Journal of Energy Storage 64 (2023) 107092

Contents lists available at ScienceDirect

Journal of Energy Storage

journal homepage: www.elsevier.com/locate/est

Research papers

Multi-step time series forecasting on the temperature of lithium-ion batteries

Zijing Wan [a], Yilin Kang [b], Renwei Ou [b], Song Xue [a], Dongwei Xu [a,*], Xiaobing Luo [a]

[a] State Key Laboratory of Coal Combustion, School of Energy and Power Engineering, Huazhong University of Science and Technology, Wuhan, Hubei 430074, China
[b] College of Computer Science, South-Central Minzu University, Wuhan 430074, China

ARTICLE INFO

Keywords:
Convolutional transformer
Multi-step time series forecasting
Battery thermal management

ABSTRACT

Lithium-ion battery temperature monitoring contributes to the higher performance of lithium batteries and reduces the risk of thermal runaway. Since the battery temperature can be approximated as a time series, this work reports a new model named convolutional transformer (Convtrans) for multi-step time series forecasting, which obtains pleasing results. To evaluate our model, we present a cross-sectional comparison of the model with the other three mainstream algorithms in multi-step time series forecasting and a vertical comparison with single-step time forecasting. On one hand, Convtrans has the minimum root mean square error with the highest prediction accuracy and can also predict the trend and shape of temperature curves compared with the other three mainstream algorithms, which means that it has the best results in multi-step time series forecasting. On the other hand, compared with single-step time series forecasting, Convtrans predicts 24 times temperature data while doesn't sacrifice much accuracy even though it costs 6 times running time. Furthermore, in the case of predicting more points, it also maintains good accuracy and can perfectly predict trends of the temperature. In all, we prove the superiority of multi-step time series forecasting over a long period. Therefore, multi-step time series forecasting based on Convtrans can sever as the battery temperature prognostic technology providing timely warnings to assist battery thermal management.

1. Introduction

The world is on its way to the electric. The high energy density, great specific power and long cycle life of lithium-ion batteries (LIBs) have made them the preferred choice of technology for electrical power energy storage [1–3]. With the development of technology, the energy density of LIBs continues to increase, challenges are encountered in monitoring and managing the state of LIBs [4,5]. The battery temperature has a dramatic effect on the state of LIBs, such as State-of-Charge (SoC) and State-of-Health (SoH) [6]. The range of battery temperature is suggested to be 15 °C–35 °C [7]. On the one hand, during the charging and discharging process, a substantial amount of heat is generated inside the LIBs due to the exothermic reaction and internal resistance. If the heat cannot be dissipated in time and accumulates inside the LIBs resulting in a sustained temperature rise, thermal runaway is mostly likely to occur. In this case, the LIBs will rupture or even explode as the temperature and pressure rise [8]. On the other hand, at extremely high and low temperatures, the capacity of LIBs would severely decline, leading to poor SoH [9–11]. Consequently, real-time temperature prognostics providing timely warnings are essential to ensure the proper operation and safety of LIBs.

However, the battery temperature prognostics is technical difficulty [12]. First, during the LIBs operation, their temperature will fluctuate wildly, which may increase the difficulty of forecasting. Furthermore, different cells and loading conditions can cause battery temperature changes, which means that both physical and chemical mechanism changes can cause temperature changes [12]. Lastly, the complexity of the heat production mechanism and the uncertainty of the environmental conditions make the forecast of battery temperature changes more difficult [13].

In the field of forecasting temperature, there have been numerous physical models to simulate the thermal field such as Newman model [14] and Bernardi model [15]. However, the physical models have their limitations in application. Facing with the various battery types, usage environments and loading conditions, the simulation method must calculate each case owing the physical model is not universal, which will result in a great workload. Furthermore, the physical model creation takes into account many factors, and at the same time requires the

* Corresponding author.
E-mail address: dwxu@hust.edu.cn (D. Xu).

https://doi.org/10.1016/j.est.2023.107092
Received 25 October 2022; Received in revised form 24 February 2023; Accepted 6 March 2023

calculation of extensive formulas, which increases the complexity of the forecasting process. In all, real-time forecasting and timely warning based on physical models is difficult.

Nowadays, data-driven methods are gradually applied to meet the needs of real-time forecasting and timely warning. Since the sensors can measure the real-time temperature of LIBs with slightly mean absolute error, data-driven methods based on statistical theory or machine learning algorithms can directly use the measured real-time data to forecast [16,17]. For example, one optimized data-driven algorithm with reasonable input parameters can forecast the temperature in various situations, reflecting their universality and transferability. Furthermore, data-drive methods can adjust output based on real-time data [18]. Hence, we can achieve real-time forecasting and timely warning by directly studying the battery temperature change instead of the internal physical and chemical reactions of the battery. The key to utilizing data-driven methods is how to choose a suitable algorithm based on the characteristics of the data [19]. During battery operation, the average temperature per cycle can be approximated as time series data. Time series data are the values of objects with the same measurement time frequency over a time range [20,21]. Machine learning methods based on battery temperature time series data are silently emerging. For example, an algorithm named Long Short-Term Memory (LSTM) is applied to forecast battery temperature, whose root mean square error (RMSE) is 0.044 [13]. Furthermore, another algorithm named gated recurrent unit (GRU) is found less error in the datasets acquired from the Prognostics Center of Excellence (PCoE) of NASA [6]. However, their algorithms are based on single-step time series forecasting (only can forecast one time step), which is a major limitation in practical applications where multi-step forecasting (can forecast multiple time steps) is needed. Although we can use N single-step models to achieve multi-step forecasting, it is only practicable to predict small steps [22,23]. Furthermore, when forecasting the temperature, although RNN-based frameworks (LSTM and GRU) can effectively forecast time series data, they will degrade performance due to long-term dependency [24–26].

