A Fully Automated Methodology for the Selection and Extraction of Energy-Relevant Features for the Energy Consumption of Battery Electric Vehicles

Patrick Petersen1 · Eric Sax1

Received: 14 September 2021 / Accepted: 25 May 2022 © The Author(s) 2022

Abstract
Compared to traditional vehicles, battery electric vehicles (BEVs) have a limited driving range. Therefore, accurately estimating the range of BEVs is an important requirement to eliminate range anxiety, which describes the driver’s fear of getting stranded. However, range estimators used in currently available BEVs are not accurate enough. To overcome this problem, more precise energy estimation techniques have been investigated. Modeling the energy consumption of BEVs is essential to obtaining an accurate estimation. For accurately estimating the energy consumption, many non-deterministic influencing factors such as weather and traffic conditions, driving style, and the travel route need to be considered. Thus, reducing the possible feature space to improve estimation is necessary. In consequence, we propose a fully automatic methodology to select and extract a subset of energy-relevant features. Utilizing existing real-world data to investigate all types of influencing factors. Taking into account different segmentation methods, data scalers, feature selection, and extraction techniques, our methodology uses the full range of combinations to identify the combination that yields the best subset of features.

Keywords Battery electric vehicle · Energy consumption · Feature engineering

Introduction and State of the Art
Electrification is one of the most important driving forces in the transformation of the automotive industry [1]. Compared to internal combustion engine vehicles (ICEVs), battery electric vehicles (BEVs) offer many advantages such as no emissions, low noise, and high energy efficiency. Thus, BEVs are seen as a promising solution for the reduction of greenhouse gases, pollution emissions, and dependence on fossil fuels [2]. Nevertheless, customers are still reluctant to adapt to BEVs due to the perceived limited range of such vehicles. The shorter range of BEVs, limited charging facilities, as well as longer charging time leads to range anxiety. Completely draining the battery is not only inconvenient for the driver but can also lead to dangerous situations, e.g., stopping on road shoulders due to an empty battery. Roughly, 9% of pedestrian fatalities occurred at locations such as shoulders [3]. Thus, car manufacturers extend the existing driving range by including improving battery technology [4] (e.g., higher density batteries [5]), advanced technologies to enable faster charging times [6], and pushing the deployment of more charging facilities.

Despite all efforts by manufacturers to increase the possible range of BEVs, drivers cannot even make the most of the actual range, because they do not fully trust the displayed range [8]. Drivers are more willing to charge their vehicle when the state of charge (SoC) is less than 15% and the remaining driving range is less than 50 km [9]. Up to 27% is used as a range safety buffer for trips of length 60 km to prevent running out of battery [10]. Therefore, accurately estimating the energy consumption, respectively, estimating the remaining range of BEVs can significantly reduce range anxiety without increasing actual battery capacity. In addition, energy-efficient routing of BEVs do also benefit from an accurate energy estimation [11].
Nevertheless, determining energy consumption and estimating the range are still a challenging problem. Due to numerous influencing factors such as driver style, road topology, and weather and traffic conditions, it is difficult to accurately estimate the energy consumption and actual range of BEVs. In addition, battery life cycles as well as the aging of the vehicle and battery need to be considered. As a result, range estimation algorithms attempt to reduce uncertainties in the estimation via various methods [12]. In the literature, they can be divided into two methods: history-based estimation and model-based estimation [13]. History-based estimation methods belong to the most common methods for estimation, especially for current available BEVs [14]. These algorithms calculate the range of a vehicle based on historical average energy consumption, current velocity, and current state of charge [7, 13]. History-based methods have the advantage of being model-free, but the accuracy of these methods is limited. They do not take into account the changing effects of current driver style, road topology, and traffic and weather conditions. In contrast to history-based methods for range estimation, model-based methods, as the name suggests, utilize a model to determine the energy consumption. Figure 1 schematically shows such a model-based method for range estimation. The exemplary model-based method considers influences such as driving style, battery and vehicle parameters (e.g., weight), traffic and weather conditions, as well as the route itself. Thus, not only the historical energy consumption but also currently available data are considered for the estimation. A vehicle energy consumption model is used to calculate the energy consumption along a route. Taking into account the state of charge, this route-based estimation can determine whether the specified destination is within reach. In addition, the remaining energy and range at the destination can be determined. This is particularly helpful for the driver, as they can assess whether a charging stop is necessary or whether they can continue driving. The main advantage of a model-based method is that the energy consumption is not based solely on historical consumption values, but can be calculated in advance based on the route. Thus, model-based estimations have proven to be more accurate than history-based methods [15]. However, model-based methods are not free of inaccuracies in the range estimation. Large deviations can occur if unforeseen traffic-related changes occur during the trip (e.g., rerouting due to accidents). It is important to mention that the accuracy of model-based methods depends upon the estimation of energy consumption. In the literature, a distinction is made between two classes of energy consumption estimations: microscopic and macroscopic models [16]. Microscopic models rely on physical equations such as driving resistances to calculate energy consumption. Figure 2 shows such simplified model.

