Robust Battery Lifetime Prediction with Noisy Measurements via Total-Least-Squares Regression

Ting Lu[a], Xiaoang Zhai[a], Sihui Chen[b], Yang Liu[c,*], Jiayu Wan[d,*], Guohua Liu[a], Xin Li[c,*]
[a] *Donghua University*, *Shanghai, China*
[b] *Southern University of Science and Technology*, *Shenzhen*, *Guangdong*, *China*
[c] *Duke Kunshan University*, *Kunshan*, *Jiangsu*, *China*
[d] *Shanghai Jiao Tong University*, *Shanghai*, *China*

***Abstract*—Machine learning technologies have gained significant popularity in rechargeable battery research in recent years, and have been extensively adopted to construct data-driven solutions to tackle multiple challenges for energy storage in embedded computing systems. An important application in this area is the machine learning-based battery lifetime prediction, which formulates regression models to estimate the remaining lifetimes of batteries given the measurement data collected from the testing process. Due to the non-idealities in practical operations, these measurements are usually impacted by various types of interference, thereby involving noise on both input variables and regression labels. Therefore, existing works that focus solely on minimizing the regression error on the labels cannot adequately adapt to the practical scenarios with noisy variables. To address this issue, this study adopts total least squares (TLS) to construct a regression model that achieves superior regression accuracy by simultaneously optimizing the estimation of both variables and labels. Furthermore, due to the expensive cost for collecting battery cycling data, the number of labeled data samples used for predictive modeling is often limited. It, in turn, can easily lead to overfitting, especially for TLS, which has a relatively larger set of problem unknowns to solve. To tackle this difficulty, the TLS method is investigated conjoined with stepwise feature selection in this work. Our numerical experiments based on public datasets for commercial Lithium-Ion batteries demonstrate that the proposed method can effectively reduce the modeling error by up to 11.95%, compared against the classic baselines with consideration of noisy measurements.**

***Index Terms*—Battery Lifetime Prediction, Total Least Squares, Noisy Measurements, Stepwise Feature Selection, Energy Storage**

1. Introduction

Rechargeable battery technology is gaining rapidly increasing attentions in the modern world, as sustainability is becoming a critical issue of global interests. As a popular class of rechargeable battery, Lithium-Ion battery has been widely deployed due to its advantages such as high energy density and low cost, which enable reliable and economic energy storage in multiple embedded computing system-based applications such as portable devices and electric vehicles [1]-[3]. The quality of lithium-ion batteries can be quantitatively evaluated through their lifetime. Especially, accurate prediction of battery lifetime using early cycle data will greatly reduce the evaluation cost. Therefore, the early prediction of battery lifetime has been extensively researched in the literature, providing strategic guidance for the manufacturing, maintenance, and use of batteries [1], [4]-[5].

A variety of machine learning techniques have been recently adopted to predict battery lifetime with high accuracy and low cost [1]. These approaches aim to train a regression model that takes the battery measurements as its inputs to predict the corresponding lifetime value. For example, a significant progress has been made in [1] by identifying a set of important features from the early-cycle battery discharge curves, and these features are strongly correlated with the battery lifetime. This seminal work enables battery lifetime prediction at an early stage and is of practical utility in real-world applications. Furthermore, various regression modeling algorithms, such as elastic net [1], Gaussian process regression [6], support vector regression [7], XGBoost [7] and deep neural networks [8]-[9], have been proposed to predict battery lifetime in practice.

As a common practice in regression modeling, most conventional approaches attempt to minimize the prediction error for battery lifetime when a regression model is trained. However, the input features (e.g., voltage, current, resistance, temperature, etc.) are often noisy because they are measured through probe techniques with measurement errors [10]. In addition, when the measured physical metrics are used to derive new features for battery lifetime prediction, the measurement errors on these physical metrics may be further propagated to the derived features [10]-[11]. Consider the discharge capacity curve as an example. While it is computed by integrating the current over the testing period, it suffers from the cumulative error posed by both time-domain sampling and current measurement. These measurement errors associated with input features have not been appropriately taken into account by the conventional regression modeling methods.

To address the aforementioned challenges, this paper proposes to use total least squares (TLS) regression method to predict the lifetime of batteries. Unlike the conventional approaches, TLS minimizes the total errors for both the input features and the predicted outcomes [12]-[13]. In other words, it is able to learn an accurate regression model with

Authors' Addresses: Ting Lu, Xiaoang Zhai and Guohua Liu are with Donghua University, Shanghai, China, 200051; emails: luting@dhu.edu.cn; xiaoang_zhai@mail.dhu.edu.cn; ghliu@dhu.edu.cn; Sihui Chen is with Southern University of Science and Technology, Shenzhen, Guangdong, China, 518055; email: 12232304@mail.sustech.edu.cn; Yang Liu and Xin Li are with Duke Kunshan University, Kunshan, Jiangsu, China, 215316; emails: yang.liu2@dukekunshan.edu.cn; xinli.ece@dukekunshan.edu.cn; Jiayu Wan is with Shanghai Jiao Tong University, Shanghai, China, 200240; email: wanjy@sjtu.edu.cn.
*Yang Liu, Jiayu Wan and Xin Li are the corresponding authors.

consideration of noisy features. Due to this advantage, TLS has been broadly used in many practical applications with an errors-in-variables (EIV) nature, such as image reconstruction and financial data forecast [13].

It is important to note that the battery testing data till the end-of-life are usually expensive to obtain, and hence, battery lifetime prediction models are often trained using a limited number of data samples. It, in turn, can potentially lead to overfitting and jeopardize the generalization ability, especially for the TLS method, which has more problem unknowns to solve than ordinary least squares (OLS) regression [14]. In order to tackle this issue, we prefer to construct the prediction model based on a subset of important features, thereby improving modeling efficiency and prediction accuracy, while avoiding overfitting [15]-[17]. Additionally, identifying the important features can also help understand the degradation mechanisms of Lithium-Ion batteries from a data-driven perspective and reduce the measurement costs induced by the less important features [18]-[19].