In order to predict more effective multi-step battery temperature time series, we use convolutional transformer (Convtrans), which has been successfully applied to several real-world datasets [27]. To compare, the results of three mainstream algorithms of single-step and multi-step battery temperature time series forecasting, including LSTM, GRU and Temporal Convolutional Network (TCN) [28,29], are also given in SI and this paper, respectively. The training set and testing set in the two kinds of forecasting methods are shown in Fig. 1. In our work, inputs (X_i) are both made of 24 time steps (T_i) in two forecasting methods. And outputs (Y_i) of multi-step time series forecasting algorithms are made of 24 time steps, while that of single-step time series forecasting algorithms are made of 1 time step. In addition, the training sets are set to the same (m = n) to compare the two forecasting methods.

2. Method

The main goal of Convtrans is to achieve multi-step temperature time series forecasting. Hence, we demonstrate our model including denoising auto encoder and convolutional transformer. The detailed architecture is shown in Fig. 2.

2.1. Input and normalization

To reduce the impact of temperature distribution changes on network model and enhance the accuracy and convergence of network model, input data must be normalized [30,31]. Let $X = \{T_0, T_1, T_2, \dots, T_{n-1}\}$ denotes the input temperature time series data with length *n*, which is normalized to [0,1], as follows:

$$X'i = X'_{min} + \frac{T_i - T_{min}}{T_{max} - T_{min}} \times \left(X'_{max} - X'_{min}\right) \tag{1}$$

where X'_i, X'_{min} and X'_{max} is the index i, minimum and maximum of X' tensor (scaled input tensor), respectively.

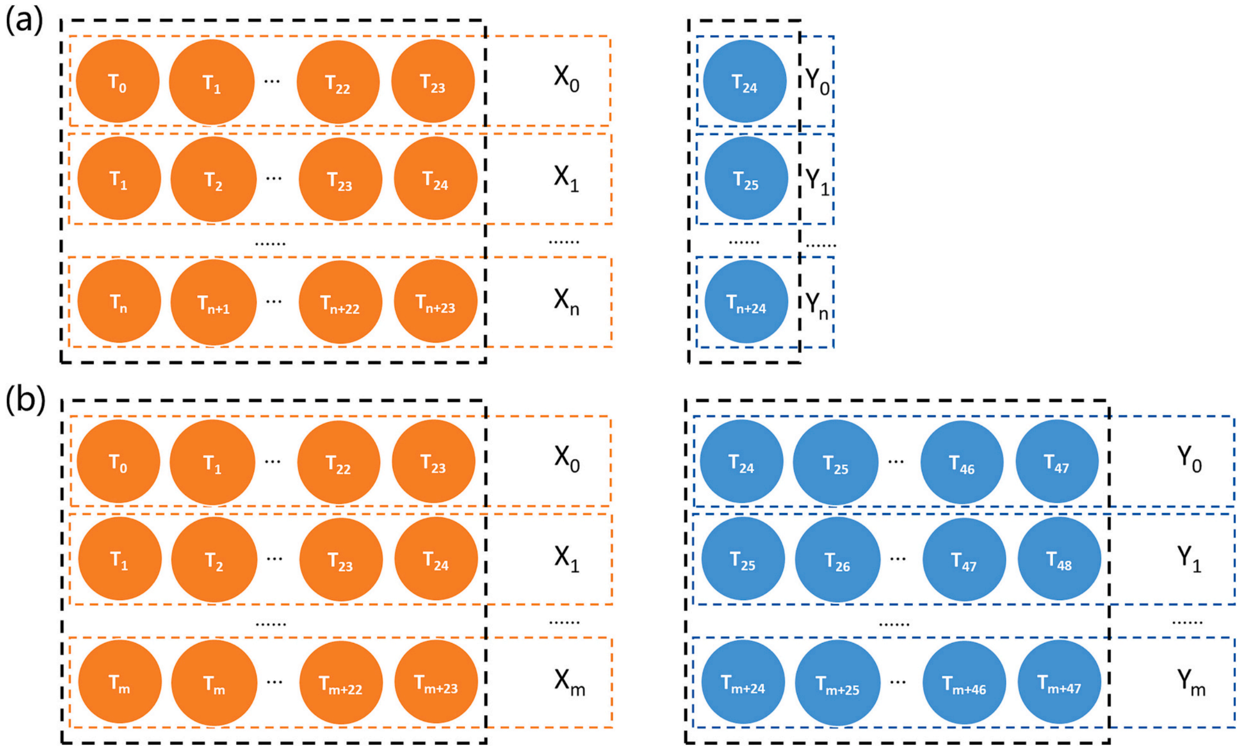

Fig. 1. The training set and testing set in the two kinds of forecasting methods: (a) single-step time series forecasting (b) multi-step time series forecasting, where X_i means the (i + 1st) input of model, Y_i means the corresponding target of mode, and T_i means the average battery temperature of i-th cycle.

