A Data-Selection Framework for Data-Efficient Battery Parameter Estimation

Weinreich, Nicolai André; Teodorescu, Remus; Larsen, Kim Guldstrand

Published in:
2025 IEEE Energy Conversion Congress & Exposition Asia (ECCE-Asia)

Publication date:
2025

Document Version
Accepted author manuscript, peer reviewed version

A Data-Selection Framework for Data-Efficient Battery Parameter Estimation

Nicolai A. Weinreich
Department of Energy
Aalborg University
Aalborg, Denmark
nawe@energy.aau.dk

Remus Teodorescu
Department of Energy
Aalborg University
Aalborg, Denmark
ret@energy.aau.dk

Kim G. Larsen
Department of Computer Science
Aalborg University
Aalborg, Denmark
kgl@cs.aau.dk

***Abstract*—Online battery parameter identification is critical for accurate monitoring of battery states. However, conventional identification methods will perform poorly when the measured cell data is not informative enough. Data selection methods can be used to detect and only use high-quality cell data segments, thus making the identification process more efficient. In this paper, we measure the quality of a data segment based on the Cramer-Rao lower bound under the Total Least Squares (TLS) framework. A Convolutional Neural Network (CNN) is then used to predict if a data segment is useful or not based on the acceleration profile of the vehicle. The CNN was trained and validated on synthetic cell data generated from simulated trips in the city of Berlin. The results show good performance of the CNN and the proposed data selection algorithm yields cell models which performs better than conventional identification methods with a data reduction of up to 56 percent.**

***Index Terms*—Data selection, battery identification, machine learning, CNN.**

I. Introduction

Online battery identification is a critical task of any Electric Vehicle (EV) Battery Management System (BMS) as it provides more accurate monitoring of internal battery states such as State of Charge (SOC), State of Temperature (SOT), and internal resistance, thus enabling safe control of the battery pack. The Recursive Least Squares (RLS) algorithm is highly popular for this task due to its low complexity and its compatibility with Kalman Filter (KF) based SOC observers [1], [2]. However, as discussed in [3] and [4], the RLS algorithm provides biased parameter estimates in the presence of current measurement noise. To alleviate this issue, the Total Least Squares (TLS) algorithm was proposed to account for input noise; however, the high complexity of the algorithm limits its use in online application. To address this, recursive versions of the TLS algorithm have been proposed, either updating one sample at a time [3] or using smaller batches of data [5]. One key consideration in using the TLS estimate is how to account for the informativeness of the measurement data since data with low information content can still introduce estimation bias. Being able to only select highly informative data to include in the estimation process will both lead to better estimates and less overall number of computations, thus improving the efficiency of the estimator. In [6], a data quality measure was proposed, showing an improved estimation accuracy by an order of magnitude; however, the proposed measure assumes a least squares estimation framework and may not be compatible in the TLS framework. Additionally, the quality measure depends on cell current and voltage measurements, meaning that the decision process is confined to low-complexity hardware which can limit its practicality.

This work was supported by the Villum Foundation for the SMART BATTERY project (No. 222860).

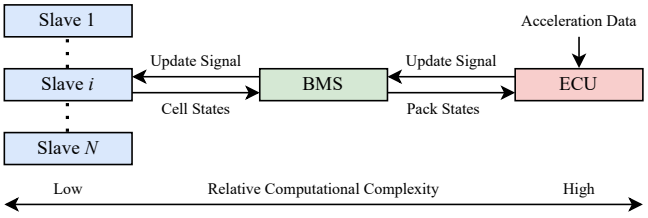

Fig. 1. Proposed parameter update strategy in this paper.

In this paper, we propose that the quality of battery data can not only be inferred from the data itself, but also the context it was generated in e.g. the driving profile of the EV. With this interpretation, it should be possible to identify time segments where the battery data is highly informative based on the driving profile during that time. Thus, the decision making can be moved from low-complexity slave modules to an Electronic Control Unit (ECU) with higher computational capability. This strategy is pictured in Fig. 1. Here, the ECU predicts if the recently collected acceleration data generated informative battery data. If this is the case, then an update signal is sent via the BMS to the slaves which in turn update their parameters using TLS. In this paper, a Convolutional Neural Network (CNN) is used to provide this prediction. The CNN is trained based on acceleration and battery data in a fully simulated environment. The Cramer-Rao Lower Bound (CRLB) is used to quantify the theoretical informativeness of the battery data. This is motivated by the fact that the CRLB is the theoretical lower bound on the variance of any unbiased parameter estimator [7]. Since the variance of the estimator is assumed to be a consequence of measurement noise, data with a low CRLB should yield parameter estimates which are more

