Electronic Supplementary Material (ESI) for Journal of Materials Chemistry A.

Supplementary Note 1. The methodologies for SoH calculation and identifying the tendency of SoH distribution

To overcome the limitation of difficulty in measuring the capacity of internal modules, which are used as labels for training DeepSUGAR, a new method of capacity calculation in State-of-Health (SoH) formula (Eqn. 1) was proposed where Q_{max} is the available maximum charge and C_r is the rated capacity.

$$SOH = 100\frac{Q_{max}}{C_r} \qquad \text{(Eqn. 1)}$$

Since the setup of the cycler, charging condition of constant power with 289.3W to 57.8V and discharging condition of constant current with 0.5C to 42.0V, is based on the pack, it is possible to move on to the next cycle even if some internal modules have not yet reached voltage 3.0V. Therefore, based on the coulomb counting method, the capacity was calculated by accumulating current until the voltage reached 43.4V, 3.1V as a threshold for the pack and its individual modules, respectively. SoH was calculated based on the current accumulation values of the first cycle and current cycle.

To identify the tendency of SoH distribution of fourteen modules more clearly, the linear regressor was used. All the parameters of LinearRegression() were set as default[1].

Supplementary Note 2. The methodologies for Temperature distribution regression in Fig. 2d

In our experimental setup, four temperatures at 2nd, 6th, 9th, and 13th modules were collected by battery management system. Based on four temperatures, the temperature distribution inside the pack was predicted using Gaussian Process Regressor (GPR) to analyze the cause of SoH distribution of modules inside the pack. The parameters of GPR were set 'kernel' as 1*RBF(length_scale = 1.0, length_scale_bounds=(1e-2, 1e2)), 'n_restarts_optimizer' as 9, 'random_state' as 0, and other parameters as default[1].

Supplementary Note 3. Conventional machine learning framework for Li-ion battery SoH estimation

This section provides a detailed description of the working process of the conventional machine learning frameworks.

1.1 Data preprocessing and feature extraction

Battery degradation is a complex phenomenon resulting from various mechanisms, such as SEI layer formation and Li-plating, occurring inside the battery cell. Estimating battery health from these mechanisms is limited as they are difficult to detect without battery disassembly. Hence, a phenomenological approach based on measurable parameters, such as voltage, capacity, and temperature, is required. Previous research have noted distinct patterns in cycling profiles with battery aging, such as a shift the voltage profile due to the decrease time required for one cycle and the peak of the capacity curve shifts as cycling progresses, as shown in Fig. 2a. To compare the performance of DeepSUGAR, we analyzed the salient patterns of cycling curves resulting from battery aging.

In the construction of conventional frameworks, we extracted 49 features as battery health indicators, as presented in Table S4. These features are classified into five groups: voltage-related, temperature-related, capacity-related, current-related, and IC-related features. Although some of the extracted features are closely associated with battery aging, others may also be negligible or have no relevance to SoH. Including irrelevant features can lead to an increase in computational cost and potential overfitting problems, the selection of optimal feature set is imperative.

1.2 Feature selection

In this paper, five feature selection methods are concerned: ElasticNet, PCA, Filter method, Lasso, and Feature Selector. The detailed model parameters are listed in Table S3, and the selected features are listed in Table S5. The Decision Tree (DT) regressor is used as the basic SoH estimation model to find out the optimal feature set. The parameter was set 'criterion' as 'squared_error', 'splitter' as 'best', 'max_depth' as 'None', 'min_samples_split' as '2', 'min_samples_leaf' as '1', 'min_weight_fraction_leaf' as '0.0', 'max_features' as 'None', 'random_state' as 'None', 'max_leaf_nodes' as 'None', 'min_impurity_decrease' = '0.0', and 'ccp_alpha' = '0.0'[1]. As a result, the feature set which was selected equally by ElasticNet and Lasso has the best performance, 0.004925 of RMSE.

1.3 Compared SoH estimation performance with DeepSUGAR

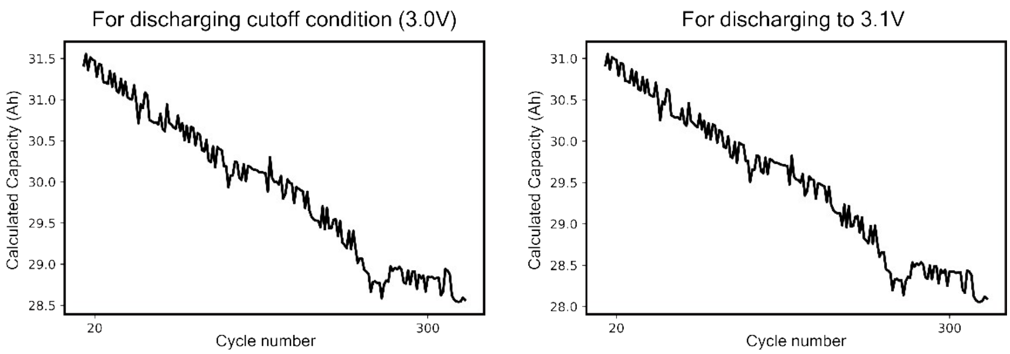
In this section, the authors compare the performance of four machine learning models for estimating SoH, GPR, NuSVR, RF, and KNN, using a 6-feature set that was selected equally by ElasticNet and Lasso. The model parameters are provided in Table S11[1].

The trained models were evaluated by comparing their performance degradation with DeepSUGAR by applying the trained and validated models using pack cycling data to module cycling data (Table S12).

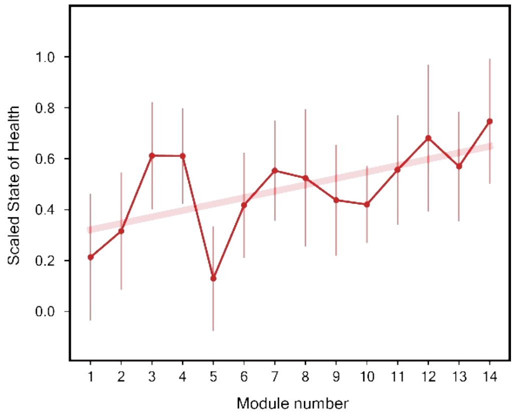
Supplementary Note 4. Data volume utilized for training DeepSUGAR.

In training DeepSUGAR for state-of-health estimation, we utilized cycling data from a 14S7P pack. The data was collected over 300 cycles augmented by random noise within a range of 0.01, 1% of the scaled voltage, current, and capacity profiles from 0 to 1. Of the total 1750 battery pack cycling profiles, the CNN model in DeepSUGAR was trained using 70% randomly extracted and validated using 70% of the unused profiles. The remaining data was used to test the model performance. Furthermore, the applicability of the model, trained using pack cycling data, to fourteen internal module data was evaluated with 2450 points that were neither used in training nor validation.

Supplementary Fig. 1. The capacity degradation during the cycling test. (A) The capacity for each cycle under discharging cutoff conditions (3.0V). (B) The calculated capacity for each cycle reaching 3.1V to encompass the range necessary to account for the uneven discharging process within the battery pack. These values were utilized to calculate SoH.



Supplementary Fig. 2. The scaled SoH distribution of internal modules over 300 cycles. Observed values are represented by the red line and the trend predicted by linear regression by translucent red line.