The individual forces during driving consist of: aerodynamic drag force $F_{\text{aero}}$, rolling resistance $F_r$, climbing force $F_c$, and initial force $F_i$ [17]. The total tractive force $F_{\text{total}}$, 

![Fig. 1] Simplified structure and functionality of a model-based range estimation method [7]

![Fig. 2] Forces acting during the motion of a vehicle [17]
which is required to be generated by the electric motor of the vehicle, is calculated by the following equation:

\[
F_{\text{total}} = F_{\text{aero}} + F_r + F_c + F_i \\
= \frac{1}{2} \cdot \rho \cdot A \cdot c_w \cdot v^2 \\
+ c_r \cdot m \cdot g \cdot \cos \alpha \\
+ m \cdot g \cdot \sin \alpha \\
+ m \cdot a,
\]  \hspace{1cm} (1)

where

- \( \rho \) = Air density [kg/m³]
- \( A \) = Vehicle equivalent cross-section [m²]
- \( c_w \) = Aerodynamic drag coefficient [-]
- \( v \) = Vehicle velocity [m/s]
- \( c_r \) = Rolling resistance coefficient [-]
- \( m \) = Total vehicle mass [kg]
- \( g \) = Gravitational acceleration [m/s²]
- \( \alpha \) = Road gradient angle [°]
- \( a \) = Vehicle acceleration [m/s²].

Accordingly, the following equation describes the required energy \( E \) for a trip with the duration \( T \) at the velocity \( v \):

\[
E = \int_0^T F_{\text{total}} \cdot v \cdot dt. \hspace{1cm} (2)
\]

Accurately calculating the energy consumption of a vehicle requires a lot of effort. Modeling not only the aforementioned parameters but also every other required parameter such as internal losses and their respective efficiency. Some of them are difficult to model, others are unknown, especially to researchers, due to limited knowledge of vehicle construction and incorporated components. In contrast, macroscopic models use various parameters to calculate the energy consumption, which are learned directly from the available real-world data. These models can therefore be described as data-driven methods, because they are not based on modeling physical conditions, but rather learn them from the data itself. Machine learning methods also apply to this category of models. They adapt to different parameters in real time to achieve improved estimation quality. The model-based range estimation method shown in Fig. 1 is a macroscopic model, respectively, data-driven model.

The selection of parameters for data-driven models is crucial for the accuracy of the estimation. Due to a variety of influencing factors, such as driving style, road topology, and traffic and weather conditions, the actual range that BEVs can travel varies greatly [18]. The use of climate control, as well as auxiliary systems, has an additional impact on the actual energy consumption. Figure 3 summarizes the main influencing factors and shows exemplary data attributes for each of the fundamental categories of influencing factors.

Thus, to determine the actual range of a vehicle, it is necessary to take all these influencing factors into account when estimating the energy required for a given route by the navigation system. However, including unimportant features for the target variable can reduce the accuracy of some models [20]. Overfitting the model when the resulting feature space is too large can lead to “curse of dimensionality” [21]. Using numerous features for the estimation can slow down model development due to increased training time and required memory. It is a common approach to reduce the high-dimensional feature space by eliminating features without losing information for the model [22];1 resulting in eliminating only the unimportant features to improve model accuracy, reduction of calculation time, and benefits of improving understanding of the model due to less-used features.

This paper continues the work presented in [23], where we assessed the current state of the art in feature engineering for identifying as well as calculating relevant features. We noticed that non-present research investigated different influencing factors concurrently. Thus, a methodology for a data-driven analysis of energy-relevant factors covering driver style, road topology, and weather and traffic conditions were

---

1 This image has been designed using resources from https://www.flaticom.com.
presented. Our previous semi-automated approach required manually engineering features as well as manually selecting a suitable segmentation method for the automatic application of feature selection and extraction methods. In this research, no prior selection is needed, because we now make full use of the whole range of combinations for deducing suitable feature subsets. Thus, analyzing all possible combinations, covering segmentation methods, data scalers, as well as selection and extraction methods. These combinations and resulting feature subsets are evaluated regarding their estimation performance of different regression models such as Multiple Linear Regression, Gradient Boosting Regressor, Random Forest Regressor, and Support Vector Regression (incl. different kernel types). In contrast to our previous research, we now focus on the evaluation of feature subsets via the estimation itself, rather than only focusing on the covered data variance; leading to an exhaustive research for the identification of the best feature subset for the estimation of energy consumption of BEVs.

The remainder of this paper is organized as follows: “Methodology” introduces our overall methodology and core concepts for a fully automated selection and extraction of energy-relevant features for the energy consumption of BEVs. In “Results and Discussion”, our prototypical implementation of the methodology is evaluated on recorded real-world data from different drivers and routes. Finally, “Conclusion and Future Work” gives concluding remarks and discusses future work based on the results of this paper.

Methodology

Open standard process models such as Knowledge Discovery in Databases (KDD) and Crossindustry standard process for data mining (CRISP-DM) describe the data science life cycle for extracting knowledge from databases [24]. Although those processes vary slightly regarding the steps which need to be considered when working with data, they generally have the following in common: Those processes cover different steps to guideline data science and machine learning projects starting by covering data understanding, data preparation up to designing a machine learning algorithm, as well as their final evaluation and deployment [25, 26]. Based on these already established process models, we have developed similar steps for our methodology, which are shown in Fig. 4.

Our methodology starts by segmenting the recorded signals of the vehicle into suitable segments for feature calculation. During the feature engineering step, data within each segment undergo relevant data preparations steps, which enables those segments to be used for the calculation of features. In the feature selection step, a subset of energy-relevant features is selected. By applying different feature extraction methods, those selected feature subsets are further reduced and combined to new features, leading to a combined subset of extracted features. In the final step, those extracted features are evaluated regarding their usability within different regression models, which are...
representative of the application within advanced energy consumption estimation algorithms.

