw form of the data, the discriminator extracts features through multiple convolutional layers and ultimately outputs a judgment result through one or more fully connected layers.

During model training, key tuning parameters include the learning rate, batch size, and number of iterations. Typically, the learning rate is set between 0.0001 and 0.0002. Depending on the specific needs of the experiment, the generator and discriminator may use different learning rates to ensure a balanced training process. In addition, the batch size has a significant impact on the stability and efficiency of model training and is typically chosen between 32 and 128, depending on the availability of computational resources and the specific needs of the task.

The number of iterations is then adjusted based on the complexity of the data and the response of the model. Thousands or even tens of thousands of iterations may be required to achieve stable generation results. The choice of optimizer is also crucial. The Adam optimizer is widely used due to its ability to adaptively adjust the learning rate. Optimizer parameters, such as beta1, are often set to 0.5 or 0.9 to regulate momentum effects during optimization.

4.3 Comparison of Prediction Results and Data Characteristics

When comparing these four methods, this experiment considered their performance in terms of prediction accuracy, data stability, and the authenticity of the augmented data. Although noise addition is the most straightforward method, it may not be as effective as the others. DTW and LSTM augmentation demonstrated their advantages in handling time-dependent data, while GAN augmentation provided high-quality and diverse data, which may be most beneficial for improving the model's generalization ability. Therefore, the choice of augmentation method depends on the specific application scenario and the particular needs of the prediction task. Table 2 presents the evaluation of these models.

Table 2 Evaluation of Models Used in This Experiment

According to Table 2, GAN-generated data has the smallest standard deviation and mean absolute deviation, indicating good adaptability. Moreover, it maintains a skewness of 1.3389, which is optimal among these models for fitting real data conditions. In contrast, noise addition, while enhancing data through noise, cannot truly capture the real trends in data changes. Although numerical data can currently be transformed into image data for augmentation using Fourier transforms, noise addition alone consumes a significant amount of memory and is less effective than GAN. GAN outperforms the other three models in augmenting lithium battery time-series data. Given the difficulty of extracting features from numerical data, this experiment used GAN for data augmentation to more precisely capture data features and trends.

4.4 Further discussion

(1)DTW enhancement is specifically designed for time-series data. It enhances the data by measuring and adjusting the similarity between time series, making it suitable for data with temporal dependencies.

(2)LSTM enhancement leverages the capability of LSTM networks to generate time-series data, which helps capture long-term dependencies. The stability of LSTM-enhanced data shows improvement compared to noise-added enhancement, though its effect may still be less pronounced than that of DTW and GAN enhancement.

(1) GAN enhancement produces high-quality data through adversarial training. The generated data closely resembles real data in features while incorporating a degree of variation, thereby increasing data diversity.

4.1 Conclusion

This section delves into the key training parameters that critically influence the performance of the lithium battery thermal runaway prediction model and provides a detailed introduction to two distinct parameter training methods based on Dynamic Time Warping (DTW) and Generative Adversarial Networks (GAN). These two approaches optimize model parameters from different perspectives — the former focuses on the characteristics of time-series data, while the latter emphasizes the features of non-time-series data.

Augmentation Method	Standard Deviation	Mean Absolute Deviation	Skewness	Kurtosis
DTW	1.7037	1.4737	0.9591	-0.08981
LSTM	1.6401	1.3255	1.3610	0.0023
GAN	1.6189	1.3088	1.3389	-0.0298

5. CONCLUSION AND MODEL ANALYSIS

5.1 Research conclusions

To evaluate the performance of the classification model, recall is commonly used as a key metric to assess classification results. Therefore, a recall report, as shown in Table 3, was calculated.

Table 3 recall rate report

	Precision	recall	F1-score	Support
No TR	0.837	0.881	0.875	18032
Mild TR	0.612	0.891	0.759	62842
Severe TR	0.832	0.765	0.867	169116
Macro Average	0.871	0.922	0.875	

Weighted Average	0.903	0.841	0.849

Table 3 provides a performance evaluation of the classification model, including three categories: No TR (No Thermal Runaway), Mild TR (Mild Thermal Runaway), and Severe TR (Severe Thermal Runaway), with their respective Precision, Recall, F1-score, and Support values. Additionally, the report includes evaluation metrics for Macro Average and Weighted Average.