Supplementary Fig. 3. The convolutional neural network for SoH estimation. (A) Model summary of the CNN architecture. (B) Key codes to construct CNN model. (C) Training and validation loss over 50 epochs obtained by training the CNN regression model.

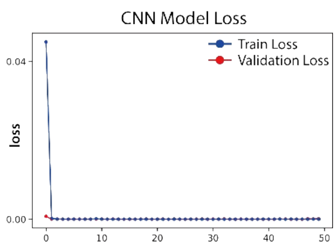
a.

Layer type	Output Shape	Parameters #
Conv2D	(None, 256, 3, 32)	160
Conv2D	(None, 256, 3, 64)	8256
MaxPooling2D	(None, 255, 2, 64)	0
Flatten	(None, 32640)	0
Dense	(None, 64)	2089024
Dense	(None, 1)	65

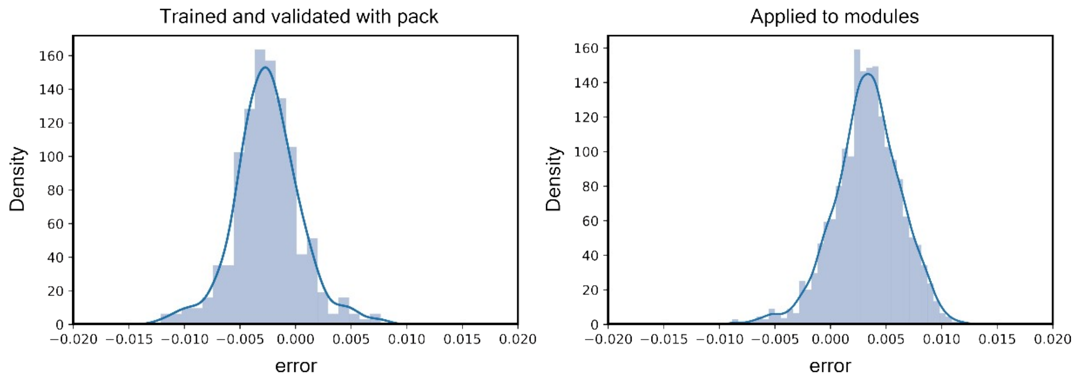
b.

```
model_cnn = Sequential()
model_cnn.add(Conv2D(32, kernel_size = (2,2), strides = (1,1),
              padding = 'same', activation = 'relu', input_shape = input_shape))
model_cnn.add(Conv2D(64, kernel_size = (2,2), strides = (1,1),
              padding = 'same', activation = 'relu'))
model_cnn.add(MaxPooling2D(pool_size = (2,2), strides = (1,1)))
model_cnn.add(Flatten())
model_cnn.add(Dense(64,activation = 'relu'))
model_cnn.add(Dense(1))
model_cnn.summary()
```

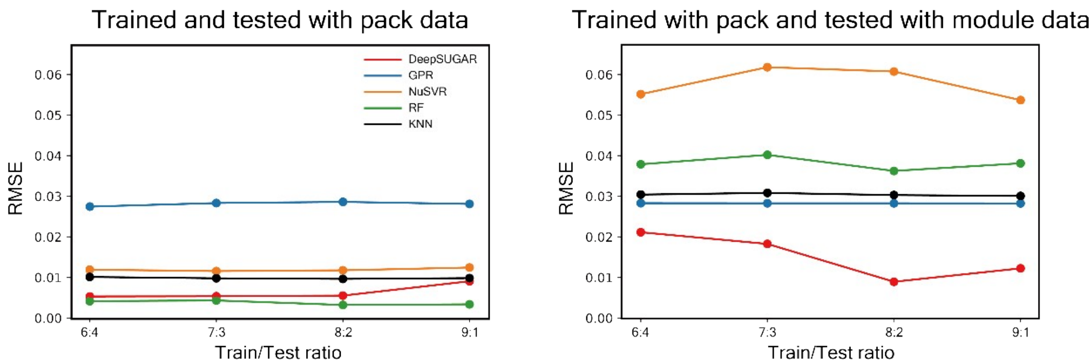
c.



Supplementary Fig. 4. (A) The error distribution plot of CNN framework trained and validated with pack cycling data. (B) The error distribution plot of CNN framework applied to module cycling data.



Supplementary Fig. 5. Comparison of SoH estimation performance between DeepSUGAR and conventional frameworks in terms of the mean RMSE of 10 iterations by varying the training-test ratio to 9:1, 8:2, 7:3, and 6:4. The RMSE values for each case are listed in Table. S17-S18.



Supplementary Fig. 6. The conditional deep convolutional generative adversarial networks for restoring cycling profiles of fourteen individual modules. (A) Model summary of generator in cDCGAN. (B) Key codes to construct generator in cDCGAN. (C) Model summary of discriminator in cDCGAN. (D) Key codes to construct discriminator in cDCGAN.

a.

Layer type	Output Shape	Parameters #
Dense	(None, 57344)	5791744
Reshape	(None, 7, 32, 256)	0
Conv2DTranspose	(None, 14, 64, 128)	295040
BatchNormalization	(None, 14, 64, 128)	512
LeakyReLU	(None, 14, 64, 128)	0
Conv2DTranspose	(None, 14, 128, 64)	73792
BatchNormalization	(None, 14, 128, 64)	256
LeakyReLU	(None, 14, 128, 64)	0
Conv2DTranspose	(None, 14, 256, 3)	1731
Activation	(None, 14, 256, 3)	0

b.

```
model_gen = Sequential()
model_gen.add(Dence(256*32*7, input_dim = z_dim))
model_gen.add(Reshape((7,32,256)))
model_gen.add(Conv2DTranspose(128, kernel_size = 3, strides = (2,2), padding = 'same'))
model_gen.add(BatchNormalization(momentum=0.8))
model_gen.add(LeakyReLU(alpha=0.2))
model_gen.add(Conv2DTranspose(64, kernel_size = 3, strides = (1,2), padding = 'same'))
model_gen.add(BatchNormalization(momentum=0.8))
model_gen.add(LeakyReLU(alpha=0.2))
model_gen.add(Conv2DTranspose(3, kernel_size = 3, strides = (1,2), padding = 'same'))
model_gen.add(Activation('tanh'))
model_gen.summary()
```