Segmentation

An important part of feature engineering is the segmentation of features. Especially, when dealing with time-series data, it is necessary to pay attention to the granularity of the features. Automotive real-world data from the vehicle contain recorded signals from the Controller Area Network (CAN); thus, time-series are directly influenced when sudden changes appear, such as changes in the velocity profile. Features calculated on the complete trip may suffer from missing out on fine-grained information, which would get lost when features are calculated on the whole trip. Thus, suitable segmentation methods need to be considered. There exist two main concepts for the segmentation of such data:

- Static segmentation.
- Dynamic segmentation (via grouping variables or micro-trips).

An example of the application of these segmentation methods is shown in Fig. 5.

A straightforward method is a static segmentation, which uses fixed intervals for segmenting the data. These intervals can be distance-based or time-based, thus resulting in segments of equal length in time or distance.

Dynamic segmentation in general applies different rules to the data. There are two commonly used methods for dynamic segmentation, based on micro-trips and via grouping variables. Dynamic segmentation is based on grouping variables to segment data when a value in at least one categorical signal is changing. Dynamic segmentation is based on micro-trips using intervals between two stops (micro-trip) along a trip when segmenting data.

In the literature, authors utilized various segmentation methods for their research and stated different pros and cons regarding those segmentation methods. For the training of an energy consumption model, De Cauwer et al. applied time-based static segmentation (2 min, 5 min and 10 min) to segment the used data [27]. A similar study conducted by Li et al. investigated the effects of different segmentation methods on the energy consumption model [19]. Due to the nature of the static segmentation method, they mention that segments may suffer from information discontinuities, especially when dealing with categorical data. Calculating features by, e.g., aggregating changing speed limits is not trivial and may result in missing important information.

Kamble et al. proposed a methodology for developing a driving cycle using micro-trips extracted from real-world data, but they mention that a dynamic segmentation based on micro-trips suffers from information discontinuities [28]. In addition, segments retrieved between two stops may result in a big difference in length for each segment: no stop during a trip results in a segment of the length of the whole trip. Thus, important fine-grained information may get lost when aggregating a whole trip into one feature.

Ericsson et al. investigated in their study the influence of driving patterns on the fuel-use and exhaust of ICEVs [29]. For segmentation, they applied dynamic segmentation based on 11 grouping variables covering driver behavior, street topology, vehicle characteristics, and traffic. In another study by Langner et al., they derived logical scenarios via separating scenarios via categorical and ordinal grouping variables [30]. Elspas et al. used a rule-based method to detect scenarios based on multivariate time-series segmentation [31]. Navigation providers such as HERE and Google utilize dynamic segmentation based on grouping variables as well to segment the street map into segments [32, 33]. In general, when applying grouping variables, the used signals are homogeneous within a segment.

Due to the sheer amount of possible segmentation methods, it is necessary to analyze which of the methods performs best when different features are calculated and the different selection and extraction methods are applied. Thus, all of them need to be considered for an exhaustive evaluation of the possible combinations.

Feature Engineering

Feature engineering prepares the input data in such a way that it is compatible with the individual requirements of the machine learning algorithm. Thus, it is one of the most crucial parts when developing machine learning algorithms. In our methodology, it consists of three parts: signal preprocessing, feature calculation, and feature preprocessing.
In general, noisy and faulty data do harm the performance of a machine learning algorithm [34]. This is especially true for automotive real-world data recorded from CAN. Therefore, during data preprocessing, smoothing the original data is necessary. A commonly used smoothing filter is the Savitzky–Golay (SG) filter, which uses the time domain for applying a low-pass filter on the signal for removing noise in time-series without distorting the original information [35].

Another important aspect of data preprocessing is dealing with different data categories.

Signals in automotive data commonly belong to signal types of nominal, categorical, or ordinal, if the order of a categorical signal is known. However, categorical signals can lead to problems for machine learning algorithms, because many algorithms cannot operate on such data directly [36]. One way to cope with categorical signals is to apply encoding techniques such as One-hot encoding. This is done by converting categorical data to numerical values. For each of the categorical variables, a new binary variable is added. Therefore, whenever a categorical variable occurs in the original signal the value of the corresponding binary variable is 1, otherwise 0. This is beneficial for segmentation methods that suffer from aggregating categorical data, such as static segmentation.

Differences in the scales across input variables may increase the difficulty of the problem being modeled. An example of this is that large input values (e.g., a spread of hundreds or thousands of units) can result in a model that learns large weight values. A model with large weight values is often unstable, meaning that it may suffer from poor performance during learning and sensitivity to input values resulting in higher generalization error.

To prevent data leakage, the applied data scalers are only applied to the training data. Afterward, the fitted scalers are applied to the test data. Once fit, the data preparation algorithms or models can then be applied to the training dataset, and to the test dataset. Those now preprocessed signals are ready to be used for calculating features.

The step of feature calculation covers the proper transformation and aggregation of data into more suitable features for a machine learning algorithm, but it is not trivial to choose the correct representation for the features. Usually, experts engineer features using their domain knowledge to transform the original data into more suitable features. Due to the sheer possibilities when doing so, this can be a time-consuming task. Additionally, when relying on an expert, it may suffer from being biased by the expert resulting in an unfavorable selection of features. Therefore, tools and libraries such as tsfresh exist that automatically calculate numerous statistical features on time-series data without being biased by expert decisions [37]. Respective signals which cover all influencing factors on the energy consumption (see Fig. 3) need to be considered when working with such tools to fully utilize their potential contribution to the benefit of a machine learning algorithm.

