International Journal of Internet, Broadcasting and Communication Vol.16 No.3 236- 241 (2024)
http://dx.doi.org/10.7236/IJIBC.2024.16.3.236

IJIBC 24-3-25

A Study on the Lifetime Prediction of Lithium-Ion Batteries Based on the Long Short-Term Memory Model of Recurrent Neural Networks

Sang-Bum Kim[1]

[1]*Professar, Department of Robotdrone Engineering, Honam University, Korea*
2021115@honam.ac.kr[1]

Abstract

Due to the recent emphasis on carbon neutrality and environmental regulations, the global electric vehicle (EV) market is experiencing rapid growth. This surge has raised concerns about the recycling and disposal methods for EV batteries. Unlike traditional internal combustion engine vehicles, EVs require unique and safe methods for the recovery and disposal of their batteries. In this process, predicting the lifespan of the battery is essential. Impedance and State of Charge (SOC) analysis are commonly used methods for this purpose. However, predicting the lifespan of batteries with complex chemical characteristics through electrical measurements presents significant challenges. To enhance the accuracy and precision of existing measurement methods, this paper proposes using a Long Short-Term Memory (LSTM) model, a type of deep learning-based recurrent neural network, to diagnose battery performance. The goal is to achieve safe classification through this model. The designed structure was evaluated, yielding results with a Mean Absolute Error (MAE) of 0.8451, a Root Mean Square Error (RMSE) of 1.3448, and an accuracy of 0.984, demonstrating excellent performance.

Keywords: *LSTM, Waste Battery, Electric vehicles, Impedance, Calibration, Reusable Battery*

1. Introduction

With the expansion of the markets for Electric Vehicles (EV) and Energy Storage Systems (ESS) both domestically and internationally, driven by government policies, the demand for batteries is increasing significantly. Consequently, there is a growing expectation of a surge in used lithium-ion batteries with diminished performance over time. Especially as electric vehicles rapidly proliferate due to reasons such as carbon neutrality and environmental improvement, the frequency of accidents, malfunctions, and end-of-life vehicle occurrences is expected to rise post-operation. This necessitates efficient management of used batteries from both safety and environmental perspectives [1]. The importance of addressing issues related to the disposal and recycling of these batteries, as well as the collection and storage of discarded batteries, is becoming increasingly critical. Therefore, different methods for predicting the lifespan of electric vehicle batteries from conventional approaches need to be proposed and applied. This paper aims to predict the lifespan of CS2 lithium-ion batteries using the Long Short-Term Memory (LSTM) model, one of the deep learning-

Manuscript Received: June. 19, 2024 / Revised: June. 25, 2024 / Accepted: June. 30, 2024
Corresponding Author: (Sang-Bum Kim) 2021115@honam.ac.kr
Tel:+82-62-940-3664, Fax: +82-62-940-5285
Professor, Department of Electronic Engineering, Honam University, Gwangju, Korea

based models, and to assess and evaluate its accuracy [2].

2. Design of the LSTM Model Structure

The Long Short-Term Memory (LSTM) model is a variant of the Recurrent Neural Network (RNN) designed to process sequence data (e.g., text, audio, or video) and address the long-term dependency problem. LSTM is more effective than traditional RNN models in handling longer sequence data and can be implemented using various libraries such as TensorFlow, Keras, and PyTorch in Python. It overcomes the limitations of conventional RNN models. While RNN models are primarily used for sequential data, they can encounter issues when the length of the input and output data is inconsistent or when the time intervals between input data points are irregular. The LSTM model is employed to resolve these issues. Consequently, the LSTM model has a more complex structure compared to traditional RNN models and includes mechanisms that allow it to retain previous information regardless of the length or time intervals of the input data, enabling more accurate predictions of sequence data. Figure 1 illustrates the training process for estimating the State of Health (SOH) of lithium-ion batteries using the LSTM model. In this study, the LSTM model is defined using the Keras library [3].

