vehicles

MDPI

Article

Trip-End Battery State of Charge Prediction for Electric Vehicles Using a Physics-Constrained Machine Learning Stacking Ensemble

Dmytro Tymoshchuk [1,*], Oleh Lyashuk [2], Pavlo Maruschak [3,*], Dmytro Mironov [2] and Vitaliy Tymoshchuk [4]

[1] Department of Artificial Intelligence Systems and Data Analysis, Ternopil Ivan Puluj National Technical University, 46001 Ternopil, Ukraine
[2] Department of Engineering of Machines, Structures and Technologies, Ternopil Ivan Puluj National Technical University, 46001 Ternopil, Ukraine; oleglashuk@ukr.net (O.L.); mironov.epz@gmail.com (D.M.)
[3] Department of Automation of Technological Processes and Production, Ternopil Ivan Puluj National Technical University, 46001 Ternopil, Ukraine
[4] Department of Computer Science, Ternopil Ivan Puluj National Technical University, 46001 Ternopil, Ukraine; vitaliy.tymoshchuk@gmail.com
* Correspondence: dmytro.tymoshchuk@gmail.com (D.T.); maruschak.tu.edu@gmail.com (P.M.)

Abstract

This study proposes an approach for predicting the state of charge (SoC) of an electric vehicle battery at the end of an individual operational interval using parameters available before the start of the trip. The input features included the initial SoC, trip distance, and ambient temperature. The driving range estimated by the onboard system was also considered, as it summarizes information about the vehicle's previous operating conditions. The direct target variable was the change in state of charge over the operational interval (Delta SoC), which was subsequently used to calculate the state of charge at the end of the interval (Current SoC). The study was conducted using operational data from a 2020 Kia Niro EV, comprising 352 operational intervals from 81 discharge cycles. The obtained performance metrics characterize the vehicle investigated in this study and the range of operating conditions represented in the dataset. Cross-vehicle generalizability was not evaluated within the scope of this study. Nevertheless, the proposed methodology is general in its structure and can be applied to other electric vehicles, provided that the models are retrained and independently validated using relevant operational data. Model performance was evaluated using repeated five-fold cycle-aware GroupKFold validation. The primary Physics-Constrained Stacking configuration combined ten regression models using non-negative weights. For Current SoC, this configuration achieved an RMSE of 2.1183 percentage points, an MAE of 1.2532 percentage points, and an R^2 of 0.9804. The absolute prediction error did not exceed 2 percentage points for 84.48% of the predictions. Normalized split conformal prediction provided a mean empirical coverage of 92.85% for nominal 90% prediction intervals. Explainable artificial intelligence (XAI) analysis showed that travel distance was the primary predictor of Delta SoC, whereas the initial SoC was the dominant predictor of Current SoC. The proposed approach enables accurate, physically consistent, and interpretable prediction of the final state of charge while also quantifying predictive uncertainty.

Keywords: electric vehicles; battery state of charge; machine learning; stacking ensemble; conformal prediction; explainable artificial intelligence; uncertainty quantification; operational data

Check for updates

Academic Editor: Qiao Wang

Received: 7 August 2026
Revised: 11 September 2026
Accepted: 21 September 2026
Published: 23 September 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The transport sector is one of the largest consumers of fossil fuels and a major source of greenhouse gas emissions. The continued growth in transportation demand intensifies both energy-related and environmental challenges. Consequently, transport electrification has become one of the key priorities of decarbonization strategies.

Battery electric vehicles (BEVs) are widely regarded as a promising foundation for low-carbon mobility. Governments and vehicle manufacturers are gradually expanding their support for BEVs while reducing their reliance on vehicles powered by internal combustion engines. Effective route optimization and charging schedule planning require models capable of predicting vehicle energy consumption during a trip and estimating the expected battery state of charge (SoC) at the destination [1,2]. Detailed physics-based powertrain models can provide high predictive accuracy. However, they require considerable effort for parameterization and may be computationally demanding when applied to the real-time management of large vehicle fleets. Response surface methods are less complex, but their performance depends on the appropriate selection of input features, regression structure, and the model's transferability across different vehicles.

The wider adoption of BEVs is constrained by their limited driving range. So-called range anxiety is directly associated with the accuracy of forecasts of future energy consumption and the remaining battery charge. The driving range of a BEV depends on the driving pattern and operating conditions [3]. Reliable predictions support informed decisions regarding route planning, charging time, and charging infrastructure selection, thereby reducing the risk of an unplanned vehicle stop. This issue is particularly important for commercial fleets, where prediction errors affect not only driver convenience but also schedule adherence, charging station utilization, and operating costs.

The level of model detail determines the trade-off between representing vehicle dynamics and maintaining computational efficiency in prediction. Therefore, modeling at the level of short trip segments or operational intervals is a promising approach, as it enables the analysis of distance traveled, ambient temperature, and the previous battery state of charge.

According to their underlying principles, models can be classified as physics-based or data-driven. Physics-based models reproduce vehicle dynamics and the operation of powertrain components, including the regenerative braking system [4]. However, they require a large number of parameters to be identified. Data-driven models establish relationships among driving parameters, operating conditions, and battery state of charge without explicitly reproducing all internal physical processes.

The accuracy of driving-range prediction depends on the reliability of lithium-ion battery SoC estimation. Because SoC cannot be measured directly, it is estimated from measurable variables such as current, voltage, and temperature [5–7]. Conventional SoC estimation approaches include ampere-hour counting, model-based methods, state observers, and algorithms from the Kalman filter family [6–11]. Ampere-hour counting is susceptible to the accumulation of current measurement errors, whereas model-based methods require the development and parameterization of a battery model. This group includes methods based on sliding-mode observers [9], approaches using the open-circuit voltage–state of charge (OCV–SoC) relationship [10], and adaptive extended Kalman filter algorithms [11]. The accuracy of these methods depends on the validity of the battery model, the identification of its parameters, the OCV–SoC relationship, and the operating conditions [7].

Current approaches to battery state estimation are advancing along several interrelated directions. One of these involves the joint estimation of SoC, state of health (SoH), and state of function (SoF) [12]. Another direction relies on machine learning and deep

learning methods to capture complex nonlinear relationships between measurable battery parameters and SoC without explicitly specifying a detailed physical model [13–15]. At the same time, model-based methods employing fractional-order models, adaptive filters, and state observers continue to be improved. In particular, fractional-order modeling combined with an adaptive square-root cubature Kalman filter was applied in [16], whereas an incommensurate fractional-order observer was used in [17]. Hybrid models represent another important direction and combine equivalent-circuit battery models, optimization algorithms, and neural networks [18]. A broader integration of physical mechanisms with data-driven learning is also being developed in intelligent transportation systems. In particular, physics-informed and human-guided reinforcement learning has been applied to autonomous driving, where physical models were integrated with learning-based decision-making to improve safety and robustness in complex driving scenarios [19]. Integrating energy consumption parameters and SoC estimation within a unified information framework can improve the accuracy of remaining-range estimation and the efficiency of charging management.

Recent studies demonstrate the rapid development of data-driven approaches to SoC estimation for lithium-ion batteries in electric vehicles. In [20], SoC estimation under different temperature conditions was investigated using current, voltage, temperature, accumulated capacity, and temporal-window features. The authors compared long short-term memory (LSTM), convolutional neural network–long short-term memory (CNN–LSTM), convolutional neural network–bidirectional long short-term memory (CNN–BiLSTM), gated recurrent unit (GRU), and convolutional neural network–gated recurrent unit–long short-term memory (CNN–GRU–LSTM) models. The CNN–GRU–LSTM achieved average MAE and RMSE values of 0.75% and 1.23% when trained jointly on data from −10, 0, 10, and 25 °C, while separate training yielded the best result at 25 °C, with an MAE of 0.41% and an RMSE of 0.61%. Shapley additive explanations (SHAP) were used to quantify the contributions of electrical and temperature-related variables to the SoC predictions.

In [21], SoC estimation for a Volkswagen e-Golf was performed using real-world battery management system signals, including current, voltage, temperature, cumulative energy, and derived temporal features. The authors compared long short-term memory (LSTM), gated recurrent unit (GRU), convolutional neural network–long short-term memory (CNN–LSTM), Informer, residual network (ResNet), ResNet–LSTM, InceptionTime, and a fusion-based hybrid model. Both LSTM and the fusion-based hybrid model achieved an R^2 of 0.996 on the test set. Explainability analysis using the local InputXGradient method and the global Pairwise Importance Estimate Extension method showed that current and voltage were the most influential predictors, while also revealing excessive reliance on correlated variables. These findings were confirmed after revising the feature set and retraining the models.

In [22], a real-time SoC prediction system based on a bidirectional long short-term memory (Bi-LSTM) model was developed using current, voltage, temperature, and their exponentially weighted moving average (EWMA) and exponentially weighted moving standard deviation (EWMS) representations, with SHAP applied for feature selection and interpretation. The model was evaluated on ten driving cycles and compared with the extended Kalman filter (EKF) and unscented Kalman filter (UKF). Bi-LSTM achieved the best MAE of 0.40%, compared with 0.66% for EKF and 1.36% for UKF, while the maximum R^2 reached 0.9999.

In [23], SoC was predicted while accounting for temperature conditions and battery aging. The input variables included current, voltage, temperature, and an aging indicator, while SoC served as the target variable. The authors compared a deep neural network (DNN), GRU, and LSTM using both experimental and simulated data. On the experimental

dataset, the DNN achieved an RMSE of 0.463%, an MAE of 0.341%, and an R^2 of 0.9999. By comparison, GRU yielded an RMSE of 1.152% and an MAE of 0.862%, whereas LSTM produced an RMSE of 1.411% and an MAE of 1.085%. On the simulated dataset, the DNN achieved an RMSE of 0.107% and an MAE of 0.040%. No dedicated explainable artificial intelligence (XAI) methods were applied. Interpretation of the results was limited to error comparisons and analysis of the effects of operating conditions.

A hybrid CNN–UKF approach for SoC estimation was introduced in [24]. The method was evaluated using data from the UDDS, US06, LA92, and an additional load profile at temperatures of 0, 10, and 25 °C. The CNN was trained to capture the nonlinear relationship between the measured battery parameters and SoC, after which the UKF was applied to refine the resulting predictions. The mean RMSE and MAE values achieved by CNN–UKF were 0.00581 and 0.00467, respectively, on the normalized SoC scale adopted by the authors. No XAI methods were used.

Reference [25] presents a combined CNN–LSTM architecture for SoC estimation. The model received time series of measured electrical and thermal battery parameters, including voltage, current, and temperature. The CNN extracted local features from the input sequences, while the LSTM captured long-term temporal dependencies. The proposed architecture achieved an RMSE of 0.92% and an MAE of 0.48%, substantially outperforming the conventional LSTM model, which yielded an RMSE of 2.97% and an MAE of 2.03%. No XAI methods were employed.

Reference [26] introduces a self-supervised Transformer for predicting the SoC of a lithium-ion battery from time series of current, voltage, and temperature. The model was evaluated under both constant and varying ambient temperatures. Under varying-temperature conditions, it achieved an RMSE of no more than 1.19% and an MAE of no more than 0.65%. Under constant-temperature conditions, the corresponding values did not exceed 0.90% and 0.44%, respectively. The authors emphasized that these results were obtained without manual feature engineering or prior signal filtering. No dedicated XAI methods were applied.

Using instantaneous and historical voltage, current, and temperature measurements, the study in [27] predicted SoC with a deep feedforward neural network. The model was trained on data from several standard driving cycles and evaluated across a wide range of temperature conditions. The authors demonstrated that a single DNN could directly map measured voltage, current, and temperature values to SoC while achieving an MAE of approximately 1% or less under the main experimental conditions. No XAI methods were applied.

Reference [28] examines regular, recurrent, and autoregressive recurrent Gaussian process regression (GPR) models for SoC estimation using voltage, current, temperature, and, in the recurrent variants, previous SoC and historical measurements. The autoregressive recurrent GPR achieved an RMSE of 0.2762% and a maximum absolute error of 0.5491%, while the recurrent GPR with one previous SoC value yielded an RMSE of 0.5482% and a maximum absolute error of 1.3872%. In addition to point predictions, GPR provided 95% confidence intervals for uncertainty quantification. No dedicated post hoc XAI methods, such as SHAP or local interpretable model-agnostic explanations (LIME), were applied.

The study reported in [29] proposed an RNARX model whose hyperparameters were optimized using the lightning search algorithm (LSA). Current, voltage, and temperature served as the input variables, while the SoC of the lithium-ion battery was the prediction target. The proposed RNARX–LSA model was compared with BPNN–LSA, RBFNN–LSA, ELM–LSA, DRNN–LSA, and RF–LSA, as well as with RNARX variants optimized using other metaheuristic algorithms. Under a constant 1 C discharge, the model achieved an RMSE of 0.6858%. For the 0.25 C HPPC profile, it yielded an RMSE of 0.4302% and an MAE

of 0.2287%. The corresponding RMSE values for the DST and FUDS cycles were 1.1373% and 1.0268%, respectively. No dedicated XAI methods were employed.

Recent studies have increasingly focused on predicting energy consumption at the individual-trip level before the trip begins, using information available prior to departure. In [30], pre-trip prediction of trip-level SoC consumption for battery electric vehicles was investigated, and a long-tail-aware cost-sensitive ensemble learning framework (LT-ACE) was proposed. The method used pre-trip features related to the driver, vehicle, route, and environment, together with long-tail-aware data augmentation, a gated attention pooling Transformer (GAPFormer), and mileage-aware cost-sensitive learning. On real-world fleet data, LT-ACE achieved an MAE of 0.492%, an RMSE of 1.373%, and an R^2 of 0.958. The method also demonstrated stable performance under limited-data settings and vehicle-type holdout validation, while open-road tests on two previously unseen vehicles provided preliminary external validation.

In [31], a sequence-aware deep learning methodology was proposed for predicting energy consumption before the start of a trip using only realistically available pre-trip data. Unlike approaches that require future driving trajectories or speed profiles, the method models dependencies between consecutive route segments. The best-performing sequence-aware GRU achieved mean MAE, RMSE, and R^2 values of 1.192, 1.647, and 0.98, respectively, across three datasets, reducing MAE by 35.1% compared with the best non-sequence-aware XGBoost model. Compared with conventional methods, the proposed approaches achieved, on average, a 65.2% improvement in prediction accuracy.

The study in [32] presented a Single Feature Prediction Model (SFPM) for pre-trip estimation of energy use in micro electric vehicles based on a single planned input, distance. The proposed Driving-Profile Extractor (DPE) transforms the planned distance into profile-based features, allowing vehicle-specific predictions to be generated without real-time signals. Uncertainty was quantified using bootstrap prediction intervals, which were also used to estimate the available driving range and the probability of successfully completing a trip. The approach was evaluated on real-world data from several vehicle models and across different trip-distance regimes, demonstrating stable and accurate predictions with well-calibrated intervals.

Another recent contribution [33] introduced a Machine Learning-Control Variable (MLCV) model for estimating trip energy consumption in electric vehicles when the available data are limited and uncertain. The method integrates machine learning with the concept of controlled experiments and uses data augmentation techniques, ensemble methods, and sampling factors as model parameters. Its performance was assessed on real-world driving data collected from actual drivers under complex urban driving conditions.

Despite these developments, a large share of SoC estimation research still relies on current or historical measurements of battery voltage, current, and temperature. In most of the examined publications, RMSE values range from approximately 0.3% to 1.6%, although the error may exceed 2% under challenging temperature conditions or when models are transferred to previously unseen operating scenarios. Dedicated XAI methods have been applied in only a limited number of studies, while most publications restrict interpretation to error analysis and comparisons among model architectures. Moreover, many existing models require real-time electrical battery signals that become available only while the vehicle is in motion.

By contrast, predicting the state of charge at the end of an individual operational interval using a limited set of parameters known before the interval begins remains insufficiently investigated. Such parameters include the initial SoC, a specified trip distance, ambient temperature, and the driving range estimated by the onboard system. Comparative evaluation of a broad set of machine learning models, the development of ensemble models

subject to physical constraints, predictive uncertainty quantification, and comprehensive XAI analysis also remain underexplored.

The main contribution of this study is an integrated framework for trip-end SoC prediction in which stacking is combined with cycle-aware validation, explicit physical constraints on Delta SoC, conformal uncertainty quantification, and XAI-based interpretation, rather than being used only as a conventional prediction-combination mechanism.

The aim of this study is to develop and comprehensively evaluate a physically consistent machine learning approach for trip-end SoC prediction by directly predicting the change in battery state of charge over an individual operational interval and determining the final SoC from parameters known before the interval begins, while also quantifying predictive uncertainty and providing XAI-based interpretation of the predictions.

2. Materials and Methods

2.1. Experimental Setup and Data Collection

The experimental data used in this study were collected during the operation of a 2020 Kia Niro EV [34]. Data acquisition covered two interconnected information streams: vehicle charging from the external power grid and the recording of operational parameters while driving. This approach enabled the development of a dataset for training machine learning models designed to predict the current charge level of the high-voltage battery.

During vehicle charging, the amount of electrical energy supplied from the external power grid was measured using a wattmeter installed between the power source and the vehicle charger. The wattmeter directly measured the electrical energy drawn from the grid during charging and displayed the result in kilowatt-hours. This made it possible to monitor the total amount of grid energy consumed by the vehicle during each charging session.

During charging, 230 V alternating current (AC) from the power grid was supplied through the charging cable to the vehicle's charging port. The electrical energy was then transferred to the on-board charger (OBC), which converted AC into direct current (DC). The OBC incorporates an electromagnetic interference input filter, a rectifier, a power factor correction stage, and an isolated DC–DC converter. Together, these components generate the DC voltage required to charge the high-voltage battery.

After conversion, the electrical energy was supplied to the high-voltage battery. The main high-voltage fuse, power contactors, precharge relay and resistor, and current sensor are integrated into the power relay assembly (PRA), which is located inside the high-voltage battery. The charging process was managed by the battery management system (BMS), which monitors cell voltages, current, cell temperatures, and battery SoC. When necessary, the BMS adjusts the charging parameters or terminates the process to ensure the safe operation of the high-voltage battery.

The electrical energy flow during vehicle charging from a single-phase power supply is illustrated in Figure 1.

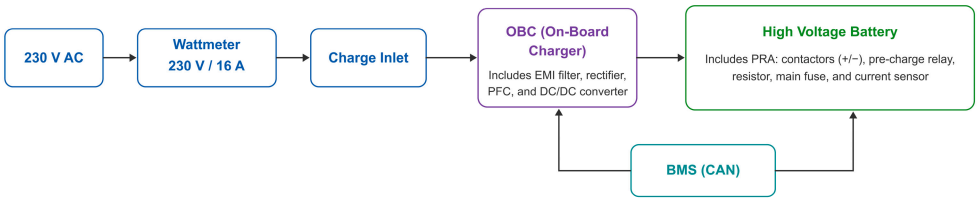


Figure 1. Energy flow during AC charging of the Kia Niro EV.

The wattmeter readings represented the total amount of electrical energy drawn by the vehicle from the power grid. However, some of this energy was dissipated as heat during the operation of the OBC, high-voltage cables, battery cooling system, and auxiliary

electrical loads. Consequently, the amount of electrical energy recorded by the wattmeter was generally greater than the actual increase in energy stored in the high-voltage battery.

Because the wattmeter used in the study was rated for a voltage of 230 V and a maximum current of 16 A, the maximum power that could be measured and supplied through the device was approximately 3.68 kW.

Accordingly, the full capacity of the OBC, which can reach 7.2 kW in the Kia Niro EV, was not utilized in this study. Charging was performed at reduced power from a household single-phase supply, reflecting real-world slow-charging conditions for an electric vehicle.

2.2. Dataset Formation and Description

The dataset used in this study was constructed from sequential vehicle operation records. The initial observations were ordered by vehicle mileage and grouped into individual battery discharge cycles. Each cycle represented a sequence of records collected between two charging events. This grouping preserved the physical and temporal progression of the battery charge level.

An individual observation in the resulting dataset corresponded to an operational interval between two consecutive records belonging to the same discharge cycle. For each operational interval, the preceding record described the vehicle state at the beginning of the interval, while the current record represented its state at the end. An operational interval was created only when both records belonged to the same cycle. This condition prevented intervals from being formed across charging events, since the initial and final states of each interval had to belong to the same continuous battery discharge sequence.

The distance traveled during each operational interval was calculated as the difference between the final and initial odometer readings. The change in battery charge level over the interval was determined as the difference between the initial and final SoC values. After the operational intervals had been formed, the final dataset comprised 352 observations from 81 discharge cycles. Each row represented one operational interval between two consecutive vehicle operating states.

The main characteristics of each operational interval were as follows:

- Previous SoC, %—battery state of charge at the beginning of the operational interval;
- Current SoC, %—battery state of charge at the end of the operational interval;
- Delta SoC, percentage points—change in the state of charge over the operational interval;
- Distance, km—distance traveled;
- Temperature, °C—ambient temperature;
- Previous driving range, km—driving range estimated by the onboard system at the beginning of the operational interval;
- Previous mileage, km and Current mileage, km—initial and final odometer readings, respectively.