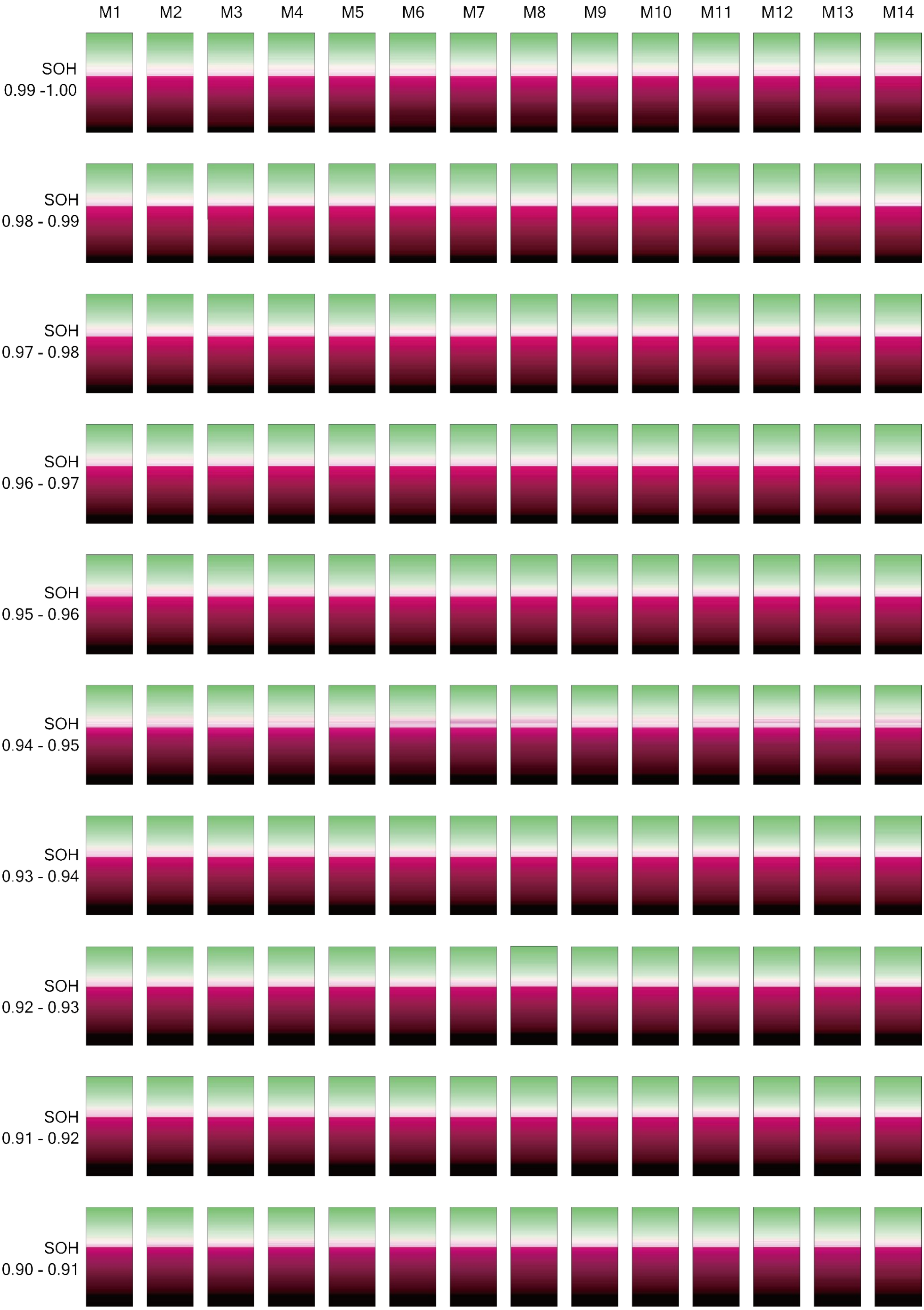
c.

Layer type	Output Shape	Parameters #
Conv2D	(None, 7, 128, 64)	2368
LeakyReLU	(None, 7, 128, 64)	0
Conv2D	(None, 4, 64, 64)	36928
LeakyReLU	(None, 4, 64, 64)	0
Conv2D	(None, 2, 32, 128)	73856
LeakyReLU	(None, 2, 32, 128)	0
Flatten	(None, 8192)	0
Dense	(None, 256)	2097408
LeakyReLU	(None, 256)	0
Dropout	(None, 256)	0
Dense	(None, 1)	257

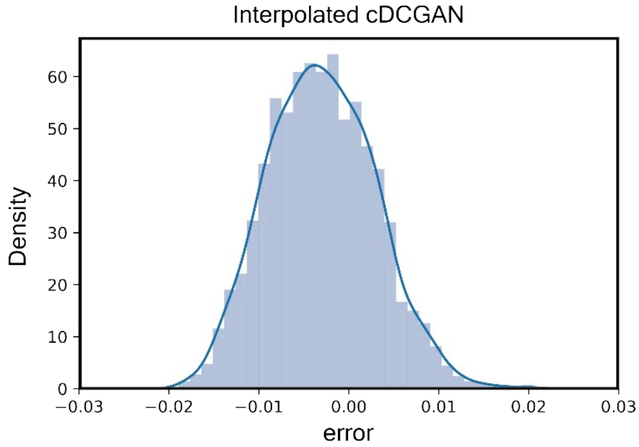
d.

```
model_dis = Sequential()
model_dis.add(Conv2D(64, kernel_size = 3, strides = 2,
              input_shape = (img_shape[0]. img_shape[1[, img_shape[2]+1), padding = 'same'))
model_dis.add(LeakyReLU(alpha=0.2))
model_dis.add(Conv2D(64, kernel_size = 3, strides = 2, padding = 'same'))
model_dis.add(LeakyReLU(alpha=0.2))
model_dis.add(Conv2D(128, kernel_size = 3, strides = 2, padding = 'same'))
model_dis.add(LeakyReLU(alpha=0.2))
model_dis.add(Flatten())
model_dis.add(Dense(256))
model_dis.add(LeakyReLU(alpha=0.2))
model_dis.add(Dropout(0.5))
model_dis.add(Dense(1, activation = 'sigmoid))
model_dis.summary()
```

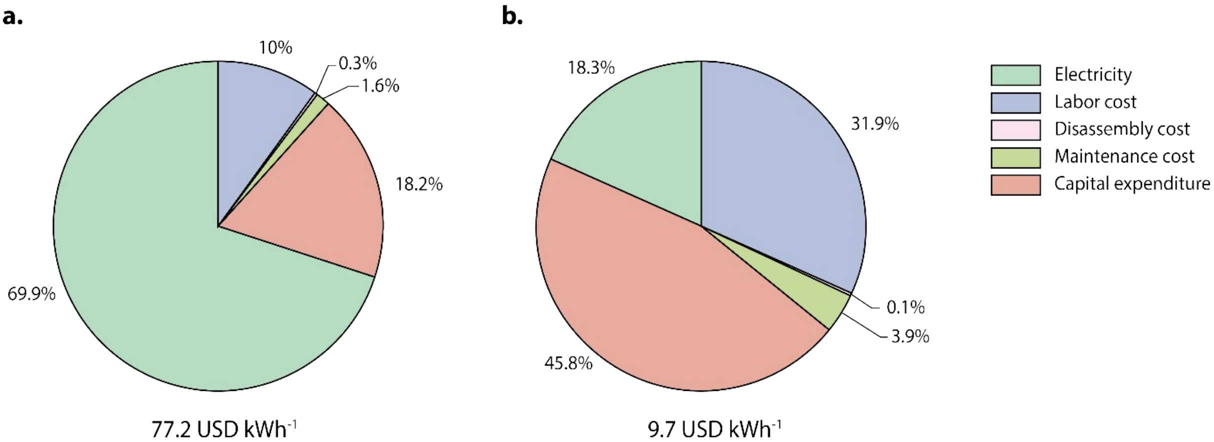
Supplementary Fig. 7. Restored V, I, Q-converted spatial channels of fourteen individual modules with cDCGAN.