**Feature Selection**

By eliminating irrelevant and redundant features, the risk of overfitting as well as the curse of dimensionality can be minimized to maximize the relevance of the remaining features. A naive approach for eliminating irrelevant features is to try out all the possible combinations to evaluate which feature subset yields the best results. Such approaches are called brute-force search or exhaustive search. However, the computational cost for an exhaustive search is extremely high, e.g., for 100 features \(2^{100} = 1.26^{30}\) different possibilities exist [38]. Therefore, feature selection approaches exist that rely on heuristics to find the best subset of features that allows for building a model with improved performance, reduced computational cost, and improved interpretability [39]. Based on how they utilize a machine learning model, feature selection methods can be classified into filter, wrapper, embedded, and hybrid methods [40, 41].

Filter methods do not rely on a machine learning model when selecting relevant features. They apply statistical measures to score the dependence between features to find the most relevant ones [42]. Examples of the possible statistical measures are correlation coefficient, variance, Chi-square test, and mutual information. The selected features benefit from a low inter-correlation between each other due to removing such features to further reduce redundancy [43]. In addition, filter methods are computationally very fast and can therefore be scaled easily.

Wrapper methods use a machine learning model, such as a predictive model, to evaluate feature subsets [39]. For those methods, finding the best subset of features can be described as a search problem, where different feature subset combinations are evaluated against each other. Genetic algorithms, random search, and sequential selection search can be used as a search strategy to find the optimal solution [44, 45]. However, based on the used machine learning model, the search for an optimal solution can be computationally high but generally achieve better results than filter methods. The most commonly used representatives for wrapper methods are sequential backward selection (SBS), sequential forward selection (SFS), and recursive feature elimination (RFE). They mainly differ in their approach to how the best feature subset is found while utilizing the machine learning model [46–48].

Embedded methods, as the name suggests, integrate feature selection as part of the machine learning model. They select features that have the most contribution to the accuracy of the machine learning model by performing feature selection during the construction and training of the model [39, 49]. Embedded methods are less prone
to overfitting compared to wrapper methods and are more accurate than filter methods. Embedded features exist in different variations such as regularization methods (e.g., lasso regression) and algorithm-based methods (e.g., decision trees) [50].

Hybrid feature selection methods combine different feature selection methods to get the best possible feature subset. An example of such hybrids is the concept by Hsu et al. which combines filter and wrapper methods [51]. The main idea of that hybrid is to utilize the efficiency of filter methods and the accuracy of wrapper methods. Nevertheless, hybrid methods are currently not covered in this study.

**Feature Extraction**

Although applying feature selection methods to reduce dimensionality to achieve meaningful feature subsets, it may be beneficial to combine them to new more meaningful features [52]. Thus, resulting in additional improvement of the accuracy, reducing the risk of overfitting, and speeding up the training of the machine learning model. This is done by feature extraction methods, which reduce existing features by creating new ones via a linear combination of the existing ones. The original features are then removed from the feature subset, because the new features should be able to cover all the information of the originals. Commonly used techniques for feature extraction are Principal Component Analysis (PCA), exploratory factor analysis (EFA), and Linear Discriminant Analysis (LDA) [53, 54]. For EFA, the Kaiser’s criterion is used to find the optimal number of factors that have eigenvalues greater than one.

**Evaluation**

For the evaluation of the selected and extracted feature subsets, the performance of different prediction models needs to be evaluated. The formula for the energy consumption (Eq. 1 resp. Eq. 2) can be described as a linear combination of the kinematic parameters [55]. The coefficients of the linear combination can then be determined by applying a multiple linear regression (MLR). Thus, MLR is a suitable model to evaluate if a model can learn the relationship between energy consumption and selected resp. extracted features. Another widely popular machine learning model is the Support Vector Machine (SVM). Applying the concepts of SVM to regression problems is called Support Vector Regression (SVR) and benefits from being robust to outliers with high estimation accuracy [56]. However, studies have shown that mixing multiple machine learning models improves estimation performance when compared to using a single machine learning model [57]. This is called Ensemble Learning and has already been used to estimate the energy consumption of BEVss [58, 59]. Gradient boosting and Random Forest Regressor are such Ensemble Learning methods that increase the robustness of the utilized regression models [60, 61]. Thus, in this study, the following regression models are used for the evaluation:

- Multiple linear regression
- Support vector regression (incl. different kernel types)
- Gradient boosting regressor
- Random forest regressor.

This allows interpretation of each feature subset regarding their potential benefits for different models architectures. To measure the performance of each of the applied models, the $R^2$ scoring metric is used [62]. $R^2$ scoring metric between the real value $y$ and the predicted value $\hat{y}$ is calculated via the following equation:

$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2},$$

where: $\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i$.

For the evaluation of the trained predictive models, it is necessary to split the training from the test data. To prevent overfitting and achieve a less biased and less optimistic estimate of the model, we suggest the application of cross-validation. There exist different cross-validation techniques such as Resubstitution, Holdout, Leave-one-out, and K-Fold Cross-Validation [63]. In addition, there exist special cross-validation techniques for time series that pay attention to the timestamp of the data. This can be done by applying cross-validation on a rolling basis. This is especially reasonable if algorithms such as LSTMs are used that use prior data points to estimate future data points. However, this is not relevant in this study due to the applied linear models which learn the relation of a single calculated feature and its corresponding energy consumption instead of considering the time series itself. Thus, we use a standard K-Fold Cross-Validation with $k = 4$. 