Previous SoC ranged from 20% to 100%, with a median of 82%. Current SoC varied from 17% to 98%, with a median of 77%. Delta SoC ranged from 0 to 48 percentage points, while its median value was 3 percentage points.

The distance covered during an individual operational interval ranged from 1 to 354 km, with a median of 13 km. Ambient temperature varied from −9 to 34 °C, with a median of 16 °C. Previous driving range ranged from 31 to 520 km, with a median of 333 km. Previous energy consumption varied from 9.5 to 58.1 kWh/100 km, with a median of 14.6 kWh/100 km. The initial odometer readings for the constructed operational intervals ranged from 82,496 to 91,909 km, while the final readings ranged from 82,513 to 92,046 km. Overall, the dataset covered vehicle operation between odometer readings of 82,496 and 92,046 km, corresponding to a total mileage increase of 9550 km.

Thus, the resulting dataset captured changes in the state of the high-voltage battery between consecutive operational observations while preserving the assignment of interrelated operational intervals to their respective discharge cycles.

2.3. Input Variables and Regression Task Formulation

Only parameters available before the start of the corresponding operational interval or specified in advance for that interval were used to develop the prediction models. The primary input feature set comprised the initial state of charge of the high-voltage battery, travel distance, ambient temperature, and the previously estimated driving range. For the i-th operational interval, the input feature vector was defined as:

$$x_i = \begin{bmatrix} SoC_i^{prev} \\ D_i \\ T_i \\ R_i^{prev} \end{bmatrix}, \quad (1)$$

where x_i is the input feature vector for the i-th operational interval, SoC_i^{prev} is the high-voltage battery state of charge at the beginning of the operational interval (Previous SoC), D_i is the distance associated with the operational interval (Distance), T_i is the ambient temperature (Temperature), and R_i^{prev} is the driving range estimated by the onboard system and available at the beginning of the operational interval (Previous driving range).

The Previous SoC variable represented the initial state of the high-voltage battery. It was known before the start of the operational interval and defined the maximum physically possible decrease in the battery charge level over that interval. For each observation, the condition $0 \leq SoC_i^{prev} \leq 100$ was satisfied. The Distance variable characterized the length of the corresponding operational interval. In the dataset used in this study, its value was calculated as the difference between the final and initial odometer readings and therefore represented the actual distance traveled. During prediction, Distance was treated as a predefined characteristic of the interval; consequently, in the experimental evaluation, the actual distance traveled served as the specified route length. In practical application before the start of a trip, the corresponding input parameter would be the planned route length, which may be obtained from a navigation system or route-planning system. Because the actual trip distance may differ from the planned distance due to route changes, detours, or other circumstances, the effect of uncertainty in Distance on predictive performance was additionally evaluated through sensitivity analysis. Ambient temperature was included in the primary feature set because thermal conditions can affect electric vehicle energy consumption, available battery energy, and the efficiency of auxiliary systems. Temperature was provided to the model as a continuous numerical feature without assigning observations to predefined temperature categories.

The Previous driving range variable represented the driving range displayed by the vehicle's onboard system at the beginning of the operational interval. It provided an aggregate estimate of the available range generated by the vehicle's built-in algorithms based on the current battery state and previous operating conditions. This value did not represent the actual distance traveled and was not used as a target variable. It was included in the model only as an additional input feature.

Previous energy consumption was not included in the primary input feature vector. The discharge cycle number, unique operational interval identifier, initial and final odometer readings, and other administrative fields were not supplied to the models as predictors. The Discharge cycle variable was used only to define independent groups during cross-validation. The odometer readings were used to calculate Distance but were not subsequently included in the primary input feature vector.

The primary regression task was to predict the change in battery state of charge over an individual operational interval. For the i-th observation, the target variable was defined as:

$$y_i = \Delta SoC_i = SoC_i^{prev} - SoC_i^{cur}, \tag{2}$$

where y_i is the target value for the i-th observation, ΔSoC_i is the actual change in state of charge over the operational interval (Delta SoC), and SoC_i^{cur} is the state of charge at the end of the operational interval (Current SoC).

The machine learning task was to determine the functional relationship between the input feature vector and the change in state of charge:

$$\widehat{\Delta SoC_i} = f_\theta\left(SoC_i^{prev}, D_i, T_i, R_i^{prev}\right), \tag{3}$$

where $\widehat{\Delta SoC_i}$ is the predicted change in battery state of charge for the i-th operational interval, and f_θ is a regression model with parameters θ estimated from the training data.

After ΔSoC_i had been predicted, the final state of charge was calculated as:

$$\widehat{SoC_i^{cur}} = SoC_i^{prev} - \widehat{\Delta SoC_i}, \tag{4}$$

where $\widehat{SoC_i^{cur}}$ is the predicted battery state of charge at the end of the i-th operational interval.

Thus, the study considered two related output quantities. Delta SoC was the direct target variable predicted by the regression models, whereas Current SoC was the final calculated output derived from the initial state of charge and its predicted change.

2.4. Data Preprocessing and Cycle-Aware Validation

Data preprocessing and model evaluation were performed with explicit consideration of the grouped structure of the dataset. Because multiple operational intervals belonged to the same discharge cycle, they were not treated as fully independent observations. Assigning intervals from the same cycle to both the training and test sets could have caused information leakage and led to overly optimistic estimates of predictive performance. Therefore, all data splitting procedures were performed at the level of complete discharge cycles.

Input data preprocessing was carried out within the modeling pipelines. The preprocessing pipelines, feature standardization, and group-based data splitting were implemented using the scikit-learn library [35]. Standardization was applied to models that were sensitive to the scale of the input features, since differences in variable ranges could affect both model training and predictive performance. The standardization parameters were estimated using only the data assigned to the training portion of the corresponding model and were then applied unchanged to the associated calibration, validation, and test data. Standardization was not applied to tree-based models because their splitting rules are not affected by the scale of numerical features.

External evaluation was performed using repeated five-fold group cross-validation with GroupKFold [36]. In each repetition, the complete set of discharge cycles was divided into five mutually exclusive groups. Four folds were used as the training portion of the corresponding outer split, while the remaining fold served as the independent test portion. All operational intervals from the same discharge cycle were assigned entirely to only one of these subsets.

To generate different splitting configurations, the unique discharge cycles were randomly shuffled and reassigned group labels before each repetition, after which GroupKFold validation was applied. A random seed of 42 was used to ensure reproducibility. Within each repetition, every operational interval was used for testing once. Because the five-fold

cross-validation procedure was repeated five times, each observation appeared in the test portion five times, once in each repetition. Overall, 25 outer data splits were generated. The final model performance metrics were aggregated across all outer evaluations.

Within each outer split, the training portion was further divided into a model-fitting set and a calibration set. GroupShuffleSplit [37] was used for this purpose, with approximately 20% of the cycles in the training portion allocated to calibration and approximately 80% used for model fitting. The actual proportions could differ slightly because of the discrete number of cycles and the unequal number of operational intervals within individual cycles.

The division into model-fitting and calibration sets was also performed at the level of complete discharge cycles. Therefore, operational intervals from the same cycle could not be assigned to both subsets. Observations in the calibration set were not used to estimate the parameters of the primary regression models, while the outer test observations were used for neither model fitting nor calibration. The overall cycle-aware data-splitting procedure, the internal validation of the stacking ensemble, and the final evaluation are illustrated in Figure 2.

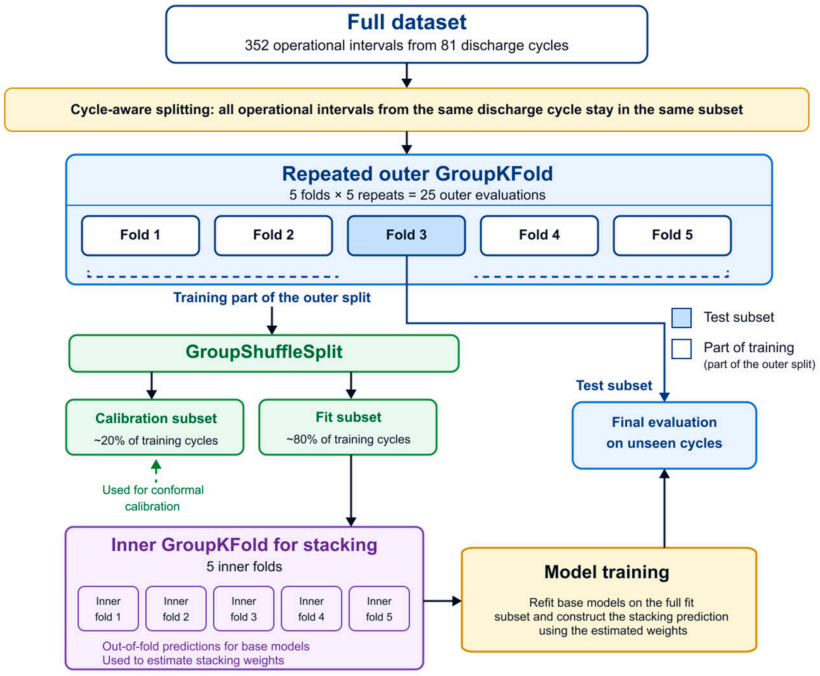


Figure 2. Cycle-aware data splitting, internal stacking validation, and final model evaluation.

To construct the stacking ensemble, the model-fitting set was further divided into five internal folds using GroupKFold. These internal folds were also formed at the level of complete discharge cycles. Within each internal split, a subset of cycles was used to train the base models, while the remaining cycles were used to generate out-of-fold predictions.

This procedure ensured that operational intervals from the same discharge cycle were not used simultaneously to train and evaluate the same model. The resulting out-of-fold predictions were then used to estimate the weights of the stacking ensemble components. After the weights had been determined, the base models were retrained on the complete model-fitting set, and their predictions were combined using the estimated weights.

The trained ensemble was applied to the calibration set to generate the predictions required for conformal calibration. The outer test set was used exclusively for the final evaluation of the models on discharge cycles that had not been involved in model fitting, stacking-weight estimation, or calibration.

The proposed framework maintained group-based data separation at every stage of the study, including outer evaluation, calibration-set formation, and internal training of the stacking ensemble. This reduced the risk of information leakage between related operational intervals and provided a more realistic assessment of the models' ability to generalize to previously unseen discharge cycles.

2.5. Baseline and Machine Learning Models

To assess whether the selected machine learning models were appropriate for the task, their predictive performance was compared with two simple baseline methods: the Persistence Baseline and the Distance-Rate Baseline. Ten regression models were evaluated separately: Random Forest, Extra Trees, Gradient Boosting, Histogram Gradient Boosting, Gaussian Process Regression, Multilayer Perceptron, XGBoost, CatBoost, Explainable Boosting Machine, and NGBoost. All of these models were also available as components of the stacking ensemble described in the following subsection.

The Persistence Baseline served as the simplest reference model, with the predicted final SoC assumed to be equal to the initial SoC. This baseline was not intended to provide a physically realistic representation of battery discharge. Instead, it was used as a basic benchmark for evaluating the predictive effectiveness of more advanced models.

The Distance-Rate Baseline assumed that the change in battery state of charge was directly proportional to the distance traveled:

$$\widehat{\Delta SoC_i^{dist}} = \widehat{\beta_D} D_i, \tag{5}$$

where $\widehat{\Delta SoC_i^{dist}}$ is the change in state of charge predicted by the Distance-Rate Baseline, and $\widehat{\beta_D}$ is the estimated change in SoC per unit distance, expressed in percentage points per kilometer.

The coefficient $\widehat{\beta_D}$ was estimated separately for each training set using least squares regression without an intercept:

$$\widehat{\beta_D} = \max\left(0, \frac{\sum_{i=1}^{N_{fit}} D_i \Delta SoC_i}{\sum_{i=1}^{N_{fit}} D_i^2}\right), \tag{6}$$

where N_{fit} is the number of operational intervals in the model-fitting set, and i is the index of an operational interval within that set.

The max(0,) operation constrained the coefficient to non-negative values and prevented the model from predicting a negative change in SoC.

Hyperparameters were optimized using GridSearchCV [38] with an internal five-fold GroupKFold validation scheme, in which the discharge cycle was used as the grouping variable. A separate grid of admissible hyperparameter values was defined for each algorithm, and the performance of each configuration was evaluated based on the mean RMSE across the internal validation folds. The main search ranges were as follows: for Random Forest and Extra Trees, n_estimators = {500, 700}, max_features = {0.85, 1.00}, and min_samples_leaf = {1, 2}; for Gradient Boosting, n_estimators = {200, 300, 400}, learning_rate = {0.025, 0.05}, max_depth = {2, 3}, and min_samples_leaf = {2, 3}; for Histogram Gradient Boosting, max_iter = {300, 400, 600}, learning_rate = {0.025, 0.045, 0.05}, max_leaf_nodes = {15, 31}, min_samples_leaf = {8, 10}, and l2_regularization = {0, 1}; for Gaussian Process Regression, the hyperparameters of the Matérn covariance kernel and white-noise component were optimized within predefined kernel bounds using multiple optimizer restarts; for MLP, hidden_layer_sizes = {(64), (64, 32), (128, 64)}, alpha = {10^{-4}, 10^{-3}}, and learning_rate_init = {0.001, 0.003}; for XGBoost,

n_estimators = {300, 600}, max_depth = {2, 3}, learning_rate = {0.025, 0.05}, subsample = {0.85, 1.00}, and colsample_bytree = {0.90, 1.00}; for CatBoost, iterations = {400, 700}, depth = {4, 5}, learning_rate = {0.025, 0.05}, and l2_leaf_reg = {3, 5}; for the Explainable Boosting Machine, interactions = {0, 3}, max_bins = {128, 256}, learning_rate = {0.025, 0.05}, and max_rounds = {3000, 5000}; and for NGBoost, n_estimators = {300, 500}, learning_rate = {0.025, 0.05}, and minibatch_frac = {0.8, 1.0}. For each model, the configuration with the lowest mean validation RMSE was selected, after which the model was retrained on the complete fit set of the corresponding outer split. The outer test set was not used during hyperparameter optimization. The same internal GroupKFold validation scheme was also used to generate the out-of-fold predictions required for estimating the stacking ensemble weights.

Because hyperparameter optimization was performed independently within each outer split, the selected configurations could differ across outer splits depending on the composition of the corresponding fit set. Therefore, the search ranges listed above defined the hyperparameter optimization space, whereas the specific parameter values selected in individual outer splits were determined exclusively by the results of the corresponding internal GroupKFold validation. The parameter values reported below correspond to the configurations most frequently selected during hyperparameter optimization across the 25 outer cycle-aware evaluations and therefore represent the most typical settings of the respective models within the adopted search ranges.

Random Forest is an ensemble of regression trees trained on different bootstrap samples, with random feature selection applied when constructing each node [39]. Model predictions were obtained by averaging the outputs of the individual trees, which reduced variance and improved robustness to the characteristics of specific training samples. The model used 700 trees, a maximum feature fraction of 0.85 for node splitting, and a minimum of two observations per leaf node.

Extra Trees also relies on an ensemble of regression trees, but the feature-splitting thresholds are selected with an additional degree of randomness [40]. This approach increases tree diversity relative to Random Forest and may further reduce prediction variance. The model comprised 700 trees, considered all available features when searching for splits, and required at least two observations per leaf node.

Gradient Boosting constructs a sequential ensemble of shallow regression trees, with each tree trained to correct the errors of the preceding ensemble [41]. The model used 300 trees with a maximum depth of 2, a learning rate of 0.025, a minimum of three observations per leaf node, and Huber loss to reduce the influence of individual large errors during training.

Histogram Gradient Boosting implements gradient boosting by first grouping continuous feature values into discrete bins. This reduces computational cost while enabling efficient modeling of nonlinear relationships. The model used 400 boosting iterations, a learning rate of 0.045, no more than 15 leaf nodes, a minimum of eight observations per leaf, and an L2 regularization parameter of 1.0.

Gaussian Process Regression treats the relationship between the input features and Delta SoC as a random function defined by a covariance kernel [42]. A Matérn kernel with a smoothness parameter of 1.5 was combined with a white-noise component. Input features were standardized, the target variable was normalized, and kernel hyperparameter optimization was restarted multiple times.

The Multilayer Perceptron is a fully connected neural network [43] with two hidden layers containing 64 and 32 neurons. The hidden layers used the rectified linear unit activation function, and the network weights were optimized with the Adam algorithm.

L2 regularization, an adaptive learning rate, and early stopping were applied to limit overfitting, with a maximum of 5000 training iterations.

XGBoost is a regularized implementation of gradient boosting over decision trees [44]. The model sequentially added trees that reduced the error of the current ensemble, while L1 and L2 regularization constrained model complexity. The configuration included 600 trees with a maximum depth of 3, a learning rate of 0.025, a subsample ratio of 0.85, and a feature sampling ratio of 0.9 for each tree.

CatBoost is another gradient-boosted decision tree model [45]. In this study, it was used to capture complex nonlinear relationships between operational parameters and Delta SoC. The model was configured with 700 boosting iterations, a tree depth of 5, a learning rate of 0.025, RMSE loss, and an L2 leaf regularization parameter of 5.0.

The Explainable Boosting Machine is a generalized additive model in which predictions are expressed as the sum of individual nonlinear feature contributions and a limited number of pairwise interactions [46]. This structure combines nonlinear predictive capability with direct interpretability of the effect of each variable. The model used three pairwise interactions, 128 bins, a learning rate of 0.025, and up to 5000 boosting rounds.

NGBoost is a probabilistic gradient boosting model [47] that estimates the parameters of the conditional distribution of Delta SoC rather than only a point value. The expected value of the predicted normal distribution was used as the point prediction. The model was trained using the logarithmic score and natural gradient and comprised 500 boosting estimators with a learning rate of 0.025.

The same cycle-aware outer splits described in Section 2.4 were used for all baseline and machine learning models. Training was performed exclusively on the corresponding fit subsets, whereas evaluation was conducted on independent outer test cycles. A random seed of 42 was used for stochastic algorithms. The computations were performed using Python 3.9.13. The main software packages included NumPy 1.26.4, pandas 2.3.3, scikit-learn 1.6.1, XGBoost 2.1.4, CatBoost 1.2.10, interpret 0.7.8, NGBoost 0.5.11, SHAP 0.49.1, Matplotlib 3.9.4, and joblib 1.5.3. The predictions generated by the individual models were subsequently used as components of the physics-constrained stacking ensemble.

2.6. Physics-Constrained Stacking Ensemble

To improve predictive accuracy and robustness, a Physics-Constrained Stacking Ensemble was developed by combining the outputs of heterogeneous base regression models. The ensemble comprised Random Forest, Extra Trees, Gradient Boosting, Histogram Gradient Boosting, Gaussian Process Regression, Multilayer Perceptron, XGBoost, CatBoost, Explainable Boosting Machine, and NGBoost.

For the i-th operational interval, the change in state of charge predicted by the m-th base model was denoted as:

$$\widehat{\Delta SoC_{i,m}} = f_m(x_i), \tag{7}$$

where $\widehat{\Delta SoC_{i,m}}$ is the change in state of charge predicted for the i-th operational interval by the m-th base model, and $f_m(\cdot)$ is the prediction function of the m-th base model.

The ensemble component weights were estimated separately within each outer split. For this purpose, out-of-fold predictions were generated on the fit set using internal five-fold GroupKFold validation.

The out-of-fold predictions from all base models were combined into the matrix:

$$P_{OOF} = \begin{bmatrix} \widehat{\Delta SoC_{1,1}^{OOF}} & \cdots & \widehat{\Delta SoC_{1,M}^{OOF}} \\ \vdots & \ddots & \vdots \\ \widehat{\Delta SoC_{N_{fit},1}^{OOF}} & \cdots & \widehat{\Delta SoC_{N_{fit},M}^{OOF}} \end{bmatrix}, \tag{8}$$

where P_{OOF} is the out-of-fold prediction matrix of size $N_{fit} \times M$, $\widehat{\Delta SoC_{i,m}^{OOF}}$ is the out-of-fold prediction of the change in state of charge for the i-th operational interval generated by the m-th base model ($i = 1, \ldots, N_{fit}$, $m = 1, \ldots, M$), N_{fit} is the number of operational intervals in the fit set, and M is the number of models in the stacking ensemble.

The vector of observed target values for this set was expressed as:

$$y_{fit} = \begin{bmatrix} \Delta SoC_1 \\ \vdots \\ \Delta SoC_{N_{fit}} \end{bmatrix}, \tag{9}$$

The stacking ensemble weights were estimated using non-negative least squares (NNLS) [48]:

$$\widehat{w} = \arg\min_{w \geq 0} \left\| y_{fit} - P_{OOF} w \right\|_2^2, \tag{10}$$

where $\widehat{w}$ is the estimated vector of stacking ensemble weights, $w = [w_1, w_2, \ldots, w_M]$ is the vector of base-model weights, and $\|\cdot\|_2^2$ denotes the squared Euclidean norm. The constraint $w \geq 0$ requires every weight to be non-negative.