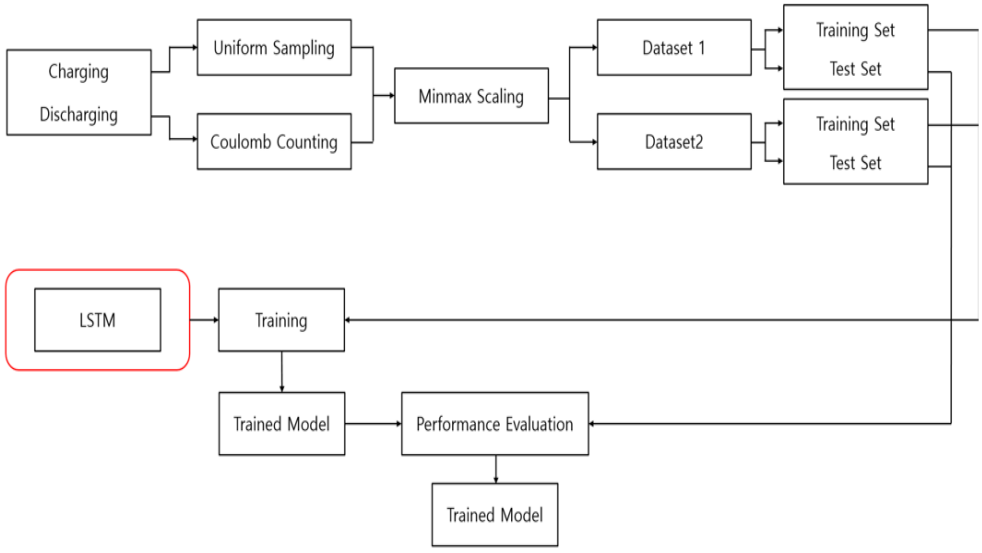

Figure 1. Design of the LSTM Model Structure for Estimating SOH of CS2 Batteries

In this paper, the LSTM model is employed to utilize input data from each cycle to predict the subsequent state based on the output of the previous state. The given battery data is based on charge-discharge cycles and includes various factors that change over time, such as the length and capacity differences of each cycle. The LSTM model can effectively model and predict these dynamic patterns in time-series data, making it suitable for estimating the relative State of Health (SOH) of battery capacity. The proposed LSTM model-based SOH prediction process, as depicted in Figure 1, consists of four main stages: Feature Selection, Data Extraction, Normalization, and LSTM Prediction.

```
def preprocess_data(df):
# convert Date_Time column to datetime type
df['Date_Time'] = pd.to_datetime(df['Date_Time'])
# sort dataframe by Date_Time
df.sort_values(by=['Date_Time'], inplace=True)
# extract feature and label data
features = df.iloc[:, :-3].values
labels = df.iloc[:, -3:-2].values
# scale feature data
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features)
# reshape feature data
features_reshaped = features_scaled.reshape(features_scaled.shape[0], features_scaled.shape[1], 1)
return features_reshaped, labels
```

Figure 2. Pseudocode for preprocessing the DataFrame (df) for the LSTM model

Figure 2 illustrates a function that preprocesses the DataFrame (df) to extract feature and label data for use in time-series prediction. The process of preprocessing the DataFrame (df) to extract feature and label data for time-series prediction is as follows. The Date_Time column is converted to datetime format, and the DataFrame is sorted based on Date_Time. All columns except the last three are extracted as feature data, while the middle column of the last three is extracted as label data. The feature data is then scaled using MinMaxScaler(). The shape of the feature data is transformed to a format suitable for use in the LSTM model. By using this function, one can obtain preprocessed feature and label data, which can be used for training the LSTM model. The LSTM model is created using the Sequential() function and consists of LSTM and Dense layers. The function create_LSTM_model(input_shape) creates an LSTM model with the given input shape. The model comprises two LSTM layers. The first LSTM layer contains 64 units and accepts the input shape with return_sequences=True to return the sequence output. The second LSTM layer consists of 32 units and does not return the sequence output with return_sequences=False. Finally, a Dense layer is added with an output size of 1. The LSTM model is compiled using the Adam optimizer and the Mean Squared Error (MSE) loss function, with Mean Absolute Error (MAE) as the performance evaluation metric. The Adam optimizer and MSE loss function are suitable for solving regression problems, which is why they are employed in the LSTM model. The Adam optimizer combines the RMSProp and Momentum methods, effectively adjusting the learning rate while updating the model's weights. The MSE loss function, commonly used in regression problems, trains the model to minimize the mean squared error between the predicted and actual values. Using the Adam optimizer and MSE loss function in the LSTM model helps minimize the error between predicted and actual values, achieving high predictive accuracy in regression tasks [4].