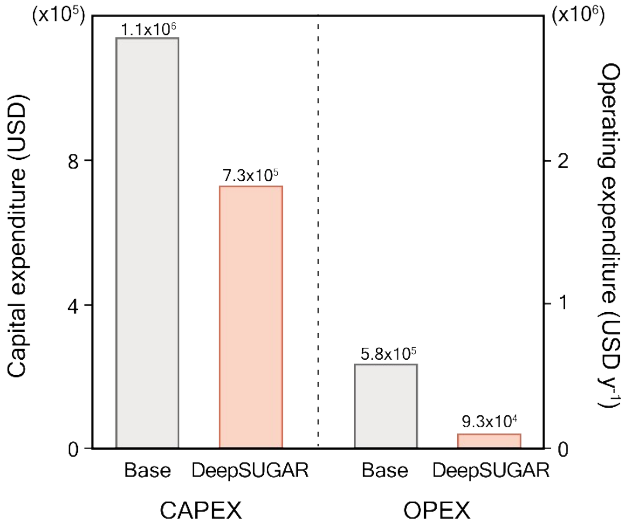
Supplementary Fig. 8. The error distribution plot of interpolated cDCGAN.



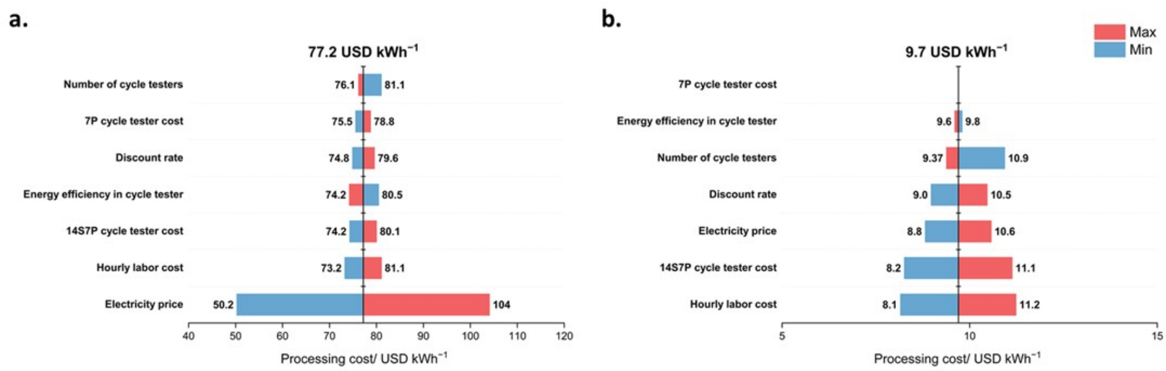
Supplementary Fig. 9. Distribution of processing cost for (a) a conventional methodology and (b) DeepSUGAR approach.



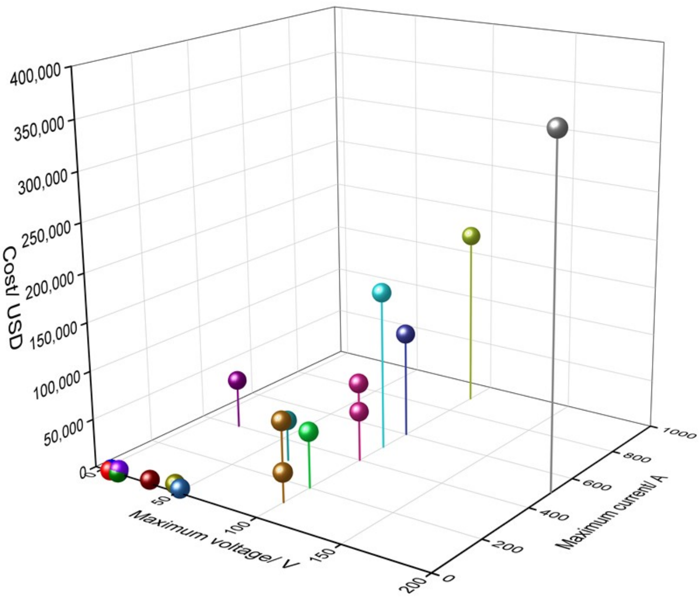
Supplementary Fig. 10. Comparison of capital and operating expenditure between the base and DeepSUGAR.



Supplementary Fig. 11. Result of sensitivity analysis in terms of processing cost in USD kWh^{-1} for (a) base and (b) DeepSUGAR.



Supplementary Fig. 12. Prices of battery cycle testers in USD per channel along with respective maximum voltage and current.



Supplementary Table 1. The SoH estimation performance of the trained and validated CNN model with pack cycling data over 10 iterations.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.962797	0.005524	0.000031	0.004185	0.004383
Try 2	0.966258	0.005261	0.000028	0.004321	0.004553
Try 3	0.964638	0.005386	0.000029	0.004145	0.00434
Try 4	0.967547	0.00516	0.000027	0.003918	0.004111
Try 5	0.968689	0.005068	0.000026	0.003827	0.00401
Try 6	0.964438	0.005401	0.000029	0.004053	0.004237
Try 7	0.967047	0.005199	0.000027	0.003903	0.004081
Try 8	0.962936	0.005514	0.00003	0.004214	0.004416
Try 9	0.965216	0.005342	0.000029	0.004594	0.004841
Try 10	0.966422	0.005248	0.000028	0.004027	0.004244
Mean	0.965599	0.00531	2.84E-05	0.004119	0.004322

Supplementary Table 2. The SoH estimation performance of CNN model applied to module cycling data over 10 iterations.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.950013	0.006315	0.00004	0.005474	0.005729
Try 2	0.933922	0.007261	0.000053	0.006569	0.006881
Try 3	0.923709	0.007802	0.000061	0.007122	0.007489
Try 4	0.949809	0.006328	0.00004	0.005581	0.005847
Try 5	0.927444	0.007609	0.000058	0.006831	0.007158
Try 6	0.90361	0.00877	0.000077	0.00805	0.008431
Try 7	0.950491	0.006285	0.00004	0.005541	0.005808
Try 8	0.935807	0.007157	0.000051	0.006533	0.006852
Try 9	0.909534	0.008496	0.000072	0.00773	0.00809
Try 10	0.923515	0.007812	0.000061	0.007191	0.007536
Mean	0.930785	0.0073835	0.0000553	0.0066622	0.0069821

Supplementary Table 3. In conventional ML frameworks, model parameters to select significant battery health indicators[1].