The non-negativity constraint prevented the predictions of some models from being offset by large negative contributions from others and simplified interpretation of the ensemble components. After estimation, the coefficients were normalized so that their sum was equal to one:

$$\widetilde{w}_m = \frac{\widehat{w}_m}{\sum_{j=1}^{M} \widehat{w}_j}, \tag{11}$$

where $\widetilde{w}_m$ is the normalized weight of the m-th base model, and $\widehat{w}_m$ is the corresponding coefficient obtained using NNLS.

The normalized weights satisfied the following conditions:

$$\widetilde{w}_m \geq 0, \quad \sum_{m=1}^{M} \widetilde{w}_m = 1, \tag{12}$$

These conditions ensured that the ensemble prediction was formed as a non-negative weighted combination of the base-model predictions.

After the weights had been determined, each base model was retrained on the complete fit set of the corresponding outer split. For a new i-th operational interval, the unconstrained prediction of the stacking ensemble was defined as:

$$\widehat{\Delta SoC_i^{raw}} = \sum_{m=1}^{M} \widetilde{w}_m \widehat{\Delta SoC_{i,m}}, \tag{13}$$

where $\widehat{\Delta SoC_i^{raw}}$ is the change in state of charge predicted by the stacking ensemble before the physical constraints are applied, $\widetilde{w}_m$ is the normalized weight of the m-th base model, and $\widehat{\Delta SoC_{i,m}}$, is the prediction generated by the m-th base model.

Despite the non-negative stacking weights, the structure of the individual base models did not guarantee that the predicted change in SoC would remain within physically admissible limits. Therefore, before the physical constraints were applied, the predicted Delta SoC could theoretically be negative or exceed the initial state of charge. In the first case, the reconstructed Current SoC would be greater than Previous SoC, which would be inconsistent with the conditions of the operational interval in the absence of charging. In the second case, the predicted charge reduction would exceed the available initial state of charge, resulting in a negative predicted Current SoC.

To eliminate such physically inadmissible results, the following physical projection was applied to the unconstrained prediction:

$$\widehat{\Delta SoC_i^{phys}} = \min\left(SoC_i^{prev}, \max\left(0, \widehat{\Delta SoC_i^{raw}}\right)\right), \tag{14}$$

where $\widehat{\Delta SoC_i^{phys}}$ is the physically admissible prediction of the change in state of charge.

The $\max(0, \cdot)$ function eliminates negative Delta SoC values, while the $\min\left(SoC_i^{prev}, \cdot\right)$ function limits the predicted change to the available initial state of charge.

Thus, the physically admissible prediction satisfied the condition:

$$0 \leq \widehat{\Delta SoC_i^{phys}} \leq SoC_i^{prev}, \tag{15}$$

where $\widehat{\Delta SoC_i^{phys}}$ is the physically admissible prediction of the change in state of charge.

The predicted final state of charge was then calculated from the corrected Delta SoC prediction using Equation (4).

The applied projection ensured that the following constraint was satisfied:

$$0 \leq \widehat{SoC_i^{cur}} \leq SoC_i^{prev} \leq 100, \tag{16}$$

Thus, the predicted final SoC could not be negative, exceed the initial state of charge, or lie above the physical upper limit of 100%.

The ensemble weights were estimated separately for each outer split. Therefore, they reflected the structure of the corresponding fit set and did not incorporate any information from the outer test cycles. After the ensemble components had been retrained, the ensemble was used to predict Delta SoC for the operational intervals in the calibration and test sets. The physical projection was then applied to the resulting predictions. The calibration-set predictions were subsequently used to construct conformal prediction intervals.

2.7. Conformal Prediction and Uncertainty Quantification

To complement the point predictions with an assessment of their uncertainty, conformal prediction was applied. Unlike conventional point prediction, this approach generated an individual interval for each operational interval within which the observed Delta SoC value was expected to fall. Split conformal prediction was used in this study. The models were trained on the fit set, while a separate calibration set was used exclusively to determine the widths of the prediction intervals [49]. The calibration and test data comprised different discharge cycles. Therefore, the test observations were used neither for model training nor for interval calibration.

For the j-th operational interval in the calibration set, the absolute prediction error was defined as:

$$r_j = \left|\Delta SoC_j - \widehat{\Delta SoC_j^{phys}}\right|, \tag{17}$$

where r_j is the absolute calibration error for the j-th operational interval, ΔSoC_j is the observed change in state of charge, $\widehat{\Delta SoC_j^{phys}}$ is the Delta SoC prediction after applying the physical constraints, $j = 1, \ldots, N_{cal}$, and N_{cal} is the number of operational intervals in the calibration set.

Because predictive uncertainty could vary across operational intervals, normalized split conformal prediction was selected as the primary approach. In this case, the absolute error was normalized by a local estimate of predictive uncertainty:

$$\eta_j = \frac{r_j}{\max(s_j,\ s_{min})}, \tag{18}$$

where η_j is the normalized conformal score for the j-th calibration operational interval, s_j, is the local estimate of predictive uncertainty, and s_{min} is the minimum allowable value of the scaling factor. In this study $s_{min} = 0.25$ was used to prevent division by a very small value and excessive inflation of the normalized errors.

The baseline local uncertainty estimate was derived from the degree of disagreement among the predictions of the stacking ensemble components. For the i-th operational interval, it was defined as:

$$d_i = \sqrt{\sum_{m=1}^{M} \widetilde{w}_m \left(\widehat{\Delta SoC_{i,m}} - \widehat{\Delta SoC_i^{raw}}\right)^2}, \tag{19}$$

A larger value of d_i indicated greater disagreement among the ensemble component predictions and, consequently, higher local uncertainty.

To account for the increase in uncertainty associated with longer operational intervals, the scale was additionally adjusted according to the distance traveled. The distance component was defined as:

$$c_i = \frac{\sqrt{\max(D_i, 0) + 1}}{\mathrm{median}\left(\sqrt{\max(D, 0) + 1}\right)}, \tag{20}$$

where c_i is the normalized distance coefficient for the i-th operational interval, D_i is the distance traveled, D is the set of distance values in the corresponding dataset subset, and median(·) denotes the median.

Adding one ensured that the expression remained positive even when the distance was zero, while the square root reduced the excessive influence of very long operational intervals.

The final local uncertainty estimate was defined as:

$$s_i = \max(s_{min}, \max(d_i, s_{min})c_i), \tag{21}$$

where s_i is the local uncertainty estimate for the i-th operational interval, d_i is the disagreement among the ensemble component predictions, and $s_{min} = 0.25$ is the predefined lower bound for the local uncertainty estimate.

The prediction disagreement was first bounded below by 0.25 and then multiplied by the normalized distance adjustment coefficient. The same lower bound was subsequently applied to the resulting value.

To construct an interval with a nominal coverage level of $1-\alpha$, the conformal quantile of the normalized scores was used. In this study, α = 0.10, corresponding to a nominal coverage level of $1 - \alpha = 0.90$. Thus, 90% conformal prediction intervals were generated.

The adjusted quantile level was defined as:

$$q_{adj} = \min\left(1, \frac{\lceil (N_{cal} + 1)(1 - \alpha) \rceil}{N_{cal}}\right), \tag{22}$$

where N_{cal} is the number of operational intervals in the calibration set.

The conformal quantile of the normalized calibration nonconformity scores was calculated as:

$$\widehat{q}_{1-\alpha} = Q_{q_{adj}}(\eta_1, \eta_2, \ldots, \eta_{N_{cal}}), \tag{23}$$

where $\widehat{q}_{1-\alpha}$ is the estimated conformal quantile of the normalized calibration nonconformity scores, and $Q_{q_{adj}}(\cdot)$ is the empirical quantile operator at level q_{adj}.

For the i-th operational interval in the test set, the half-width of the conformal prediction interval was defined as:

$$h_i = \widehat{q}_{1-\alpha} \max(s_i, s_{min}), \tag{24}$$

The initial bounds of the prediction interval for Delta SoC were defined as:

$$\begin{aligned} L_i^{\Delta,raw} &= \Delta\widehat{SoC}_i^{phys} - h_i \\ U_i^{\Delta,raw} &= \Delta\widehat{SoC}_i^{phys} + h_i \end{aligned}, \tag{25}$$

where $L_i^{\Delta,raw}$ and $U_i^{\Delta,raw}$ are the lower and upper interval bounds before the physical constraints are applied.

Because the change in SoC could neither be negative nor exceed the initial state of charge, the interval bounds were additionally restricted to the physically admissible range:

$$\begin{aligned} L_i^{\Delta} &= \max\left(0, L_i^{\Delta,raw}\right) \\ U_i^{\Delta} &= \min\left(SoC_i^{prev}, U_i^{\Delta,raw}\right) \end{aligned}, \tag{26}$$

where L_i^{Δ} and U_i^{Δ} are the physically adjusted lower and upper bounds of the prediction interval for Delta SoC, respectively.

If the upper bound became smaller than the lower bound after applying the constraints, it was set equal to the lower bound. This procedure ensured that the following condition was satisfied:

$$0 \le L_i^{\Delta} \le U_i^{\Delta} \le SoC_i^{prev}, \tag{27}$$

The prediction interval for Current SoC was obtained by transforming the bounds of the Delta SoC interval:

$$\begin{aligned} L_i^{cur} &= \max\left(0, SoC_i^{prev} - U_i^{\Delta}\right) \\ U_i^{cur} &= \min\left(SoC_i^{prev}, SoC_i^{prev} - L_i^{\Delta}\right) \end{aligned}, \tag{28}$$

where L_i^{cur} and U_i^{cur} are the lower and upper bounds of the prediction interval for Current SoC.

Because a larger Delta SoC corresponds to a lower Current SoC, the upper bound of the Delta SoC interval was used to determine the lower bound of the Current SoC interval, whereas the lower bound of the Delta SoC interval was used to determine its upper bound.

The quality of uncertainty estimation was assessed using empirical coverage and prediction interval width. For Delta SoC, empirical coverage was defined as:

$$C_{\Delta} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathrm{I}\left(L_i^{\Delta} \le \Delta SoC_i \le U_i^{\Delta}\right), \tag{29}$$

where C_{Δ} is the empirical coverage of the prediction intervals for Delta SoC, N_{test} is the number of operational intervals in the test set, and $\mathrm{I}(\cdot)$ is the indicator function, which equals 1 when the condition is satisfied and 0 otherwise.

Similarly, empirical coverage for Current SoC was defined as:

$$C_{cur} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} \mathrm{I}(L_i^{cur} \leq SoC_i^{cur} \leq U_i^{cur}), \tag{30}$$

where C_{cur} is the empirical coverage of the prediction intervals for Current SoC.

The prediction interval width for an individual operational interval was defined as:

$$\begin{aligned} W_i^{\Delta} &= U_i^{\Delta} - L_i^{\Delta} \\ W_i^{cur} &= U_i^{cur} - L_i^{cur}, \end{aligned} \tag{31}$$

where W_i^{Δ} and W_i^{cur} are the prediction interval widths for Delta SoC and Current SoC, respectively.

To summarize prediction interval widths within the test set, aggregate measures were calculated from all W_i^{cur} and W_i^{Δ} values. These measures included the mean, median, and maximum widths for both W_{cur} and W_{Δ}. A narrower interval, provided that nominal coverage was maintained, indicated more precise uncertainty quantification.

For a comprehensive evaluation of the prediction intervals, the interval score was also used. This metric simultaneously accounted for interval width and penalized cases in which the observed target value fell outside the interval bounds. For the *i*-th operational interval in the test set, the interval score was defined as:

$$IS_{\alpha,i} = (U_i - L_i) + \frac{2}{\alpha}(L_i - y_i)\mathrm{I}(y_i < L_i) + \frac{2}{\alpha}(y_i - U_i)\mathrm{I}(y_i > U_i), \tag{32}$$

where $IS_{\alpha,i}$ is the interval score for the i-th operational interval in the test set, and L_i and U_i are general notations for the lower and upper bounds of the prediction interval for the corresponding target variable.

The first term represents the prediction interval width. The second and third terms impose penalties when the observed value falls below the lower bound or above the upper bound, respectively. For prediction intervals containing the observed value, the interval score was equal to the interval width. When the observed value lay outside the interval, the score increased in proportion to the magnitude of the uncovered deviation. The penalty coefficient $2/\alpha$ was equal to 20. This substantially increased the interval score for narrow prediction intervals that failed to cover the observed value.

The mean interval score for the corresponding test set was calculated as:

$$\overline{IS}_{\alpha} = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} IS_{\alpha,i}, \tag{33}$$

A lower value of $\overline{IS}_{\alpha}$ indicated a better balance between narrow prediction intervals and adequate empirical coverage. This metric was therefore used to compare normalized split conformal prediction with alternative conformal methods.

In addition to the primary normalized split conformal prediction approach, the comparative analysis included unnormalized split conformal prediction, capped normalized conformal prediction, Mondrian conformal prediction with distance-based grouping, and conformalized quantile regression. Unnormalized split conformal prediction used a single quantile of the absolute calibration errors, so all test operational intervals received the same baseline interval half-width. In capped normalized conformal prediction, the local scale was limited by an upper quantile to prevent excessive widening of individual intervals. In the Mondrian approach, the calibration and test observations were divided into three groups according to the distance traveled, and the conformal quantile was estimated sepa-

rately for each group. Conformalized quantile regression used initial conditional quantile predictions followed by conformal calibration [50]. These variants were compared in terms of empirical coverage and prediction interval width, while normalized split conformal prediction remained the primary uncertainty quantification method.

2.8. Performance Evaluation and Statistical Analysis

Predictive performance was evaluated on the outer test sets generated using repeated cycle-aware GroupKFold validation. Five outer folds and five repetitions were used, yielding 25 performance estimates for each model across the outer splits. Within each outer split, the model was trained exclusively on the corresponding fit set, while the test cycles were used only for final evaluation. The metrics were calculated separately for the directly predicted change in state of charge, Delta SoC, and the reconstructed final state of charge, Current SoC.

For the i-th operational interval in the test set, the Delta SoC prediction error was defined as:

$$e_i^{\Delta} = \Delta SoC_i - \widehat{\Delta SoC}_i, \tag{34}$$

where ΔSoC_i is the observed change in state of charge, $\widehat{\Delta SoC}_i$ is the predicted change in state of charge after applying the physical constraints specified for the corresponding model, $i = 1, \ldots, N_{test}$, and N_{test} is the number of operational intervals in the corresponding outer test set.

The Current SoC prediction error was defined analogously:

$$e_i^{cur} = SoC_i^{cur} - \widehat{SoC}_i^{cur}, \tag{35}$$

where SoC_i^{cur} is the observed state of charge at the end of the i-th operational interval, and $\widehat{SoC}_i^{cur}$ is the corresponding predicted state of charge.

The primary accuracy metric was the root mean square error:

$$RMSE = \sqrt{\frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (y_i - \widehat{y}_i)^2}, \tag{36}$$

where y_i is the observed value of the target variable being evaluated for the i-th operational interval, and $\widehat{y}_i$ is the corresponding predicted value.

Depending on the task being analyzed, y_i denoted either ΔSoC_i or SoC_i^{cur}. RMSE values were expressed in percentage points. Because individual errors were squared before averaging, large deviations had a greater influence on *RMSE* than on *MAE*. Taking the square root then returned the metric to the units of the target variable.

The mean absolute error was defined as:

$$MAE = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} |y_i - \widehat{y}_i|, \tag{37}$$

MAE represented the average magnitude of the prediction error regardless of its direction and was less sensitive to individual large deviations than *RMSE*.

The coefficient of determination was calculated as:

$$R^2 = 1 - \frac{\sum_{i=1}^{N_{test}} (y_i - \widehat{y}_i)^2}{\sum_{i=1}^{N_{test}} (y_i - \overline{y})^2}, \tag{38}$$

where $\overline{y}$ is the mean observed value of the target variable in the corresponding test set.

To additionally evaluate the relative magnitude of the prediction error, the mean absolute percentage error was used:

$$MAPE = \frac{100\%}{N_{test}} \sum_{\substack{i=1 \\ |y_i|>\varepsilon}}^{N_{test}} \left| \frac{y_i - \widehat{y}_i}{y_i} \right|, \tag{39}$$

where ε is a small positive constant added to ensure numerical stability.

The symmetric mean absolute percentage error was defined as:

$$sMAPE = \frac{100\%}{N_{test}} \sum_{\substack{i=1 \\ (|y_i|+|\widehat{y}_i|)>\varepsilon}}^{N_{test}} \frac{2|y_i - \widehat{y}_i|}{|y_i| + |\widehat{y}_i| + \varepsilon}, \tag{40}$$

Unlike *MAPE*, this metric treated the observed and predicted values symmetrically.

Systematic prediction bias was evaluated using the mean error:

$$ME = \frac{1}{N_{test}} \sum_{i=1}^{N_{test}} (\widehat{y}_i - y_i), \tag{41}$$

A positive *ME* indicated that the model predominantly overestimated the target variable, whereas a negative *ME* indicated predominant underestimation. An ME value close to zero suggested the absence of pronounced systematic prediction bias. However, by itself, it did not guarantee small errors for individual operational intervals.

The maximum absolute error was defined as:

$$MaxError = \max_{1 \le i \le N_{test}} |y_i - \widehat{y}_i|, \tag{42}$$

This metric was used to evaluate the worst individual prediction case.

For each model and each metric, values were first obtained for all 25 outer splits. The aggregated results were reported as the mean and standard deviation:

$$\begin{aligned} \overline{G} &= \frac{1}{K} \sum_{k=1}^{K} G_k \\ SD_G &= \sqrt{\frac{1}{K-1} \sum_{k=1}^{K} \left(G_k - \overline{G}\right)^2}, \end{aligned} \tag{43}$$

where G_k is the value of the corresponding metric on the k-th outer split, $\overline{G}$ is the mean metric value, SD_G is its standard deviation, and $K = 25$ is the total number of outer evaluations.

For *RMSE*, *MAE*, *MAPE*, *sMAPE*, $|ME|$ and *MaxError*, lower values indicated better performance, whereas higher values were considered better for R^2.

For the mean metric values obtained across the 25 outer splits, 95% confidence intervals were calculated using Student's t-distribution. The interval bounds were determined from the mean value of the corresponding metric, its standard error, and the critical value of the t-distribution with $K - 1$ degrees of freedom, where $K = 25$. These intervals characterized the uncertainty of the model's mean performance across the outer cycle-aware splits.

The uncertainty of the aggregated accuracy measures was additionally assessed using cluster bootstrap at the discharge-cycle level. Because each operational interval appeared in the outer test sets across five repetitions, repeated predictions for the same operational interval were first averaged. Thus, repeated predictions for the same operational interval were not treated as independent observations during bootstrap evaluation. After averaging the repeated predictions, resampling was performed at the level of complete discharge cycles, thereby further preserving the dependence among operational intervals belonging

to the same discharge cycle. Bootstrap samples were then generated with replacement from the complete set of discharge cycles in 2000 iterations. In each iteration, all operational intervals belonging to the selected cycles were included, and RMSE and MAE were recalculated for both Delta SoC and Current SoC. The bounds of the 95% confidence interval were determined using the empirical 0.025 and 0.975 quantiles of the resulting bootstrap distribution.

To further evaluate the robustness of Physics-Constrained Stacking across different operating conditions, a stratified performance analysis by Distance and Temperature was performed after averaging the five repeated outer-test predictions for each operational interval. Based on Distance, the operational intervals were divided into short ($\leq$9 km), medium (10–18 km), and long (>18 km) trips, whereas Temperature was categorized into cold (<5 °C), moderate (5–<20 °C), and warm ($\geq$20 °C) conditions. For each group, Current SoC RMSE, MAE, R^2, and the proportion of predictions with an absolute error not exceeding 2 percentage points were calculated separately.

Pairwise statistical comparisons of the models were performed using the Current SoC RMSE values obtained on the same outer splits. Accordingly, the 1760 pooled test predictions were not treated as independent observations in the pairwise statistical tests. Instead, the comparison was based on paired RMSE values obtained from the same 25 outer splits for each pair of models. Because estimates from repeated cross-validation are also not fully independent due to overlapping training sets across splits, the corrected resampled *t*-test was used alongside the Wilcoxon signed-rank test to account for this dependence when estimating the standard error. For each pair of models, the difference between their Current SoC RMSE values was calculated for each of the 25 outer splits. A negative difference indicated a lower error and better performance of the first model, whereas a positive difference indicated an advantage of the second model.

Because the distribution of the paired differences in Current SoC RMSE was not necessarily normal, the Wilcoxon signed-rank test was used as the primary nonparametric comparison method. The null hypothesis assumed no systematic difference between the models, whereas the two-sided alternative hypothesis indicated the presence of such a difference. To control the family-wise Type I error rate across multiple pairwise comparisons, the *p*-values from the Wilcoxon signed-rank tests were adjusted using the Holm method. Differences with an adjusted $p < 0.05$ were considered statistically significant. The adjusted *p*-values were interpreted together with the direction and magnitude of the difference, because statistical significance alone did not indicate a practically meaningful advantage of one model over another.