---

**Fig. 6** Fourfold cross-validation for training and test data
Allowing us to evaluate four models on different parts of the data. Figure 6 shows the concept for the fourfold Cross-Validation.

Thus, the data are partitioned into a fixed number of folds. In each iteration, threefolds are used for training and onefold for testing the models on different parts of the data. After four iterations, we calculate the average of each iteration’s accuracies $E_i$ to retrieve our final score $E$ for evaluation of the models. This is done for each introduced model to evaluate the general performance of feature subsets in different models.

### Results and Discussion

For demonstrating the feasibility of our suggested methodology for the fully automated selection and extraction of energy-relevant features for the energy consumption of BEVs, we utilized a collection of different test drives including different drivers and routes, but with the same type of vehicle. Table 1 provides an overview of the used test drives.

In this study, we utilized signals from the vehicles’ CAN such as the vehicle’s velocity and acceleration, the drivers’ interaction with the vehicle (use of gas pedal), roads slope and altitude, information about the State of Charge, and the battery temperature including the outside temperature. For smoothing the noisy real-world data from the CAN, we applied the SG filter and sampled each signal with 10 Hz. Historical information about traffic conditions was obtained by the data provider HERE\(^2\) including speed regulations. However, additional information about historical weather conditions such as wind, perception, as well as visibility was not available during the experiment. Data providers such as the open-source database Meteostat\(^3\) did not offer suitable historical weather data for this research.

In this study, the following signals 13 have been included: brake pedal usage, gas pedal usage, mean speed of other road participants on road segments, max speed of other road participants on road segments, slope of road segments, speed limit of road segments, steering wheel angle, street class of road segments, temperature inside, temperature outside, vehicle acceleration, vehicle deceleration, and vehicle velocity. Finally, the categorical signals speed limit and street class have been one-hot encoded, resulting in 24 signals. For each signal, the following 14 different statistical features were calculated by utilizing the tsfresh library’s methods (version 0.17.0) [37]: abs_energy, absolute_maximum, absolute_sum_of_changes, count_above_mean, count_below_mean, maximum, mean, median, minimum, root_mean_square, standard_deviation, sum_values, variance, variation_coefficient. Leading to a total of 336 features for this study.

For the segmentation methods, we included different variations. For the static segmentation, we included distance-based lengths of $l \in \{500 \text{ m}, 1000 \text{ m}, ..., 5000 \text{ m}\}$ and time-based lengths of $t \in \{60 \text{ s}, 120 \text{ s}, ..., 600 \text{ s}\}$. Resulting in 20 static segmentation variations. For the dynamic segmentation, one micro-trip-based segmentation and three variations of grouping variables (speed limit, street class, speed limit + street class) were used. Thus, a total of 24 variations of segmentation methods were investigated.

In this study, we utilized data scalers, filter methods, and feature extraction methods from the scikit-learn library (version 1.0.0) with default parameters [64].

To investigate the different potential impacts of data scaling methods, numerous methods were considered. We investigated the following six variations for data scaling: None, Min-Max-Scaler, Standard-Scaler, Robust-Scaler, Quantile-Transformer, and Power-Transformer.

For the feature selection step, six methods were investigated. For filter methods, correlation-based methods (CF) as well as mutual information (MI) methods were implemented. Wrapper methods were covered by SBS, SFS, and RFE. For the embedded methods, only Least Absolute Shrinkage and

---

\(^2\) HERE website: [https://www.here.com/](https://www.here.com/).

\(^3\) Meteostat website: [https://meteostat.net](https://meteostat.net).
Selection Operator (LASSO) was implemented. For the extraction of features, PCA and EFA were utilized. Table 2 summarizes the implemented variations of each of the steps of our methodology, as well as the final result of investigated combinations.

As already mentioned in Sect. 2.5 to evaluate each of the 1728 combinations, we used fourfold cross-validation with $R^2$ scoring metric to calculate the average score $E$ for five regression models: multiple linear regression (MLR), gradient boosting regressor (GBR), random forest regressor (RFR), support vector regression with a polynomial kernel ($SVR_{poly}$), and support vector regression with a Radial Basis Function kernel ($SVR_{RBF}$). The execution time of our experiment is measured on a PC with an Intel Core i5-8600K processor, which runs at a frequency of 3.60 GHz, and 32 GB of ram. The total execution time of our methodology and the 1728 combinations was roughly 19 days. Table 3 shows the execution time for the segmentation method with the most data points (500 m, 78,000 points) and the least data points (micro-trips, 3800 points).

It can be seen that running the wrapper methods SFS and SBS took the longest to calculate. These findings are consistent with research, showing that the search for an optimal solution can be computationally high for wrapper methods.

A combined result score of the five model scores was calculated to evaluate the overall performance of a combination for every utilized model. Considering a threshold of $R^2 > 0.9$, roughly 98% of the evaluated combinations did not meet that criterion. Resulting in 50 remaining combinations for further investigation. Figure 7 shows an overview of the feature selection methods for the remaining top 50 combinations.