Model	Model parameter
ElasticNet	*alpha=0.9, l1_ratio=0.5, fit_intercept=True, precompute=False, max_iter=1000, copy_X=True, tol=0.0001, warm_start=False, positive=False, random_state=None, selection='cyclic'*
PCA	*cumulative explained_variance_ratio > 0.82*
Filter method	*variance threshold=0.05, correlation > 0.70*
Lasso	*alpha = 0.7, fit_intercept = True, precompute = False, copy_x = True, max_iter = 1000, tol = 1e^{-4}, warm_start = False, positive = False, random_state = None, selection = 'cyclic'*
Feature Selector	*correlation threshold=0.5*

Supplementary Table 4. Extracted battery health indicators for estimating SoH with conventional ML models

Feature Type	State	No.	Description
Voltage related	Charge	F1	Initial voltage value
		F2	Area of voltage profile
		F3	Final voltage value
		F4	Time at which the last voltage was measured
		F5	Area of 30 points in front
		F6	Slope of 30 points in front
		F7	Skewness of 30 points in front
		F8	Kurtosis of 30 points in front
	Discharge	F9	Initial voltage value
		F10	Area of voltage profile
		F11	Final voltage value
		F12	Time at which the last voltage was measured
		F13	Area of 30 points in front
		F14	Slope of 30 points in front
		F15	Skewness of 30 points in front
		F16	Kurtosis of 30 points in front
Temperature related	Charge	F17	Initial Temperature value
		F18	Final temperature value
		F19	Area of temperature profile
		F20	Slope of temperature profile
		F21	Area of 30 points in front
		F22	Slope of 30 points in front
		F23	Skewness of 30 points in front
		F24	Kurtosis of 30 points in front
		F25	Final temperature value
		F26	Area of temperature profile

	Discharge	F27	Slope of temperature profile
		F28	Slope of 30 points in front
		F29	Area of 30 points in front
		F30	Skewness of 30 points in front
		F31	Kurtosis of 30 points in front
Capacity related	Charge	F32	Initial capacity value
		F33	Final capacity value
		F34	Maximum capacity value
		F35	Time at which the maximum capacity was measured
		F36	Slope of 30 points in front
	Discharge	F37	Time at which the minimum capacity was measured
		F38	Slope of 30 points in front
Current related	Charge	F39	Initial current value
		F40	Final current value
		F41	Area of current profile
		F42	Slope of current profile
		F43	Slope of 30 points in front
IC related	Discharge	F44	First peak value of IC profile
		F45	Time at which the first peak was measured
		F46	Second peak value of IC profile
		F47	Time at which the second peak was measured
		F48	Third peak value of IC profile
		F49	Time at which the third peak was measured

Supplementary Table 5. Battery health indicators selected by the models listed in Table. S3 and the mean SoH estimation performance of DecisionTreeRegressor over 10 iterations (STable.6-STable.10). The ElasticNet and Lasso algorithm select the same feature set.

Model	Selected features	R-sqared	RMSE	MSE	MAE	MAPE
Before Selection	F1~F49	0.953818	0.005209	3.49E-05	0.0033326	0.0034918
ElasticNet	F10, F33, F34, F44, F46, F48	0.964522	0.004925	2.68E-05	0.0029381	0.0030806
PCA	F3, F9, F11, F15, F16, F20, F49	0.804319	0.012403	0.000156	0.0084431	0.0088301
Filter method	F1, F2, F9, F11, F23, F24, F32, F36, F45, F46, F47, F48, F49	0.945869	0.006426	4.18E-05	0.0046438	0.0048758
Lasso	F10, F33, F34, F44, F46, F48	0.964522	0.004925	2.68E-05	0.0029381	0.0030806
Feature Selector	F1, F9, F11, F23, F24, F32, F46, F47, F48	0.697145	0.015427	0.000243	0.0107437	0.0112388

Supplementary Table 6. SoH estimation performances of DecisionTreeRegressor trained with full feature set before selection over 10 iterations. The mean performances are listed in Table S5.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.976443	0.004548	0.000021	0.003519	0.003665
Try 2	0.978001	0.004371	0.000019	0.003231	0.003374
Try 3	0.968004	0.004968	0.000025	0.003187	0.003342
Try 4	0.983722	0.003672	0.000013	0.002854	0.002952
Try 5	0.982453	0.003696	0.000014	0.002681	0.002791
Try 6	0.975477	0.003988	0.000016	0.002822	0.002956
Try 7	0.972816	0.004816	0.000023	0.003551	0.003749
Try 8	0.985615	0.003591	0.000013	0.002409	0.002528
Try 9	0.96242	0.00506	0.000026	0.003374	0.003514
Try 10	0.753231	0.01338	0.000179	0.005698	0.006047
Mean	0.953818	0.005209	3.49E-05	0.0033326	0.0034918

Supplementary Table 7. SoH estimation performances of DecisionTreeRegressor trained with feature set selected by ElasticNet over 10 iterations. Lasso selected the same feature set with ElasticNet. The mean performances are listed in Table S5.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.984963	0.003328	0.000011	0.002113	0.002219
Try 2	0.984619	0.003563	0.000013	0.002506	0.002594
Try 3	0.926778	0.007147	0.000051	0.003638	0.003848
Try 4	0.986764	0.003334	0.000011	0.002378	0.002461
Try 5	0.955209	0.005901	0.000035	0.002918	0.003069
Try 6	0.976617	0.004074	0.000017	0.003061	0.003198
Try 7	0.935037	0.007428	0.000055	0.003866	0.004074
Try 8	0.972795	0.004417	0.00002	0.0032	0.003364
Try 9	0.941637	0.006372	0.000041	0.00299	0.003157
Try 10	0.980804	0.003684	0.000014	0.002711	0.002822
Mean	0.964522	0.004925	2.68E-05	0.0029381	0.0030806

Supplementary Table 8. SoH estimation performances of DecisionTreeRegressor trained with feature set selected by PCA over 10 iterations. The mean performances are listed in Table S5.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.809337	0.012709	0.000162	0.00855	0.008899
Try 2	0.790563	0.012293	0.000151	0.007804	0.008173
Try 3	0.808581	0.01296	0.000168	0.008841	0.00929
Try 4	0.783039	0.013001	0.000169	0.009318	0.00981
Try 5	0.867579	0.009965	0.000099	0.00673	0.007068
Try 6	0.694078	0.015603	0.000243	0.010378	0.010825
Try 7	0.811438	0.012068	0.000146	0.008443	0.008815
Try 8	0.804356	0.012116	0.000147	0.008349	0.008677
Try 9	0.866617	0.010545	0.000111	0.007656	0.008023
Try 10	0.807605	0.012767	0.000163	0.008362	0.008721
Mean	0.804319	0.012403	0.000156	0.0084431	0.0088301

Supplementary Table 9. SoH estimation performances of DecisionTreeRegressor trained with feature set selected by Filter method over 10 iterations. The mean performances are listed in Table S5.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.950625	0.006174	0.000038	0.004104	0.0043
Try 2	0.954042	0.006555	0.000043	0.004845	0.005063
Try 3	0.921014	0.008213	0.000067	0.005325	0.005598
Try 4	0.94959	0.00582	0.000034	0.004635	0.004829
Try 5	0.958114	0.005573	0.000031	0.004351	0.004569
Try 6	0.930362	0.007025	0.000049	0.005266	0.005573
Try 7	0.948314	0.006421	0.000041	0.004347	0.004558
Try 8	0.945552	0.005989	0.000036	0.004641	0.004898
Try 9	0.934003	0.00708	0.00005	0.004984	0.005215
Try 10	0.967072	0.005407	0.000029	0.00394	0.004155
Mean	0.945869	0.006426	4.18E-05	0.0046438	0.0048758

Supplementary Table 10. SoH estimation performances of DecisionTreeRegressor trained with feature set selected by Feature Selector over 10 iterations. The mean performances are listed in Table S5.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.732014	0.014281	0.000204	0.010536	0.011079
Try 2	0.479841	0.019286	0.000372	0.014138	0.014773
Try 3	0.655977	0.017221	0.000297	0.011627	0.012184
Try 4	0.78068	0.01419	0.000201	0.010063	0.010496
Try 5	0.643511	0.017843	0.000318	0.01051	0.011017
Try 6	0.808868	0.0126	0.000159	0.009347	0.00969
Try 7	0.812737	0.013181	0.000174	0.00884	0.009263
Try 8	0.6357	0.01686	0.000284	0.01116	0.01165
Try 9	0.750157	0.013425	0.00018	0.010355	0.010812
Try 10	0.671963	0.015378	0.000236	0.010861	0.011424
Mean	0.697145	0.015427	0.000243	0.0107437	0.0112388

Supplementary Table 11. In conventional ML frameworks, regression model parameters to estimate SoH[1].