To account for the dependence among the Current SoC RMSE values obtained through repeated cross-validation, the corrected resampled *t*-test was additionally applied [51]. This correction was required because different outer splits partially used the same operational intervals for training, meaning that the resulting metric values could not be considered fully independent. In the corrected test, the variance of the paired differences was increased to account for the ratio between the test-set and training-set sizes. This provided a more conservative assessment of statistical significance than the conventional paired *t*-test.

The practical magnitude of the differences between models was characterized using the paired effect size Cohen's d_z. For each pair of models, it was calculated as the ratio of the mean paired difference in Current SoC RMSE to the standard deviation of those differences. The sign of d_z reflected the direction of the advantage according to the adopted subtraction order, while its absolute value characterized the magnitude of the difference between the models. This made it possible to assess not only statistical significance but also the practical magnitude of the observed difference.

For the probabilistic interpretation of the paired differences in Current SoC RMSE, the Bayesian correlated *t*-test was applied [52]. It used the Nadeau–Bengio corrected standard error, analogous to that employed in the corrected resampled *t*-test, while the posterior distribution was derived from Student's *t*-distribution. The region of practical equivalence was defined as ± 0.10 percentage points. Based on the resulting posterior distribution, the probability that the first model was superior, the probability that the models were practically equivalent, and the probability that the second model was superior were determined. Because lower Current SoC RMSE values indicated better predictive accuracy, the direction of superiority was interpreted according to the order used to calculate the paired difference.

The combined use of *t*-based confidence intervals, cycle-level cluster bootstrap confidence intervals, the Wilcoxon signed-rank test with Holm correction, the corrected resampled *t*-test, the paired effect size Cohen's d_z and the Bayesian correlated *t*-test provided a comprehensive comparison of the models in terms of result stability, statistical significance, the practical magnitude of the differences, and the probability of practical equivalence.

The final ranking of the models was based primarily on the mean Current SoC RMSE obtained on the outer test sets. However, identification of the best-performing model was not limited to a single metric. MAE, R^2, result stability across the outer splits, confidence intervals, pairwise statistical test results, practical effect size, and the characteristics of the conformal prediction intervals were also considered. The combined evaluation of these indicators made it possible to distinguish a marginal improvement in an individual metric from a robust and practically meaningful model advantage.

2.9. Ablation and Sensitivity Analysis Methodology

To evaluate the contribution of individual base models to the Physics-Constrained Stacking result, a component ablation study was conducted. In this analysis, the input feature set remained unchanged, while only the composition of the stacking ensemble was modified. The primary objective was to determine whether each base model made a useful contribution to the final prediction and whether the ensemble could be simplified without a substantial loss of accuracy.

The full ensemble composition was denoted as:

$$M_{full} = \{f_1, f_2, \ldots, f_M\} \tag{44}$$

where M_{full} is the complete set of stacking ensemble components, f_m is the m-th base regression model, $m = 1, \ldots, M$, and M is the total number of base models in the full ensemble.

The full ensemble comprised Random Forest, Extra Trees, Gradient Boosting, Histogram Gradient Boosting, Gaussian Process Regression, Multilayer Perceptron, XGBoost, CatBoost, Explainable Boosting Machine, and NGBoost. All components used the same input feature set and were trained on the same fit sets for the corresponding outer split.

For each ablation configuration, the component weights were not transferred from the full ensemble. Instead, they were re-estimated using the out-of-fold predictions of the models included in the corresponding configuration.

The primary ablation approach was leave-one-component-out analysis. For each base model, a reduced ensemble was formed by removing that model from the full set:

$$M_{-m} = M_{full} \setminus \{f_m\} \tag{45}$$

where M_{-m} is the ensemble composition after removing the m-th base model, and f_m is the excluded model.

For each reduced ensemble, the weights were re-estimated using the corresponding subset of internal out-of-fold predictions. The test predictions of the already trained components were then combined using the newly estimated weights.

The contribution of the m-th model to ensemble accuracy was evaluated from the change in RMSE:

$$\Delta RMSE_m = RMSE_{-m} - RMSE_{full} \tag{46}$$

where $\Delta RMSE_m$ is the change in RMSE after removing the m-th base model, $RMSE_{-m}$ is the RMSE of the reduced ensemble without that model, and $RMSE_{full}$ is the RMSE of the full Physics-Constrained Stacking ensemble.

A positive value of $\Delta RMSE_m$ indicated that removing the component reduced predictive accuracy, meaning that the model made a useful contribution to the ensemble prediction. A value close to zero indicated a limited unique contribution or suggested that the predictive information provided by the component could be reproduced by the other models. A negative value indicated that RMSE decreased after the component was removed, which could suggest redundancy or an adverse effect of its predictions on the ensemble's generalization performance. The change in MAE was evaluated analogously.

The relative change in RMSE was defined as:

$$\Delta RMSE_m^{rel} = \frac{RMSE_{-m} - RMSE_{full}}{RMSE_{full}} \cdot 100\% \tag{47}$$

The change in the coefficient of determination was defined as:

$$\Delta R_m^2 = R_{full}^2 - R_{-m}^2, \tag{48}$$

where ΔR_m^2 is the change in the coefficient of determination after removing the m-th model, R_{full}^2 is the value for the full ensemble, and R_{-m}^2 is the value for the reduced ensemble.

A positive value indicated a deterioration in performance after removal of the corresponding component. All configurations were evaluated on the same outer splits. For each variant, RMSE, MAE, and R^2 were calculated for both Delta SoC and Current SoC. Stacking-component ablation made it possible to distinguish the formal complexity of the ensemble from the actual utility of its components. When removal of a particular model led to a consistent deterioration in the metrics, that model was considered an important ensemble component. When a reduced configuration achieved accuracy comparable to that of the full ensemble, this indicated that the stacking ensemble could be simplified without a substantial loss of predictive performance. A targeted feature-ablation analysis was additionally conducted to assess the dependence of Physics-Constrained Stacking on Previous driving range. The primary Physics-Constrained Stacking configuration included Previous SoC, Distance, Ambient Temperature, and Previous driving range. In the reduced configuration, Previous driving range was removed from the input feature set, while the stacking ensemble composition, cycle-aware evaluation protocol, inner GroupKFold procedure, NNLS weighting, and physical projection remained unchanged. After removing this feature, all base models were retrained, and the stacking weights were re-estimated using the inner out-of-fold predictions. The reduced configuration was evaluated using the same procedure as the primary model configuration, and the change in predictive performance was quantified in terms of Current SoC RMSE relative to the main Physics-Constrained Stacking result.

A sensitivity analysis was also conducted to assess the effect of uncertainty in the specified route length on the predictive performance of Physics-Constrained Stacking. The analysis was performed without retraining the models: for each of the 25 outer test splits, the same fitted base models and the corresponding NNLS stacking weights obtained during

the primary cycle-aware evaluation were used. At inference time, the Distance value was independently perturbed by −10%, −5%, +5%, and +10% relative to its original value, while Previous SoC, Ambient Temperature, and Previous driving range were kept unchanged. For each perturbation level, predictions from the base models were recomputed, the stacking prediction was formed using the stored weights, the same physical projection was applied, and Current SoC was determined. For each scenario, Current SoC RMSE, MAE, and R^2 were calculated, together with the changes in RMSE and MAE relative to the baseline scenario with unmodified Distance. In addition, the mean absolute change in predicted Current SoC relative to the baseline prediction was determined. Before conducting the sensitivity analysis, it was verified that inference with the unmodified Distance exactly reproduced the predictions obtained in the primary experiment. A sensitivity analysis of the stacking-weight estimation strategy was also performed to determine whether the variability of NNLS weights across outer splits affected the predictive stability of the ensemble. The primary NNLS configuration was compared with equal-weight averaging and a Ridge-based meta-learner. For all three variants, the same inner out-of-fold predictions from the base models and the same outer-test predictions as in the primary experiment were used, so the base models were not retrained. In the equal-weight configuration, the predictions of the ten base models were averaged using identical weights. For the Ridge configuration, a linear meta-learner was trained separately within each outer split on the corresponding matrix of inner out-of-fold predictions, while the regularization parameter was selected exclusively through internal cycle-aware validation without using the outer test data. After the ensemble prediction had been formed, the same physical projection was applied for all weighting schemes. The comparison was based on Current SoC RMSE, MAE, R^2, and the variability of RMSE across the 25 outer cycle-aware splits. For paired comparison with the primary NNLS configuration, split-wise differences in Current SoC RMSE were also analyzed.

2.10. Explainable Artificial Intelligence Analysis

A comprehensive set of explainable artificial intelligence methods was used to interpret the predictions, covering both global and local explanations of model behavior. The analysis was conducted for the predicted change in state of charge, Delta SoC, and the reconstructed final state of charge, Current SoC. SHAP and permutation importance were used to assess the consistency of the conclusions.

Global interpretation characterized the average influence of the input features on the predictions, whereas local analysis explained how a prediction was formed for an individual operational interval. The global stability of the explanations was evaluated across the outer test splits, which were not used to train the corresponding models. To generate the final local explanations, the primary Physics-Constrained Stacking configuration was retrained on the complete dataset, after which selected operational intervals were analyzed.

The primary interpretation method was SHAP, which represents a model prediction as the sum of a baseline value and the contributions of individual features [53]. For the i-th operational interval, the Delta SoC prediction was represented as:

$$\widehat{\Delta SoC_i} = \phi_0 + \sum_{j=1}^{p} \phi_{i,j}, \tag{49}$$

where $\widehat{\Delta SoC_i}$ is the predicted change in state of charge for the i-th operational interval, ϕ_0 is the baseline prediction value corresponding to the expected model output for the background dataset, $\phi_{i,j}$ is the SHAP value of the j-th feature for the i-th operational interval, $j = 1, \ldots, p$, and p is the number of input features.

A positive value of $\phi_{i,j}$ indicated that the corresponding feature increased the predicted Delta SoC relative to the baseline value. A negative value indicated a decrease in the prediction. The absolute magnitude of the SHAP value characterized the strength of the feature's influence for a specific operational interval.

The global SHAP importance of the j-th feature was defined as the mean absolute value of its local contributions:

$$I_j^{SHAP} = \frac{1}{N}\sum_{i=1}^{N}|\phi_{i,j}|, \tag{50}$$

where I_j^{SHAP} is the global SHAP importance of the j-th feature, and N is the number of operational intervals used for explanation.

A larger value of I_j^{SHAP} indicated a stronger average influence of the feature on the model prediction. Because absolute values were used, this measure characterized the magnitude of the influence but not its overall direction.

To determine the average direction of the influence, the mean SHAP value was additionally calculated as:

$$\overline{\phi}_j = \frac{1}{N}\sum_{i=1}^{N}\phi_{i,j}, \tag{51}$$

where $\overline{\phi}_j$ is the mean SHAP contribution of the j-th feature. A positive value indicated that the feature predominantly increased the predicted Delta SoC, whereas a negative value indicated that it predominantly decreased the prediction.

The distribution of SHAP values was analyzed using summary plots. These plots simultaneously showed the magnitude and direction of the feature contributions, as well as their dependence on the corresponding feature values. SHAP dependence plots were used to examine how the contribution of a particular feature changed with its observed value.

SHAP waterfall plots were used for a detailed analysis of individual predictions. A local prediction was represented as the sequential addition of feature contributions to the baseline value:

$$\widehat{\Delta SoC_i} = \phi_0 + \phi_{i,1} + \cdots + \phi_{i,p}, \tag{52}$$

The local plots displayed the observed and predicted values, absolute error, operational interval number, and its identifier. The tenth most accurate and tenth least accurate predictions, ranked by the absolute Current SoC error, were explained separately. This enabled a comparison of the structure of feature contributions in the two cases.

The stability of global SHAP importance was evaluated across the outer splits. For each feature, the mean absolute SHAP value was calculated separately on each outer test set. The mean, variability, and confidence interval of the resulting importance values were then analyzed. This made it possible to determine whether the feature ranking remained consistent across different cycle-aware data splits.

To quantitatively assess the uncertainty of global SHAP importance, 95% bootstrap confidence intervals were additionally calculated for the mean absolute SHAP values. To account for the dependence among operational intervals belonging to the same discharge cycle, bootstrap resampling was performed at the discharge-cycle level. In each bootstrap iteration, discharge cycles were sampled with replacement from the set of cycles represented in the corresponding SHAP explanation dataset, after which all operational intervals belonging to the selected cycles were included in the bootstrap sample. If the same discharge cycle was selected more than once, the corresponding operational intervals were likewise included in the bootstrap sample multiple times. For each resampled dataset, the mean absolute SHAP value was recalculated for each predictor. A total of 1000 bootstrap iterations were performed. The lower and upper bounds of the 95% confidence interval

were defined by the empirical 0.025 and 0.975 quantiles of the corresponding bootstrap distribution. This cycle-level cluster bootstrap preserved the dependence among operational intervals within the same discharge cycle and was consistent with the cycle-aware validation strategy adopted in the study.

Because Current SoC was calculated from Previous SoC and the predicted Delta SoC, the SHAP contributions of the features to the Current SoC prediction generally had the opposite direction to their contributions to the Delta SoC prediction. However, the direct contribution of Previous SoC to Current SoC was also taken into account. Therefore, the interpretation of Current SoC considered both the direct effect of Previous SoC and the feature contributions to the Delta SoC prediction, rather than simply reversing the signs of all SHAP contributions.

Permutation importance was used as an independent method for validating the global feature ranking. The values of one feature were randomly permuted across the operational intervals in the test set, while all other features remained unchanged. Feature importance was defined as the deterioration in the metric after permutation:

$$I_j^{perm} = M_j^{perm} - M^{base}, \tag{53}$$

where I_j^{perm} is the permutation importance of the j-th feature, M^{base} is the metric value for the original test data, and M_j^{perm} is the metric value after randomly permuting the j-th feature.

For RMSE and MAE, a positive value of I_j^{perm} indicated a deterioration in accuracy after disrupting the relationship between the feature and the target variable. The greater this deterioration, the more important the feature was considered.

When multiple permutation repetitions were performed, the mean permutation importance was defined as:

$$\bar{I}_j^{perm} = \frac{1}{B}\sum_{b=1}^{B} I_{j,b}^{perm}, \tag{54}$$

where $\bar{I}_j^{perm}$ is the mean permutation importance, $I_{j,b}^{perm}$ is the importance value obtained in the b-th repetition, and B is the number of permutation repetitions.

To compare permutation importance with SHAP importance, the absolute importance values were additionally normalized. Because a negative permutation importance indicated that randomly permuting the feature did not increase RMSE, such values were set to zero before normalization:

$$I_j^{perm,+} = \max\left(0, \bar{I}_j^{perm}\right), \tag{55}$$

where $I_j^{perm,+}$ is the non-negative permutation importance of the j-th feature.

The normalized permutation importance was defined as:

$$I_j^{perm,norm} = \frac{I_j^{perm,+}}{\sum_{k=1}^{p} I_k^{perm,+}}, \tag{56}$$

where $I_j^{perm,norm}$ is the normalized permutation importance of the j-th feature, and p is the number of input features.

After normalization, the importance values across all features summed to one.

The mean absolute SHAP values were normalized analogously:

$$I_j^{SHAP,norm} = \frac{I_j^{SHAP}}{\sum_{k=1}^{p} I_k^{SHAP}}, \tag{57}$$

The normalized values were used only to compare the relative shares of importance and feature rankings between SHAP and permutation importance. The original mean absolute SHAP values, expressed in percentage points, were retained to interpret the absolute magnitude of each feature's contribution.

Permutation importance assessed the extent to which model accuracy depended on individual features, whereas SHAP characterized their contributions to the formation of predictions. The combined use of global and local XAI methods made it possible to evaluate the magnitude, direction, and nature of the influence of the input features, explain individual predictions, and assess the stability of the interpretations across the outer splits. Agreement between the feature rankings obtained using SHAP and permutation importance was considered additional evidence of the reliability of the resulting explanations.

3. Results and Discussion

3.1. Predictive Performance of Baseline and Machine Learning Models

The predictive performance of the investigated models was evaluated across 25 outer cycle-aware splits generated using five-fold GroupKFold validation with five repetitions. The primary ranking criterion was the mean RMSE for Current SoC. MAE, R^2, the standard deviation of RMSE, and its 95% confidence interval based on the t-distribution were also considered. The results for the two baseline approaches, ten individual machine learning models, and Physics-Constrained Stacking are presented in Table 1.

Table 1. Predictive performance of the baseline, individual machine learning, and stacking models for Current SoC.

Model	RMSE, p.p. *	95% CI for RMSE	MAE, p.p.	R^2
Physics-Constrained Stacking	2.1183 ± 0.4622	1.9275–2.3091	1.2532	0.9804
Extra Trees	2.2815 ± 0.6785	2.0015–2.5616	1.2776	0.9772
CatBoost	2.3207 ± 0.6493	2.0527–2.5888	1.2907	0.9766
Explainable Boosting Machine	2.3545 ± 0.7361	2.0506–2.6583	1.2985	0.9757
Multilayer Perceptron	2.3618 ± 0.4452	2.1780–2.5455	1.4557	0.9758
Gaussian Process Regression	2.4451 ± 1.2827	1.9156–2.9746	1.3098	0.9701
Distance-Rate Baseline	2.4504 ± 0.4466	2.2661–2.6347	1.5009	0.9737
Random Forest	2.4876 ± 0.8367	2.1422–2.8330	1.3544	0.9727
NGBoost	2.5127 ± 0.5883	2.2699–2.7555	1.4373	0.9727
Gradient Boosting	2.6604 ± 0.9646	2.2622–3.0585	1.3337	0.9685
XGBoost	2.8974 ± 1.0304	2.4720–3.3227	1.5928	0.9614
Histogram Gradient Boosting	3.2407 ± 1.1326	2.7732–3.7082	1.7753	0.9521
Persistence Baseline	6.9606 ± 1.3470	6.4045–7.5166	4.4376	0.7917

* RMSE is reported as mean ± standard deviation across 25 outer cycle-aware evaluations. The confidence intervals are t-based 95% intervals for the mean RMSE.

Physics-Constrained Stacking achieved the lowest mean Current SoC RMSE among all evaluated approaches. Its RMSE was 2.1183 percentage points, its MAE was 1.2532 percentage points, and its mean coefficient of determination was 0.9804. Thus, the ensemble explained approximately 98% of the variability in the observed Current SoC values across the outer test cycles. The best-performing individual model was Extra Trees, with an RMSE of 2.2815 percentage points and an MAE of 1.2776 percentage points. Compared with Extra Trees, Physics-Constrained Stacking reduced the mean RMSE by 0.1632 percentage points, or approximately 7.15%. CatBoost was the second most accurate individual model, with an RMSE of 2.3207 percentage points. Relative to CatBoost, the ensemble reduced RMSE by approximately 8.72%. At the same time, the difference between Physics-Constrained Stacking and the strongest individual models was moderate. Its statistical significance and practical magnitude could not be determined solely from the

mean metric values or the overlap of the confidence intervals. The corresponding pairwise comparisons are presented later in the subsection devoted to the statistical analysis of the models.

The results of the individual algorithms showed that the highest accuracy was achieved by models capable of capturing nonlinear relationships among the operational parameters. Extra Trees, CatBoost, Explainable Boosting Machine, and Multilayer Perceptron formed a relatively compact group, with RMSE values ranging from 2.2815 to 2.3618 percentage points. Their R^2 values ranged from 0.9757 to 0.9772. This indicates that different nonlinear modeling approaches were able to reproduce the main structure of the relationship between the initial battery state, the characteristics of the operational interval, and its final SoC with sufficiently high accuracy.

Gaussian Process Regression achieved an RMSE of 2.4451 percentage points and an MAE of 1.3098 percentage points. Its mean accuracy remained competitive, although the standard deviation of RMSE was 1.2827 percentage points, the highest among the strongest-performing models. This indicates that Gaussian Process Regression was more sensitive to the specific composition of the cycles in the fit and test sets. Random Forest and NGBoost produced similar results, with RMSE values of 2.4876 and 2.5127 percentage points, respectively. Gradient Boosting performed worse in terms of RMSE, although its MAE was 1.3337 percentage points. The combination of a relatively low MAE and a higher RMSE indicates that its average absolute error remained moderate, while several larger deviations had a stronger effect on the squared-error metric. XGBoost and Histogram Gradient Boosting showed lower accuracy than the other nonlinear models, with RMSE values of 2.8974 and 3.2407 percentage points, respectively. Nevertheless, even these models substantially outperformed the Persistence Baseline, whose RMSE reached 6.9606 percentage points.

The comparison of the baseline methods confirmed the importance of accounting for distance traveled. The Persistence Baseline assumed that the state of charge remained unchanged, meaning that its predicted change in SoC was always zero. This approach yielded an MAE of 4.4376 percentage points and $R^2 = 0.7917$, representing the poorest performance in the study. The Distance-Rate Baseline, which modeled the change in SoC as proportional to the distance traveled, reduced RMSE to 2.4504 percentage points and increased R^2 to 0.9737. Thus, using Distance alone reduced RMSE by 64.80% compared with the Persistence Baseline. The Distance-Rate Baseline was also competitive with several more complex models. It outperformed Random Forest, NGBoost, Gradient Boosting, XGBoost, and Histogram Gradient Boosting in terms of mean RMSE. This result indicates a strong and largely systematic relationship between distance traveled and the change in SoC. However, Physics-Constrained Stacking further reduced RMSE by 13.55% compared with the Distance-Rate Baseline. Therefore, ambient temperature, Previous SoC, Previous driving range, and the nonlinear combination of these features provided additional predictive information that could not be captured by a simple proportional model.