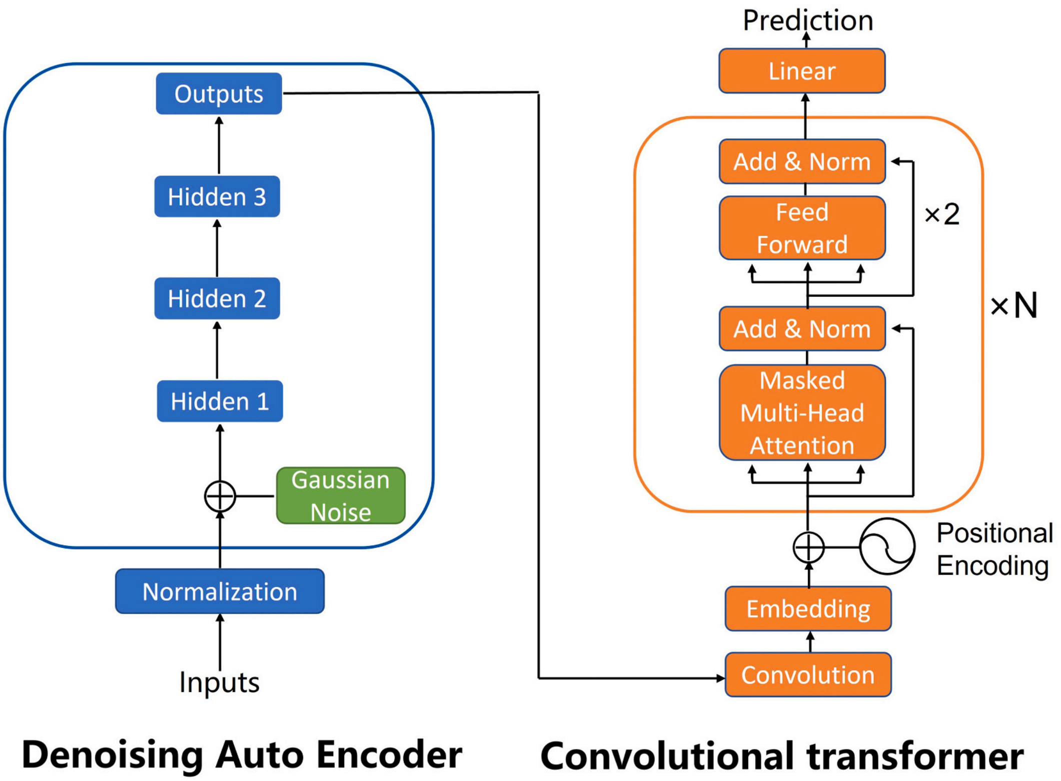

Fig. 2. Convtrans network for multi-step time series forecasting on the temperature.

2.2. Denoising

The raw input is always riddled with noise, especially when the external ambient temperature fluctuates. If the raw data without denoising is directly input to the neural network, it will severely damage the prediction accuracy of the network model. To preserve stability and robustness, we use Denoising Auto Encoder (DAE) [32] to denoise the normalized input data, which is an unsupervised method to learning useful information.

Let $X'_t = \{X'_{t+1}, X'_{t+2}, \ldots, X'_{t+m}\} \in X'$, randomly select m data in the normalized data set and keep their values, and set all the rest to 0. Furthermore, Gaussian noise is added to the input to obtain the corrupted input X_{cor} before denoising [33]. Then denoise the corrupted input:

$$z = a\left(W^T X_{cor} + b\right) \tag{2}$$

where z, a(·), W, b denotes output of DAE encoder, activation function (ReLU), weight, bias, respectively.

Lastly, to obtain the reconstructed the input, map the latent representation back to the input sequence, as follows:

$$X_{recon} = W^{'} z + b^{'} \tag{3}$$

where W′, b′, denotes weight, bias of the output layer in DAE, respectively. In addition, the objective function is defined as follows:

$$\xi_d = \frac{1}{n} \sum_{n}^{1} \ell(X_{cor} - X_{recon}) + \lambda\left(\| W\|_F^2 - \| W^{'} \|_F^2\right) \tag{4}$$

where $\ell(\cdot)$ denotes the loss function. Obviously, we use the L2 norm to prevent overfitting. Next the finally denoised output data $X_{recon} = \{T'_0, T'_1, T'_2, \ldots, T'_{n\text{-}1}\}$ is fed to the next decoder as the new datasets.

2.3. Convolutional transformer

Convtrans adds convolution to transformer. Since the canonical transformer is a sequence-to-sequence (seq2seq) architecture, which is inherently suitable for multi-step time series forecasting, we choose the transformer network. However, the similarity of a time point in the input X_i in canonical transformer is computed based on their pointwise values, without making full use of the rest time points (the rest 23 points in this paper) [27]. Because causal convolution shares weight parameters and have no access to future information, Convtrans can concern the local context information (like shape), which benefits multi-step time series forecasting. Here, We use one-dimensional convolution as follows:

$$X_{conv\_k} = h(k-i)^{*} X_{recon}(i) = \sum_{i=0}^{N} h(k-i) X_{recon}(i) \tag{5}$$

where X_{conv}, h, X_{recon} denotes the convolutional output, convolution kernel and input, respectively. The applied kernel size k, in_channel, out_channel, padding, dilation and stride is 9, 2, 256, 8, 1 and 1, respectively. So when we let batch size = 16, the shape of data changes from (2, 24) to (256, 33), which means the output contains more local context information.

Next, we take the convolutional output into transformer network. Here we only employ encoder to train the convolutional output and learn the function relationship between inputs and outputs. Encoder has N encoder layers, and we take $N = 3$. The embedding sub-layer employs word2vec to create a lookup table [34], whose shape is (512, 256). To fully use the position information of the time series data, we use positional encoding [35]:

$$\text{PE}\ (\text{t}, 2\text{k}) = sin\left(\text{t}/10000^{2\text{k}/\text{dmodel}}\right) \tag{6}$$

$$\text{PE}\ (\text{t}, 2\text{k}+1) = cos\left(\text{t}/10000^{2\text{k}/\text{dmodel}}\right) \tag{7}$$

where t and d_{model} denote the time step and the dimension of input.