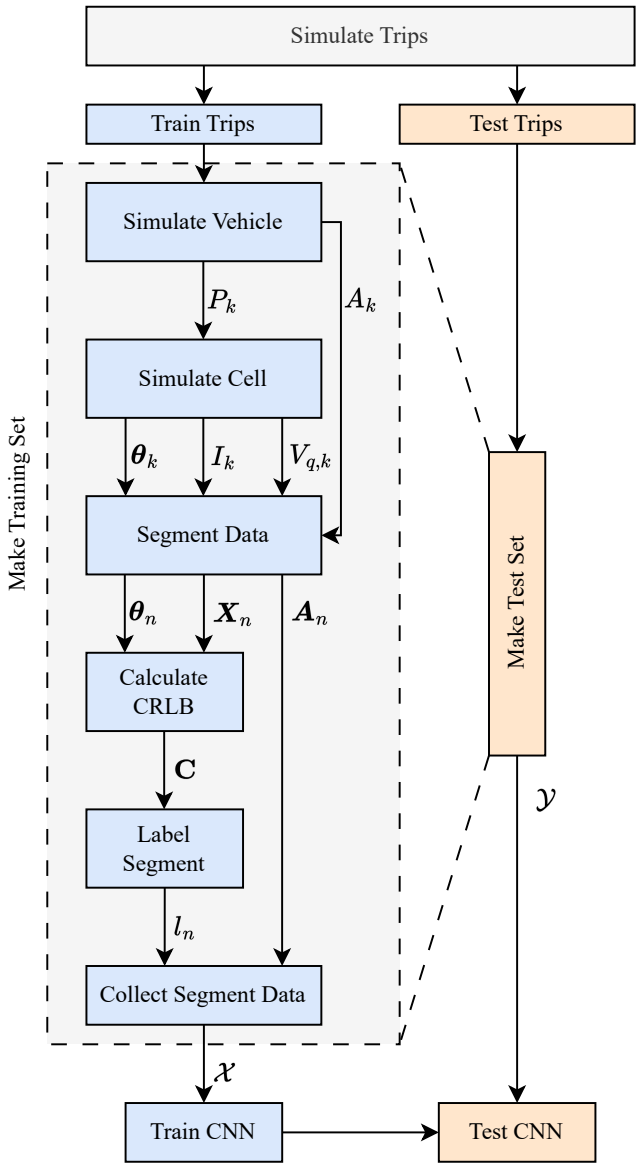

Fig. 2. Diagram of the methodology in this paper.

robust against noise. The CRLB is calculated assuming an Autoregressive with exogenous variables (ARX) battery model and that the parameters are estimated using TLS.

The paper is organized as follows: Section II presents the overall methodology of the study including the simulation methodology, the training of the CNN, and the algorithms used for parameter estimation. Section III presents the results of the CNN classifier and parameter identification algortihms. The paper is concluded in Section V. The associated code repository for this paper can be found at https://github.com/nawe-aau/ecce-asia-2025.

II. Methodology

The main methodology of the paper is pictured in Fig. 2. Trips are simulated within a city network and then subsequently split between trips used for training and trips used

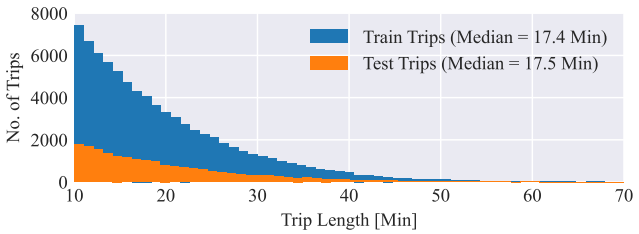

Fig. 3. distribution of trip lengths for the 100.000 simulated trips used in this paper.