The overall ranking of the models by mean Current SoC RMSE is shown in Figure 3. The horizontal error bars represent the 95% *t*-based confidence intervals for the mean RMSE across the 25 outer splits. Physics-Constrained Stacking is ranked first, although the intervals of several top-performing models partially overlap. This further justifies the need for pairwise statistical analysis based on the same outer splits.

Because Current SoC was calculated as the difference between Previous SoC and the predicted Delta SoC, the Current SoC prediction error had the same absolute magnitude as the Delta SoC prediction error but the opposite sign. Therefore, RMSE, MAE, and other direction-independent metrics were identical for Delta SoC and Current SoC.

Pairwise statistical analysis based on the same 25 outer cycle-aware splits generally confirmed the advantage of Physics-Constrained Stacking over the strongest individual

models, although the advantage remained moderate for its closest competitors. Based on the mean values reported in Table 1, the reduction in RMSE relative to Extra Trees, CatBoost, Explainable Boosting Machine, Multilayer Perceptron, Gaussian Process Regression, and Random Forest ranged from 0.1632 to 0.3693 percentage points. Physics-Constrained Stacking achieved a lower RMSE in 72–84% of the outer splits, indicating that its advantage was consistent across different compositions of the fit and test cycles. For all pairwise comparisons, the p-values from the Wilcoxon signed-rank tests remained below 0.05 after Holm correction. The corrected resampled t-test also confirmed statistically significant differences, although the corresponding p-values for Extra Trees, CatBoost, and Gaussian Process Regression were close to the 0.05 threshold. The lower bounds of the corrected 95% confidence intervals for ΔRMSE remained above zero, but were only 0.002–0.009 percentage points for the closest competitors. The d_z values ranged from 0.52 to 0.79, corresponding to moderate or moderately large paired effects. The Bayesian probability that Physics-Constrained Stacking was superior ranged from 0.824 to 0.929, whereas the probability of practical equivalence within the ROPE ranged from 0.047 to 0.138. Thus, the statistical analysis supported Physics-Constrained Stacking but did not indicate unconditional dominance over all strong individual models. The greatest uncertainty remained in the comparisons with Extra Trees, CatBoost, and Gaussian Process Regression, whereas the advantage over Random Forest was more pronounced.

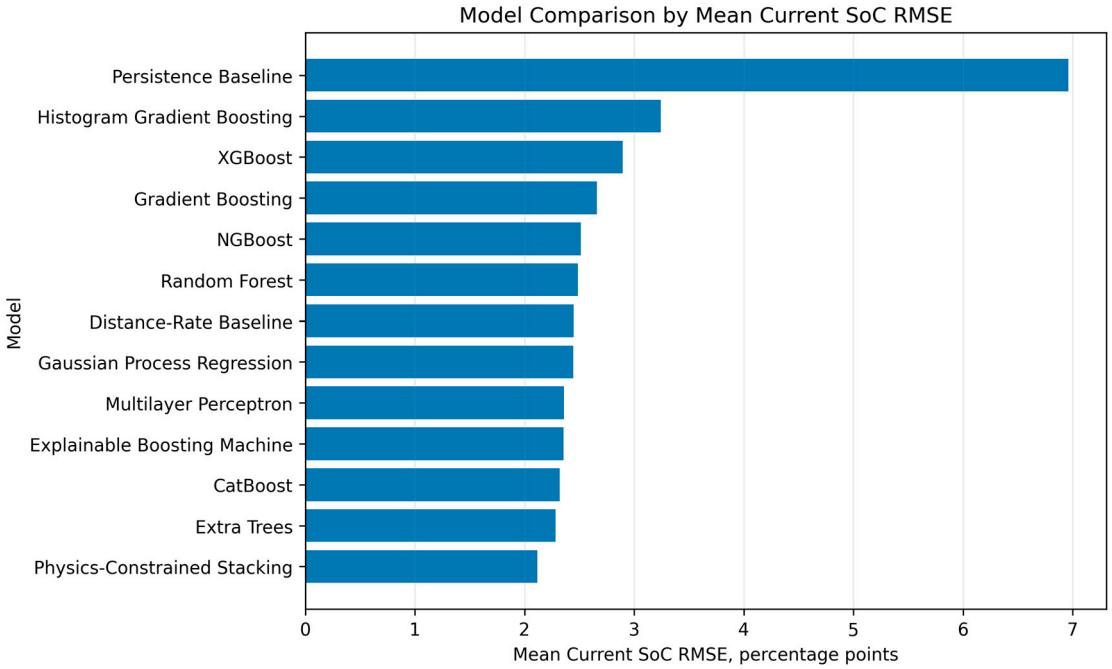


Figure 3. Comparison of the evaluated models by mean Current SoC RMSE.

Although Physics-Constrained Stacking statistically outperformed Extra Trees, the absolute reduction in Current SoC RMSE was 0.1632 percentage points, corresponding to a relative improvement of approximately 7.15%. Thus, the observed improvement was quantitatively supported, but its practical effect should be regarded as moderate. This difference alone does not demonstrate a substantial operational advantage in every use scenario, and the direct impact of such an error reduction on route-planning or charging decisions was not evaluated in this study. Therefore, the advantage of Physics-Constrained Stacking over Extra Trees is best interpreted primarily as a statistically supported but moderate improvement in predictive accuracy. At the same time, Physics-Constrained Stacking serves in this study as the central predictive component of a broader trip-end

SoC prediction framework, whose practical value is determined not only by accuracy but also by the comprehensiveness of the evaluation procedure and prediction analysis. Thus, Physics-Constrained Stacking demonstrated the best overall combination of performance indicators among the investigated approaches. It achieved the lowest RMSE and MAE, the highest mean R^2, negligible systematic bias, and relatively stable results across the outer cycle-aware splits. At the same time, the competitive performance of Extra Trees, CatBoost, Explainable Boosting Machine, and Multilayer Perceptron, together with the strong result of the Distance-Rate Baseline, indicates that the prediction task was strongly dependent on distance traveled. However, additionally accounting for nonlinear relationships among the input features further improved predictive accuracy.

3.2. Prediction Accuracy and Residual Analysis of Physics-Constrained Stacking

To examine the behavior of Physics-Constrained Stacking in greater detail, the agreement between the predicted and observed Current SoC values, the residual distribution, and the proportion of predictions within specified absolute-error thresholds were analyzed. The analysis was based on predictions from the outer test sets generated across the 25 cycle-aware splits. Because each of the 352 operational intervals appeared in the test set once during each of the five repetitions, the pooled dataset contained 1760 outer test predictions. The pooled set of 1760 predictions was not treated as 1760 independent observations because each of the 352 operational intervals was represented by five predictions obtained from different repetitions of cross-validation. In this section, the pooled predictions were used for descriptive analysis of the residual distribution, visualization of the predictions, and calculation of the proportions of predictions falling within specified absolute-error thresholds. They were not treated as independent units for statistical inference.

The main accuracy and residual characteristics of Physics-Constrained Stacking are presented in Table 2. Unless otherwise specified for metrics calculated from the pooled prediction dataset, the values in the table are reported as the mean results across the 25 outer splits.

Table 2. Prediction accuracy and residual characteristics of Physics-Constrained Stacking for Current SoC.

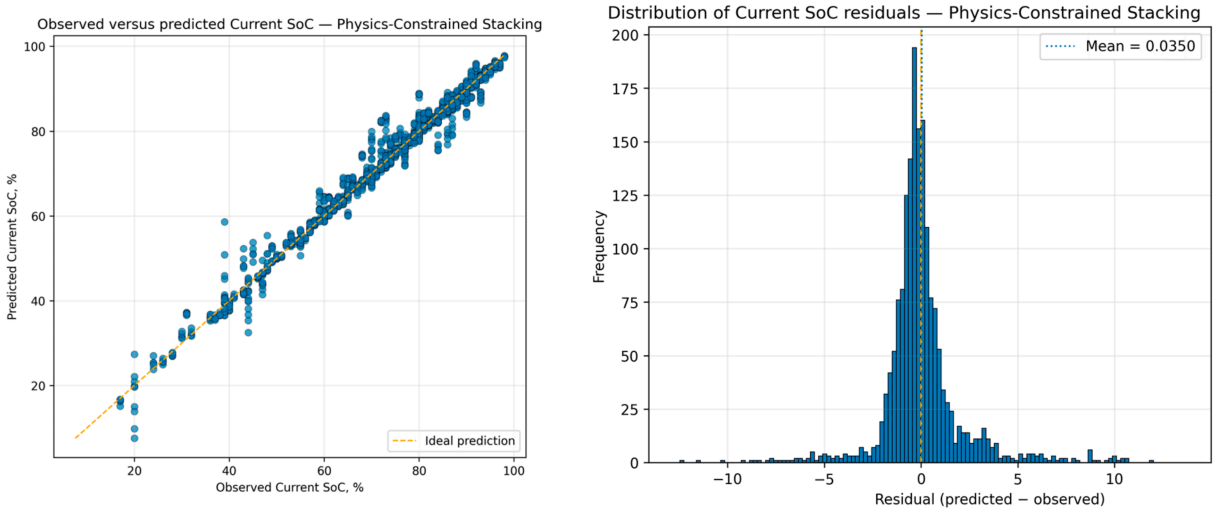
Indicator	Mean	SD	95% CI
RMSE, p.p.	2.1183	0.4622	1.9275–2.3091
MAE, p.p.	1.2532	0.2347	1.1564–1.3501
R^2	0.9804	0.0098	0.9763–0.9844
MAPE, %	1.9967	0.5182	1.7828–2.2107
sMAPE, %	2.0000	0.5177	1.7863–2.2137
ME, p.p. *	0.0350	0.3536	−0.1109–0.1810
Residual SD, p.p.	2.1074	0.4516	1.9210–2.2938
MaxError, p.p. *	9.3680	2.8110	8.2077–10.5284
Absolute error $\leq$ 1 p.p., %	64.25	6.73	61.47–67.02
Absolute error $\leq$ 2 p.p., %	84.48	4.41	82.66–86.30
Absolute error $\leq$ 3 p.p., %	89.60	3.32	88.24–90.97

* ME was calculated as predicted minus observed Current SoC. MaxError represents the mean of the maximum absolute errors obtained separately in the 25 outer test splits.

Analysis of the proportion of predictions within specified absolute-error thresholds confirmed the high accuracy of the ensemble. For Physics-Constrained Stacking, an average of 64.25% of predictions had an absolute error $\left|\widehat{SoC_i^{cur}} - SoC_i^{cur}\right|$ not exceeding 1 percentage point, 84.48% had an error not exceeding 2 percentage points, and 89.60% had an error not exceeding 3 percentage points. The mean error for Physics-Constrained Stacking was 0.0350 percentage points. Its proximity to zero indicates the absence of pronounced

systematic overestimation or underestimation of Current SoC. The low MAPE and sMAPE values, both approximately 2.00%, further confirmed the low relative prediction error.

The agreement between the predicted and observed Current SoC values is shown in Figure 4a. Most points were located close to the line of perfect agreement. This indicates that the ensemble reproduced Current SoC with sufficient accuracy across most of its observed range. More pronounced deviations were concentrated in a limited number of operational intervals and did not determine the overall prediction pattern.



(**a**) (**b**)

Figure 4. Prediction accuracy and residual analysis of Physics-Constrained Stacking for Current SoC: (**a**) observed versus predicted values, (**b**) residual distribution.

As shown in Figure 4b, the residual distribution was concentrated around zero. The mean ME across the 25 outer splits was 0.0350 percentage points. Its positive sign corresponded to a slight average overestimation of the predicted Current SoC. However, the 95% confidence interval for ME ranged from −0.1109 to 0.1810 percentage points and included zero. Therefore, the results did not indicate persistent systematic overestimation or underestimation of the predictions. The mean residual standard deviation was 2.1074 percentage points, which was close to the RMSE because the mean bias was practically zero.

At the same time, the residual distribution contained several extreme values. The largest absolute error in the pooled set of 1760 repeated outer predictions was 12.4592 percentage points. For this operational interval, the observed Current SoC was 20.0%, whereas the predicted value was 7.5408%. This corresponded to an underestimation of Current SoC by 12.4592 percentage points. This deviation exceeded the mean MaxError of 9.3680 percentage points reported in Table 2 because the table presents the average of the maximum absolute errors calculated separately for each of the 25 outer splits. The presence of such isolated large deviations supports the use of not only average performance metrics but also MaxError analysis, prediction intervals, and local XAI explanations.

The observed and predicted Current SoC values were additionally ordered by increasing observed state of charge, as shown in Figure 5. The predicted curve generally followed the shape of the observed sequence across the entire Current SoC range. The most pronounced local discrepancies corresponded to a limited number of operational intervals with large absolute errors.

Because Current SoC was defined as the difference between Previous SoC and the predicted Delta SoC, the absolute Current SoC and Delta SoC errors were identical for

the same operational interval. Their residuals had the same absolute magnitude but opposite signs.

Given the substantial variability in Distance and Temperature, the predictive performance of Physics-Constrained Stacking was additionally evaluated across different operating regimes. To avoid treating repeated outer-test predictions as independent observations, the five predictions for each of the 352 operational intervals were first averaged, so that each operational interval was represented only once in the analysis. Based on Distance, the observations were divided into short (≤9 km), medium (10–18 km), and long (>18 km) operational intervals. For Temperature, cold (<5 °C), moderate (5–<20 °C), and warm (≥20 °C) conditions were considered. The results are presented in Table 3.

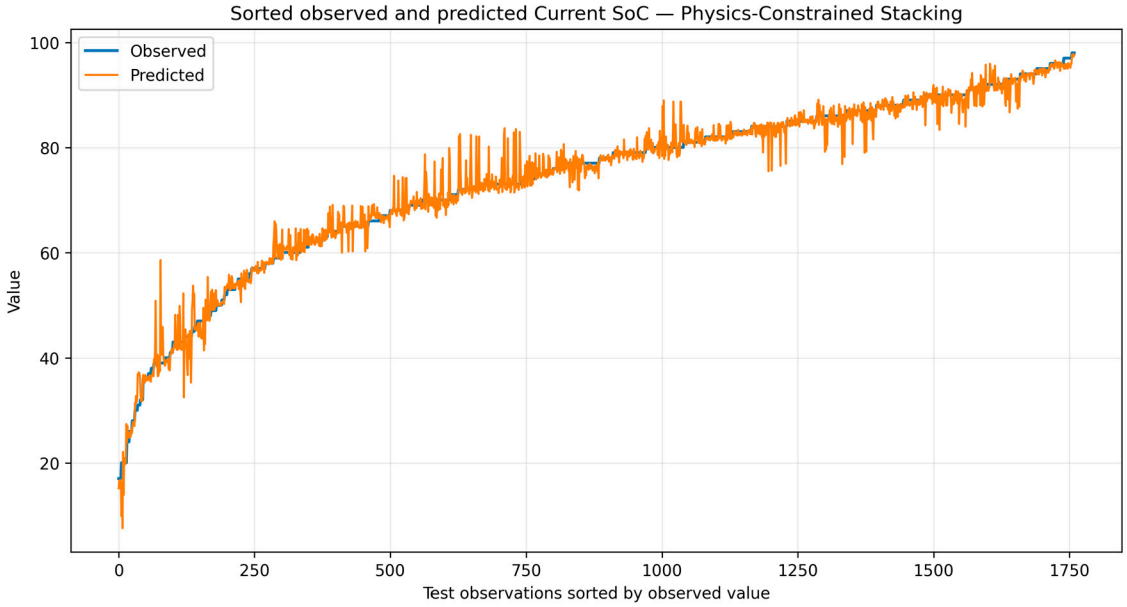


Figure 5. Sorted observed and predicted Current SoC values for Physics-Constrained Stacking.

Table 3. Current SoC prediction performance of Physics-Constrained Stacking across Distance and Temperature regimes *.

Regime Variable	Regime	Intervals n	RMSE, p.p.	MAE, p.p.	R^2	Absolute Error ≤ 2 p.p., %
Distance	Short (≤9 km)	118	1.6004	1.0231	0.9883	90.68
Distance	Medium (10–18 km)	118	1.3639	0.7489	0.9939	94.92
Distance	Long (>18 km)	116	2.8641	1.8416	0.9692	70.69
Temperature	Cold (<5 °C)	56	1.6165	1.0771	0.9686	83.93
Temperature	Moderate (5–<20 °C)	165	2.0692	1.1712	0.9809	88.48
Temperature	Warm (≥20 °C)	131	2.1764	1.2912	0.9877	82.44

* Repeated outer-test predictions were averaged for each operational interval before regime-specific metrics were calculated; therefore, each of the 352 operational intervals contributed once to this analysis.

The analysis by Distance showed that prediction accuracy was lower for long operational intervals. Current SoC RMSE was 1.6004 percentage points for short intervals and 1.3639 percentage points for medium intervals, increasing to 2.8641 percentage points for long intervals. For long intervals, MAE also increased to 1.8416 percentage points, while the proportion of predictions with an absolute error not exceeding 2 percentage points decreased to 70.69%, compared with 90.68% for short intervals and 94.92% for medium intervals. Thus, the overall performance metrics partially masked the reduction in prediction accuracy observed for longer operational intervals.

Differences among the Temperature regimes were less pronounced. Current SoC RMSE was 1.6165 percentage points under cold conditions, 2.0692 percentage points under moderate conditions, and 2.1764 percentage points under warm conditions. At the same time, R^2 remained high across all three temperature regimes, ranging from 0.9686 to 0.9877. Thus, the model maintained a strong agreement between predicted and observed Current SoC values across all investigated temperature ranges, although the absolute error was somewhat higher under moderate and warm conditions.

The higher error observed for long operational intervals indicates that Distance was a more pronounced source of variability in predictive performance than Temperature.

The results showed that Physics-Constrained Stacking combined low mean RMSE and MAE with a near-zero ME and a high proportion of predictions within 1–3 percentage points. However, the presence of a small number of substantially larger errors indicated that predictive accuracy was not uniform across all operational intervals. This supports further analysis of the stacking ensemble composition, uncertainty quantification using conformal prediction, and local explanation of the most and least accurate predictions.

3.3. Composition and Physical Consistency of the Stacking Ensemble

Physics-Constrained Stacking combined the predictions of ten base models using non-negative weights that were estimated separately for each of the 25 outer cycle-aware splits from internal out-of-fold predictions. This procedure allowed the ensemble composition to adapt to the structure of the corresponding fit set without using information from the outer test cycles.

The mean weights of the ensemble components are presented in Table 4. In addition to the mean value, the table reports the standard deviation, weight range, and the number of outer splits in which the corresponding model received a nonzero weight.

Table 4. Distribution of non-negative stacking weights across the outer cycle-aware splits.

Stacking Component	Mean Weight	SD	Median	Minimum	Maximum	Outer Splits with Non-Zero Weight
Multilayer Perceptron	0.4223	0.2193	0.4460	0.0000	0.7623	24 of 25
Gaussian Process Regression	0.2569	0.2171	0.2136	0.0000	0.6997	20 of 25
Extra Trees	0.1301	0.2012	0.0049	0.0000	0.7377	13 of 25
NGBoost	0.0796	0.1529	0.0000	0.0000	0.5006	7 of 25
CatBoost	0.0527	0.1047	0.0000	0.0000	0.3505	6 of 25
XGBoost	0.0257	0.0564	0.0000	0.0000	0.2315	6 of 25
Explainable Boosting Machine	0.0198	0.0653	0.0000	0.0000	0.2498	3 of 25
Gradient Boosting	0.0112	0.0398	0.0000	0.0000	0.1729	2 of 25
Histogram Gradient Boosting	0.0018	0.0088	0.0000	0.0000	0.0442	1 of 25
Random Forest	0.0000	0.0000	0.0000	0.0000	0.0000	0 of 25

Multilayer Perceptron was the dominant ensemble component. Its mean weight was 0.4223, and it received a nonzero weight in 24 of the 25 outer splits. The maximum weight reached 0.7623. This indicates that the model consistently provided useful information for the final prediction, although its relative contribution varied depending on the composition of the cycles in the fit set. Gaussian Process Regression provided the second-largest contribution, with a mean weight of 0.2569. It received a nonzero weight in 20 outer splits, and its maximum weight was 0.6997. The high mean weights of Multilayer Perceptron and Gaussian Process Regression indicate that most of the ensemble prediction was formed by models based on different approximation principles. Multilayer Perceptron captured complex nonlinear relationships using a multilayer neural network, whereas Gaussian Process Regression used a probabilistic representation of the functional relationship defined by a covariance kernel. Extra Trees ranked third, with a mean weight of 0.1301. However,

its standard deviation was 0.2012, its median weight was only 0.0049, and its maximum weight reached 0.7377. Thus, the contribution of Extra Trees was less stable than those of the two leading components. In some outer splits, the model had almost no influence on the ensemble, whereas in others it became the dominant component. This variability indicates that the usefulness of its tree-based structure depended on the specific composition of the training cycles. NGBoost and CatBoost had mean weights of 0.0796 and 0.0527, respectively. Their median weights were zero, and nonzero contributions were observed in only 7 and 6 outer splits. This means that NNLS did not use these models systematically but assigned them positive weights when their predictions contained information complementary to that provided by the dominant components. The mean weights of XGBoost, Explainable Boosting Machine, Gradient Boosting, and Histogram Gradient Boosting were below 0.03. Random Forest received a zero weight in all 25 outer splits. A zero weight does not imply that the model was unsuitable for prediction when used independently. Rather, it indicates that, given the predictions of the other components, Random Forest did not provide an additional unique contribution that reduced the ensemble's internal out-of-fold RMSE.