The Masked Multi-Head Self-Attention sub-layer consists of Multi-Head Self-Attention and mask. Multi-Head Self-Attention aims to learn the similarity of input data [36–38]. To learn it, Self-Attention uses dot-product to indicate the magnitude of similarity. Let Q, K, V denote query, key, and value, Scaled Dot-Product Attention [39] is defined as follows:

$$\mathrm{Attention}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{Softmax}\left(\frac{\mathrm{QK}^{\mathrm{T}}}{\sqrt{\mathrm{d}_K}} + \mathrm{M}\right)\mathrm{V} \tag{8}$$

where Softmax maps the input as a real number between 0 and 1, then normalizes it to ensure that the sum is 1. And d_K denotes the dimension of K, which avoids minimal gradients and generates softer attention [40]. In addition, the size of Q, K, V is (24, 256), (33, 256), (33, 256) in our model, respectively. Note the mask matrix M is used to mask out future information by adding negative infinity to future data, while historical data remains intact.

To learn more useful information of similarity, Multi-Head defines as follows:

$$\mathrm{head}_\mathrm{i} = \mathrm{Attention}\left(\mathrm{QW}_\mathrm{i}{}^{\mathrm{Q}}, \mathrm{KW}_\mathrm{i}{}^{\mathrm{K}}, \mathrm{VW}_\mathrm{i}{}^{\mathrm{K}}\right) \tag{9}$$

$$\text{Multi-Head}(\mathrm{Q},\mathrm{K},\mathrm{V}) = \mathrm{Concat}(\mathrm{head}_0, \mathrm{head}_1, \ldots, \mathrm{head}_{\mathrm{n}-1})W^O \tag{10}$$

where $\mathrm{W}_\mathrm{i}^\mathrm{Q}$, $\mathrm{W}_\mathrm{i}^\mathrm{K}$ and $\mathrm{W}_\mathrm{i}^\mathrm{V}$ are corresponding weight matrix, W^O is a trainable splicing matrix.

Feed-Forward sub-layer strengthens the ability to express this representation by using non-linear variation [41]. In this paper, Feed-Forward sub-layer has two different mappings (linear and ReLU non-linear) as follows:

$$\mathrm{FFN}(\mathrm{x}) = \mathrm{ReLU}(\mathrm{xW}_1 + \mathrm{b}_1)\mathrm{W}_2\mathrm{b}_2 \tag{11}$$

$$\mathrm{ReLU}(\mathrm{x}) = max(0, \mathrm{xW}_1 + \mathrm{b}_1) \tag{12}$$

Therefore, the created Feed-Forward sub-layer is applied to x (temperature in each time step) identically to mitigate overfitting, prevent gradient explosion, and reduce computational effort.

The last Add & Norm sub-layer mitigates the gradient disappearance and the degradation of the weight matrix [42,43] as follows:

$$\mathrm{Z} = \mathrm{f}(\mathrm{x}) + \mathrm{x} \tag{13}$$

where Z denotes output, f(x) denotes learned function.

Finally, we reduce dimensionality to one dimension to obtain trained results by using linear layer as follows:

$$T_{prediction} = W_{linear}Z + b_{linear} \tag{14}$$

where $\mathrm{T}_\mathrm{prediction}$, $\mathrm{W}_\mathrm{linear}$, $\mathrm{b}_\mathrm{linear}$ denotes the output, weight, bias of the linear layer, respectively.

Furthermore, the Convtrans model is programmed employing Pytorch [44]. In Convtrans model, loss is evaluated by using the loss function of RMSE as follows:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{\mathrm{n}}(Tprediction - Ttrue)^2}{n}} \tag{15}$$

where n, $\mathrm{T}_\mathrm{prediction}$ and T_true denote total number of steps in one prediction, the predicted temperature and the true temperature, respectively. In addition, some hyper-parameters are given: the initial learning rate is 0.004; the Learning rate decay is 0, and the optimizer is SGD.

3. Datasets

Here we select 2018-04-12_batch8_CH43 dataset (containing 830,319 sample data) from previous researchers [17] to do the single-step and multi-step time series forecasting. Note the temperature data we used are the cell surface temperatures. The charging temperature range of the lithium-ion phosphate (LFP)/graphite cells is 0 °C–55 °C and the discharging temperature range is −30 °C–55 °C. In their experiment, these cells are cycled in a forced convection temperature chamber setting to 30 °C (Amerex Instruments). In order to overcome the problem of time-consuming calculations, we use the average temperature of each cycle as input time series data shown in Fig. 3. In the whole data periods, the temperature fluctuation is very small at the beginning, but with the increase of the number of cycles, the temperature fluctuation range is gradually increasing, and after 1000 cycles, the temperature is generally on the rise. Furthermore, the temperature of each cycle is theoretically closely tied to the temperature of the previous cycles, which means that the temperature of next time point will be influenced by the historical temperature points' shape. This is consistent with the superiority of Convtrans algorithm. In this way, our forecasting results based on Convtrans algorithm can give a reasonable values and trends of the next 24 cycles as a reference.

4. Results and analysis

First, we show the prediction of the four algorithms with the split ratio 8:2 in Fig. 4. It is not difficult to see that Convtrans has the best prediction with the true experimental results in multi-step forecasting. To further evaluate the comprehensive performance of the algorithm, the RMSE, running time and correlation coefficient (R) are calculated in Fig. 5. To make RMSE and R comparable in the four algorithms, we calculate them in whole test sets by adding all calculated RMSE and R in each prediction up respectively, and then averaging the cumulative results.

According to the calculated RMSE results, the Convtrans has the minimum RMSE among the four algorithms, reaching 0.120, while that of LSTM reach 0.202, which is the maximum shown in Fig. 5(a). As for the time cost, due to the complexity of the algorithm, Convtrans algorithm takes the most time shown in Fig. 5(b). During the training process, Convtrans computation time reaches in about 1 min in our lab computer (CPU: Intel i5-9400U@2.90 GHz, Memory: 16 GB). Note that the time cost of prediction has no big difference for the four algorithms.