All the implemented feature selection methods are present in the top 50 combinations, but LASSO and CF are the most represented methods. These findings suggest that selecting relevant features by using LASSO and CF yield better results compared to other selection methods. Especially, for MI only two combinations remained, thus it is the least represented method of them all.

Table 4 shows the detailed results for the best ten combinations. It shows that every segmentation method, besides micro-trips, is represented in the top ten combinations. We speculate that this might be due to the nature of a micro-trip-based segmentation, which aggregation larger parts of a trip for calculating features. Thus, it may suffer from missing out on fine-grained information, which would get lost when features are calculated on larger parts of a trip. Every combination in the top ten utilized the Quantile-Transformer as a data scaler during preprocessing. This suggests that for scaling data, the application of the Quantile-Transformer should be considered as an additional preprocessing step. All selection methods and extraction methods are present in the top ten combinations. However, EFA is the most represented one and PCA is only included in the combination with a correlation-based filter and dynamic segmentation with grouping variables. From this standpoint, EFA can be considered the preferred extraction method. For the top ten combinations in Table 4, on average, roughly 23 features have been selected by the applied selection methods. Thus, reducing the 336 features by roughly 93%. It can therefore be assumed that roughly 23 features are sufficient for covering relevant data of vehicle signals if those features are carefully selected. Furthermore, an average of six factors could be extracted by the applied feature extraction methods for the selected features in Table 4. It can be noted that SBS selected the most and SFS the least number of features; resulting in 1 extracted feature for SBS and 16 for SFS. A note of caution is due here, since using only one extracted feature for covering the various aspects of relevant vehicle signals may be undesired due to the limited interpretability of that extracted feature. One unanticipated finding was that $SVR_{poly}$ performed

| Step | Execution Time |
|------|----------------|
| Feature Engineering | |
| 500 m | Micro-Trips |
| 21 min 11 s | 10 min 20 s |
| Selection | |
| CF | 9 s | 4 s |
| MI | 58 s | 4 s |
| RFE | 4 min 3 s | 9 s |
| SFS | 35 min 28 s | 02 min 01 s |
| SBS | 12 h 38 min 47 s | 39 min 27 s |
| LASSO | 1 s | 1 s |
| Extraction | |
| PCA* | 2 h 33 min 25 s | 15 min 33 s |
| EFA* | 3 h 16 min 43 s | 16 min 47 s |
| Total | 19 h 30 min 45 s | 1 h 24 min 26 s |

* Including segmentation and one data scaler
** Combined execution time of all selected feature sets
worse than the other regression models; indicating that a polynomial kernel function should not be considered as a regression model for the energy consumption. However, no significant performance differences could be seen for the other regression models. The overall combined results of every combination in the top 10 are better than 0.93. It is important to highlight the fact that only regression models were used in this study. It remains unclear to which degree different model architectures may yield different results in regard to performance and selected or extracted features. However, according to these data, we can infer that even though wrapper methods such as SBS for SFS yield good results, other selections methods should be considered due to reduced execution time and yielding similar results. Thus, the total execution time can be reduced drastically for our methodology by leaving out wrapper methods. To further investigate the best combination, Fig. 8 shows a scree plot for the Kaiser’s criterion applied to the 16 selected features of the first entry in Table 4.

![Scree plot for the 16 selected features of the best combination](image)

Fig. 8 Scree plot for the 16 selected features of the best combination

It shows that six factors have eigenvalues $\geq 1$ and thereby meet the Kaiser’s criterion. To investigate how much data can be covered by the selected number of factors is known as the extracted variance. A sufficient extracted variance for the factors needs to cover 75–90% of the data [65]. In Fig. 9, the extracted variance for each factor as well as the cumulative extracted variance of the 9 factors is shown.

Even though 6 factors meet the Kaiser’s criterion and the overall performance, regarding the $R^2$ scoring metric gave the best results, the extracted variance of 66% does not meet the suggested lower threshold of 75%.

As seen in the results, no prior decisions can be made when deciding on a segmentation, data scaling, selection, and extraction method. These findings suggest that each possible combination needs to be considered and evaluated reasonably to retrieve the best feature subset. Thus, confirming our suggested methodology for a fully automated selection and extraction of energy-relevant features for the energy consumption of BEVs.
Conclusion and Future Work

The main contribution of this work is a fully automated methodology for the selection and extraction of energy-relevant features for the energy consumption of BEVs. For this purpose, we design our methodology to cover all important steps: segmentation, feature engineering, feature selection, feature extraction, and evaluation. Due to many existing methods in each of the steps, we investigated a selection of them for utilizing the full range of combinatorial possibilities. Based on 336 calculated features, a total of 1728 combinations of different segmentation, selection, and extraction methods as well as data scalers were investigated.

By evaluating every combination, only roundabout 2% (= 50 in total) of all combinations did meet that criterion of an averaged $R^2$ metric score for all considered regression models to be $> 0.9$. On average, the 336 features could be reduced to roughly 23 selected features and 6 extracted features for the top 10 combinations. No clear conclusion could be drawn regarding which segmentation, data scaling, selection, and extraction method should be considered all the time. Only the Quantile-Transformer data scaler was represented in all the top ten combinations. Thus, no prior decisions can be made when deciding on one of the methods. The presented results proved our motivation for our methodology to make use of the full range of combinatorial possibilities. However, if execution time is an important aspect of one’s research, wrapper methods should not be considered due to high computational cost and yielding similar results when compared to filter methods.