In practice, feature selection can be accomplished through involving regularization terms (e.g., L1-norm, L2-norm or L0-norm) in the loss function to compress the magnitudes of the model coefficients corresponding to less important features [20]-[22]. While the model parameters considering L1- and L2-norm regularizations can be efficiently solved using various optimization algorithms, L0-norm regularization cannot be easily implemented due to its NP-hard nature [23]. Therefore, a sub-optimal solution is usually pursued by removing least informative features (or preserving most informative features) based on a quantified criterion of their importance. This type of methods can be constructed by using greedy strategies, such as stepwise feature selection [24] and orthogonal matching pursuit (OMP) [25], or carefully-designed searching structures, such as the tree-based approach in [26]. Note that the criterion for feature ranking is closely related to the specific modeling algorithm. While the linear-regression-based algorithms, such as OMP, identify the important features based on their linear correlations with the prediction target, the important features for decision-tree-based algorithms (e.g., XGBoost) can be computed by the number of splits applied to them in the decision tree structure [27]. Inspired by the aforementioned works in the literature, stepwise feature selection is investigated in this paper to automatically discover a subset of critical features for TLS, thereby improving the accuracy and robustness of prediction.

Numerical experiments have been comprehensively conducted on two publicly available datasets derived from commercial lithium-ion batteries [1], [28], where additive zero-mean Gaussian noise is deployed to emulate the noisy measurements in practical scenarios. The proposed TLS method is compared to multiple representative baselines to demonstrate its superior prediction accuracy, especially when the measurement noises are large, while robustly identifying the critical features.

The remainder of this paper is organized as follows. In Section 2, the relevant technical background is briefly reviewed. In Section 3, the mathematical formulations for TLS are presented in the context of battery lifetime prediction, and the implementation details are further discussed in Section 4. In Section 5, our experimental results are analyzed to demonstrate the efficacy of TLS. Finally, we conclude in Section 6.

2. Background

Considering a set of features $\mathbf{x} = [x_1\ x_2\ \cdots\ x_N]^T$, where N denotes the total number of features and the superscript T represents the transpose of a vector or matrix, our objective is to learn a mapping from $\mathbf{x}$ to the battery lifetime y [23],[29]:

$$y = \sum_{m=1}^{M} w_m \cdot g_m(\boldsymbol{x}) - \varepsilon_y, \tag{1}$$

where $g_m(\mathbf{x})$ is the m-th basis function, w_m denotes the m-th model coefficient, ε_i stands for the modeling error following a zero-mean Gaussian distribution $N(0, \sigma_y^2)$, and M represents the total number of basis functions.

Given a set of data samples $\{(\mathbf{x}_k, y_k);\ k = 1, \cdots, K\}$, the model coefficients $\{w_m;\ m = 1, \cdots, M\}$ may be determined by minimizing the total squared error [14]:

$$\min_{\mathbf{w}} \quad \|\mathbf{y} - \mathbf{G} \cdot \mathbf{w}\|_2^2, \tag{2}$$

where $\|\bullet\|_2$ denotes the L_2 norm of a vector and

$$\mathbf{G} = \begin{bmatrix} g_1(\mathbf{x}_1) & g_2(\mathbf{x}_1) & \cdots & g_M(\mathbf{x}_1) \\ g_1(\mathbf{x}_2) & g_2(\mathbf{x}_2) & \cdots & g_M(\mathbf{x}_2) \\ \vdots & \vdots & \vdots & \vdots \\ g_1(\mathbf{x}_K) & g_2(\mathbf{x}_K) & \cdots & g_M(\mathbf{x}_K) \end{bmatrix} \tag{3}$$

$$\mathbf{w} = [w_1 \quad w_2 \quad \cdots \quad w_M]^T \tag{4}$$

$$\mathbf{y} = [y_1 \quad y_2 \quad \cdots \quad y_K]^T. \tag{5}$$

The aforementioned approach is referred to as OLS regression in the literature and it aims to find the maximum-likelihood solution $\mathbf{w}$ for the unknown model coefficients [14].

In practice, each basis function $g_m(\mathbf{x})$ may be noisy, as the elements in feature vector $\mathbf{x}$ is obtained through the physical measurements of batteries (e.g., voltage, current, temperature, etc.), which are usually associated with measurement errors. In this case, Eq. (1) should be re-written as:

$$y + \varepsilon_y = \sum_{m=1}^{M} w_m \cdot \left[g_m(\mathbf{x}) + \varepsilon_{g,m}\right], \tag{6}$$

where $\varepsilon_{g,m}$ represents the measurement error associated with the m-th basis function $g_m(\mathbf{x})$. In this paper, we assume that $\varepsilon_{g,m}$ follows a zero-mean Gaussian distribution $N(0, \sigma_{g,m}^2)$. In this scenario, the prediction of OLS becomes less reliable as it ignores the errors of the input features. In contrast, TLS can generate a robust linear regression model that accommodates the noises on both the prediction target y and the input basis functions $\{g_m(\mathbf{x});\ m = 1, \cdots, M\}$.

3. Proposed solution

In order to compute the solution of the unknown model coefficients in (6), we formulate an optimization problem of $\mathbf{w}$ based on maximum-likelihood estimation. To achieve this goal, we make two further assumptions. First, both ε_y and $\{\varepsilon_{g,m};\ m = 1, \cdots, M\}$ can be normalized to standard Gaussian distribution $N(0, 1)$ by appropriately scaling y and $\{g_m(\mathbf{x});\ m = 1, \cdots, M\}$ respectively. Second, ε_y and $\{\varepsilon_{g,m};\ m = 1, \cdots, M\}$ are statistically independent.

With these two assumptions, it is straightforward to show that the likelihood of observing a sample $(\mathbf{x}_k, y_k)$ is equal to:

$$p(\mathbf{x}_k|y_k) = \frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\varepsilon_{y,k}^2}{2}\right) \times \prod_{m=1}^{M}\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{\varepsilon_{g,m,k}^2}{2}\right) = \frac{1}{(\sqrt{2\pi})^{M+1}}\exp\left[-\frac{\varepsilon_{y,k}^2}{2} - \sum_{m=1}^{M}\frac{\varepsilon_{g,m,k}^2}{2}\right], \quad (7)$$

where $\varepsilon_{y,k}$ and $\varepsilon_{g,m,k}$ denote the k-th samples for ε_y and $\varepsilon_{g,m}$ respectively. Furthermore, by assuming that all samples in the dataset $\{(\mathbf{x}_k, y_k);\ k = 1, \cdots, K\}$ are statistically independent, the likelihood for observing these K samples is equal to:

$$\prod_{k=1}^{K} p(\mathbf{x}_k, y_k) \propto \prod_{k=1}^{K}\exp\left[-\frac{\varepsilon_{y,k}^2}{2} - \sum_{m=1}^{M}\frac{\varepsilon_{g,m,k}^2}{2}\right] = \exp\left[-\sum_{k=1}^{K}\frac{\varepsilon_{y,k}^2}{2} - \sum_{k=1}^{K}\sum_{m=1}^{M}\frac{\varepsilon_{g,m,k}^2}{2}\right]. \quad (8)$$