The mean out-of-fold RMSE values of the ensemble components also differed. Multilayer Perceptron achieved the lowest mean value, with an internal out-of-fold RMSE of 2.4153 percentage points. The corresponding values for CatBoost, Extra Trees, NGBoost, and Gaussian Process Regression were 2.5225, 2.5470, 2.5862, and 2.6276 percentage points, respectively. However, the magnitude of the NNLS weight was not determined solely by the individual RMSE of a model. For example, Gaussian Process Regression had a slightly higher out-of-fold RMSE than Extra Trees or CatBoost but received a substantially larger mean weight. This indicates that ensemble construction depended not only on the individual accuracy of each component but also on how its error structure differed from those of the other models. A model could receive a higher weight when its predictions improved performance specifically for operational intervals in which the other components were less accurate. At the same time, the observed split-to-split variability in the NNLS weights raises the question of whether changes in ensemble composition lead to corresponding instability in the final predictions. To examine this issue, the primary NNLS weighting procedure was compared with equal-weight averaging and Ridge-based meta-learning using the same 25 outer cycle-aware splits. The base-model predictions were kept unchanged, so the comparison isolated the effect of the weighting strategy itself. The results are presented in Table 5.

Table 5. Comparison of stacking meta-learners on the same 25 outer cycle-aware splits.

Meta-Learner	Current SoC RMSE, p.p.*	Current SoC MAE, p.p. *	Current SoC R^2	ΔRMSE vs. NNLS, p.p.
Ridge	2.0944 ± 0.4000	1.2413 ± 0.2122	0.9808	−0.0239
NNLS	2.1183 ± 0.4622	1.2532 ± 0.2347	0.9804	0.0000
Equal-weight averaging	2.3068 ± 0.6948	1.2839 ± 0.2780	0.9764	+0.1885

* RMSE and MAE are reported as mean ± standard deviation across 25 outer cycle-aware evaluations. ΔRMSE was calculated relative to the primary NNLS configuration.

Ridge-based meta-learning achieved the lowest mean Current SoC RMSE, at 2.0944 percentage points, compared with 2.1183 percentage points for NNLS. The absolute difference was only 0.0239 percentage points. Ridge yielded a lower RMSE in 16 of the 25 outer splits, whereas NNLS performed better in 9 splits. However, the 95% confidence interval for the paired Ridge − NNLS difference ranged from −0.1131 to +0.0653 percentage points and included zero, while the Wilcoxon signed-rank test did not indicate a statistically significant difference ($p = 0.396$). Therefore, the observed numerical advantage of Ridge does not provide evidence of systematic superiority over NNLS.

Equal-weight averaging resulted in a higher mean Current SoC RMSE of 2.3068 percentage points, which was 0.1885 percentage points higher than that of NNLS. It outperformed NNLS in only 8 of the 25 outer splits, whereas NNLS achieved a lower RMSE in 17 splits. In addition, RMSE variability across the outer splits was greater for equal-weight averaging: the standard deviation was 0.6948 percentage points, compared with 0.4622 for NNLS and 0.4000 for Ridge. Thus, fixed equal weighting did not provide more stable predictive performance despite the complete stability of the individual weights.

The results indicate that variability in the individual NNLS weights does not directly translate into corresponding instability in the final ensemble predictions. This may be explained by the fact that the predictions of individual base models contain partially interchangeable or correlated predictive information, allowing different weight combinations to produce similar final predictions. Ridge regularization slightly reduced the split-to-split variability in RMSE; however, its mean accuracy did not differ statistically significantly from that of NNLS. At the same time, the Ridge coefficients also varied across outer splits and could take negative values, whereas NNLS retained non-negative normalized weights and therefore provided a more straightforward interpretation of the ensemble composition. For this reason, NNLS was retained as the primary meta-learner in Physics-Constrained Stacking.

From a deployment perspective, the observed variability in the weights indicates that the relative contributions of individual base models may change after retraining on new operational data. Therefore, the individual stacking weights should not be interpreted as fixed vehicle-specific coefficients. Instead, they are data-dependent parameters that should be re-estimated whenever the ensemble is retrained. At the same time, the similar predictive performance of NNLS and Ridge, together with the lower variability of NNLS compared with equal-weight averaging, indicates that changes in the weight composition across the investigated splits did not lead to instability in the final predictive performance.

After combining the predictions of the base models, physical constraints were applied to the predicted Delta SoC to ensure that the predicted decrease in state of charge was non-negative and did not exceed Previous SoC. Because Current SoC was calculated as the difference between Previous SoC and the predicted Delta SoC, these constraints also guaranteed that the predicted Current SoC was neither below 0% nor above Previous SoC.

To assess the actual effect of the physical constraints, Raw Stacking and Physics-Constrained Stacking were compared on the same outer test splits. Before the physical constraints were applied, the stacking ensemble produced no negative Delta SoC values or values exceeding Previous SoC in any of the 25 outer splits. Accordingly, no predicted Current SoC values were below 0% or above Previous SoC. The mean proportion of Raw Stacking predictions that violated any of these physical conditions was 0.00%.

Because no physically inadmissible predictions were obtained, the projection did not alter the predicted values. Therefore, the RMSE and MAE values of Raw Stacking and Physics-Constrained Stacking were identical. This indicates that, within the available outer test data, the non-negative weighted combination of the base models already produced physically admissible predictions. However, the nonnegativity of the NNLS weights alone does not guarantee compliance with the physical bounds for new data, because individual base models may theoretically predict a negative Delta SoC or a value exceeding Previous SoC. Therefore, retaining the physical projection is appropriate as a mechanism that guarantees prediction admissibility during practical application of the ensemble.

The absence of changes in RMSE and MAE after projection also shows that the improved accuracy of Physics-Constrained Stacking relative to the individual models was not caused by clipping extreme physically inadmissible predictions. Instead, the improvement

resulted directly from the weighted combination of the base-model predictions, whose errors partially compensated for one another.

Thus, the effective composition of Physics-Constrained Stacking was considerably sparser than its formally specified set of ten models. The final prediction was driven primarily by Multilayer Perceptron, Gaussian Process Regression, and Extra Trees, whereas the other components were used selectively or received zero weights. At the same time, the physical projection did not affect predictive accuracy during outer validation because all predictions already satisfied the admissible bounds. Its primary function was not to correct the results of the current experiment but to guarantee the physical consistency of predictions during subsequent practical application of the ensemble. Thus, the fact that the physical projection was not activated for the current outer test data is a property of the Raw Stacking predictions obtained for the dataset considered and does not imply that the proposed methodology lacks a physical constraint. At the same time, the improvement in predictive performance relative to the individual models was achieved by the stacking ensemble rather than by the physical projection, because none of the predictions generated during the outer evaluations required correction. The available dataset did not contain a sufficient number of extreme or strongly out-of-distribution operating conditions to empirically assess how often the physical projection would be activated in such scenarios. Therefore, the practical benefit of the physical constraint under extreme conditions cannot be quantified using the current dataset. Future practical validation will specifically include long trips, low initial SoC values, and other challenging operating regimes in which unconstrained models may be more prone to producing physically inadmissible predictions.

3.4. Reliability and Adaptivity of Prediction Intervals

Normalized split conformal prediction achieved a mean C_{cur} of 92.85% for Physics-Constrained Stacking at a nominal coverage level of 90%. Thus, the prediction intervals achieved the target coverage on average and exhibited moderate conservatism.

The main results of normalized split conformal prediction across the 25 outer splits are presented in Table 6.

Table 6. Performance of normalized split conformal prediction intervals for Current SoC.

Indicator	Mean	SD	95% CI	Minimum–Maximum
C_{cur}, %	92.85	5.23	90.69–95.01	74.65–100.00
Mean W_{cur}, p.p.	9.62	2.66	8.53–10.72	4.66–17.16
Mean split-specific median W_{cur}, p.p.	5.41	1.59	4.75–6.06	2.12–10.43
Mean split-specific maximum W_{cur}, p.p.	71.44	16.89	64.47–78.41	26.63–95.00
Mean interval score	12.17	2.31	11.22–13.13	8.68–17.16
Conformal quantile	8.62	3.27	7.27–9.97	3.76–18.96
Calibration sample size	54.36	9.16	50.58–58.14	39–67

The Current SoC intervals were obtained by transforming the physically admissible Delta SoC bounds. Therefore, C_{Δ} and C_{cur}, as well as W_{Δ} and W_{cur}, had identical numerical values.

Across individual outer splits, C_{cur} ranged from 74.65% to 100.00%, with a standard deviation of 5.23 percentage points. Thus, achieving the nominal coverage level on average did not imply identical coverage for every specific combination of calibration and test cycles. This variability may have been associated with differences in the composition of the discharge cycles and the relatively small calibration-set sizes, which ranged from 39 to 67 observations. This variability across individual outer splits indicates that the mean empirical coverage should be interpreted as a characteristic of the overall repeated

cycle-aware evaluation procedure rather than as a guarantee of the same coverage level for every specific combination of calibration and test partitions. In practical applications, the reliability of the resulting prediction intervals may therefore depend on how representative the available calibration data are of the operating conditions for which predictions are made. The mean W_{cur} across the 25 outer splits was 9.62 percentage points. The mean of the median prediction interval widths across the outer splits was 5.41 percentage points, while the mean maximum width was 71.44 percentage points. The lower median width relative to the mean W_{cur} indicated that the prediction intervals were relatively narrow for most operational intervals, whereas individual observations received substantially wider bounds.

The mean interval score for normalized split conformal prediction was 12.17. This metric simultaneously accounted for prediction interval width and penalized cases in which the observed Current SoC fell outside the interval bounds. The resulting interval score should be interpreted together with C_{cur} and W_{cur}, because coverage above the nominal level was accompanied by substantial widening of the prediction intervals for some predictions.

The variation in prediction interval width across operational intervals is shown in Figure 6. In the figure, the observed and predicted Current SoC values are ordered by the observed state of charge, and the shaded region represents the corresponding 90% prediction intervals. Most observed values lie within the constructed bounds. At the same time, the interval widths vary substantially across individual predictions.

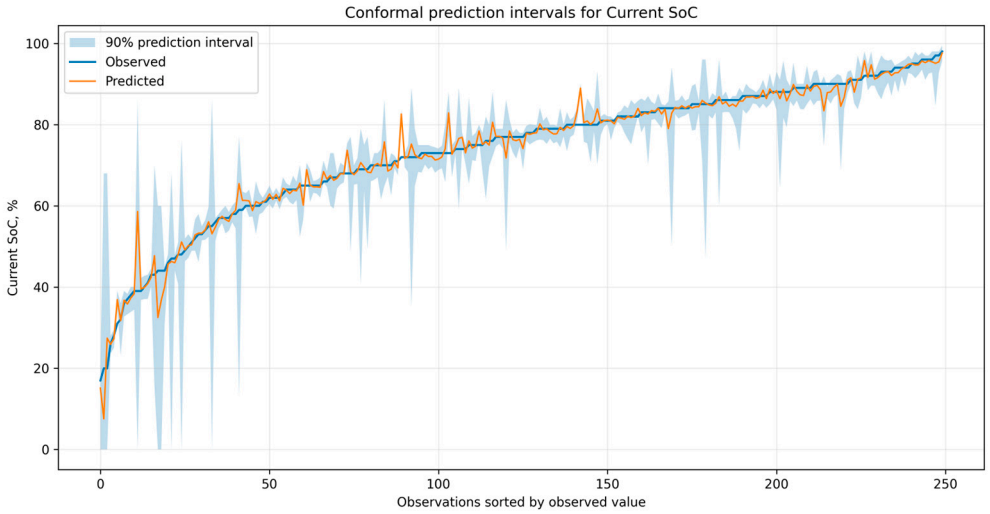


Figure 6. Normalized conformal prediction intervals for Current SoC.

As shown in Figure 6, normalized split conformal prediction did not use the same prediction interval width for all predictions. For observations with a lower local uncertainty scale, the interval bounds were narrower, whereas the prediction intervals widened as disagreement among the ensemble component predictions or Distance increased. This behavior is consistent with the local scaling procedure described in Section 2.7, which accounted for both disagreement among the stacking ensemble components and Distance. At the same time, this trend is consistent with the regime-specific performance analysis in Section 3.2, where higher Current SoC RMSE and MAE values were obtained for long operational intervals than for short and medium intervals. Thus, longer operational intervals were associated with both larger point-prediction errors and wider prediction intervals.

From a practical perspective, this adaptive behavior is useful because a wider prediction interval directly indicates greater predictive uncertainty. In such cases, assessment of the possible final Current SoC should take into account not only the point prediction but

also the full range of the prediction interval. The same applies to observations with greater disagreement among the stacking ensemble components, for which the local uncertainty estimate is higher. At the same time, very wide intervals have limited practical value for operational decision-making because they define a broad range of possible final Current SoC values. Therefore, such intervals should primarily be interpreted as an indication of increased predictive uncertainty rather than as equally informative estimates of the final state of charge.

The comparison of the five conformal methods showed that all of them achieved a mean C_{cur} above the nominal level of 90%. However, the methods differed substantially in terms of W_{cur} and interval score. The results are presented in Table 7.

Table 7. Comparison of conformal prediction methods for Current SoC.

Method	Mean C_{cur}, %	SD, p.p.	Mean W_{cur}, p.p.	Median Split-Specific Median W_{cur}, p.p.	Overall Maximum W_{cur}, p.p.	Mean Interval Score
Capped normalized conformal	92.29	6.12	7.72	5.46	57.97	10.59
Unnormalized split conformal	93.37	5.62	7.97	7.36	19.56	11.42
Mondrian conformal by Distance	96.03	4.18	9.73	8.91	48.55	11.45
Conformalized quantile regression	94.83	3.80	8.90	7.00	49.99	11.92
Normalized split conformal	92.85	5.23	9.62	5.46	95.00	12.17

The Maximum W_{cur} values in Tables 6 and 7 were calculated at different aggregation levels. In Table 6, the maximum prediction interval width was first determined separately for each of the 25 outer splits. The resulting maximum values were then summarized using the mean, standard deviation, 95% confidence interval, and minimum–maximum range. Therefore, 71.44 percentage points represents the mean of the 25 split-specific maximum widths, whereas the range of 26.63–95.00 percentage points represents the minimum and maximum of these split-specific maxima. Table 7 reports the largest prediction interval width obtained for the corresponding conformal method across all test predictions from all 25 outer splits. Thus, the value of 95.00 percentage points for normalized split conformal prediction in Table 7 corresponds to the upper bound of the Maximum W_{cur} range reported in Table 6.

As shown in Table 7, none of the conformal methods performed best across all evaluation criteria. Mondrian conformal by Distance achieved the highest mean C_{cur} but produced wider prediction intervals. Unnormalized split conformal had the smallest maximum width, although its median interval width was greater than that of the normalized approaches. Capped normalized conformal achieved the lowest W_{cur} and the lowest mean interval score. Normalized split conformal had the same median prediction interval width as capped normalized conformal but was more sensitive to individual large values of the local uncertainty scale, which increased its mean and maximum interval widths.

Consistent with the study design, normalized split conformal prediction was used as the primary method because it provided individualized scaling of prediction intervals without imposing an upper bound on the local uncertainty estimate. The mean empirical coverage achieved by this approach exceeded the nominal level of 90%. Prediction intervals became narrower for predictions with a lower local uncertainty scale and widened as disagreement among the ensemble components or Distance increased. From a practical perspective, this adaptivity is useful because a wider interval indicates greater predictive uncertainty. In such cases, assessment of the expected final Current SoC should consider not only the point prediction but also the full range of the prediction interval. This is particularly important for longer operational intervals and for observations with greater disagreement among the stacking ensemble components. At the same time, very wide

intervals have limited practical value for operational decision-making because they provide only a coarse estimate of the possible final Current SoC.

The comparison of the conformal methods also showed that unrestricted local scaling may lead to excessively wide intervals for individual observations. Capped normalized conformal reduced both the mean and maximum prediction interval widths and achieved the lowest mean interval score among the methods considered, indicating a more favorable trade-off between coverage and sharpness for the dataset used in this study. Nevertheless, uncapped normalized split conformal prediction was retained as the primary method because it does not constrain the local uncertainty estimate with an artificial upper bound and therefore more fully reflects cases of elevated predictive uncertainty. Capped normalized conformal can, however, be considered a practical modification for applications in which excessive widening of individual prediction intervals is undesirable.

3.5. Ablation Results and Interpretation

The contribution of individual Physics-Constrained Stacking components was evaluated by sequentially removing each base model from the full ensemble. For each leave-one-component-out configuration, the stacking ensemble was reconstructed using the same procedure as the full ensemble and then evaluated on the same 25 outer cycle-aware splits. The ablation analysis results are presented in Table 8.

Table 8. Leave-one-component-out ablation of Physics-Constrained Stacking.

Removed Component	Number of Remaining Components	RMSE, p.p.	ΔRMSE, p.p. *	Relative ΔRMSE, %	MAE, p.p.	R^2
None, full stacking	10	2.1183	0.0000	0.00	1.2532	0.9804
Multilayer Perceptron	9	2.2958	+0.1775	+8.38	1.2864	0.9766
Extra Trees	9	2.1429	+0.0246	+1.16	1.2662	0.9799
Gaussian Process Regression	9	2.1356	+0.0173	+0.82	1.2639	0.9802
Random Forest	9	2.1183	0.0000	0.00	1.2532	0.9804
Gradient Boosting	9	2.1183	0.0000	0.00	1.2543	0.9804
Histogram Gradient Boosting	9	2.1170	−0.0013	−0.06	1.2531	0.9804
CatBoost	9	2.1165	−0.0018	−0.08	1.2538	0.9804
Explainable Boosting Machine	9	2.1159	−0.0024	−0.11	1.2520	0.9804
XGBoost	9	2.1121	−0.0062	−0.29	1.2491	0.9806
NGBoost	9	2.1062	−0.0121	−0.57	1.2529	0.9806

* ΔRMSE and relative ΔRMSE were calculated with respect to the full Physics-Constrained Stacking configuration.

The ΔRMSE values were calculated relative to the full Physics-Constrained Stacking configuration. A positive ΔRMSE indicated a deterioration in performance after removing a component, whereas a negative value indicated an improvement in the mean result.

The ablation analysis showed that Multilayer Perceptron was the most critical component of Physics-Constrained Stacking. Its removal caused the largest deterioration across all major metrics, indicating that this model made a decisive contribution to the final prediction. Extra Trees and Gaussian Process Regression also provided additional predictive value, although their contributions were substantially smaller. The deterioration observed after removing these models remained moderate, indicating that they played supporting roles. Removing Random Forest had no effect on the result, which was consistent with its zero mean NNLS weight. Similarly, Gradient Boosting made only a minimal contribution because ensemble accuracy remained unchanged after its removal. This indicated that some components were functionally redundant in the full configuration.

For Histogram Gradient Boosting, CatBoost, Explainable Boosting Machine, XGBoost, and NGBoost, a slight decrease in RMSE was observed after removal. The most noticeable,

although still very small, effect was obtained for NGBoost. However, all of these differences were negligible. Therefore, they should not be interpreted as convincing evidence that the corresponding models systematically degraded predictive performance. Rather, they indicated functional overlap among the ensemble components and suggested that re-optimization of the NNLS weights could compensate for the removal of individual models.

Overall, the results confirmed that the accuracy of Physics-Constrained Stacking was driven primarily by Multilayer Perceptron, while Extra Trees and Gaussian Process Regression provided additional useful contributions. The remaining models either had minimal influence or were partially redundant within the full ensemble configuration.

A targeted feature-ablation analysis was additionally conducted to assess the contribution of Previous driving range to the predictive performance of Physics-Constrained Stacking. In the reduced configuration, Previous driving range was excluded from the predictor set, after which the model was retrained using Previous SoC, Distance, and Ambient Temperature under the same cycle-aware evaluation procedure. The results are presented in Table 9.

Table 9. Effect of excluding Previous driving range from Physics-Constrained Stacking.