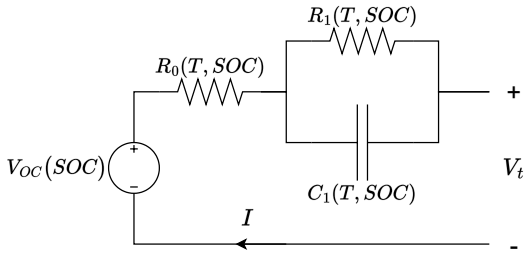

Fig. 4. Schematic of a dynamic 1RC ECM.

for validation. For each training trip, the vehicle and battery cell data are simulated and divided into smaller segments. The resulting simulated cell data for each segment are used to calculate the CRLB which is used to label the segment as good or bad. This label is then collected with the acceleration data for that segment and used to construct the training set. The same procedure is used to construct the test set. In the following, the specific steps will be described in detail.

A. Trip Simulation

The trips used for this study are constructed from a 24 hour traffic simulation in the city of Berlin. The simulation was performed using the simulation framework TRACKSIM as described in [8]. The construction of the city network and the traffic modeling are detailed in [9]. The traffic simulation was performed with a sampling rate of 1 Hz. Out of a total of 2.2 million trips generated, a random selection of 100.000 trips were used of which 80.000 were used for training and 20.000 were used for validation. The distribution of the lengths of the simulated trips can be seen in Fig. 3.

B. Vehicle Simulation

For each trip, the vehicle and battery pack was simulated by TRACKSIM. A modified model of a 1'st generation Chevy Volt was used based on the description in [10]. The battery pack consists of 12 8p3s battery modules and has a nominal energy capacity of 52 kWh. The vehicle simulation yields the acceleration of the vehicle A_k and battery power demand P_k for time steps $k = 1, 2, \ldots, K$ where K is the length of the trip.

Algorithm 1 $[\hat{\boldsymbol{\theta}}_k, \mathbf{P}_k]$ = RLS$[\hat{\boldsymbol{\theta}}_{k-1}, \mathbf{P}_{k-1}, \tilde{V}_{q,k}, \tilde{\boldsymbol{\phi}}_k, \lambda]$

1: Calculate gain: $\mathbf{L}_k = \mathbf{P}_{k-1}\tilde{\boldsymbol{\phi}}_k / (\lambda + \tilde{\boldsymbol{\phi}}_k^T \mathbf{P}_{k-1}\tilde{\boldsymbol{\phi}}_k)$
2: Update parameter: $\hat{\boldsymbol{\theta}}_k = \hat{\boldsymbol{\theta}}_{k-1} + \mathbf{L}_k(\tilde{V}_{q,k} - \hat{\boldsymbol{\theta}}_{k-1}^T \tilde{\boldsymbol{\phi}}_k)$
3: Update covariance: $\mathbf{P}_k = (\mathbf{I} - \mathbf{L}_k \tilde{\boldsymbol{\phi}}_k^T)\mathbf{P}_{k-1}$

Algorithm 2 $[\hat{\boldsymbol{\theta}}_{TLS}]$ = TLS$[\tilde{\boldsymbol{y}}, \tilde{\boldsymbol{X}}]$

1: Construct augmented observation matrix: $\boldsymbol{H} = \begin{bmatrix} \tilde{\boldsymbol{X}} & \tilde{\boldsymbol{y}} \end{bmatrix}$
2: Perform SVD: $\boldsymbol{H} = \boldsymbol{U}\boldsymbol{\Lambda}\boldsymbol{V} = \boldsymbol{U}\boldsymbol{\Lambda}\begin{bmatrix} \mathbf{V}_{pp} & \boldsymbol{v}_{pq} \\ \boldsymbol{v}_{qp} & v_{qq} \end{bmatrix}$
3: Compute parameter: $\hat{\boldsymbol{\theta}}_{TLS} = -\boldsymbol{v}_{pq}/v_{qq}$

C. Battery Pack & Cell Modeling

The battery pack is simulated using TRACKSIM to obtain the required input current, I, and terminal voltage, V_t, for each cell in the pack based on the required battery power during the trip. Each cell is simulated as a dynamic 1RC Electrical Circuit Model (ECM) as shown in Fig. 4 where V_{oc} is the SOC dependent Open Circuit Voltage (OCV). Defining $V_q = V_t - V_{oc}$, the circuit in Fig. 4 can be characterized by the transfer function

$$V_q(s)/I(s) = -(R_0 + R_1 + R_0R_1C_1s)/(1 + R_0C_1s) \quad (1)$$

where the temperature and SOC dependencies are omitted for brevity. Using the bilinear transform, the transfer function can be discretized into the ARX model

$$V_{q,k} = \boldsymbol{\theta}_k^T \boldsymbol{\phi}_k, \quad k = 1, 2, \dots K \quad (2)$$