Model	Model parameter
GPR	*Kernel = 1.0*Matern(length_scale = 1.0, length_scale_bounds = (1e-1, 10.0), nu = 2.5), alpha = 0.1, optimizer = fmin_l_bfgs_b, n_restarts_optimizer = 10, normalize_y = True, copy_X_train = True, random_state = None*
NuSVR	*nu=0.5, C=1.0, kernel='rbf', degree=3, gamma='scale', coef0=0.0, shrinking=True, tol=0.001, cache_size=200, verbose=False, max_iter=-1*
RF	*n_estimators=100, *, criterion='squared_error', max_depth=None, min_samples_split=2, min_samples_leaf=1, min_weight_fraction_leaf =0.0, max_features=1.0, max_leaf_nodes=None, min_impurity_ decrease=0.0, bootstrap=True, oob_score=False, n_jobs=None, random_state=None, verbose=0, warm_start=False, ccp_alpha=0.0, max_samples=None*
KNN	*n_neighbors=10, *, weights='uniform', algorithm='auto', leaf_size=30, p=2, metric='manhattan', metric_params=None, n_jobs=None*

Supplementary Table 12. The SoH estimation performances of DeepSUGAR, GPR, NuSVR, RF, and KNN with a novel feature set selected by ElasticNet. The train and validation with pack cycling data were evaluated over 10 iterations (Table S13-S16).

Model	Data	R-squared	RMSE	MSE	MAE	MAPE
DeepSUGAR	Train/Validate with pack	0.965599	0.00531	2.84E-05	0.004119	0.004322
	Applied to module	0.930785	0.0073835	0.0000553	0.0066622	0.0069821
	PIPs	3.60543	39.049	94.7183	61.7431	61.5479
GPR	Train/Validate with pack	-0.020942	0.027987	0.000785	0.0241441	0.0253725
	Applied to module	-0.000025	0.028247	0.000798	0.024454	0.025701
	PIPs	99.88062	0.929	1.65605	1.28354	1.29471
NuSVR	Train/Validate with pack	0.8280557	0.011683	0.000138	0.0094284	0.0099374
	Applied to module	-4.57748	0.066709	0.060434	0.060434	0.06258
	PIPs	652.7986	-470.992	-43692.8	-540.978	-529.742
RF	Train/Validate with pack	0.9754125	0.004106	1.82E-05	0.0024836	0.0026171
	Applied to module	-1.007911	0.040026	0.032545	0.032545	0.033476
	PIPs	203.3318	874.817	178719	1210.4	1179.13
KNN	Train/Validate with pack	0.8687586	0.009993	0.000101	0.0078577	0.0082539
	Applied to module	-0.117051	0.029854	0.024672	0.024672	0.026257
	PIPs	113.4734	198.749	24327.7	213.985	218.116

Supplementary Table 13. The SoH estimation performance of GPR with a novel feature set selected by ElasticNet from pack cycling data over 10 iterations. The mean performances are listed in Table S12.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	-0.002693	0.028275	0.000799	0.024682	0.025781
Try 2	0.000521	0.028326	0.000802	0.024235	0.025314
Try 3	-0.040926	0.030257	0.000916	0.026876	0.02849
Try 4	0.053652	0.026455	0.0007	0.021919	0.022957
Try 5	-0.143684	0.030045	0.000903	0.02629	0.02804
Try 6	0.002341	0.027145	0.000737	0.022997	0.02433
Try 7	-0.022764	0.026907	0.000724	0.023764	0.024882
Try 8	-0.037233	0.027403	0.000751	0.023569	0.024543
Try 9	-0.033225	0.025779	0.000665	0.022047	0.022962
Try 10	0.014595	0.029273	0.000857	0.025062	0.026426
Mean	-0.0209416	0.027987	0.000785	0.0241441	0.0253725

Supplementary Table 14. The SoH estimation performance of NuSVR with a novel feature set selected by ElasticNet from pack cycling data over 10 iterations. The mean performances are listed in Table S12.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.856323	0.011031	0.000122	0.008766	0.009269
Try 2	0.806079	0.011472	0.000132	0.00905	0.009481
Try 3	0.869687	0.011668	0.000136	0.00959	0.01011
Try 4	0.793776	0.012576	0.000158	0.010384	0.010904
Try 5	0.771373	0.013044	0.00017	0.010611	0.011188
Try 6	0.844073	0.011036	0.000122	0.008985	0.009534
Try 7	0.868163	0.010098	0.000102	0.008141	0.008532
Try 8	0.791506	0.012712	0.000162	0.010201	0.010825
Try 9	0.811398	0.01256	0.000158	0.010147	0.010649
Try 10	0.868179	0.010634	0.000113	0.008409	0.008882
Mean	0.8280557	0.011683	0.000138	0.0094284	0.0099374

Supplementary Table 15. The SoH estimation performance of RF with a novel feature set selected by ElasticNet from pack cycling data over 10 iterations. The mean performances are listed in Table S12.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.971077	0.004781	0.000023	0.002813	0.002973
Try 2	0.992392	0.002684	0.000007	0.002071	0.002167
Try 3	0.985465	0.003293	0.000011	0.002602	0.002712
Try 4	0.971966	0.00453	0.000021	0.002269	0.002416
Try 5	0.976769	0.004099	0.000017	0.002599	0.002746
Try 6	0.98825	0.002967	0.000009	0.002259	0.002389
Try 7	0.939666	0.006677	0.000045	0.003063	0.003232
Try 8	0.963466	0.004889	0.000024	0.002789	0.002946
Try 9	0.985136	0.003364	0.000011	0.001939	0.002038
Try 10	0.979938	0.003777	0.000014	0.002432	0.002552
Mean	0.9754125	0.004106	1.82E-05	0.0024836	0.0026171

Supplementary Table 16. The SoH estimation performance of KNN with a novel feature set selected by ElasticNet from pack cycling data over 10 iterations. The mean performances are listed in Table S12.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.83065	0.011128	0.000124	0.008845	0.009336
Try 2	0.877697	0.009459	0.000089	0.007132	0.007436
Try 3	0.902179	0.008707	0.000076	0.007115	0.007448
Try 4	0.871336	0.010193	0.000104	0.008224	0.008643
Try 5	0.887291	0.009348	0.000087	0.00736	0.007773
Try 6	0.886631	0.009208	0.000085	0.007268	0.007682
Try 7	0.840321	0.01119	0.000125	0.008521	0.008885
Try 8	0.864671	0.009908	0.000098	0.007561	0.007871
Try 9	0.887087	0.00951	0.00009	0.007428	0.007817
Try 10	0.839723	0.011279	0.000127	0.009123	0.009648
Mean	0.8687586	0.009993	0.000101	0.0078577	0.0082539