Feature Configuration	Current SoC RMSE, p.p.	ΔRMSE, p.p.	Relative ΔRMSE, %
Main Physics-Constrained Stacking	2.1183	0.0000	0.00
Without Previous driving range	2.1667	+0.0484	+2.28

Excluding previous driving range increased the Current SoC RMSE from 2.1183 to 2.1667 percentage points, corresponding to an increase of 0.0484 percentage points, or approximately 2.28%. Thus, removing this feature resulted in only a slight deterioration in prediction accuracy. This indicates that, for the dataset used in this study, the predictive performance of Physics-Constrained Stacking depended only marginally on Previous driving range when Previous SoC, Distance, and Ambient Temperature were already included. Therefore, Previous driving range, which is an estimate generated by the vehicle's onboard system, provided only a limited additional predictive contribution in this configuration.

3.6. Sensitivity to Uncertainty in the Specified Distance

Because the planned route length is used as an input in practical applications and may differ from the actual distance traveled, the sensitivity of Physics-Constrained Stacking to uncertainty in Distance was additionally evaluated. For this purpose, the Distance values in the outer test data were perturbed by −10%, −5%, +5%, and +10% without retraining the base models or re-estimating the NNLS stacking weights. The results are presented in Table 10.

Table 10. Sensitivity of Physics-Constrained Stacking to uncertainty in the specified Distance.

Distance Perturbation, %	Current SoC RMSE, p.p.	ΔRMSE vs. Baseline, p.p.	Current SoC MAE, p.p.	ΔMAE vs. Baseline, p.p.	R^2	Mean Absolute Change in Predicted Current SoC, p.p.
−10	2.1757	+0.0574	1.2911	+0.0378	0.9796	0.3401
−5	2.1335	+0.0152	1.2634	+0.0101	0.9804	0.1727
0	2.1183	0.0000	1.2532	0.0000	0.9804	0.0000
+5	2.1183	+0.0000	1.2563	+0.0031	0.9801	0.1664
+10	2.1484	+0.0301	1.2871	+0.0338	0.9793	0.3236

As shown in Table 10, Physics-Constrained Stacking demonstrated low sensitivity to moderate uncertainty in Distance. With the original Distance values, the mean Current

SoC RMSE was 2.1183 percentage points. Reducing Distance by 5% increased the RMSE to 2.1335 percentage points, an increase of only 0.0152 percentage points, whereas increasing Distance by 5% produced virtually no change in the mean RMSE. Under −10% and +10% perturbations, the RMSE values were 2.1757 and 2.1484 percentage points, respectively, corresponding to increases of 0.0574 and 0.0301 percentage points relative to the baseline.

The effect of uncertainty in Distance was also evaluated in terms of changes in the predicted Current SoC itself. The mean absolute change in prediction was 0.1727 and 0.1664 percentage points for the −5% and +5% perturbations, respectively, and 0.3401 and 0.3236 percentage points for the −10% and +10% perturbations. Thus, even when the input Distance value was perturbed by ±10%, the mean absolute change in predicted Current SoC was only approximately 0.32–0.34 percentage points.

The relationship between Current SoC RMSE and Distance perturbation is shown in Figure 7.

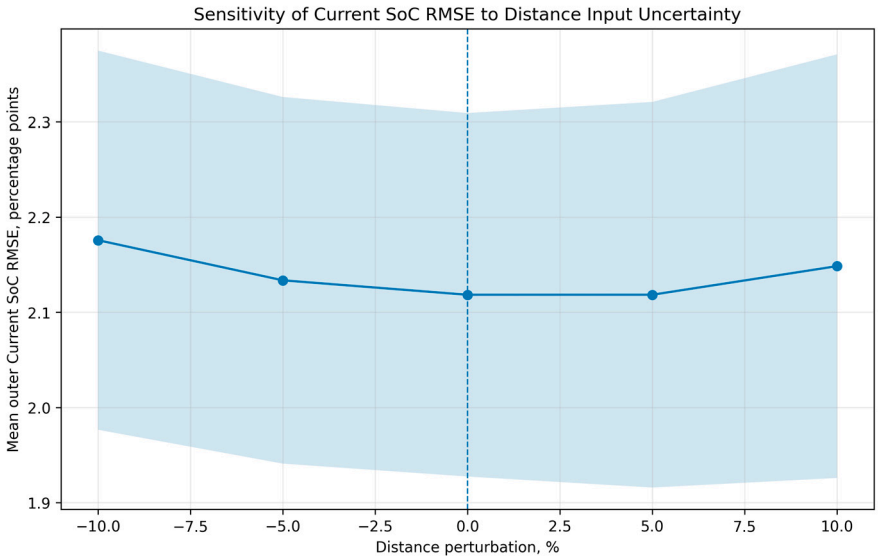


Figure 7. Sensitivity of Current SoC RMSE to uncertainty in the specified Distance.

Figure 7 shows that the mean Current SoC RMSE remained close to the baseline level across the entire range of perturbations considered, and deviations in the input Distance of ±5–10% did not cause a substantial deterioration in aggregate predictive performance. For all perturbation scenarios, the 95% confidence intervals for the mean change in RMSE included zero, indicating that the observed differences did not provide evidence of a pronounced systematic degradation in predictive performance under moderate uncertainty in this input.

At the same time, prediction changes could be larger for individual operational intervals; therefore, these results should be interpreted as an assessment of aggregate robustness rather than as a guarantee of uniform sensitivity for every individual trip. Overall, the findings support the robustness of the proposed trip-end SoC prediction framework to moderate uncertainty in the specified route length.

3.7. Global and Local Explainability of Model Predictions

Explainable Artificial Intelligence analysis for Physics-Constrained Stacking was performed separately for the directly predicted Delta SoC and the reconstructed Current SoC. Global feature importance was assessed using mean absolute SHAP values with 95% confidence intervals obtained through cycle-level cluster bootstrap, as well as permutation importance. SHAP summary plots and SHAP dependence plots were used to determine

the direction and distribution of feature effects, whereas waterfall plots provided local explanations for individual predictions.

The aggregated feature-importance estimates are presented in Table 11. The mean absolute SHAP values are expressed in percentage points of the corresponding predicted target variable. The normalized SHAP importance and permutation importance values characterized the relative contribution of each predictor and summed to one separately for each explanation method and target variable.

Table 11. Global SHAP and permutation importance for Physics-Constrained Stacking.

Target	Predictor	Mean Absolute SHAP, p.p. *	95% Bootstrap CI, p.p.	Normalized SHAP Importance	Normalized Permutation Importance
Delta SoC	Distance	2.6918	1.9259–3.7780	0.7791	0.8904
Delta SoC	Temperature	0.4039	0.3286–0.5014	0.1169	0.0683
Delta SoC	Previous driving range	0.2701	0.2295–0.3142	0.0782	0.0270
Delta SoC	Previous SoC	0.0892	0.0740–0.1062	0.0258	0.0143
Current SoC	Previous SoC	11.4766	9.9126–13.3790	0.7786	0.7570
Current SoC	Distance	2.5018	1.9171–3.3553	0.1697	0.2195
Current SoC	Temperature	0.4834	0.4010–0.5836	0.0328	0.0169
Current SoC	Previous driving range	0.2774	0.2397–0.3143	0.0188	0.0067

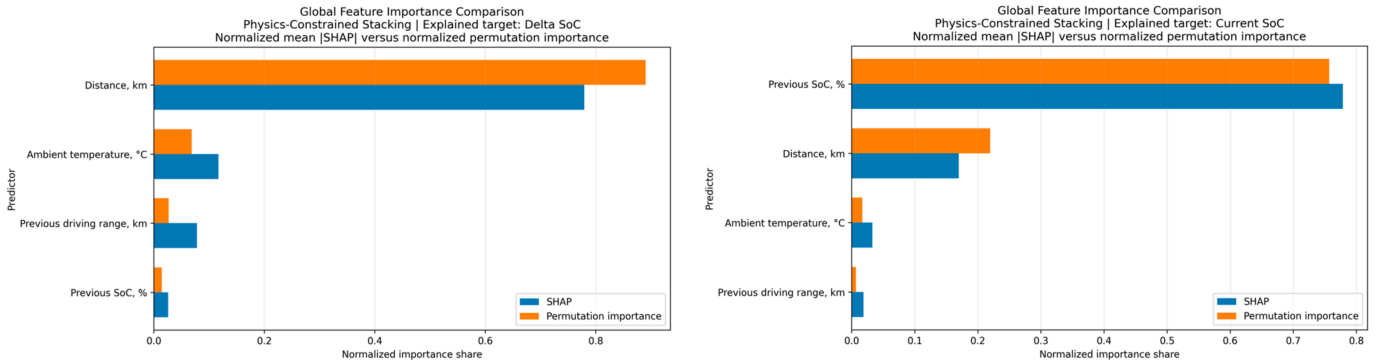
* Mean absolute SHAP values characterize the average absolute contribution of each predictor to the corresponding model output. The 95% confidence intervals were estimated using cycle-level cluster bootstrap.

According to Table 11, Distance was the dominant predictor of Delta SoC. Its mean absolute SHAP value was 2.6918 percentage points, which was more than six times greater than that of Temperature, the second-ranked feature. Previous driving range provided a smaller additional contribution, while Previous SoC had the weakest influence on the directly predicted Delta SoC. The dominance of Distance was stable under bootstrap evaluation. The lower bound of its 95% bootstrap confidence interval was 1.9259 percentage points and remained substantially higher than the upper bounds of the intervals for the other features. Thus, the leading role of distance traveled was not driven by individual observations or by random variation in the composition of the bootstrap samples. For Current SoC, the feature ranking was different. Previous SoC had the largest mean absolute SHAP value at 11.4766 percentage points. Distance ranked second at 2.5018 percentage points, whereas the contributions of Temperature and Previous driving range were substantially smaller. The dominance of Previous SoC was expected because Current SoC was reconstructed as the difference between Previous SoC and the predicted Delta SoC. Therefore, the initial state of charge directly determined the baseline level of the final SoC, while the other features primarily adjusted the predicted decrease in charge. A similar approach to interpreting model explanations has been applied in recent explainable deep learning research for power-system applications, where SHAP was used to identify global and local feature contributions and to assess whether model behavior was consistent with physically meaningful characteristics of the system under study [54]. In that work, the most influential SHAP features corresponded to key electrical descriptors of power-quality disturbances. Although the classification of power-quality disturbances differs substantially from EV battery SoC prediction, both studies demonstrate the value of SHAP not only for ranking predictor importance but also for assessing whether the model relies on physically interpretable relationships. In the present study, this consistency is reflected, in particular, by the dominance of Distance for Delta SoC and Previous SoC for Current SoC.

The agreement between SHAP and permutation importance is shown in Figure 8.

For Delta SoC, both methods clearly identified Distance as the most important predictor. Its normalized SHAP importance was 0.7791, while its normalized permutation importance reached 0.8904. Temperature ranked second, whereas Previous driving range and Previous SoC made substantially smaller contributions. Although the absolute importance shares produced by the two methods differed somewhat, the resulting feature rankings were identical.

For Current SoC, both approaches identified Previous SoC as the dominant predictor. Its normalized SHAP importance was 0.7786, and its normalized permutation importance was 0.7570. Distance ranked second, with corresponding values of 0.1697 and 0.2195. Temperature and Previous driving range made only minor contributions to the Current SoC prediction.

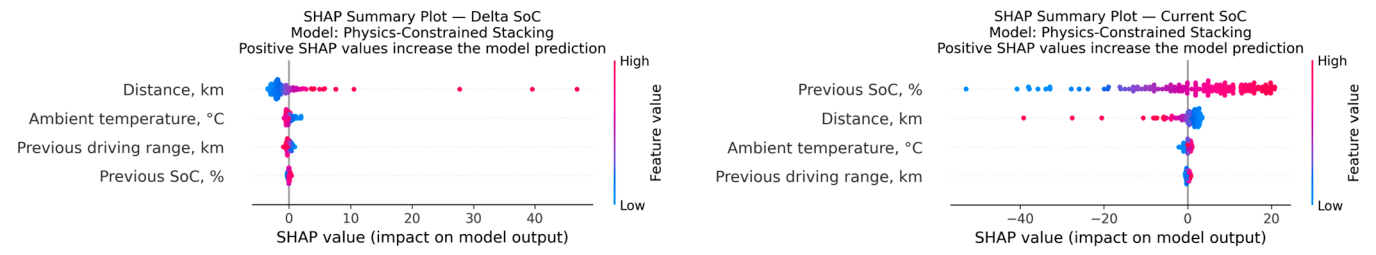


(**a**) (**b**)

Figure 8. Comparison of normalized SHAP and permutation importance for Physics-Constrained Stacking: (**a**) Delta SoC and (**b**) Current SoC.

Thus, SHAP and permutation importance produced consistent rankings of the main predictors for both target variables. Distance played the dominant role in predicting Delta SoC, whereas Previous SoC dominated the prediction of Current SoC. The differences between the normalized importance shares reflected the distinct evaluation principles of the two methods but did not alter the overall conclusion regarding the prediction structure.

Figure 9 shows the distributions of SHAP values for all input features of Physics-Constrained Stacking separately for Delta SoC and Current SoC.



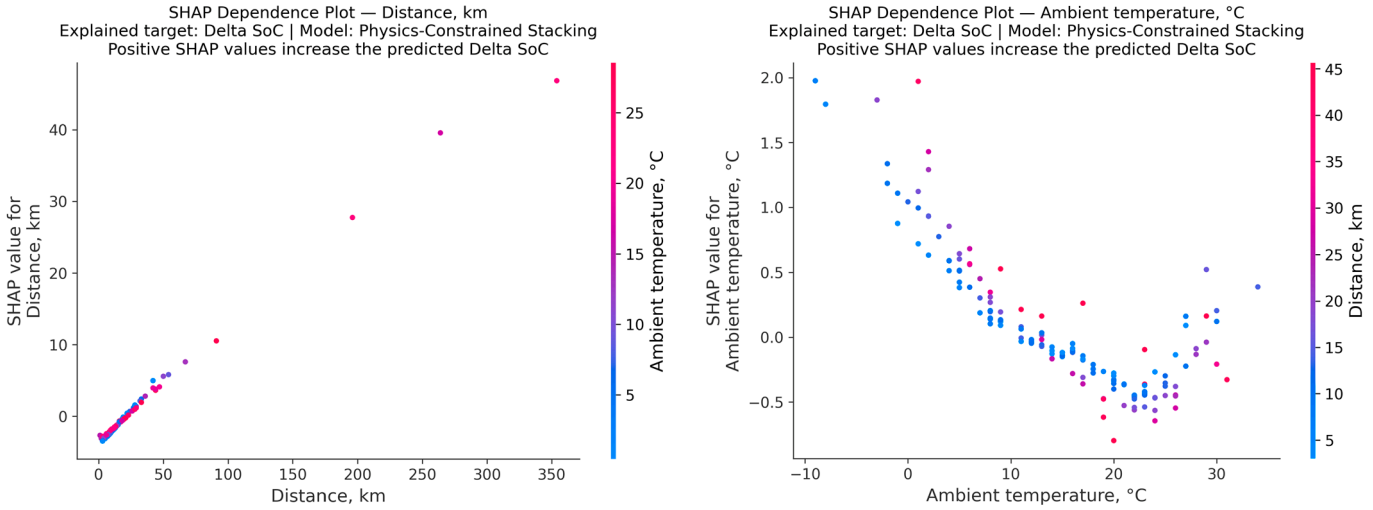
(**a**) (**b**)

Figure 9. SHAP summary plots for Physics-Constrained Stacking: (**a**) Delta SoC and (**b**) Current SoC.

In Figure 9a, Distance had the widest range of SHAP values and was the primary predictor of Delta SoC. Low Distance values were predominantly associated with negative SHAP values, whereas high values of this feature were associated with positive contributions to the prediction. Thus, the model associated longer operational intervals with a larger predicted Delta SoC. The effects of Temperature, Previous driving range, and Previous SoC were substantially smaller because their SHAP values were concentrated near zero.

In Figure 9b, Previous SoC had the greatest influence on Current SoC. High Previous SoC values predominantly produced positive SHAP values, whereas low values resulted in substantial negative contributions. This was consistent with the reconstruction of Current SoC from Previous SoC and the predicted Delta SoC. Distance ranked second in importance. High Distance values generally reduced the predicted Current SoC, whereas shorter operational intervals produced contributions close to zero or slightly positive. Temperature and Previous driving range had substantially smaller effects on the prediction.

Figure 10 shows the SHAP dependence plots for the two most important predictors of Delta SoC, Distance and Temperature.



(**a**) (**b**)

Figure 10. SHAP dependence plots for Physics-Constrained Stacking: (**a**) Distance and (**b**) Ambient temperature for Delta SoC.

As shown in Figure 10a, the contribution of Distance to the predicted Delta SoC increased almost monotonically with distance traveled. For short operational intervals, SHAP values were predominantly negative or close to zero, whereas for longer intervals they became positive and increased substantially. Thus, Physics-Constrained Stacking associated greater Distance with a larger predicted Delta SoC.

In Figure 10b, the effect of Temperature was distinctly nonlinear. At low temperatures, particularly below 0 °C, SHAP values were predominantly positive, corresponding to an increase in the predicted Delta SoC. Within the approximate range of 10–15 °C, the contribution of Temperature approached zero, whereas at temperatures of approximately 18–25 °C, it became predominantly negative. At still higher temperatures, SHAP values partially returned to positive levels, although the dispersion among individual observations increased. This indicates that the model captured a nonlinear relationship between ambient temperature and the predicted change in SoC.

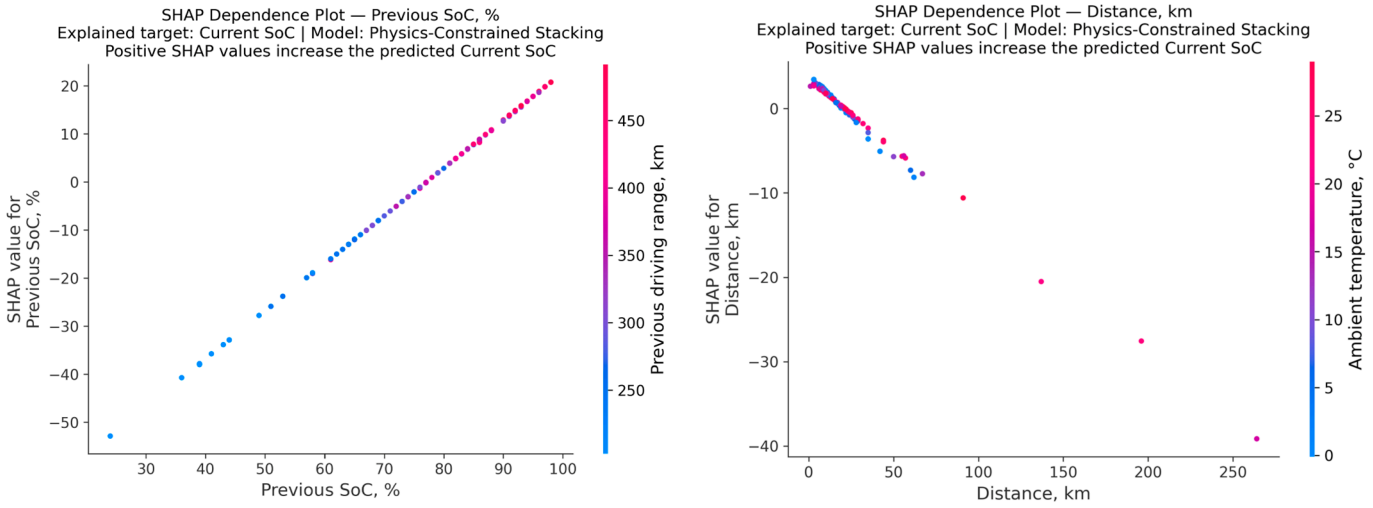
The point color in Figure 10a represents Temperature, whereas in Figure 10b it represents Distance. However, these color gradients should be interpreted only as a possible indication of interaction between the predictors, rather than as independent evidence of a SHAP interaction effect.

Figure 11 shows the SHAP dependence plots for the two most important predictors of Current SoC, Previous SoC and Distance.

The contribution of Previous SoC to the predicted Current SoC increased almost linearly with the initial state of charge. Low Previous SoC values produced substantial negative SHAP values, whereas high values of this feature corresponded to positive contributions. In Figure 11b, the SHAP values for Current SoC decreased consistently as

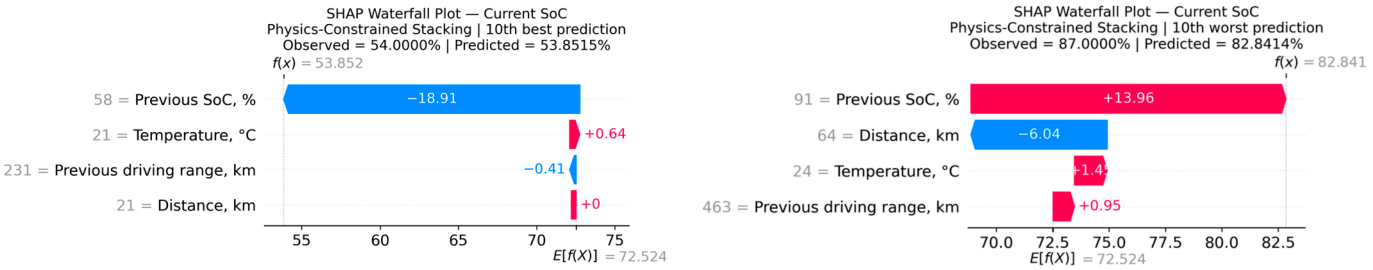
Distance increased. Short operational intervals had contributions close to zero or slightly positive, whereas longer intervals produced increasingly negative contributions. Thus, Physics-Constrained Stacking associated greater distance traveled with a lower predicted Current SoC. The point color in Figure 11a represents Previous driving range, whereas in Figure 11b it represents Temperature.