where $\boldsymbol{\theta}_k = [\theta_{1,k}, \theta_{2,k}, \theta_{3,k}]^T$, $\boldsymbol{\phi}_k = [V_{q,k-1}, I_k, I_{k-1}]^T$ and

$$\begin{bmatrix} \theta_{1,k} \\ \theta_{2,k} \\ \theta_{3,k} \end{bmatrix} = \begin{bmatrix} -(\Delta t - 2R_0C_1)/(\Delta t + 2R_0C_1) \\ -(\Delta tR_0 + \Delta tR_1 + 2R_0R_1C_1)/(\Delta t + 2R_0C_1) \\ -(\Delta tR_0 + \Delta tR_1 - 2R_0R_1C_1)/(\Delta t + 2R_0C_1) \end{bmatrix} \quad (3)$$

with Δt being the time between samples. The true terminal voltage is then given by

$$V_{t,k} = V_{q,k} + V_{oc,k}, \quad k = 1, 2, \dots, K. \quad (4)$$

The derivations of the above are omitted, but details can be found in [3].

After obtaining the true current and voltage data, additive Gaussian measurement noise is used to corrupt the data. The noisy current and voltage data is given by

$$\tilde{I}_k = I_k + \varepsilon_{I,k}, \quad \varepsilon_{I,k} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_I^2), \quad (5)$$

$$\tilde{V}_{q,k} = V_{q,k} + \varepsilon_{V,k}, \quad \varepsilon_{V,k} \overset{i.i.d.}{\sim} \mathcal{N}(0, \sigma_V^2). \quad (6)$$

D. RLS and TLS

The conventional RLS method iteratively updates the parameter estimate $\hat{\boldsymbol{\theta}}$ based on recently obtained measurements, $\tilde{V}_{q,k}$ and $\tilde{\boldsymbol{\phi}}_k = [\tilde{V}_{q,k-1}, \tilde{I}_k, \tilde{I}_{k-1}]^T$, and a user-defined forgetting factor $0 \ll \lambda < 1$. The method is described in Algorithm 1. Due to noisy measurements in $\tilde{\boldsymbol{\phi}}_k$ the parameter estimate will become biased since this is not accounted for in conventional LS methods. To solve this, TLS methods have been proposed to account for noise in both the input and output measurements. The TLS method is described in Algorithm 2 where

$$\tilde{\boldsymbol{y}} = \begin{bmatrix} \tilde{V}_{q,1} & \tilde{V}_{q,2} & \cdots & \tilde{V}_{q,K} \end{bmatrix}^T \quad (7)$$

$$\tilde{\mathbf{X}} = \begin{bmatrix} \tilde{\boldsymbol{\phi}}_1 & \tilde{\boldsymbol{\phi}}_2 & \cdots & \tilde{\boldsymbol{\phi}}_K \end{bmatrix}^T. \quad (8)$$

While the TLS method eliminates bias due to noise on the input, it requires K data points to construct $\tilde{\boldsymbol{y}}$ and $\tilde{\mathbf{X}}$ instead of only the most recent data. This introduces the following challenges: 1) It assumes that $\boldsymbol{\theta}$ is constant over this time interval which will lead to biased estimates for systems with dynamic parameters. 2) The high computational complexity of calculating the SVD makes the method impractical for large K. Both of these challenges can be alleviated by dividing the measurement data into smaller segments and then obtain the TLS estimate for each segment. However, this may lead to unreliable estimates due to low-information segments, thus motivating the use of data selection.

E. Data Segmentation and Labeling

After obtaining A_k, $V_{q,k}$, and I_k for all k, the K-length signals are divided into $N = \lfloor K/S \rfloor$ consecutive segments where S is the segment length. We will denote $\boldsymbol{A}_n$ as the collection of acceleration data for the n'th segment. Likewise, we will denote $\boldsymbol{y}_n$ and $\boldsymbol{X}_n$ as in (7) and (8) with the true current and voltage data from the n'th segment.