Furthermore, the trend of the forecast is also an important indicator in our assessment of the multi-step forecast. So the value of R, which is used to study the degree of correlation between the predicted and true values, represents the degree of accuracy of the predicted trend. R of Convtrans is 0.421 for the ratio of 8:2 shown in Fig. 5(c). As a comparison, the other three algorithms have negative coefficients, which means that when the temperature is falling, their predicted temperature is perhaps rising.

To support this, we give other two predictions shown in Fig. 6. Obviously, there is a dip of the true 24 temperature points in Fig. 6(a). However, only the Convtrans prediction shows the shape of dip near the cycle 1015, while other three algorithms' predicted trends are very different from the true trend. In detail, LSTM-predicted temperature trend shows a peak at near cycle 1015, TCN-predicted temperature fluctuates up and down. Although GRU-predicted trend is somewhat consistent with the true, the predicted decline of the temperature near cycle 1030 departs from the uptrend of the true values. The comparison is more obvious in Fig. 6(b). The Convtrans-predicted temperature curve trend is consistent with the real one, even the shape is basically the same. Therefore, we can obtain the conclusion that the Convtrans can predict the suitable trend. This conclusion fits with Convtrans advantage of paying more attention to the shape of the surrounding points.

In addition, when we use the split ratio 5:5, Convtrans still has the minimum RMSE and the maximum R among the four algorithms as in Fig. S1, showing that Convtrans still has the aforementioned effective prediction.

On the other hand, to find out whether multi-step time series forecasting is a more suitable application than single-step time series forecasting, we compare the single-step (LSTM, GRU and TCN) and multi-step time series forecasting (Convtrans). Here, the X_i used in the input

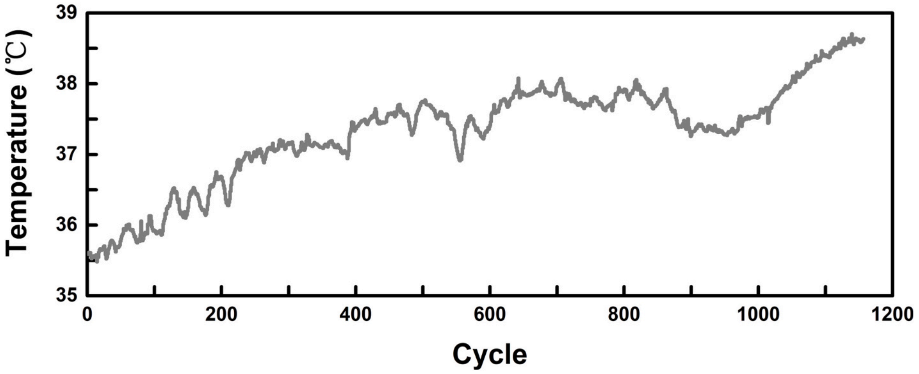

Fig. 3. Cycle performance on temperature for the battery.

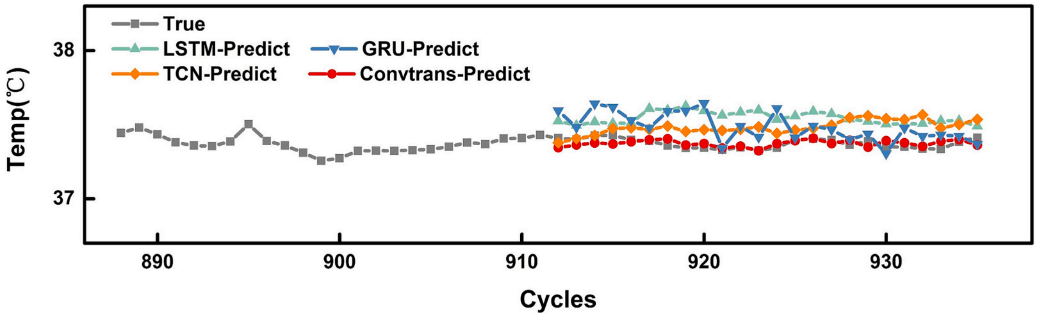

Fig. 4. Four algorithms' forecasting results for the next 24 cycles. The ratios of training data and prediction data are 8:2.

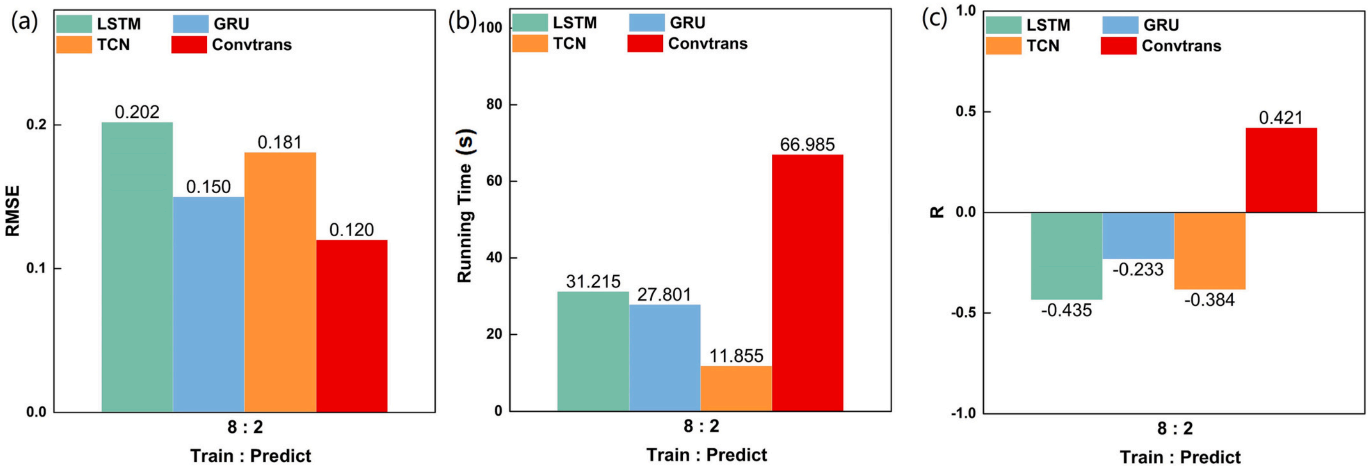

Fig. 5. Four algorithms' forecasting (a) RMSE in whole test sets and (b) Running time (c) R in the split ratio 8:2.