Hence, the maximum-likelihood solution $\mathbf{w}$ can be found by solving the following optimization problem:

$$\min_{\boldsymbol{\varepsilon},\mathbf{E},\mathbf{w}} \quad \|\boldsymbol{\varepsilon}\|_2^2 + \|\mathbf{E}\|_F^2 \quad \text{s.t.} \quad \mathbf{y} + \boldsymbol{\varepsilon} = (\mathbf{G} + \mathbf{E}) \cdot \mathbf{w}, \quad (9)$$

where $\|\bullet\|_F$ denotes the Frobenius norm of a matrix and

$$\boldsymbol{\varepsilon} = [\varepsilon_{y,1} \quad \varepsilon_{y,2} \quad \cdots \quad \varepsilon_{y,K}]^T \quad (10)$$

$$\mathbf{E} = \begin{bmatrix} \varepsilon_{g,1,1} & \varepsilon_{g,2,1} & \cdots & \varepsilon_{g,M,1} \\ \varepsilon_{g,1,2} & \varepsilon_{g,2,2} & \cdots & \varepsilon_{g,M,2} \\ \vdots & \vdots & \ddots & \vdots \\ \varepsilon_{g,1,K} & \varepsilon_{g,2,K} & \cdots & \varepsilon_{g,M,K} \end{bmatrix}. \quad (11)$$

Note that minimizing the cost function in (9) is equivalent to maximizing the likelihood in (8). Such an approach is referred to as the TLS regression in the literature [12].

The optimization problem in (9) is non-convex due to its quadratic equality constraints [30]. Hence, it is difficult to find the optimal solution both efficiently (i.e., with low computational cost) and robustly (i.e., with guaranteed global convergence). To mitigate this issue, we transform the original non-convex problem to an equivalent version that can be easily solved.

First, we note that Eq. (9) is reduced to the following convex optimization problem if the vector $\mathbf{w}$ is known:

$$\min_{\boldsymbol{\varepsilon},\mathbf{E}} \quad \|\boldsymbol{\varepsilon}\|_2^2 + \|\mathbf{E}\|_F^2 \quad \text{s.t.} \quad \mathbf{y} + \boldsymbol{\varepsilon} = (\mathbf{G} + \mathbf{E}) \cdot \mathbf{w}, \quad (12)$$

where ε and E are the problem unknowns. Eq. (12) minimizes a convex quadratic cost function subject to a set of linear equality constraints. Its optimal solution can be analytically derived by using the KKT condition [30]. For this purpose, we calculate the Lagrangian of (12):

$$L(\boldsymbol{\varepsilon}, \mathbf{E}, \boldsymbol{\lambda}) = \|\boldsymbol{\varepsilon}\|_2^2 + \|\mathbf{E}\|_F^2 + \boldsymbol{\lambda}^T[(\mathbf{G} + \mathbf{E}) \cdot \mathbf{w} - \mathbf{y} - \boldsymbol{\varepsilon}], \quad (13)$$

where $\boldsymbol{\lambda}$ is a vector containing K Lagrange multipliers. By taking the partial derivatives of $L(\boldsymbol{\varepsilon}, \mathbf{E}, \boldsymbol{\lambda})$ with respect to $\boldsymbol{\varepsilon}$ and $\mathbf{E}$ and setting them to zero, we have:

$$2 \cdot \boldsymbol{\varepsilon} - \boldsymbol{\lambda} = \mathbf{0} \quad (14)$$

$$2 \cdot \mathbf{E} + \boldsymbol{\lambda} \cdot \mathbf{w}^T = \mathbf{0}. \quad (15)$$

The optimal solutions of $\boldsymbol{\varepsilon}$ and $\mathbf{E}$ must satisfy (14)-(15) and the linear equality constraints:

$$\mathbf{y} + \boldsymbol{\varepsilon} = (\mathbf{G} + \mathbf{E}) \cdot \mathbf{w}. \quad (16)$$

Solving the linear system of (14)-(16) yields:

$$\boldsymbol{\varepsilon} = \frac{\mathbf{G}\cdot\mathbf{w} - \mathbf{y}}{\|\mathbf{w}\|_2^2 + 1} \quad (17)$$

$$\mathbf{E} = -\frac{(\mathbf{G}\cdot\mathbf{w} - \mathbf{y})\cdot\mathbf{w}^T}{\|\mathbf{w}\|_2^2 + 1}. \quad (18)$$

Substituting (17)-(18) into (9), the cost function of TLS regression can be expressed as:

$$\|\boldsymbol{\varepsilon}\|_2^2 + \|\mathbf{E}\|_F^2 = \frac{\|\mathbf{G}\cdot\mathbf{w} - \mathbf{y}\|_2^2}{\|\mathbf{w}\|_2^2 + 1}. \quad (13)$$

Note that the cost function in (13) has been minimized over $\boldsymbol{\varepsilon}$ and $\mathbf{E}$, and it is now a non-convex function of $\mathbf{w}$.

To efficiently minimize the cost function in (13) over $\mathbf{w}$, we construct the following optimization problem [30]:

$$\min_{\mathbf{w},t} \quad \frac{\|\mathbf{G}\cdot\mathbf{w} - t\cdot\mathbf{y}\|_2^2}{\|\mathbf{w}\|_2^2 + t^2} \quad \text{s.t.} \quad t = 1. \quad (20)$$

It is equivalent to:

$$\min_{\boldsymbol{\theta}} \quad \frac{\boldsymbol{\theta}^T\cdot\mathbf{B}\cdot\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2^2} \quad \text{s.t.} \quad \theta_{M+1} = 1, \quad (21)$$

where

$$\mathbf{B} = \begin{bmatrix} \mathbf{G}^T\mathbf{G} & -\mathbf{G}^T\mathbf{y} \\ -\mathbf{y}^T\mathbf{G} & \mathbf{y}^T\mathbf{y} \end{bmatrix} \quad (22)$$

$$\boldsymbol{\theta} = \begin{bmatrix} \mathbf{w} \\ t \end{bmatrix}, \quad (23)$$

and θ_{M+1} is the (M+1)-th element of $\boldsymbol{\theta}$. Studying (21), we notice that its cost function is independent of the magnitude of $\boldsymbol{\theta}$. Hence, solving the following optimization problem without the constraint $\theta_{M+1} = 1$ yields the same optimal cost function value:

$$\min_{\boldsymbol{\theta}} \quad \frac{\boldsymbol{\theta}^T\cdot\mathbf{B}\cdot\boldsymbol{\theta}}{\|\boldsymbol{\theta}\|_2^2} \quad \text{s.t.} \quad \boldsymbol{\theta} \neq \mathbf{0}. \quad (24)$$