Each segment is labeled as either good or bad based on a measure on its inherent usefulness in estimating $\boldsymbol{\theta}$. In this paper, the CRLB is used. The CRLB is defined as

$$\mathbf{C} = \mathbf{F}^{-1} \quad (9)$$

where $\mathbf{I}$ is the Fisher Information Matrix (FIM). Using the noise assumptions in Section II-C, the FIM of the n'th segment for the TLS problem is given by [4]

$$\mathbf{F}_n = \frac{(S+1)\mathbf{D}}{\sigma_y^2} + \frac{2(S-1)\mathbf{D}\boldsymbol{\theta}_n\boldsymbol{\theta}_n^T\mathbf{D}}{\sigma_y^4} + \frac{\mathbf{X}_n^T\mathbf{X}_n}{\sigma_y^2} \quad (10)$$

where

$$\sigma_y^2 = (1 + \theta_1^2)\sigma_V^2 + (\theta_2^2 + \theta_3^2)\sigma_I^2 \quad (11)$$

and $\mathbf{D} = diag\{\sigma_V^2, \sigma_I^2, \sigma_I^2\}$. The CRLB for each parameter is then given by the diagonals of $\mathbf{C}$. Note that the CRLB requires knowledge about the true parameters and regression variables. Thus, it can not be computed in practice. However, it can still be used to classify the usefulness of the segment which in turn can be predicted based on measurable data. Moreover, the CRLB assumes that the true parameter $\boldsymbol{\theta}_n$ is constant over the n'th segment. It was found that defining $\boldsymbol{\theta}_n$ as the average parameter vector over the segment produced meaningful CRLB values in practice. Let $CRLB(\theta_i)$ be the CRLB of θ_i. A segment is deemed good for estimating θ_i if

$$CRLB(\theta_i) < thr_i \quad (12)$$

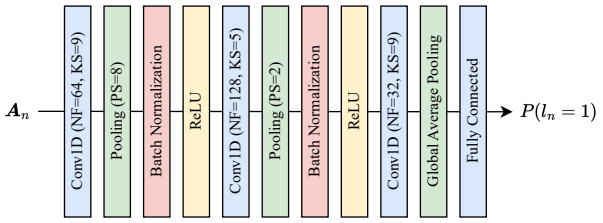

Fig. 5. Architecture of the CNN classifier used in this paper. NF = number of filters, KS = kernel size, PS = pool size.

Algorithm 3 $[\hat{\boldsymbol{\theta}}_n]$ = DS-TLS$[\hat{\boldsymbol{\theta}}_{n-1}, \boldsymbol{A}_n, \tilde{\boldsymbol{y}}_n, \tilde{\boldsymbol{X}}_n, \tau]$

1: Get confidence: $C_n = P(l_n = 1|\boldsymbol{A}_n) = CNN(\boldsymbol{A}_n)$

2: Update parameter: $\hat{\boldsymbol{\theta}}_n = \begin{cases} TLS[\tilde{\boldsymbol{y}}_n, \tilde{\boldsymbol{X}}_n], & C_n > \tau, \\ \hat{\boldsymbol{\theta}}_{n-1}, & \text{otherwise.} \end{cases}$

where thr_i is a user-defined threshold for the particular parameter. The label of the segment is defined as

$$l_n = \begin{cases} 1, & CRLB(\theta_i) < thr_i, \quad \forall i \\ 0 & \text{otherwise} \end{cases} \tag{13}$$

In this paper, thr_i is defined as the 20'th percentile of the histogram constructed by the values of $CRLB(\theta_i)$ for all segments in all training trips. It was found that this choice of percentile yielded an acceptable number of good segments in the final training set while ensuring that the variance of the parameter estimate errors are reduced by an order of magnitude.

F. Classifier Training & Testing

The training set $\mathcal{X}$ and test set $\mathcal{Y}$ are constructed by collecting the acceleration data $\boldsymbol{A}_n$ and label l_n for every segment in the training trips and test trips respectively. The segment classification is cast as a binary time-series classification problem. The classifier is constructed as a 3-layer CNN with the architecture described in Fig. 5. The number of filters, kernel sizes, pool sizes, and additional regularization parameters were found through hyperparameter tuning. To account for imbalance in the training set, the binary focal cross-entropy was used as the optimization function [11]. The classifier provides a confidence measure

$$C_n = P(l_n = 1|\boldsymbol{A}_n) \tag{14}$$

where $P(\cdot)$ indicates probability. A segment is classified as good if $C_n > \tau$ where τ is a user-defined threshold. The classifier is evaluated based on the precision and recall defined as

$$\text{Precision} = \frac{TP}{TP + FP}, \quad \text{Recall} = \frac{TP}{TP + FN} \tag{15}$$

where TP is the number of true positives ($l_n = 1$), FP is the number of false positives, and FN is the number of false negatives ($l_n = 0$).