Figure 12 presents local SHAP for two Current SoC predictions generated by Physics-Constrained Stacking.



(**a**) (**b**)

Figure 11. SHAP dependence plots for Physics-Constrained Stacking: (**a**) Previous SoC and (**b**) Distance for Current SoC.



(**a**) (**b**)

Figure 12. Local SHAP of Physics-Constrained Stacking predictions for Current SoC: (**a**) the 10th most accurate prediction and (**b**) the 10th prediction with the largest absolute error.

Figure 12a illustrates a prediction with a small absolute error. The final value was formed relative to the model baseline $E[f(X)]$, which corresponded to the mean expected prediction of Physics-Constrained Stacking for the SHAP background dataset. For this operational interval, Previous SoC produced the main negative shift from the baseline. The contributions of Temperature, Previous driving range, and Distance were substantially smaller and partially offset one another. As a result of their combined effects, the predicted value remained close to the observed Current SoC. Figure 12b shows a prediction with a relatively large absolute error. Relative to the same baseline value, a high Previous SoC substantially shifted the prediction toward a higher Current SoC. At the same time, Distance produced a considerable negative contribution that partially offset the effect of the initial state of charge. Temperature and Previous driving range further increased the prediction, although their contributions were insufficient to compensate for the effect of Distance. Consequently, the model produced a Current SoC value below the observed value. This

indicated that, for this operational interval, the negative contribution of Distance was too large relative to the actual decrease in SoC.

The comparison of the local explanations shows that the waterfall plots represent the formation of an individual prediction by adding the contributions of the individual features to the model baseline. Previous SoC determined the primary direction of the prediction shift, whereas Distance was the main adjustment factor. Temperature and Previous driving range had secondary effects that, depending on the specific operational interval, could partially reinforce or offset the contributions of the dominant features. These local explanations were consistent with the results of the global XAI analysis.

At the same time, the results should be interpreted in light of the limitations of the available dataset. The study is based on operational data from a single 2020 Kia Niro EV and includes 352 operational intervals from 81 discharge cycles. The applied cycle-aware GroupKFold validation provides an assessment of the models' ability to generalize to discharge cycles that were not used during training; however, it does not evaluate their generalizability to other vehicles. In particular, this study did not assess model transfer to other electric vehicle models, different traction battery types and capacities, different levels of battery degradation, other drivers, driving styles, road profiles, or a broader range of climatic and seasonal conditions. Therefore, the reported accuracy metrics characterize prediction performance for the vehicle investigated and the range of operating conditions represented in the dataset and should not be directly generalized to the broader electric vehicle fleet. At the same time, the proposed framework for input-data formation, cycle-aware validation, ensembling, physical constraints on predictions, conformal uncertainty quantification, and XAI-based analysis is general in structure and could potentially be applied to other electric vehicles. Confirming such applicability would require separate model training or adaptation and independent validation using data from multiple vehicles, drivers, and operating conditions.

4. Conclusions

This study developed and comprehensively evaluated an approach for predicting the final state of charge of an electric vehicle battery at the level of individual operational intervals. The models used Previous SoC, Distance, Temperature, and Previous driving range, with the actual distance traveled serving as a proxy for the specified route length in the experimental evaluation. Delta SoC was predicted directly and subsequently used to calculate Current SoC. Evaluation was performed using repeated cycle-aware GroupKFold validation, which reduced the risk of information leakage between related observations.

The primary NNLS-based Physics-Constrained Stacking configuration achieved the best overall combination of performance indicators in the primary comparison with the baseline and individual machine-learning models. For Current SoC, it achieved an RMSE of 2.1183 percentage points, an MAE of 1.2532 percentage points, and an R^2 of 0.9804. The absolute error did not exceed 2 percentage points for 84.48% of the predictions, while the mean error was 0.0350 percentage points, indicating the absence of pronounced systematic bias. The ensemble reduced RMSE by approximately 7.15% compared with Extra Trees, the best-performing individual model.

The weight analysis and ablation study showed that Multilayer Perceptron made the largest contribution to the ensemble performance. Gaussian Process Regression and Extra Trees provided additional useful contributions, whereas several other models were functionally redundant. The physical projection did not alter predictive accuracy on the test data but guaranteed the admissibility of predictions during practical application.

Normalized split conformal prediction achieved a mean empirical coverage of 92.85% for prediction intervals with a nominal coverage level of 90%. The mean prediction interval

width for Current SoC was 9.62 percentage points, while the median width was 5.41 percentage points. These results confirmed the value of supplementing point predictions with individualized uncertainty estimates.

The XAI analysis showed that Distance was the dominant predictor of Delta SoC, whereas Previous SoC dominated the prediction of Current SoC. Temperature had a smaller but distinctly nonlinear effect. The consistent results obtained using SHAP and permutation importance confirmed the stability of the resulting interpretations.

Thus, the proposed approach provides accurate, physically consistent, and interpretable prediction of final SoC using a limited set of parameters known before the beginning of an operational interval. The main limitations of this study are the use of data from a single electric vehicle and the relatively limited number of discharge cycles. Therefore, the reported results characterize the model's ability to generalize to new discharge cycles within the investigated vehicle and the represented operating conditions, but they do not confirm its direct transferability to other electric vehicle models, traction battery types and health states, drivers, or a broader range of climatic and road conditions. Future research will focus on validating the proposed approach using data from multiple electric vehicles and drivers and on evaluating its generalizability across different operating conditions.

Author Contributions: Conceptualization, D.T. and O.L.; methodology, D.T.; software, D.T. and V.T.; validation, D.T., P.M. and D.M.; formal analysis, D.M. and O.L.; investigation, D.T.; resources, P.M.; data curation, D.T. and O.L.; writing—original draft preparation, D.T., P.M., O.L., V.T. and D.M.; writing—review and editing, D.T., P.M., O.L., D.M. and V.T.; visualization, D.T.; supervision, P.M.; project administration, O.L.; funding acquisition, P.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The raw data supporting the conclusions of this article will be made available by the authors on request. The data that support the findings of this study are available in the Mendeley Data repository at https://data.mendeley.com/datasets/kj7z4fb7fr/1 (accessed on 5 August 2026). The dataset is subject to an embargo, and the files will become publicly available after 1 July 2028. Until the end of the embargo period, this dataset is available from the corresponding author upon reasonable request.

Conflicts of Interest: The authors declare no conflicts of interest.

Abbreviations

The following abbreviations are used in this manuscript:

BMS	Battery management system
EV	Electric vehicle
MAE	Mean absolute error
MAPE	Mean absolute percentage error
ME	Mean error
NNLS	Non-negative least squares
OBC	On-board charger
RMSE	Root mean square error
SHAP	Shapley additive explanations
sMAPE	Symmetric mean absolute percentage error
SoC	State of charge
XAI	Explainable artificial intelligence

References

1. De Cauwer, C.; Verbeke, W.; Coosemans, T.; Faid, S.; Van Mierlo, J. A Data-Driven Method for Energy Consumption Prediction and Energy-Efficient Routing of Electric Vehicles in Real-World Conditions. *Energies* **2017**, *10*, 608. [CrossRef]
2. Nageshrao, S.P.; Jacob, J.; Wilkins, S. Charging cost optimization for EV buses using neural network based energy predictor. *IFAC-PapersOnLine* **2017**, *50*, 5947–5952. [CrossRef]
3. Galvin, R. Energy consumption effects of speed and acceleration in electric vehicles: Laboratory case studies and implications for drivers and policymakers. *Transp. Res. Part D Transp. Environ.* **2017**, *53*, 234–248. [CrossRef]
4. Maia, R.; Silva, M.; Araújo, R.; Nunes, U. Electrical vehicle modeling: A fuzzy logic model for regenerative braking. *Expert Syst. Appl.* **2015**, *42*, 8504–8519. [CrossRef]
5. Chen, H.-C.; Li, S.-S.; Wu, S.-L.; Lee, C.-Y. Design of a Modular Battery Management System for Electric Motorcycle. *Energies* **2021**, *14*, 3532. [CrossRef]
6. Adaikkappan, M.; Sathiyamoorthy, N. Modeling, state of charge estimation, and charging of lithium-ion battery in electric vehicle: A review. *Int. J. Energy Res.* **2021**, *46*, 2141–2165. [CrossRef]
7. Shrivastava, P.; Soon, T.K.; Idris, M.Y.I.B.; Mekhilef, S. Overview of model-based online state-of-charge estimation using Kalman filter family for lithium-ion batteries. *Renew. Sustain. Energy Rev.* **2019**, *113*, 109233. [CrossRef]
8. Uzair, M.; Abbas, G.; Hosain, S. Characteristics of Battery Management Systems of Electric Vehicles with Consideration of the Active and Passive Cell Balancing Process. *World Electr. Veh. J.* **2021**, *12*, 120. [CrossRef]
9. Rezaei, O.; Moghaddam, H.A.; Papari, B. A fast sliding-mode-based estimation of state-of-charge for Lithium-ion batteries for electric vehicle applications. *J. Energy Storage* **2022**, *45*, 103484. [CrossRef]
10. Xiao, R.; Hu, Y.; Jia, X.; Chen, G. A novel estimation of state of charge for the lithium-ion battery in electric vehicle without open circuit voltage experiment. *Energy* **2022**, *243*, 123072. [CrossRef]
11. Wang, D.; Yang, Y.; Gu, T. A hierarchical adaptive extended Kalman filter algorithm for lithium-ion battery state of charge estimation. *J. Energy Storage* **2023**, *62*, 106831. [CrossRef]
12. Shen, P.; Ouyang, M.; Lu, L.; Li, J.; Feng, X. The Co-estimation of State of Charge, State of Health, and State of Function for Lithium-Ion Batteries in Electric Vehicles. *IEEE Trans. Veh. Technol.* **2018**, *67*, 92–103. [CrossRef]
13. Chin, C.; Gao, Z. State-of-Charge Estimation of Battery Pack under Varying Ambient Temperature Using an Adaptive Sequential Extreme Learning Machine. *Energies* **2018**, *11*, 711. [CrossRef]
14. Guo, S.; Ma, L. A comparative study of different deep learning algorithms for lithium-ion batteries on state-of-charge estimation. *Energy* **2023**, *263*, 125872. [CrossRef]
15. Zou, R.; Duan, Y.; Wang, Y.; Pang, J.; Liu, F.; Sheikh, S.R. A novel convolutional informer network for deterministic and probabilistic state-of-charge estimation of lithium-ion batteries. *J. Energy Storage* **2023**, *57*, 106298. [CrossRef]
16. Chen, L.; Yu, W.; Cheng, G.; Wang, J. State-of-charge estimation of lithium-ion batteries based on fractional-order modeling and adaptive square-root cubature Kalman filter. *Energy* **2023**, *271*, 127007. [CrossRef]
17. Chen, L.; Guo, W.; Lopes, A.M.; Wu, R.; Li, P.; Yin, L. State-of-charge estimation for lithium-ion batteries based on incommensurate fractional-order observer. *Commun. Nonlinear Sci. Numer. Simul.* **2022**, *118*, 107059. [CrossRef]
18. Li, F.; Zuo, W.; Zhou, K.; Li, Q.; Huang, Y.; Zhang, G. State-of-charge estimation of lithium-ion battery based on second order resistor-capacitance circuit-PSO-TCN model. *Energy* **2024**, *289*, 130025. [CrossRef]
19. Zhang, Q.; Yan, Y.; Wang, J.; Xing, Y.; Liu, Y.; Yin, G. Enhancing Autonomous Driving With Physics-Informed and Human-Guided Reinforcement Learning. *IEEE Trans. Syst. Man Cybern. Syst.* **2026**, 1–15. [CrossRef]
20. Shahriar, S.M.; Bhuiyan, E.A.; Nahiduzzaman, M.; Ahsan, M.; Haider, J. State of Charge Estimation for Electric Vehicle Battery Management Systems Using the Hybrid Recurrent Learning Approach with Explainable Artificial Intelligence. *Energies* **2022**, *15*, 8003. [CrossRef]
21. Chan, H.T.J.; Rubeša-Zrim, J.; Pichler, F.; Salihi, A.; Mourad, A.; Šimić, I.; Časni, K.; Veas, E. Explainable Artificial Intelligence for State of Charge Estimation of Lithium-Ion Batteries. *Appl. Sci.* **2025**, *15*, 5078. [CrossRef]
22. Pimentel, J.; McEwan, A.A.; Yu, H.Q. A Novel Real-Time Battery State Estimation Using Data-Driven Prognostics and Health Management. *Appl. Sci.* **2025**, *15*, 8538. [CrossRef]
23. El Fallah, S.; Kharbach, J.; Vanagas, J.; Vilkelytė, Ž.; Tolvaišienė, S.; Gudžius, S.; Kalvaitis, A.; Lehmam, O.; Masrour, R.; Hammouch, Z.; et al. Advanced State of Charge Estimation Using Deep Neural Network, Gated Recurrent Unit, and Long Short-Term Memory Models for Lithium-Ion Batteries under Aging and Temperature Conditions. *Appl. Sci.* **2024**, *14*, 6648. [CrossRef]
24. Ma, H.; Bao, X.; Lopes, A.; Chen, L.; Liu, G.; Zhu, M. State-of-Charge Estimation of Lithium-Ion Battery Based on Convolutional Neural Network Combined with Unscented Kalman Filter. *Batteries* **2024**, *10*, 198. [CrossRef]
25. Song, X.; Yang, F.; Wang, D.; Tsui, K.-L. Combined CNN-LSTM Network for State-of-Charge Estimation of Lithium-Ion Batteries. *IEEE Access* **2019**, *7*, 88894–88902. [CrossRef]

26. Hannan, M.A.; How, D.N.T.; Lipu, M.S.H.; Mansor, M.; Ker, P.J.; Dong, Z.Y.; Sahari, K.S.M.; Tiong, S.K.; Muttaqi, K.M.; Mahlia, T.M.I.; et al. Deep learning approach towards accurate state of charge estimation for lithium-ion batteries using self-supervised transformer model. *Sci. Rep.* **2021**, *11*, 19541. [CrossRef] [PubMed]
27. Chemali, E.; Kollmeyer, P.J.; Preindl, M.; Emadi, A. State-of-charge estimation of Li-ion batteries using deep neural networks: A machine learning approach. *J. Power Sources* **2018**, *400*, 242–255. [CrossRef]
28. Sahinoglu, G.O.; Pajovic, M.; Sahinoglu, Z.; Wang, Y.; Orlik, P.V.; Wada, T. Battery State-of-Charge Estimation Based on Regular/Recurrent Gaussian Process Regression. *IEEE Trans. Ind. Electron.* **2018**, *65*, 4311–4321. [CrossRef]
29. Hannan, M.A.; Lipu, M.S.H.; Hussain, A.; Ker, P.J.; Mahlia, T.M.I.; Mansor, M.; Ayob, A.; Saad, M.H.; Dong, Z.Y. Toward Enhanced State of Charge Estimation of Lithium-ion Batteries Using Optimized Machine Learning Techniques. *Sci. Rep.* **2020**, *10*, 4687. [CrossRef] [PubMed]
30. Cai, S.; Chen, J.; Liu, Y.; Yuan, S.; Huang, R.; Zhang, P.; Hu, J. Pre-trip energy consumption prediction for battery electric vehicles via long-tail-aware cost-sensitive ensemble learning. *Appl. Energy* **2026**, *426*, 128616. [CrossRef]
31. Huang, H.; Gao, K.; Wang, Y.; Najafi, A.; Zhang, Z.; He, H. Sequence-aware energy consumption prediction for electric vehicles using pre-trip realistically accessible data. *Appl. Energy* **2025**, *401*, 126673. [CrossRef]
32. Choi, I.; Rah, C.; Kim, M.; Kim, H.; Kim, S.-j. Pre-trip energy-use prediction for micro electric vehicles from a single input: Driving-Profile Extraction and the Single Feature Prediction Model. *Energy* **2025**, *340*, 139293. [CrossRef]
33. Liu, Y.; Zhang, Q.; Lyu, C.; Liu, Z. Modelling the energy consumption of electric vehicles under uncertain and small data conditions. *Transp. Res. Part A Policy Pract.* **2021**, *154*, 313–328. [CrossRef]
34. Tymoshchuk, D.; Lyashuk, O. Electric Vehicle Trip-End State of Charge Prediction Dataset Based on Operational Data. *Mendeley Data* **2026**, *V1*. [CrossRef]
35. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-Learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
36. Scikit-Learn Developers. GroupKFold. Scikit-Learn Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupKFold.html (accessed on 5 August 2026).
37. Scikit-Learn Developers. GroupShuffleSplit. Scikit-Learn Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GroupShuffleSplit.html (accessed on 5 August 2026).
38. Scikit-Learn Developers. GridSearchCV. Scikit-Learn Documentation. Available online: https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html (accessed on 5 August 2026).
39. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
40. Geurts, P.; Ernst, D.; Wehenkel, L. Extremely Randomized Trees. *Mach. Learn.* **2006**, *63*, 3–42. [CrossRef]
41. Friedman, J.H. Greedy Function Approximation: A Gradient Boosting Machine. *Ann. Stat.* **2001**, *29*, 1189–1232. [CrossRef]
42. Rasmussen, C.E.; Williams, C.K.I. *Gaussian Processes for Machine Learning*; MIT Press: Cambridge, MA, USA, 2006. [CrossRef]
43. Rumelhart, D.E.; Hinton, G.E.; Williams, R.J. Learning Representations by Back-Propagating Errors. *Nature* **1986**, *323*, 533–536. [CrossRef]
44. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016*; Association for Computing Machinery: New York, NY, USA, 2016; pp. 785–794. [CrossRef]
45. CatBoost Developers. CatBoostRegressor. CatBoost Documentation. Available online: https://catboost.ai/docs/en/concepts/python-reference_catboostregressor (accessed on 5 August 2026).
46. Lou, Y.; Caruana, R.; Gehrke, J.; Hooker, G. Accurate Intelligible Models with Pairwise Interactions. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Chicago, IL, USA, 11–14 August 2013*; Association for Computing Machinery: New York, NY, USA, 2013; pp. 623–631. [CrossRef]
47. Duan, T.; Anand, A.; Ding, D.Y.; Thai, K.K.; Basu, S.; Ng, A.; Schuler, A. NGBoost: Natural Gradient Boosting for Probabilistic Prediction. In *Proceedings of the 37th International Conference on Machine Learning, Virtual Event, 13–18 July 2020*; Daumé, H., III, Singh, A., Eds.; Proceedings of Machine Learning Research (PMLR): Cambridge, MA, USA, 2020; Volume 119, pp. 2690–2700. Available online: https://proceedings.mlr.press/v119/duan20a.html (accessed on 5 August 2026).
48. Lawson, C.L.; Hanson, R.J. Solving Least Squares Problems. In *Classics in Applied Mathematics*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1995; Volume 15. [CrossRef]
49. Lei, J.; G'Sell, M.; Rinaldo, A.; Tibshirani, R.J.; Wasserman, L. Distribution-Free Predictive Inference for Regression. *J. Am. Stat. Assoc.* **2018**, *113*, 1094–1111. [CrossRef]
50. Romano, Y.; Patterson, E.; Candès, E.J. Conformalized Quantile Regression. In *Advances in Neural Information Processing Systems 32*; Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2019; pp. 3543–3553. Available online: https://papers.nips.cc/paper/8613-conformalized-quantile-regression.pdf (accessed on 5 August 2026).
51. Nadeau, C.; Bengio, Y. Inference for the Generalization Error. *Mach. Learn.* **2003**, *52*, 239–281. [CrossRef]

52. Benavoli, A.; Corani, G.; Demšar, J.; Zaffalon, M. Time for a Change: A Tutorial for Comparing Multiple Classifiers through Bayesian Analysis. *J. Mach. Learn. Res.* **2017**, *18*, 1–36.
53. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In *Advances in Neural Information Processing Systems 30*; Guyon, I., von Luxburg, U., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., Garnett, R., Eds.; Curran Associates, Inc.: Red Hook, NY, USA, 2017; pp. 4765–4774. Available online: https://proceedings.neurips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf (accessed on 5 August 2026).
54. Altun, B.E.; Alpsalaz, F.; Uzel, H.; Türkay, Y. Explainable DL Based Classification for Power Quality Disturbances in Renewable-Energy-Integrated Distribution Networks. *IET Renew. Power Gener.* **2026**, *20*, e70269. [CrossRef]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.