Eq. (24) represents the well-known problem of Rayleigh quotient minimization [12]. Its optimal solution $\boldsymbol{\theta}^*$ is the eigenvector corresponding to the least eigenvalue of $\mathbf{B}$. Once $\boldsymbol{\theta}^*$ is known, we can scale $\boldsymbol{\theta}^*$ by its (M+1)-th element θ^*_{M+1} to determine the optimal solution $\boldsymbol{\theta}$ for (21):

$$\boldsymbol{\theta} = \frac{\boldsymbol{\theta}^*}{\theta^*_{M+1}}. \quad (25)$$

Thus, the optimal solution $\mathbf{w}$ for unknown model coefficients can be obtained by taking the first M elements of $\boldsymbol{\theta}$:

$$\mathbf{w} = [\theta_1 \quad \theta_2 \quad \cdots \quad \theta_M]^T. \quad (26)$$

The major steps for TLS are summarized in Algorithm 1.

Algorithm 1: Total-Least-Squares (TLS) Regression

1. Start from a set of basis functions $\{g_m(\mathbf{x});\ m = 1, \cdots, M\}$ and a set of samples $\{(\mathbf{x}_k, y_k);\ k = 1, \cdots, K\}$.
2. Calculate the matrix $\mathbf{G}$ in (3) and the vector $\mathbf{y}$ in (5).
3. Construct the matrix $\mathbf{B}$ based on (22).
4. Compute the eigenvector $\boldsymbol{\theta}^*$ of $\mathbf{B}$ corresponding to its least eigenvalue.
5. Calculate the vector $\boldsymbol{\theta}$ in (25) by normalizing $\boldsymbol{\theta}^*$.
6. Compute the vector $\mathbf{w}$ in (26), containing the optimal model coefficients.

4. Implementation details

In this section, we discuss several implementation issues, including (i) stepwise feature selection, and (ii) cross-validation. Finally, the complete algorithm for TLS with stepwise feature selection is summarized.

4.1 Stepwise feature selection

Through observing the optimization problem in (9), we notice that TLS has more problem unknowns to solve than OLS, which may result in overfitting if the amount of training

data is insufficient. In order to address this issue, we should appropriately identify the most informative features and use them for regression modeling. To implement this goal, we adopt the L0-norm regularization from the literature [23], [26], which leads to a sparse solution by directly constraining the number of non-zero model coefficients. As a result, the features associated with zero model coefficients are ignored.

Considering the L0-norm regularization, the optimization problem for TLS in (9) can be re-written as:

$$\begin{aligned} \min_{\boldsymbol{\varepsilon},\mathbf{E},\mathbf{w}} \quad & \|\boldsymbol{\varepsilon}\|_2^2 + \|\mathbf{E}\|_F^2 \\ \text{s.t.} \quad & \mathbf{y} + \boldsymbol{\varepsilon} = (\mathbf{G} + \mathbf{E}) \cdot \mathbf{w}, \\ & \|\mathbf{w}\|_0 \le \eta \end{aligned} \tag{27}$$

where $\|\bullet\|_0$ denotes the L0-norm of a vector (i.e., the number of non-zero elements in the vector). The hyper-parameter η represents the maximum number of non-zero elements in $\mathbf{w}$. It can be determined via cross-validation, as shown in Section 0. The optimization problem in (27) is NP-hard [23], which poses challenges on obtaining the globally optimal solution in a highly efficient manner.

In order to find a reasonably good solution for the optimization problem in (27), a stepwise feature selection strategy is adopted to iteratively select the most informative features. Suppose that we need to select η features in total. We initialize the subset Ω of selected features as an empty set Ø (i.e., Ω = Ø). The first most informative feature is selected among all candidate features to minimize the root mean-squared error (RMSE) on the training dataset. Let Φ denote the set of all candidate features. Once the first feature is determined, it is removed from the set Φ and added to the set Ω. Next, the second most important feature is selected among all remaining features in Φ to minimize the RMSE in combination with all selected features in Ω. The aforementioned procedure is repeated until $|\Omega| = \eta$, where $|\bullet|$ denotes the cardinality of a set (i.e., the number of elements in the set). The major steps of TLS with stepwise feature selection are summarized in Algorithm 2.

Algorithm 2: Total Least Squares (TLS) Regression with Stepwise Feature Selection

1. Start from a set of basis functions $\{g_m(\mathbf{x}); m = 1, \cdots, M\}$, a set of training samples $\{(\mathbf{x}_k, y_k); k = 1, \cdots, K\}$, and a given value of η.
2. Initialize the set of selected features as Ω = Ø, and the candidate feature set $\Phi = \{g_m(\mathbf{x}); m = 1, \cdots, M\}$.
3. **for** $m = 1, 2, \cdots, \eta$
4. **for** each $g_i(\mathbf{x})$ in Φ
5. Apply Algorithm 1 to create a TLS model including all features in Ω and $g_i(\mathbf{x})$.
6. Compute the RMSE on the training dataset as $r(g_i)$.
7. **endfor**
8. Select the feature $g^*(\mathbf{x})$ with the minimum RMSE among $\{r(g_i); g_i(\mathbf{x}) \in \Phi\}$.
9. Set $\Omega = \Omega \cup \{g^*(\mathbf{x})\}$ and $\Phi = \Phi \setminus \{g^*(\mathbf{x})\}$.
10. **endfor**
11. Apply Algorithm 1 to create the TLS model including all features in Ω.

4.2 Cross-validation

The number of selected features (i.e., η in (27)) is determined using the cross-validation method, where the training dataset is partitioned into L subsets, such that L-1 subsets are used for model training and the remaining subset is used to evaluate the modeling error. The aforementioned train-test split is repeated for L times, where different testing subsets are adopted for different runs. The appropriate value of η is determined by minimizing the total cross-validation error estimated from all L runs. In this work, leave-one-out cross-validation is adopted, which means that L is equal to the number of training data samples. The major steps of the leave-one-out cross-validation method are summarized in Algorithm 3.