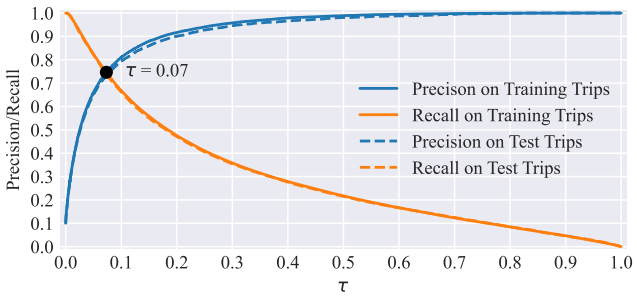

Fig. 6. Precision and recall of the CNN classifier over various choices of τ.

G. Data Selective TLS

They key idea behind DS-TLS is to only update the parameter estimate if the confidence that the segment is informative is above a certain threshold. The method is described in Algorithm 3. As the performance of the algorithm depends on the classifier and τ it may be possible that no parameter estimates occur and thus $\hat{\boldsymbol{\theta}}_n = \hat{\boldsymbol{\theta}}_0$ for all n. Thus, having a reasonable initial estimate may enhance the DS-TLS method. To investigate this, we define two variants of the DS-TLS method:

- *DS-TLS-1*: $\hat{\boldsymbol{\theta}}_n$ is only updated if $C_n > \tau$ as described in Algorithm 3.
- *DS-TLS-2*: The first segment is always used to update $\hat{\boldsymbol{\theta}}_n$ using TLS while Algorithm 3 is used for every subsequent segment.

III. Results

This section presents simulation results of the CNN classifier and the DS-TLS method described in Section 3.

A thermo-coupled ECM described in [12] is used to simulate each cell in the battery pack for each trip. The precise model description of the cells can be found in the code repository. Each vehicle is assumed to be at rest at the start of the trip with a cell temperature of 25 degrees Celsius and a random SOC between 0.5 and 0.8. As the cells in the battery pack are assumed to be identical, only data from one cell is used. The true dynamic ECM parameters are converted to the equivalent ARX parameters using (3) for further simulation. To account for discretization error between (1) and (2), the simulated battery voltage was re-calculated using (2) with the true dynamic ARX parameters and the input current calculated by TRACKSIM. Additive Gaussian noise is added with $\sigma_I = 0.7A$ and $\sigma_V = 0.001V$. A segment length of $S = 120s$ was chosen to limit the computational costs of the TLS method while still providing enough data for effective parameter estimation.

A. CNN Results

The performance of the CNN on the training and test set is pictured in Fig. 6. The precison and recall curves on the training and test sets are similar, indicating a low amount of

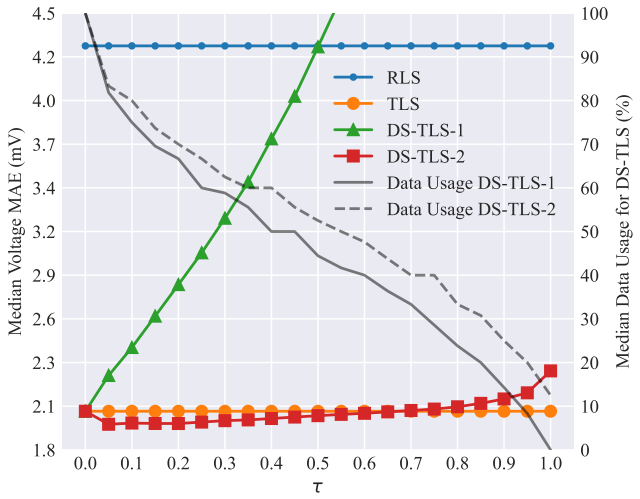

Fig. 7. Median voltage MAE and data usage over all test trips for various choices of τ.

overfitting. Additionally, the figure clearly shows the precision/recall trade-off i.e. for high thresholds, the probability of the segment being good when the CNN classifies it as good increases but it also misses a larger number of good segments. In practice, having a higher precision means that the chosen segments will likely lead to better model estimates, while lower recall means longer wait times between model updates. As is seen on the figure, the precision and recall curves intersect at $\tau \approx 0.07$ where both the precision and recall are $\approx 75\%$. The precision at $\tau = 0$ (all segments are selected) is 10% which is due to the imbalance in the training and test set i.e. around 10% of all segments are good.