Although the current study investigates several features regarding influencing factors on energy consumption, features of weather conditions are mainly missing. Thus, integrating data providers for historical weather information may provide additional insights into relevant weather condition features on energy consumption. In addition, the scope of this study did not consider any parameter optimization for the utilized methods. However, this might prove beneficial and yield different results, therefore, need further investigation.

Future work may extend the repertoire of methods and algorithms in each of the methodology steps such as extending automated calculated features, including hybrid feature selection methods, applying different rotation approaches for feature extraction, as well as adding additional models for the evaluation. Thus, extending and investigating more combinatorial possibilities when applying a fully automated methodology for the selection and extraction of energy-relevant features. Furthermore, future work may investigate the potential differences in feature subsets when applying our methodology to specialized data. Identifying different energy-relevant features for varying driver types and road types may provide useful insights on designing specialized machine learning models for special conditions to further improve the estimation of energy and range for BEVs.

Acknowledgements We thank Jan Sternagel for his help in implementing the concept. Special thanks to Philipp Rigoll for his constructive criticism of the manuscript. We also gratefully thank Dr. Ing. h.c. F. Porsche AG for providing the data for this study.

Funding Open Access funding enabled and organized by Projekt DEAL.

Availability of data and materials Not applicable.

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Code availability Not applicable.

Ethical approval Not applicable.

Consent to participate Not applicable.

Consent for publication Not applicable.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licenses/by/4.0/.
References

1. Pfeil F. Megatrends und die dritte Revolution der Automobilindustrie: Eine Analyse der Transformation der autonomen Wertschöpfung auf Basis des Diamantmodells. Research Papers on Marketing Strategy Julius-Maximilians-Universität (13). 2018.

2. Andwari AM, Pesiridis A, Rajoo S, Martinez-Botas R, Esfahanian V. A review of battery electric vehicle technology and readiness levels. Renew Sustain Energy Rev. 2017;78:414–30. https://doi.org/10.1016/j.rser.2017.04.063.

3. National Center for Statistics and Analysis: Pedestrians: 2017 data

4. Burg JTJ, Moore EA, Ezzat H, Kirchain R, Roth R. Improvements in electric vehicle battery technology influence vehicle lightweighting and material substitution decisions. Appl Energy. 2021;283:116269. https://doi.org/10.1016/j.apenergy.2020.116269.

5. Habib AA, Hasan MK, Mahmud M, Motakabber S, Ibrahimy MA, Islam S. A review: energy storage system and balancing circuits for electric vehicle application. IET Power Electron. 2021;14(1):1–13.

6. Makeen P, Ghali HA, Memon S. A review of various fast charging power and thermal protocols for electric vehicles represented by lithium-ion battery systems. Future Transport. 2022;2(1):281–99. https://doi.org/10.3390/futuretransp20010015.

7. Enthaler A, Gauterin F. Method for reducing uncertainties of predictive range estimation algorithms in electric vehicles. In: 2015 IEEE 82nd vehicular technology conference (VTC2015-Fall), 2015. pp. 1–5. https://doi.org/10.1109/VTCFall.2015.7391023.

8. Zhang Z, Tian R. Studying battery range and range anxiety for electric vehicles based on real travel demands. In: Proceedings of the human factors and ergonomics society annual meeting 2021, vol. 65, no. 1, pp. 332–336. Los Angeles: SAGE Publications Sage CA.

9. Pevec D, Babic J, Carvalho A, Ghiassi-Farrokhfal Y, Kettery W, Podobnik V. Electric vehicle range anxiety: an obstacle for the personal transportation (re)evolution? In: 2019 4th international conference on smart and sustainable technologies (SpliTech), 2019. pp. 1–8. https://doi.org/10.23919/SpliTech.2019.8783178.

10. Yuan Q, Hao W, Su H, Bing G, Gui X, Safikhani A. Investigation on range anxiety and safety buffer of battery electric vehicle drivers. J Adv Transport 2018;12.

11. De Cauwer C, Verbeke W, Van Mierlo J, Coosemans T. A model for range estimation and energy-efficient routing of electric vehicles in real-world conditions. IEEE Trans Intell Transport Syst. 2020;21(7):2787–800. https://doi.org/10.1109/ITITS.2019.2918019.

12. Kruppok K, Kriesten R, Sax E. Calculation of route-dependent energy saving potentials to optimize ev’s range. In: Internationales Stuttgart symposium, 2018. pp. 1349–1363. https://doi.org/10.1007/978-3-658-21194-3_106.

13. Hong J, Park S, Chang N. Accurate remaining range estimation for electric vehicles. In: 2016 21st Asia and South Pacific design automation conference (ASP-DAC), 2016. pp. 781–786. https://doi.org/10.1109/ASPDAC.2016.7428106.

14. Miri I, Fotouhi A, Ewin N. Electric vehicle energy consumption modelling and estimation—a case study. Int J Energy Res. 2021;45(1):501–20. https://doi.org/10.1002/er.5700.

15. Oliva JA, Weihrauch C, Bertram T. Model-based remaining driving range prediction in electric vehicles by using particle filtering and Markov chains. World Electr Veh J. 2013;6(1):204–13. https://doi.org/10.1109/EVS.2013.6914989.

16. Kubička M, Sciarratta A, Cela A, Mounier H, Thibault L, Niculescu S.-I. About prediction of vehicle energy consumption for eco-routing, 2016. pp. 1096–101 . https://doi.org/10.1109/ITSC.2016.7795693.