Algorithm 3: Leave-One-Out Cross-Validation

1. Start from a set of basis functions $\{g_m(\mathbf{x}); m = 1, \cdots, M\}$ and a set of samples $\{(\mathbf{x}_k, y_k); k = 1, \cdots, K\}$.
2. **for** $k = 1, 2, \cdots, K$
3. **for** $\eta = 1, 2, \cdots, H$
4. Apply Algorithm 2 to train the TLS model where the k-th sample is used to evaluate the mean squared error (MSE) $e(k, \eta)$ and all other samples are used for model training.
5. **endfor**
6. **endfor**
7. **for** $\eta = 1, 2, \cdots, H$
8. Compute the overall RMSE as: $\text{RMSE}_\eta = \sqrt{\frac{\sum_k e(k,\eta)}{K}}$.
9. **endfor**
10. Determine the appropriate value of η: $\eta^* = \arg\min_\eta \text{RMSE}_\eta$.

4.3 Summary

The complete algorithm flow for TLS regression with L0-norm regularization, including leave-one-out cross-validation and stepwise feature selection, are summarized in Algorithm 4. In step 1-2, it receives and preprocesses the training data. In step 3, it applies Algorithm 3 to determine the number of features to be selected. In step 4, it applies Algorithm 2 to select the features for modeling and compute the vector of model coefficients.

Algorithm 4: Total Least Squares (TLS) Regression with L0-Norm Regularization

11. Start from a set of basis functions $\{g_m(\mathbf{x}); m = 1, \cdots, M\}$ and a set of samples $\{(\mathbf{x}_k, y_k); k = 1, \cdots, K\}$.
12. Apply Algorithm 3 to determine the appropriate value of η.
13. Apply Algorithm 2 to create the TLS model.

5. Numerical results

5.1 Experimental setup

In this section, two public datasets based on commercial Lithium-Ion batteries are used to demonstrate the efficacy of our proposed TLS method for battery lifetime prediction [1], [28]. The first dataset, referred to as "Dataset 1", was

generated by Peter M. Attia et al. [28], which is composed of the measurement data of 233 commercial LiFePO4/graphite batteries. In our experiments, we only use the measurement data extracted from 45 batteries that have been tested until the end-of-life (EOL) cycle (80% of the nominal capacity). The second dataset, referred to as "Dataset 2", was generated by Severson et al. [1], which consists of 124 commercial LiFePO4/graphite batteries cycled to EOL under fast-charging conditions. During the cycling test of these batteries, several important metrics, such as voltage, current, discharge capacity, temperature, impedance, charge time, etc., are measured in real time. The batteries in Dataset 1 differ from Dataset 2 as they were tested using a more complex fast-charging protocol and the internal resistance values of these batteries were not measured.

Based on the availability of measurement data and domain expertise, five features in total are extracted for regression modeling, which are indexed by x_1, x_2, x_3, x_4 and x_5. Note that all these five features are available for Dataset 2 while only x_1 and x_2 are available for Dataset 1 due to the its limited measurements. The feature names, physical meanings and their availabilities are summarized in Table 1. To reduce the nonlinearity of our modeling task, we take the logarithm for both the battery lifetime and the first feature x_1, following the common practice in the literature [1]. With these nonlinear transformations, we adopted a linear model template

$$\log(y) \approx w_1 \log(x_1) + w_2 x_2 + w_3 \tag{28}$$

for Dataset 1, and

$$\log(y) \approx w_1 \log(x_1) + w_2 x_2 + w_3 x_3 + w_4 x_4 + w_5 x_5 + w_6 \tag{29}$$

for Dataset 2. To improve numerical stability, we normalize the predicted outcome $\log(y)$ and all features $\{\log(x_1), x_2, x_3, x_4, x_5\}$ so that they have zero mean and unit variance over the training dataset.

Table 1: Features for battery lifetime modeling

Feature Name	Description	Available Dataset
x_1	Variance of the difference in the discharge capacity curves as a function of voltage between the 10-th and 100-th cycles	Dataset 1, Dataset 2
x_2	Slope of the capacity fade curve fitted by a linear function	Dataset 1, Dataset 2
x_3	Discharge capacity of the 2-nd cycle	Dataset 2
x_4	Integral of temperature over time from the 2-nd cycle to the 100-th cycle	Dataset 2
x_5	Difference of internal resistances between the 2-nd cycle and the 100-th cycle	Dataset 2

Through the experiments, we evaluate the improvement brought by the proposed method compared against other existing techniques when measurement noises are involved. However, it is usually difficult to identify the ground truth and noise from physical measurements in practice. Therefore, we assume that the public data are noise-free, and emulate the measurement noises by injecting artificially generated random noise following zero-mean Gaussian distributions, which are $N(0, \sigma_{n,i}^2)$ for the i-th input feature and $N(0, \sigma_{n,y}^2)$ for the prediction target. In our experiments, we assume that the standard deviation of the measurement noise is proportional to that of a feature/target (i.e., $\sigma_{n,I} = t \times \sigma_i$ and $\sigma_{n,y} = t \times \sigma_y$), where t is a scaling factor ranging from 0 to 0.95, referred to as the noise level in this paper. For testing and comparison purposes, we implemented two regression modeling methods, including (i) TLS and (ii) OLS with and without stepwise feature selection. To evaluate the modeling accuracy, we divided each dataset into (i) a training dataset for determining the unknown model coefficients, and (ii) a testing dataset for evaluating the modeling accuracy based on RMSE [31]. Note that the Gaussian noise is added to the training dataset only.

5.2 Dataset 1

Two experiments are conducted in this example to compare TLS against OLS. In the first experiment, we randomly select 5% samples as the testing data while keeping the remaining 95% samples as the training data. The noise level is increased from 0 to 0.95 with a stride of 0.05, for each of which 100 groups of noise patterns are generated and the random train-test split is repeated for 200 times. Thus, totally 200×100 RMSEs can be obtained at each noise level, and their median is reported to avoid the bias induced by random fluctuations. In the second experiment, we fix the noise level to 0.6 and vary the proportion of the training data from 35% to 95% to study the impact of training data size. Again, the random train-test split is repeated 200 times combined with 100 groups of randomly generated noise pattern, where the median value of the 200×100 RMSEs is reported for each proportion of training data.