is consist of 24 time points in the three algorithms for the comparability (whole prediction is shown in Fig. S2). All the three single-step algorithms have pleased prediction in the ratio 8:2 shown in Fig. 7. However, predicting only the temperature at the next moment is difficult to apply practically. By comparison, our Convtrans network predict more time points on the basis of the same amount of input time point data and does not sacrifice accuracy. As for time cost, the running time of TCN is the minimum. This shows that TCN algorithm sacrifices accuracy for fast running time. Finally, in terms of predicting trends, single-step time series prediction is unable to obtain correlation coefficient R like multi-step time because only one temperature point is available for each prediction. In all, compared with the single-step forecasting, Convtrans costs 5 times the running time, yet predicts 24 times temperature data without sacrificing accuracy.

To verify whether the Convtrans algorithm is able to predict more data points, we show the predicted 48 temperature points based on the previous 48 data points in Fig. S3. It is not difficult to find that with more predicted points, Convtrans is still able to accurately predict the trend and even the shape of the temperature curve, with correlation coefficient R as high as 0.573. As for accuracy, it's RMSE is 0.111, which means that predicting more points does not sacrifice accuracy. In addition, we also verify that all our conclusions hold for the other datasets (labeled as 2019-01-24_batch9_CH21 in the dataset of published article [17]) as shown in Fig. S4, Tables S1 and S2. So Convtrans is a suitable method in forecasting the temperature of LIBs.

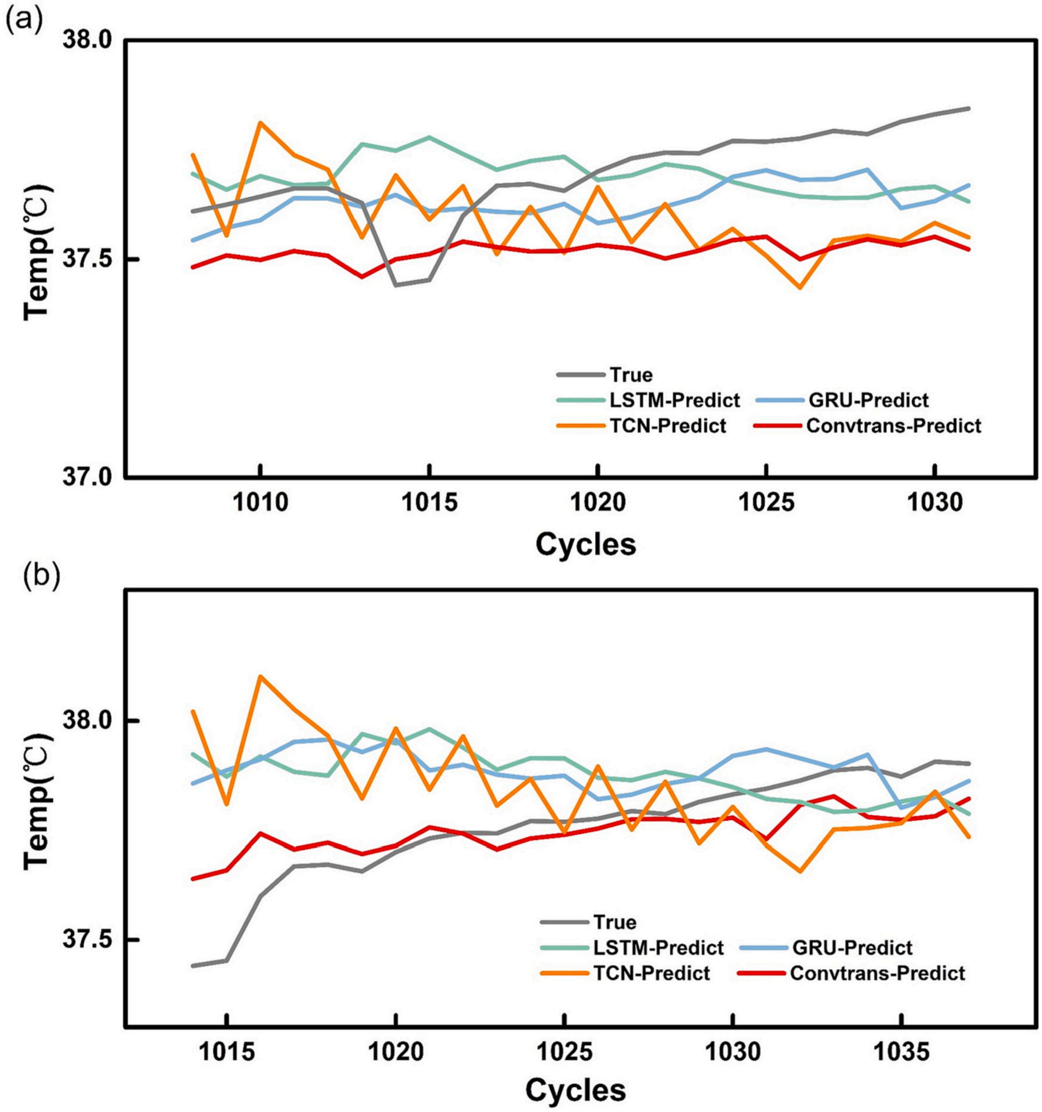

Fig. 6. Four algorithms' forecasting results for the next 24 cycles starting with the (a) 1008th and (b) 1014th cycle in the ratio of 8:2.