Based on this report, the classification model performs well across all categories, particularly for Severe TR, which achieves a perfect recall rate. Overall, the model exhibits high recall, especially for Mild TR and Severe TR, indicating its strong effectiveness in detecting and identifying these categories while minimizing missed detections.

The weighted average recall is slightly lower than the macro average recall, likely due to the reduced influence of categories with lower support (e.g., Mild TR) in the weighted average calculation.

5.2 CONCLUSION

The lithium battery data enhancement framework established in this study integrates data augmentation methods such as Generative Adversarial Networks (GAN) and Dynamic Time Warping (DTW), improving the accuracy and generalization capability of lithium battery performance prediction models. Through comparative experiments, the proposed data enhancement method effectively enhances prediction results and enables early warning of thermal runaway in lithium batteries, demonstrating significant improvements in key performance indicators compared to non-enhanced data. Furthermore, by conducting an in-depth analysis of lithium battery data characteristics, this research not only enhances the accuracy of performance prediction models but also broadens the understanding of battery behavior dynamics, which holds substantial practical value for battery health monitoring and lifespan prediction.

Future research can explore several directions for further advancement. First, continuous optimization of models and algorithms remains essential. Novel GAN architectures and more advanced training strategies may further improve the quality of generated data and the efficiency of model training. For instance, exploring conditional Generative Adversarial Networks (cGAN) could provide more precise control variables, enabling the generation of more targeted augmented data. Second, applying the proposed method to broader scenarios, such as research on fuel cells and supercapacitors, could validate its universality and effectiveness across different energy storage technologies. Additionally, the transition from theory to practice is crucial—translating these research findings into practical products or solutions is a vital step in advancing battery management system (BMS) technology.

In summary, this study not only provides valuable academic insights but also offers practical technical support for the application and management of lithium batteries and other battery types.

REFERENCES

[1] He X,Deng K, Wang X, Li Y, ZhangY and Wang M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In SIGIR, 639–648.

[2] Ma, J, Zhou C, Yang H, Cui P, Wang X and Zhu W. 2020. Disentangled self-supervision in sequential recommenders. In KDD, 483–491.

[3] Fu W, Qian L, Zhu X. GAN-based intrusion detection data enhancement[C]//2021 33rd Chinese Control and Decision Conference (CCDC). IEEE, 2021: 2739-2744.

[4] Peng J, Zheng Z, Zhang X, et al. A data-driven method with feature enhancement and adaptive optimization for lithium-ion battery remaining useful life prediction[J]. Energies, 2020, 13(3): 752.

[5] McAuley J, Pandey R, and Leskovec J. 2015. Inferring networks of substitutable and complementary products. In KDD, 785–794.

[6] QiuR, Huang Z, Yin H, and Wang Z. 2022. Contrastive learning for representation degeneration problem in sequential recommendation. In WSDM, 813–823.

[7] Rendle S, Freudenthaler C, and Schmidt-Thieme, L. 2010. Factorizing personalized markov chains for next basket recommendation. In WWW, 811–820.

[8] He, R.; and McAuley, J. 2016. Fusing similarity models with markov chains for sparse sequential recommendation. In ICDM, 191–200. IEEE.

[9] He, X.; Deng, K.; Wang, X.; Li, Y.; Zhang, Y.; and Wang, M. 2020. Lightgcn: Simplifying and powering graph convolution network for recommendation. In SIGIR, 639–648.
[10] Qu, S.; Yuan, F.; Guo, G.; Zhang, L.; and Wei, W. 2022. CmnRec: Sequential Recommendations with Chunk accelerated Memory Network. IEEE Transactions on Knowledge and Data Engineering.
[11] Rendle, S.; Freudenthaler, C.; Gantner, Z.; and Schmidt Thieme, L. 2012. BPR: Bayesian personalized ranking from implicit feedback. arXiv preprint arXiv:1205.2618.

Funding Information
This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

Author Contributions
The manuscript has a single author who was solely responsible for the research design, data collection and analysis, and the drafting and revision of the manuscript.

Conflict of Interest Statement
The author declares no conflicts of interest.

Ethical Statement
This study adhered to all relevant ethical standards for academic research. Where applicable, any research involving humans or animals was conducted in accordance with ethical guidelines.