Figure 1 shows the RMSE of TLS and OLS under different noise levels for the first experiment. It can be observed that the RMSE of OLS is less than that of TLS when the noise level is low. As the noise level increases, the RMSEs of both TLS and OLS increase. However, the RMSE of TLS increases less rapidly than OLS, and the RMSE of OLS exceeds that of TLS when the noise level is greater than 0.35. This observation demonstrates that the proposed TLS method can ensure a more reliable prediction with intense noise.

Figure 2 shows the RMSEs of TLS and OLS given different proportions of training data. It reveals three important observations. First, the RMSEs of both TLS and OLS are reduced, as more training samples are available and, hence, more information is incorporated for model training. Second, when only a small number of training samples are used, OLS outperforms TLS. Note that TLS needs to solve $\boldsymbol{\varepsilon}$, $\mathbf{E}$ and $\mathbf{w}$ in (9), while OLS only needs to solve $\mathbf{w}$ in (2). As TLS involves more problem unknowns, it is less robust to over-fitting and requires more training samples to reliably determine all unknowns. Third, when the training dataset is sufficiently large, TLS is more accurate than OLS.

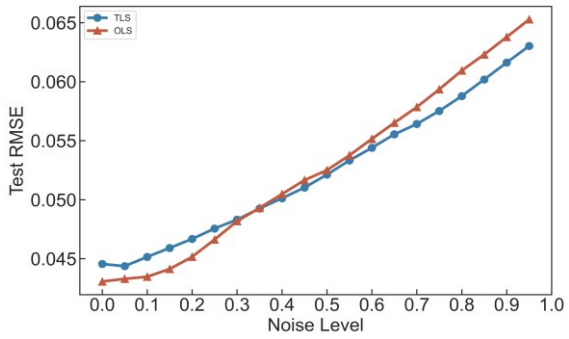

Figure 1: RMSEs of TLS and OLS under different noise levels on Dataset 1.

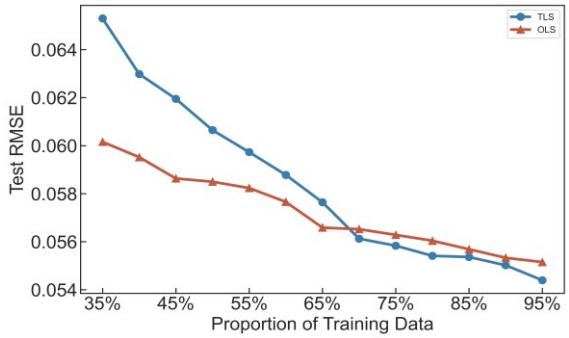

Figure 2: RMSEs of TLS and OLS on Dataset 1 given different proportions of training data when the noise level is set to 0.6.

5.3 Dataset 2

In this example, the efficacy of stepwise feature selection is studied based on Dataset 2. Similar to the previous example, two experiments are conducted to evaluate the impact of noise level and training data size respectively. In the first experiment, we randomly select 10% samples for testing while keeping the remaining 90% samples for training, which is repeated for 100 times. As the noise level is increased from 0 to 0.95 with a stride of 0.05, 50 groups of noise patterns are generated for each combination of the train-test split and noise level. The median of the 100×50 RMSEs is reported at each noise level. In the second experiment, we fix the noise level to 0.5, and vary the proportion of training data from 30% to 90% to investigate the impact of training data size for both TLS and OLS. In this experiment, the median of 100×50 RMSEs is reported for each proportion of training data.

Figure 3 shows the RMSEs of TLS and OLS with and without stepwise feature selection under different noise levels for the first experiment. Note that the RMSEs of both methods are almost identical when the noise level is 0. The RMSEs substantially increase as the noise level increases. Compared to OLS, the RMSEs of the TLS-based methods increase less rapidly, and stepwise feature selection can further reduce the RMSE . When the noise level is set to 0.75, TLS with stepwise feature selection can reduce the RMSE by 1.18%, 9.30% and 11.95% respectively, compared to TLS, OLS with stepwise feature selection and OLS. Two conclusions can be drawn from these observations. First, the TLS-based methods can generate more accurate regression models with consideration of measurement noises. Second, stepwise feature selection can effectively identify the important features to improve the prediction accuracy.

The RMSEs given different proportions of training data are shown in Figure 4. Note that the RMSE is reduced when the proportion of training data is increased. Furthermore, the TLS-based methods start to outperform the OLS-based methods when the proportion of training data is greater than 55%. It once again demonstrates that the TLS-based methods need more training data to avoid overfitting as they have more problem unknowns to solve. Stepwise feature selection can further reduce the RMSE, demonstrating its efficacy on learning a robust model with limited data samples.

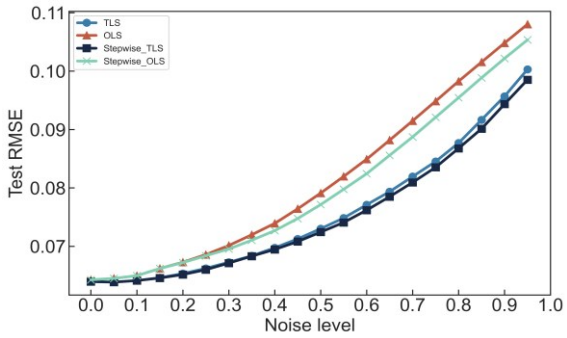

Figure 3: RMSEs of the TLS- and OLS-based methods with and without stepwise feature selection on Dataset 2 under different noise levels.

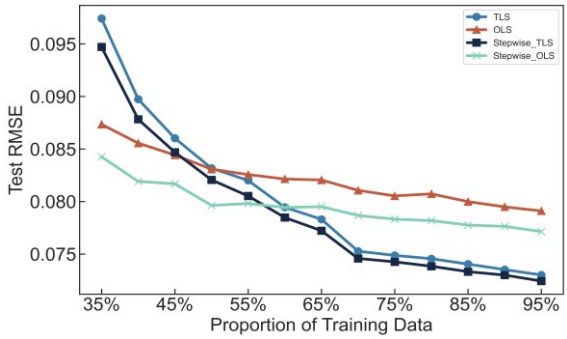

Figure 4: RMSEs of the TLS- and OLS-based methods with and without stepwise feature selection on Dataset 2 given different proportions of training data when the noise level is set to 0.5.