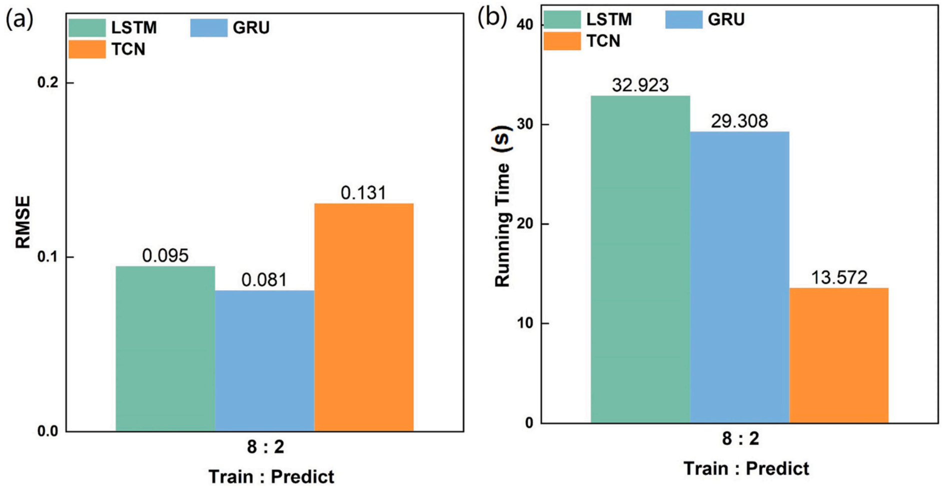

Fig. 7. Three algorithms' single-step forecasting (a) RMSE in whole test sets and (b) Running time in the split ratio 8:2.

5. Conclusion

Since the average temperature of each cycle is consistent with the characteristics of time series, we use data-driven method to predict the temperature of LIBs in this research. To predict more temperature data, a multi-step time series forecasting battery temperature technology is developed based on Convtrans algorithm, which has smaller prediction errors and provides accurate trends and even essentially similar shape. Furthermore, compared to single-step time series forecasting, although it spends five times the running time, Convtrans algorithm does not sacrifice accuracy and obtain more predicted temperature data. In all, this paper proposes a new model based on Convtrans for battery temperature warning in time, which is of guidance for the design of battery thermal management systems.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

Acknowledgements

D.X. acknowledges support from the National Natural Science Foundation of China (No. 51806072).

Appendix A. Supplementary data

Here are the predictions of three algorithms (LSMF, GRU and TCN) for the single time series forecasting and the predictions of Convtrans in the case of predicting more points (48). The prediction using other datasets is also provided. Supplementary data to this article can be found online at https://doi.org/10.1016/j.est.2023.107092.

References

[1] C.-C. Chang, S.-Y. Huang, W.-H. Chen, Thermal and solid electrolyte interphase characterization of lithium-ion battery, Energy 174 (2019) 999–1011.

[2] Y. Tian, G. Zeng, A. Rutt, et al., Promises and challenges of next-generation "Beyond Li-ion" batteries for electric vehicles and grid decarbonization, Chem. Rev. 121 (3) (2021) 1623–1669.

[3] L. Wei, Z. Lu, F. Cao, et al., A comprehensive study on thermal conductivity of the lithium-ion battery, Int. J. Energy Res. 44 (12) (2020) 9466–9478.

[4] A. Masias, J. Marcicki, W.A. Paxton, Opportunities and challenges of lithium ion batteries in automotive applications, ACS Energy Lett. 6 (2) (2021) 621–630.

[5] X. Zeng, M. Li, D. Abd El-Hady, et al., Commercialization of lithium battery technologies for electric vehicles, Adv. Energy Mater. 9 (27) (2019), 1900161.

[6] Y. Jiang, Y. Yu, J. Huang, et al., Li-ion battery temperature estimation based on recurrent neural networks, Sci.China Technol.Sci. 64 (6) (2021) 1335–1344.

[7] A. Pesaran, S. Santhanagopalan, G. Kim, Addressing the Impact of Temperature Extremes on Large Format Li-ion Batteries for Vehicle Applications (Presentation) [R], National Renewable Energy Lab. (NREL), Golden, CO (United States), 2013.

[8] Q. Wang, P. Ping, X. Zhao, et al., Thermal runaway caused fire and explosion of lithium ion battery, J. Power Sources 208 (2012) 210–224.

[9] S.A. Khateeb, S. Amiruddin, M. Farid, et al., Thermal management of Li-ion battery with phase change material for electric scooters: experimental validation, J. Power Sources 142 (1–2) (2005) 345–353.

[10] S.S. Zhang, K. Xu, T.R. Jow, The low temperature performance of Li-ion batteries, J. Power Sources 115 (1) (2003) 137–140.

[11] D. Burow, K. Sergeeva, S. Calles, et al., Inhomogeneous degradation of graphite anodes in automotive lithium ion batteries under low-temperature pulse cycling conditions, J. Power Sources 307 (2016) 806–814.

[12] X. Hu, L. Xu, X. Lin, et al., Battery lifetime prognostics, Joule 4 (2) (2020) 310–346.

[13] S. Zhu, C. He, N. Zhao, et al., Data-driven analysis on thermal effects and temperature changes of lithium-ion battery, J. Power Sources 482 (2021), 228983.

[14] K.E. Thomas, J. Newman, Thermal modeling of porous insertion electrodes, J. Electrochem. Soc. 150 (2) (2003) A176.

[15] J. Zhang, J. Huang, Z. Li, et al., Comparison and validation of methods for estimating heat generation rate of large-format lithium-ion batteries, J. Therm. Anal. Calorim. 117 (1) (2014) 447–461.