B. Identification Results

In this section, we compare the identification results of the methods described in Sections II-D and II-G. For each test trip, the cell voltage was estimated using the ARX model described in Section II-C with parameters identified using RLS, TLS, DS-TLS-1, and DS-TLS-2. To quantify the estimation performance the Mean Absolute Error (MAE) is used. The MAE for the i'th test trip is defined as

$$MAE_i = \frac{1}{K_i - S} \sum_{k=S+1}^{K_i} |\tilde{V}_{t,k}^{(i)} - \hat{V}_{t,k}^{(i)}|, \quad i = 1, 2, \ldots, 20.000 \tag{16}$$

where K_i is the length of the i'th trip and $\hat{V}_{t,k}^{(i)}$ is the estimated voltage of the trip. Note that only voltage errors after the first segment is considered. This is done to facilitate fair comparison between the methods since the RLS algorithm needs some time to converge while the TLS, DS-TLS-1, and DS-TLS-2 algorithms need the data from the first segment to obtain the earliest possible parameter update. The initial parameter for each method is defined as $\hat{\boldsymbol{\theta}}_0 = \mathbf{0}$. Additionally, the RLS algorithm was initialized with $\mathbf{P}_0 = diag\{10^6, 10^6, 10^6\}$ and a forgetting factor of $\lambda = 0.999$. For DS-TLS-1 and DS-TLS-2, the percentage of data used during the trip is defined as the number of chosen segments divided by the total number of segments. The median MAE over all test trips is used to compare the different algorithms. The results are pictured in Fig. 7. It is observed that the DS-TLS-1 algorithm performs worse as τ increases, which is due to fewer parameter updates. Thus most of the estimated voltages are based on a poorly initialized $\hat{\boldsymbol{\theta}}_0$. However, as can be seen for DS-TLS-2, having a rough estimate based on the first segment and then selecting segments afterwards enhances the overall estimation accuracy for $0 < \tau \leq 0.5$ with the biggest performance enhancement being in the interval of $0.05 \leq \tau \leq 0.1$ which is consistent with the results presented in Section III-A. The performance of the DS-TLS-2 steadily increases for $\tau > 0.2$ as the time between parameter updates becomes larger. It was found that the DS-TLS-2 algorithm still outperformed the TLS algorithm in the median with $\tau = 0.65$ while using only $\approx 44\%$ of the data.

IV. Discussion

As was seen in the results, it possible to make the online model identification problem more efficient by selecting which segments of data to include in the identification process. Moreover, the utility of using proxy data like acceleration instead of battery data to perform the selection has been demonstrated. Other types of data can be considered in the selection process, such as velocity and possibly geographical data. Linking the quality of the generated battery data to types of locations on the road network is an interesting future research direction.

The relationship between battery data quality and proxy data is highly complex, making it a suitable problem for deep learning methods; However, the added complexity related to training and inference, limits it to higher-complexity hardware such as an ECU or an advanced BMS rather than the slave boards. Additionally, model bias needs to be taken into consideration. While Fig. 6 shows low overfitting on the test set, the model may still be biased towards trips simulated from the given road network e.g. Berlin and may perform worse on trips in another city or trips in a non-urban environment such as highways. Additionally, the dependence on the CRLB for labeling could be a limiting factor of the CNN as it is highly dependent on the assumed model and noise structure. To mitigate these concerns, one could investigate more simple methods for assessing data quality such as defining specific features from the data or heuristic methods based on the conditions the data was generated in such as SOC and temperature. Model-agnostic quality measures should also be examined to develop selection strategies which fit a broader range of models and estimation algorithms.

The results indicate that being selective in which segments to use combined with a reasonable initial estimate significantly improves the performance of the model. Nonetheless, the performance of the selection algorithms is highly dependent on the choice of τ. The optimal threshold is likely dependent on the specific classification model and the given trip and

may also change over the duration of the trip. Thus, selection algorithms which include a variable threshold should be investigated in the future.

V. Conclusion

Data selection can be a powerful tool to increase the data-efficiency of the battery identification process. As was demonstrated in the results, an up to 56% decrease in median data usage over the test trips can be achieved with better estimation performance than conventional methods. By moving the decision-making from the slaves to a central controller with access to vehicle-level data, this can free up valuable resources in the slaves to be used in other tasks while improving overall performance. However, this is highly dependent on the confidence-threshold used in the algorithm which is currently fixed in this paper. A clear future research direction is investigating strategies for updating the threshold over time based on user needs. Additionally, alternative measures of data quality to the CRLB which are more agnostic to the assumed model and noise structure should be considered.