Furthermore, we have counted the number of selected features and the frequency that each feature is selected among the 100×50 random trials when the proportion of training data is set to 90% and noise level is set to 0.5. The histogram of the number of selected features is shown in Figure 5. Although all features are selected in most trials, there exist a number of trails where a subset of features are appropriately selected, leading to improved prediction accuracy.

The frequency that each feature is selected is further shown in Figures 6. Studying Figure 6, we have several important observations. First, the feature x_1 is selected in all trials. It is consistent with the domain knowledge since x_1 is broadly considered as an important feature for battery lifetime prediction in the community [1]. Second, as OLS is more sensitive to noise than TLS, it is more likely for OLS to neglect the four features x_2, x_3, x_4 and x_5 due to large noise. In other words, since OLS cannot appropriately handle measurement noises, it cannot take advantage of the noisy features (i.e., x_2, x_3, x_4 and x_5), thereby being less accurate than TLS in this example.

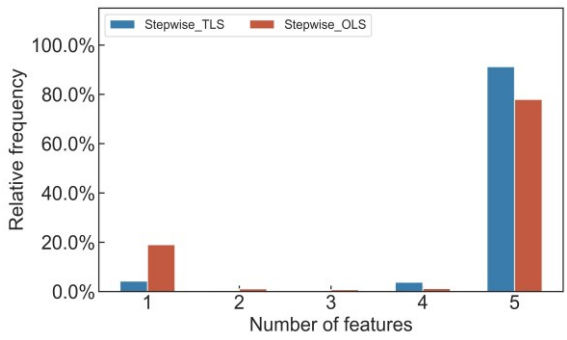

Figure 5: Number of features selected by stepwise feature selection among 100×50 random trials.

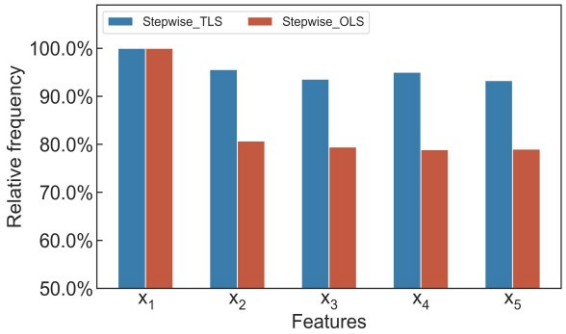

Figure 6: Frequency that each feature is selected among 100×50 trials.

6. Conclusions

In this paper, we propose a novel TLS method to predict battery lifetime based on regression modeling. TLS can effectively take into account the measurement noises associated with both input features and battery lifetime. While the TLS problem is cast as a non-convex optimization, a numerical solver is developed to determine the unknown model coefficients both efficiently (i.e., with low computational cost) and robustly (i.e., with guaranteed global convergence). Furthermore, stepwise feature selection is adopted to appropriately select a subset of important features with limited training data and alleviate the impact of overfitting. Numerical experiments based on two public datasets for commercial Lithium-Ion batteries show that TLS can effectively reduce the modeling error by up to 11.95%, compared against the classic baselines. It, in turn, demonstrates the efficacy of TLS as a promising approach for battery lifetime prediction in practice.

References

[1] K.A. Severson, P.M. Attia, N. Jin, N. Perkins, B. Jiang, Z. Yang, M.H. Chen, M. Aykol, P.K. Herring, D. Fraggedakis, M.Z. Bazant, S.J. Harris, W.C. Chueh, R.D. Braatz, Data-driven prediction of battery cycle life before capacity degradation, Nat Energy. 4 (2019) 383–391. https://doi.org/10.1038/s41560-019-0356-8.

[2] B. Scrosati, J. Garche, Lithium batteries: Status, prospects and future, J Power Sources. 195 (2010) 2419–2430. https://doi.org/10.1016/j.jpowsour.2009.11.048.

[3] Y. Echevarría, C. Blanco, L. Sánchez, Learning human-understandable models for the health assessment of Li-ion batteries via Multi-Objective Genetic Programming, Eng Appl Artif Intell. 86 (2019) 1–10. https://doi.org/10.1016/J.ENGAPPAI.2019.08.013.

[4] S. Chetoui, S. Reda, Workload- and User-aware Battery Lifetime Management for Mobile SoCs, in: 2021 Design, Automation & Test in Europe Conference & Exhibition (DATE), 2021: pp. 1679–1684. https://doi.org/10.23919/DATE51398.2021.9474111.

[5] M. Rossi, A. Toppano, D. Brunelli, Real-time optimization of the battery banks lifetime in hybrid residential electrical systems, in: 2014 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2014: pp. 1–6. https://doi.org/10.7873/DATE.2014.152.

[6] Y. Zhang, Q. Tang, Y. Zhang, J. Wang, U. Stimming, A.A. Lee, Identifying degradation patterns of lithium ion batteries from impedance spectroscopy using machine learning, Nat Commun. 11 (2020). https://doi.org/10.1038/s41467-020-15235-7.

[7] J. Zhu, Y. Wang, Y. Huang, R. Bhushan Gopaluni, Y. Cao, M. Heere, M.J. Mühlbauer, L. Mereacre, H. Dai, X. Liu, A. Senyshyn, X. Wei, M. Knapp, H. Ehrenberg, Data-driven capacity estimation of commercial lithium-ion batteries from voltage relaxation, Nat Commun. 13 (2022) 1–10. https://doi.org/10.1038/s41467-022-29837-w.

[8] C.W. Hsu, R. Xiong, N.Y. Chen, J. Li, N.T. Tsou, Deep neural network battery life and voltage prediction by using data of one cycle only, Appl Energy. 306 (2022). https://doi.org/10.1016/j.apenergy.2021.118134.

[9] J. Couture, X. Lin, Image- and health indicator-based transfer learning hybridization for battery RUL prediction, Eng Appl Artif Intell. 114 (2022) 105120. https://doi.org/10.1016/J.ENGAPPAI.2022.105120.

[10] J. Taylor, A. Barai, T.R. Ashwin, Y. Guo, M. Amor-Segan, J. Marco, An insight into the errors and uncertainty of the lithium-ion battery characterisation experiments, J Energy Storage. 24 (2019) 100761. https://doi.org/10.1016/j.est.2019.100761.