Supplementary Table 17. The SoH estimation performance of DeepSUGAR and conventional frameworks in terms of RMSE over 10 iterations with the 6:4 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.004931	0.029348	0.009072	0.012309	0.003795
Try 2	0.00541	0.026447	0.010361	0.011252	0.003769
Try 3	0.007417	0.028661	0.010552	0.011726	0.002253
Try 4	0.004049	0.026681	0.009422	0.011648	0.006657
Try 5	0.005094	0.027943	0.010452	0.012285	0.008774
Try 6	0.004449	0.029504	0.011788	0.011743	0.003163
Try 7	0.008342	0.030404	0.010278	0.013105	0.002979
Try 8	0.004504	0.024537	0.010017	0.011238	0.00367
Try 9	0.004519	0.026066	0.010321	0.011856	0.004037
Try 10	0.004313	0.024916	0.008826	0.011945	0.002415
Mean	0.0053028	0.0274507	0.0101089	0.0119107	0.0041512

Supplementary Table 18. The SoH estimation performance of DeepSUGAR and conventional frameworks in terms of RMSE over 10 iterations with the 7:3 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.004179	0.027406	0.008812	0.010886	0.002717
Try 2	0.005041	0.028969	0.009021	0.011465	0.004254
Try 3	0.005302	0.029627	0.010843	0.011091	0.004385
Try 4	0.005704	0.028273	0.010922	0.01124	0.004221
Try 5	0.008252	0.027617	0.009585	0.011055	0.007327
Try 6	0.004887	0.02757	0.010116	0.012666	0.006305
Try 7	0.005106	0.028441	0.008561	0.012682	0.00329
Try 8	0.004298	0.029685	0.009165	0.010775	0.004027
Try 9	0.00441	0.025782	0.010094	0.011747	0.003271
Try 10	0.006394	0.029889	0.010758	0.011945	0.003449
Mean	0.0053573	0.0283259	0.0097877	0.0115552	0.0043246

Supplementary Table 19. The SoH estimation performance of DeepSUGAR and conventional frameworks in terms of RMSE over 10 iterations with the 8:2 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.005613	0.025302	0.008301	0.009928	0.004169
Try 2	0.005624	0.027367	0.009658	0.012501	0.003873
Try 3	0.004353	0.028134	0.009633	0.011376	0.003278
Try 4	0.004636	0.029148	0.009534	0.012723	0.003279
Try 5	0.004924	0.029733	0.009211	0.013863	0.004201
Try 6	0.01005	0.028555	0.010582	0.010329	0.001924
Try 7	0.004617	0.032109	0.011133	0.011154	0.00372
Try 8	0.006526	0.031119	0.008226	0.01292	0.00284
Try 9	0.003819	0.02696	0.010949	0.010946	0.002096
Try 10	0.004995	0.027681	0.009057	0.011822	0.002751
Mean	0.0055157	0.0286108	0.0096284	0.0117562	0.0032131

Supplementary Table 20. The SoH estimation performance of DeepSUGAR and conventional frameworks in terms of RMSE over 10 iterations with the 9:1 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.006895	0.028918	0.00924	0.015077	0.002939
Try 2	0.009219	0.027194	0.009132	0.010294	0.004817
Try 3	0.004703	0.023364	0.008669	0.014035	0.002253
Try 4	0.004772	0.02853	0.011522	0.010667	0.002483
Try 5	0.005793	0.029291	0.008297	0.011522	0.002534
Try 6	0.004986	0.026154	0.011771	0.012701	0.004326
Try 7	0.005002	0.029035	0.007842	0.011232	0.004203
Try 8	0.003744	0.026789	0.00908	0.011092	0.004636
Try 9	0.007153	0.028999	0.009933	0.015141	0.003116
Try 10	0.038371	0.032772	0.012869	0.012565	0.002407
Mean	0.0090638	0.0281046	0.0098355	0.0124326	0.0033714

Supplementary Table 21. The SoH estimation performance of DeepSUGAR and conventional frameworks to the module data in terms of RMSE over 10 iterations with the 6:4 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.004927	0.02831	0.029867	0.036718	0.033571
Try 2	0.011745	0.028299	0.032061	0.060335	0.046451
Try 3	0.010743	0.02825	0.031976	0.051482	0.038167
Try 4	0.010625	0.028374	0.0302	0.078772	0.04254
Try 5	0.008476	0.028415	0.03058	0.0533	0.029966
Try 6	0.129931	0.028247	0.03302	0.058248	0.034439
Try 7	0.006358	0.028263	0.028976	0.051782	0.037662
Try 8	0.011695	0.028303	0.02864	0.054958	0.035214
Try 9	0.009904	0.028262	0.030278	0.050539	0.042935
Try 10	0.0069	0.028278	0.028508	0.05538	0.037835
Mean	0.0211304	0.0283001	0.0304106	0.0551514	0.037878

Supplementary Table 22. The SoH estimation performance of DeepSUGAR and conventional frameworks to the module data in terms of RMSE over 10 iterations with the 7:3 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.094251	0.028257	0.03311	0.063209	0.037473
Try 2	0.007862	0.028262	0.032346	0.069352	0.040969
Try 3	0.005809	0.028251	0.030288	0.062962	0.031306
Try 4	0.012046	0.028303	0.029031	0.062231	0.047064
Try 5	0.015864	0.028257	0.031023	0.065919	0.047523
Try 6	0.008698	0.028255	0.032084	0.062181	0.036072
Try 7	0.007999	0.028248	0.032716	0.076055	0.044348
Try 8	0.010457	0.028345	0.029231	0.042248	0.045104
Try 9	0.010954	0.02828	0.030503	0.067193	0.031388
Try 10	0.008645	0.028289	0.027949	0.046687	0.040797
Mean	0.0182585	0.0282747	0.0308281	0.0618037	0.0402044

Supplementary Table 23. The SoH estimation performance of DeepSUGAR and conventional frameworks to the module data in terms of RMSE over 10 iterations with the 8:2 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.010968	0.028264	0.028893	0.066909	0.037914
Try 2	0.006756	0.028251	0.029853	0.059117	0.031052
Try 3	0.009008	0.028291	0.032215	0.04724	0.045008
Try 4	0.009606	0.028248	0.03306	0.073241	0.043706
Try 5	0.01434	0.028247	0.030869	0.062554	0.031486
Try 6	0.004982	0.028247	0.030312	0.076698	0.034529
Try 7	0.010063	0.028313	0.031287	0.045938	0.03708
Try 8	0.005182	0.028254	0.028515	0.061255	0.031294
Try 9	0.008679	0.028308	0.02872	0.04623	0.032662
Try 10	0.009764	0.028299	0.029207	0.068254	0.037704
Mean	0.0089348	0.0282722	0.0302931	0.0607436	0.0362435

Supplementary Table 24. The SoH estimation performance of DeepSUGAR and conventional frameworks to the module data in terms of RMSE over 10 iterations with the 9:1 of training-test data ratio. The mean performances are shown in Supplementary Fig. 5.