References

[1] I. Lopez-Granados, J. M. Sosa, G. Vazquez, A. R. Lopez, and D. Langarica, "A Brief Review of Battery Model Parameter Identification Methods," in *2021 IEEE International Autumn Meeting on Power, Electronics and Computing (ROPEC)*. Ixtapa, Mexico: IEEE, Nov. 2021, pp. 1–6. [Online]. Available: https://ieeexplore.ieee.org/document/9667980/

[2] M. Hossain, M. E. Haque, and M. T. Arif, "Kalman filtering techniques for the online model parameters and state of charge estimation of the Li-ion batteries: A comparative analysis," *Journal of Energy Storage*, vol. 51, p. 104174, Jul. 2022. [Online]. Available: https://linkinghub.elsevier.com/retrieve/pii/S2352152X22002067

[3] Z. Wei, C. Zou, F. Leng, B. H. Soong, and K.-J. Tseng, "Online Model Identification and State-of-Charge Estimate for Lithium-Ion Battery With a Recursive Total Least Squares-Based Observer," *IEEE Transactions on Industrial Electronics*, vol. 65, no. 2, pp. 1336–1346, Feb. 2018, conference Name: IEEE Transactions on Industrial Electronics. [Online]. Available: https://ieeexplore.ieee.org/document/8003310/?arnumber=8003310&tag=1

[4] P. Kumar, S. Sundaresan, and B. Balasingam, "Theoretical Performance Evaluation of Battery Equivalent Circuit Model Parameter Estimators," *IEEE J. Emerg. Sel. Top. Ind. Electron.*, vol. 5, no. 3, pp. 1109–1119, Jul. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10356631/

[5] B. Balasingam and K. R. Pattipati, "On the Identification of Electrical Equivalent Circuit Models Based on Noisy Measurements," *IEEE Transactions on Instrumentation and Measurement*, vol. 70, pp. 1–16, 2021, conference Name: IEEE Transactions on Instrumentation and Measurement. [Online]. Available: https://ieeexplore.ieee.org/document/9383295/?arnumber=9383295

[6] J. Fogelquist and X. Lin, "Data selection framework for battery state of health related parameter estimation under system uncertainties," *eTransportation*, vol. 18, 2023.

[7] T. M. Cover and J. A. Thomas, *Elements of information theory*, 2nd ed. Hoboken, N.J: Wiley-Interscience, 2006, oCLC: ocm59879802.

[8] N. A. Weinreich, X. Sui, R. Teodorescu, and K. G. Larsen, "TRACK-SIM: A multi-level simulation framework for near-life battery data generation," in *2025 26th European Conference on Power Electronics and Applications (EPE'25 ECCE Europe)*, Aalborg, Denmark, Apr. 2025.

[9] K. Schrab, R. Protzmann, and I. Radusch, "A Large-Scale Traffic Scenario of Berlin for Evaluating Smart Mobility Applications," in *Smart Energy for Smart Transport*, E. G. Nathanail, N. Gavanas, and G. Adamos, Eds. Cham: Springer Nature Switzerland, 2023, pp. 276–287, series Title: Lecture Notes in Intelligent Transportation and Infrastructure. [Online]. Available: https://link.springer.com/10.1007/978-3-031-23721-8_24

[10] G. L. Plett, *Battery Management Systems, Volume 2: Equivalent Circuit Methods*, ser. Artech House Power engineering series. Boston: Artech house, 2016.

[11] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollar, "Focal Loss for Dense Object Detection," 2017, pp. 2980–2988. [Online]. Available: https://openaccess.thecvf.com/content_iccv_2017/html/Lin_Focal_Loss_for_ICCV_2017_paper.html

[12] Y. Zheng, Y. Che, X. Hu, X. Sui, and R. Teodorescu, "Online Sensorless Temperature Estimation of Lithium-Ion Batteries Through Electro-Thermal Coupling," *IEEE/ASME Trans. Mechatron.*, vol. 29, no. 6, pp. 4156–4167, Dec. 2024. [Online]. Available: https://ieeexplore.ieee.org/document/10457049/