The MAE metric is employed because the output values from the LSTM model are continuous, making MAE a suitable metric for evaluating the model's performance by calculating the difference between the predicted and actual values. Additionally, MAE is robust to outliers, making it appropriate for this model. In this paper, it is necessary to preprocess the standardized data into a form that can be input into the LSTM model. As illustrated in Figure 3, this process involves adjusting the scale of the input data and reshaping it

into a three-dimensional form to match the input format required by the LSTM model [5-6].

```
#create_LSTM_model(input_shape) :

def create_LSTM_model(input_shape):
model = Sequential()
model.add(LSTM(64, input_shape=input_shape, return_sequences=True))
model.add(LSTM(32, return_sequences=False))
model.add(Dense(1))
model.compile(optimizer='adam', loss='mse', metrics=['mae'])
return model
```

Figure 3. Pseudocode according to the description of the structure of the LSTM model

The code in Figure 4 consists of functions that perform data preprocessing to modify the data structure into a format suitable for the LSTM model. The following steps are undertaken in this process. Through this function, the data is preprocessed into a format suitable for input into the LSTM model. This involves adjusting the scale of the input data and reshaping it into a three-dimensional format to match the input format required by the LSTM model.

```
def preprocess_data(df):
df['Date_Time'] = pd.to_datetime(df['Date_Time'])
# DataFrame 을 Date_Time 으로 정렬
df.sort_values(by=['Date_Time'], inplace=True)
features = df.iloc[:, :-3].values
labels = df.iloc[:, -3:-2].values
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
features_scaled = scaler.fit_transform(features)
features_reshaped = features_scaled.reshape(features_scaled.shape[0], features_scaled.shape[1], 1)
return features_reshaped, labels
```

Figure 4. Pseudocode for changing the structural format of an LSTM model

3. Implementation of the LSTM Model

The proposed LSTM model-based SOH prediction in this study consists of four main stages: Feature Selection, Data Extraction, Normalization, and LSTM Prediction. MinMaxScaler is used to normalize all vector values to a range between 0 and 1. This normalization process is crucial to prevent performance degradation due to significant differences or unit discrepancies among the vector values used as inputs during the model's training process.

Figure 5 shows the code structure for predicting battery capacity (SOH) using the LSTM (Long Short-Term Memory) model. First, the create_LSTM_model() function is used to create the LSTM model, which consists of two LSTM layers and one Dense layer. The model is compiled using the Mean Squared Error (MSE) loss

function and the Mean Absolute Error (MAE) metric. Next, the preprocess_data() function is employed to preprocess the data, which is then divided into X_train, y_train, X_test, and y_test. The model is trained using the model.fit() function. The trained model is used to predict the X_test data, and the predictions are compared with y_test. The prediction results can be visualized as shown in Figure 6 [7-8].

```
def create_LSTM_model(input_shape):
# Define the function create_LSTM_model(input_shape)
# Create the model model = Sequential()
# Add LSTM layer: model.add(LSTM(64, input_shape=input_shape, return_sequences=True))
# Add another LSTM layer: model.add(LSTM(32, return_sequences=False))
# Add Dense layer: model.add(Dense(1)) model.add(Dense(1))
# Compile the model: model.compile(optimizer='adam', loss='mse', metrics=['mae'])
# Return the created model: return model
```