[11] A. Moradpour, M. Kasper, J. Hoffmann, F. Kienberger, Measurement Uncertainty in Battery Electrochemical Impedance Spectroscopy, IEEE Trans Instrum Meas. 71 (2022) 1–9. https://doi.org/10.1109/TIM.2022.3198466.

[12] A. Beck, A. Ben-Tal, M. Teboulle, Finding a global optimal solution for a quadratically constrained fractional quadratic problem with applications to the regularized total least squares, SIAM Journal on Matrix Analysis and Applications. 28 (2006) 425–445. https://doi.org/10.1137/040616851 (accessed August 4, 2023).

[13] H. Zhu, G. Leus, G.B. Giannakis, Sparsity-cognizant total least-squares for perturbed compressive sampling, IEEE Transactions on Signal Processing. 59 (2011) 2002–2016. https://doi.org/10.1109/TSP.2011.2109956.

[14] C.M. Bishop, N.M. Nasrabadi, Pattern recognition and machine learning, Springer, 2006.

[15] Z.P. Najafi-Haghi, F. Klemme, H. Jafarzadeh, H. Amrouch, H.-J. Wunderlich, Robust Resistive Open Defect Identification Using Machine Learning with Efficient Feature Selection, in: 2023 Design, Automation & Test in Europe Conference & Exhibition (DATE), IEEE, 2023: pp. 1–2. https://doi.org/10.23919/DATE56975.2023.10136961.

[16] X. Sun, K. Chakrabarty, R. Huang, Y. Chen, B. Zhao, H. Cao, Y. Han, X. Liang, L. Jiang, System-level hardware failure prediction using deep learning, in: Proc Des Autom Conf, Institute of Electrical and Electronics Engineers Inc., 2019. https://doi.org/10.1145/3316781.3317918.

[17] B. Jiang, J. Yang, H. Zhang, E.F.Y. Young, A Fast Machine Learning-based Mask Printability Predictor for OPC Acceleration, in: Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC, Institute of Electrical and Electronics Engineers Inc., 2019: pp. 450–455. https://doi.org/10.1145/3287624.3287682.

[18] G. Huang, J. Hu, Y. He, J. Liu, M. Ma, Z. Shen, J. Wu, Y. Xu, H. Zhang, K. Zhong, X. Ning, Y. Ma, H. Yang, B. Yu, H. Yang, Y. Wang, Machine Learning for Electronic Design Automation: A Survey, ACM Transact Des Autom Electron Syst. 26 (2021). https://doi.org/10.1145/3451179.

[19] M. Liu, R. Pan, F. Ye, X. Li, K. Chakrabarty, X. Gu, Fine-grained Adaptive Testing Based on Quality Prediction, in: ACM Transact Des Autom Electron Syst, Association for Computing Machinery, 2020. https://doi.org/10.1145/3385261.

[20] Y. Wang, W. Wen, L. Song, H.H. Li, Classification accuracy improvement for neuromorphic computing systems with one-level precision synapses, in: 2017 22nd Asia and South Pacific Design Automation Conference (ASP-DAC), IEEE, 2017: pp. 776–781. https://doi.org/10.1109/ASPDAC.2017.7858418.

[21] E.M. Elmandouh, A.G. Wassal, Guiding formal verification orchestration using machine learning methods, ACM Transact Des Autom Electron Syst. 23 (2018). https://doi.org/10.1145/3224206.

[22] J. Kwon, M.M. Ziegler, L.P. Carloni, A learning-based recommender system for autotuning design flows of industrial high-performance processors, in: Proc Des Autom Conf, Institute of Electrical and Electronics Engineers Inc., 2019. https://doi.org/10.1145/3316781.3323919.

[23] X. Li, Finding deterministic solution from underdetermined equation: Large-scale performance variability modeling of analog/RF circuits, IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems. 29 (2010) 1661–1668. https://doi.org/10.1109/TCAD.2010.2061292.

[24] U. Gupta, J. Campbell, U.Y. Ogras, R. Ayoub, M. Kishinevsky, F. Paterna, S. Gumussoy, Adaptive performance prediction for integrated GPUs, in: IEEE/ACM International Conference on Computer-Aided Design, Digest of Technical Papers, ICCAD, Institute of Electrical and Electronics Engineers Inc., 2016. https://doi.org/10.1145/2966986.2966997.

[25] Y.C. Pati, R. Rezaiifar, P.S. Krishnaprasad, Orthogonal matching pursuit: Recursive function approximation with applications to wavelet decomposition, in: Proceedings of 27th Asilomar Conference on Signals, Systems and Computers, IEEE, 1993: pp. 40–44. https://doi.org/10.1109/ACSSC.1993.342465.

[26] Y. Liu, X. Li, Predictive Modeling for Advanced Virtual Metrology: A Tree-Based Approach, IEEE International Conference on Emerging Technologies and Factory Automation, ETFA. 2018-Septe (2018) 845–852. https://doi.org/10.1109/ETFA.2018.8502480.

[27] H. Geng, T. Chen, Q. Sun, B. Yu, Techniques for CAD Tool Parameter Auto-tuning in Physical Synthesis: A Survey (Invited Paper), in: Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC, Institute of

Electrical and Electronics Engineers Inc., 2022: pp. 635–640. https://doi.org/10.1109/ASP-DAC52403.2022.9712495.

[28] P.M. Attia, A. Grover, N. Jin, K.A. Severson, T.M. Markov, Y.H. Liao, M.H. Chen, B. Cheong, N. Perkins, Z. Yang, P.K. Herring, M. Aykol, S.J. Harris, R.D. Braatz, S. Ermon, W.C. Chueh, Closed-loop optimization of fast-charging protocols for batteries with machine learning, Nature. 578 (2020) 397–402. https://doi.org/10.1038/s41586-020-1994-5.

[29] F. Wang, W. Zhang, S. Sun, X. Li, C. Gu, Bayesian model fusion: large-scale performance modeling of analog and mixed-signal circuits by reusing early-stage data, in: Proceedings of the 50th Annual Design Automation Conference, 2013: pp. 1–6. https://doi.org/10.1145/2463209.2488812.

[30] A. Beck, Introduction to nonlinear optimization: Theory, algorithms, and applications with MATLAB, SIAM, 2014.

[31] W. Zhang, X. Li, S. Saxena, A. Strojwas, R. Rutenbar, Automatic clustering of wafer spatial signatures, in: Proceedings of the 50th Annual Design Automation Conference, 2013: pp. 1–6. https://doi.org/10.1145/2463209.2488821.