	DeepSUGAR	GPR	KNN	NuSVR	RF
Try 1	0.004571	0.02825	0.029842	0.041394	0.039589
Try 2	0.007822	0.028252	0.029652	0.049059	0.036221
Try 3	0.005407	0.028247	0.030151	0.046275	0.034885
Try 4	0.010774	0.028259	0.029004	0.051005	0.03612
Try 5	0.01683	0.028248	0.030015	0.055483	0.038893
Try 6	0.036937	0.028251	0.030029	0.068411	0.040231
Try 7	0.008221	0.028247	0.030175	0.067483	0.033962
Try 8	0.008777	0.028248	0.031704	0.049089	0.036172
Try 9	0.012908	0.02825	0.029337	0.06114	0.04608
Try 10	0.010093	0.028255	0.030738	0.04759	0.03903
Mean	0.012234	0.0282507	0.0300647	0.0536929	0.0381183

Supplementary Table 25. The SoH estimation performance of interpolated cDCGAN over 10 iterations.

	R-squared	RMSE	MSE	MAE	MAPE
Try 1	0.896593	0.008527	0.006975	0.006975	0.007377
Try 2	0.888394	0.008859	0.007452	0.007452	0.007843
Try 3	0.938082	0.006598	0.005137	0.005137	0.005391
Try 4	0.940197	0.006485	0.005205	0.005205	0.005469
Try 5	0.862419	0.009836	0.008181	0.008181	0.008539
Try 6	0.936525	0.006681	0.005439	0.005439	0.005695
Try 7	0.893555	0.008652	0.006448	0.006448	0.006805
Try 8	0.893605	0.008649	0.007189	0.007189	0.007574
Try 9	0.871583	0.009503	0.007208	0.007208	0.007592
Try 10	0.858235	0.009984	0.008204	0.008204	0.008627
Mean	0.897919	0.008377	0.006744	0.006744	0.007091

Supplementary Table 26. Diagnosis processes and assumptions for time requirement.

Level	Procedure	Base	DeepSUGAR	Unit
Pack	**Inspection & handling**	5	5	min
	Connection of the electrical test equipment	5	5	
	Disconnection from electrical test equipment	5	5	
	Final inspection	5	5	
	Initial voltage set & balance **Battery characterization**	420	420	
Module	**Disassembly**	10	-	
	Inspection & handling	5	-	
	Connection of the electrical test equipment	5	-	
	Disconnection from electrical test equipment	5	-	
	Final inspection	5	-	
	Initial voltage set & balance **Battery characterization**	420	-	

Supplementary Table 27. Detailed results of comparison between the conventional and proposed processing methods.

Indicator	Base	DeepSUGAR	Unit
Annual processing capacity	9,479	19,173	kWh y^{-1}
Annual power consumption	5,326,588	355,106	kWh y^{-1}
[a]CAPEX	1,135,889	725,222	USD
[b]OPEX	584,701	93,436	USD y^{-1}
Processing cost	77.2	9.7	USD kWh^{-1}
Annual carbon dioxide emissions	2,530,129	168,675	kgCO_2-eq y^{-1}

[a]CAPEX: Capital expenditure

[b]OPEX: Operating expenditure

Supplementary Table 28. Information about labor cost and electricity price for countries in the case study and the result of economic performance.

<table>
<tr><th>Information</th><th>Unit</th><th>[a]KOR</th><th>[b]CHN</th><th>[c]USA</th><th>[d]DEU</th><th>[e]FRA</th><th>[f]GBR</th><th>[g]CAN</th><th>Reference</th></tr>
<tr><td>[1]Labor cost</td><td>USD h^{-1}</td><td>16.32</td><td>6.50</td><td>24.82</td><td>47.63</td><td>45.35</td><td>29.61</td><td>33.07</td><td>Website[2-5]</td></tr>
<tr><td>[2]Electricity price</td><td>USD kWh^{-1}</td><td>0.096</td><td>0.086</td><td>0.156</td><td>0.555</td><td>0.262</td><td>0.330</td><td>0.105</td><td>Website[6]</td></tr>
<tr><td>Processing cost [Base]</td><td rowspan="2">USD kWh^{-1}</td><td>77.2</td><td>66.8</td><td>115.0</td><td>350.0</td><td>184.6</td><td>215.1</td><td>90.4</td><td rowspan="2">Own calculation</td></tr>
<tr><td>Processing cost [Deep SUGAR]</td><td>9.7</td><td>7.7</td><td>12.4</td><td>24.1</td><td>18.3</td><td>16.6</td><td>13.0</td></tr>
</table>

[1]Hourly labor cost per employee for manufacturing field

[2]Average electricity price in 2021 (December) and 2022 (September) in business sector

[a]KOR: Republic of Korea

[b]CHN: China

[c]USA: United States of America

[d]DEU: Germany

[e]FRA: France

[f]GBR: United Kingdom

[g]CAN: Canada

Supplementary Table 29. Parameters used in economic and carbon footprint analysis.

Indicator	Base	Unit
Annual operating hours	8,000	h
Number of pack cycle tester	10	-
Installation period	1	y
Discount rate	10	%
BoP cost ratio	10	%
Maintenance cost	1	%
[1]Electricity price	0.096	USD kWh^{-1}
[2]Carbon intensity	0.475	$kgCO_2$-eq kWh^{-1}

[1]Average electricity price in the Republic of Korea from 2021 December to 2022 September in business sector[7].

[2]Average carbon intensity in the Republic of Korea from 2000 to 2019[8].

Supplementary Table 30. Parameters used in sensitivity analysis.

Parameter	Minimum	Base	Maximum	Unit
Number of cycle testers	1	10	20	-
Energy efficiency of cycle tester	80.0	85.0	90.0	%
14S7P cycle tester cost	58,100	83,000	107,900	USD
7P cycle tester cost	32,900	47,000	61,100	USD
Hourly labor cost	8.16	16.32	24.48	USD h^{-1}
Electricity price	0.05	0.10	0.14	USD kWh^{-1}
Discount rate	7.5	10.0	12.5	%

Nomenclature

SoH	State-of-Health
GPR	Gaussian Process Regressor
DT	Decision Tree
CAPEX	Capital expenditure
OPEX	Operating expenditure
KOR	Republic of Korea
CHN	China
USA	United States of America
DEU	Germany
FRA	France
GBR	United Kingdom
CAN	Canada

References

1. G. V. Fabian Pedregosa, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, Jake Vanderplas, Alexandre Passos, David Cournapeau, Matthieu Brucher, Matthieu Perrot, Édouard Duchesnay, *Scikit-learn: Machine Learning in Python*, JMLR, 2011.
2. Hourly labor cost in manufacturing in KOR, http://datakorea.datastore.or.kr/profile/industry/C/, (accessed May 1st, 2023).
3. KOR, http://datakorea.datastore.or.kr/profile/industry/C/, (accessed May 1st, 2023).
4. CHN, DEU, FRA, GBR, CAN, https://ilostat.ilo.org/topics/labour-costs/, (accessed May 1st, 2023).
5. USA, https://tradingeconomics.com/united-states/wages-in-manufacturing, (accessed May 1st, 2023).
6. Electriciy price, https://www.globalpetrolprices.com/electricity_prices/, (accessed May 1st, 2023).
7. electricity price of KOR, https://www.globalpetrolprices.com/electricity_prices/, (accessed May 1st, 2023).
8. Carbon intensity of KOR, https://ourworldindata.org/grapher/carbon-intensity-electricity?time=2000, (accessed May 1st, 2023).