Figure 5. Comparison of modular internal resistance meter and jig resistance

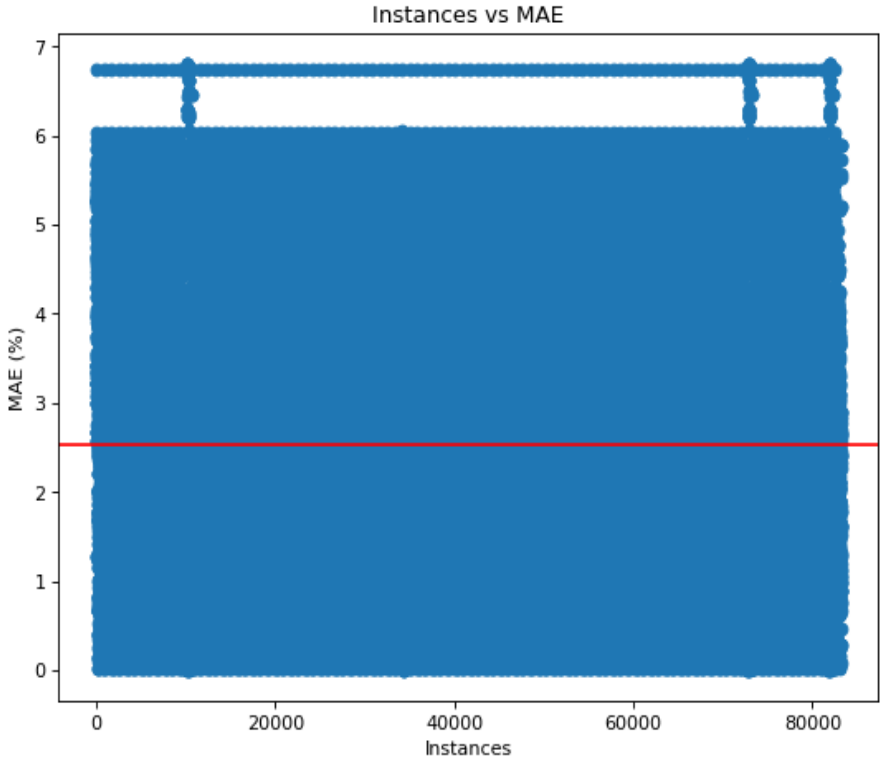

Figure 6. Visualize the relationship between Instances vs MAE

4. Conclusion

The input shape is defined by subtracting 3 from the number of features in the input data using df35.shape[1] - 3. The input shape is defined as a tuple of (number of features in the input data - 3, 1). The preprocess_data() function is called to preprocess the training and testing data. The training data is assigned to X_train and y_train, and the testing data is assigned to X_test and y_test. An LSTM model is then created. The

create_LSTM_model() function is called to create the LSTM model, which is assigned to the model variable. The model.fit() function is called to train the model, using X_train and y_train as inputs. The number of epochs is set to 100, the batch size is 32, and the validation data ratio is 0.2. The training process history information is stored in the history variable. The test data is then predicted. The model.predict() function is called to compute the predictions for the test data, which are assigned to the y_pred variable. The model is then evaluated.

The mean_squared_error() function is called to calculate the MAE and RMSE values for y and y_test. As shown in Table 1, the LSTM model used for predicting usage resulted in an MAE of 2.521 and an RMSE of 1.345, indicating lower error values compared to the CNN model and improved accuracy.

Table 1. LSTM model-based life expectancy result

Type	MAE	RMSE	Accuracy
LSTM	2.521	1.345	0.984

5. Acknowledgement

This result was supported by "Regional Innovation Strategy (RIS)" through the National Research Foundation of Korea (NRF) funded by the Ministry of Education (MOE) (2021RIS-002)

References

[1] T.H. Chen, T.Y. Hsieh, N.C. Yang, J. S. Yang and C. J. Liao, "Evaluation of advantages of an energy storage system using recycled EV batteries," International Journal of an energy storage systems, Vol. 45, Issue. 1, pp. 264-270, February 2013.
DOI: https://doi.org/10.1016/j.ijepes.2012.08.037

[2] S.B. Kim and S.H. Lee, "Design and Development of a Public Waste Battery Diagnostic Device," The International Journal of Advanced Culture Technology(IJACT), Vol. 10, No. 3, pp. 281-286, July 2022.
DOI:https://www.earticle.net/Article/A417941

[3] S.B. Kim and S.H. Lee, "Development of Aging Diagnosis Device Through Real-time Battery Internal Resistance Measurement," *The International Journal of Internet, Broadcasting and Communication(IJIBC)*, Vol. 14, No. 2, pp. 129-135, February 2022.
DOI: https://doi.org/10.7236/IJIBC.2022.14.2.129

[4] M.J. Choi and S.B. Kim, "A Study on Impedance Change Trend and Battery Life Analysis through Battery Performance Deterioration Factors," *The International Journal of Internet, Broadcasting and Communication(IJIBC)*, Vol. 15, No. 3, pp. 141-146, August 2023.
DOI: https://doi.org/10.7236/IJIBC.2023.15.3.141

[5] Y. U. Mun and S. B. Kim, "A study on the prediction of SOH estimation of waste lithium-ion batteries based on SVM model," IJCC 2023.

[6] S. B. Kim, and S. H. Lee, "Research on Hyper-parameter Optimization Method based on Genetic Algorithm of Deep Neural Network," O2O IICCC 2020.

[7] M.J. Choi and S.B. Kim, "A Study on The Life Prediction of Lithium Ion Batteries Based on a Convolutional Neural Network Model," *The International Journal of Internet, Broadcasting and Communication(IJIBC)*, Vol. 15, No. 3, pp. 130-133, August 2023.
DOI: https://doi.org/10.7236/IJIBC.2023.15.3.130