[16] P. Guo, Z. Cheng, L. Yang, A data-driven remaining capacity estimation approach for lithium-ion batteries based on charging health feature extraction, J. Power Sources 412 (2019) 442–450.

[17] K.A. Severson, P.M. Attia, N. Jin, et al., Data-driven prediction of battery cycle life before capacity degradation, Nat. Energy 4 (5) (2019) 383–391.

[18] L. Lu, X. Han, J. Li, et al., A review on the key issues for lithium-ion battery management in electric vehicles, J. Power Sources 226 (2013) 272–288.

[19] S.W. Chen, S.S. Shivakumar, S. Dcunha, et al., Counting apples and oranges with deep learning: a data-driven approach, IEEE Robot.Autom.Lett. 2 (2) (2017) 781–788.

[20] S. Rani, G. Sikka, Recent techniques of clustering of time series data: a survey, Int. J. Comput. Applic. 52 (15) (2012) 1–9.

[21] A.A. Cook, G. Mısırlı, Z. Fan, Anomaly detection for IoT time-series data: a survey, IEEE Internet Things J. 7 (7) (2019) 6481–6494.

[22] L. Yunpeng, H. Di, B. Junpeng, et al., Multi-step ahead time series forecasting for different data patterns based on LSTM recurrent neural network, in: 2017 14th web information systems and applications conference (WISA), IEEE, 2017, pp. 305–310.

[23] Z. Tian, H. Chen, Multi-step short-term wind speed prediction based on integrated multi-model fusion, Appl. Energy 298 (2021), 117248.

[24] Y. Mo, Q. Wu, X. Li, et al., Remaining useful life estimation via transformer encoder enhanced by a gated convolutional unit, J. Intell. Manuf. 32 (7) (2021) 1997–2006.

[25] J. Hao, X. Wang, B. Yang, et al., Modeling recurrence for transformer, in: Proceedings of NAACL-HLT, 2019, pp. 1198–1207.

[26] E. Egonmwan, Y. Chali, Transformer and seq2seq model for paraphrase generation, in: Proceedings of the 3rd Workshop on Neural Generation and Translation, 2019, pp. 249–255.

[27] S. Li, X. Jin, Y. Xuan, et al., Enhancing the locality and breaking the memory bottleneck of transformer on time series forecasting, in: Proceedings of the 33rd International Conference on Neural Information Processing Systems 471, 2019, pp. 5243–5253.

[28] D. Zhou, B. Wang, Battery health prognosis using improved temporal convolutional network modeling, J. Energy Storage 51 (2022), 104480.

[29] Y. Liu, J. Li, G. Zhang, et al., State of charge estimation of lithium-ion batteries based on temporal convolutional network and transfer learning, IEEE Access 9 (2021) 34177–34187.

[30] V. Joshi, M. Le Gallo, S. Haefeli, et al., Accurate deep neural network inference using computational phase-change memory, Nat. Commun. 11 (1) (2020) 1–13.

[31] R. Hu, S. Wen, Z. Zeng, et al., A short-term power load forecasting model based on the generalized regression neural network with decreasing step fruit fly optimization algorithm, Neurocomputing 221 (2017) 24–31.

[32] D. Chen, W. Hong, X. Zhou, Transformer network for remaining useful life prediction of lithium-ion batteries, IEEE Access 10 (2022) 19621–19628.

[33] X.G. Fan, Y. Zeng, Y.L. Zhi, et al., Signal-to-noise ratio enhancement for Raman spectra based on optimized Raman spectrometer and convolutional denoising autoencoder, J. Raman Spectrosc. 52 (4) (2021) 890–900.

[34] X. Peng, G. Long, T. Shen, et al., BiteNet: bidirectional temporal encoder network to predict medical outcomes, in: 2020 IEEE International Conference on Data Mining (ICDM), IEEE, 2020, pp. 412–421.

[35] X. Chu, Z. Tian, B. Zhang, et al., Conditional positional encodings for vision transformers, 2021 arXiv preprint arXiv:2102.10882.

[36] L. Wu, S. Li, C.-J. Hsieh, et al., SSE-PT: sequential recommendation via personalized transformer, in: Proceedings of the Fourteenth ACM Conference on Recommender Systems, F, 2020.

[37] H. Yuan, Z. Cai, H. Zhou, et al., TransAnomaly: video anomaly detection using video vision transformer, IEEE Access 9 (2021) 123977–123986.

[38] J. Zhang, H. Zhao, J. Li, TRS: transformers for remote sensing scene classification, Remote Sens. 13 (20) (2021) 4143.

[39] W. Wang, F. Wei, L. Dong, et al., Minilm: deep self-attention distillation for task-agnostic compression of pre-trained transformers, Adv. Neural Inf. Proces. Syst. 33 (2020) 5776–5788.

[40] A. Vaswani, N. Shazeer, N. Parmar, et al., Attention is all you need, in: Proceedings of the 31st International Conference on Neural Information Processing Systems, 2017, pp. 6000–6010.

[41] K. Yuan, S. Guo, Z. Liu, et al., Incorporating convolution designs into visual transformers, in: Proceedings of the IEEE/CVF International Conference on Computer Vision, 2021, pp. 579–588.

[42] K. Greff, R.K. Srivastava, J. Schmidhuber, Highway and residual networks learn unrolled iterative estimation, 2016 arXiv preprint arXiv:1612.07771.

[43] W. Shang, K. Sohn, D. Almeida, et al., Understanding and improving convolutional neural networks via concatenated rectified linear units, in: Proceedings of the International Conference on Machine Learning, F, PMLR, 2016.

[44] A. Paszke, S. Gross, F. Massa, et al., Pytorch: an imperative style, high-performance deep learning library, Adv. Neural Inf. Proces. Syst. 32 (2019).