17. Doppelbauer M. Grundlagen der Elektromobilität: Technik, Praxis, Energie Und Umwelt. Springer Fachmedien Wiesbaden, Wiesbaden 2020. https://doi.org/10.1007/978-3-658-29730-5.

18. Smuts M, Scholtz B, Wesson J. A critical review of factors influencing the remaining driving range of electric vehicles. In: 2017 1st international conference on next generation computing applications (NextComp), 2017. pp. 196–201. https://doi.org/10.1109/NEXTCOMP.2017.8016198.

19. Li W, Stanula P, Egede P, Kara S, Herrmann C. Determining the main factors influencing the energy consumption of electric vehicles in the usage phase. Procedia CIRP. 2016;48:352–7. https://doi.org/10.1016/j.procir.2016.03.014.

20. Kuhn M, Johnson K. Applied predictive modeling. Springer, New York; 2013. https://doi.org/10.1007/978-1-4614-6849-3.

21. Bellman RE. Adaptive control processes. Princeton University Press; 1961. https://doi.org/10.1515/9781400874668.

22. Verleysen DMF. The curse of dimensionality in data mining and time series prediction. In: Computational intelligence and bio-inspired systems, 2005. pp. 758–70. https://doi.org/10.1007/11494669_93.

23. Petersen P, Thorgeirsson A, Scheubner S, Otten S, Gauertin F, Sax E. Training and validation methodology for range estimation algorithms. In: Proceedings of the 5th international conference on vehicle technology and intelligent transport systems—VEHITS, 2019. pp. 434–43. https://doi.org/10.5220/0007717004340443.

24. Pistorius F, Baumann D, Seidel L, Sax E. Intuitive time-series-analysis-toolbox for inexperienced data scientists. In: 2020 international conference on computational science and computational intelligence (CSCI), 2020. pp. 401–06. https://doi.org/10.1109/CSC151800.2020.00075.

25. Frawley WJ, Platetsky-Shapiro G, Matheus CJ. Knowledge discovery in databases: an overview. AI Mag. 1992;13(3):57–57.

26. Shearer C. The CRISP-DM model: the new blueprint for data mining. J Data Warehouse. 2000;5(4):13–22.

27. De Cauwer C, Van Mierlo J, Coosemans T. Energy consumption prediction for electric vehicles based on real-world data. Energies. 2015;8(8):8573–93. https://doi.org/10.3390/en8088573.

28. Kamble SH, Mathew TV, Sharma GK. Development of real-world driving cycle: case study of Pune, India. Transp Res Part D Transp Environ. 2009;14(2):132–40.

29. Ericsson E. Independent driving pattern factors and their influence on fuel-use and exhaust emission factors. Transp Res Part D Transp Environ. 2001;6(5):325–45. https://doi.org/10.1016/S1361-9209(01)00003-7.

30. Langner J, Grolig H, Otten S, Holzäpfel M, Sax E. Logical scenario derivation by clustering dynamic-length-segments extracted from real-world-driving-data. In: Proceedings of the 5th international conference on vehicle technology and intelligent transport systems—VEHITS, 2019. pp. 458–67. https://doi.org/10.5220/0007723304580467.

31. Elsps P, Kloose Y, Isele S, Bach J, Sax E. Time series segmentation for driving scenario detection with fully convolutional networks. In: Proceedings of the 7th international conference on vehicle technology and intelligent transport systems—VEHITS, 2019. pp. 458–67. https://doi.org/10.5220/0007723304580467.

32. HERE: Guide-Routing API. https://developer.here.com/documentation/routing/dev_guide/topics/resource-type-route-link.html. Accessed 9 Sept 2021.

33. Google: Overview-Directions API. https://developers.google.com/maps/documentation/directions/overview?hl=en#DirectionsResponseElements. Accessed 9 Sept 2021.
34. Zhu X, Wu X. Class noise vs. attribute noise: a quantitative study. Artif Intell Rev. 2004;22(3):177–210. https://doi.org/10.1007/s10462-004-0751-8
35. Savitzky A, Golay MJE. Smoothing and differentiation of data by simplified least squares procedures. Anal Chem. 1964;36(8):1627–39. https://doi.org/10.1021/ac60214a047
36. Hancock JT, Khoshgoftaar TM. Survey on categorical data for neural networks. J Big Data. 2020;7:1–41.
37. Christ M, Braun N, Neuffer J, Kempa-Liehr AW. Time series feature extraction on basis of scalable hypothesis tests (tsfresh-a python package). Neurocomputing. 2018;307:72–7. https://doi.org/10.1016/j.neucom.2018.03.067
38. Shardlow M. An analysis of feature selection techniques. Univ Manchester. 2016;1(2016):1–7.
39. Aggarwal CC, Han J, Yu PS. Active learning: a survey. Data Classif Algor Appl. 2014. https://doi.org/10.1201/b17320.
40. Chandrashekar G, Sahin F. A survey on feature selection methods. Comput Electr Eng. 2014;40(1):16–28. https://doi.org/10.1016/j.compleceng.2013.11.024.
41. Kutlu Y, Kuntalp D. Feature extraction for ECG heartbeats using higher order statistics of WPD coefficients. Comput Methods Progr Biomed. 2012;105(3):257–67. https://doi.org/10.1016/j.cmpb.2011.10.002.
42. Bosin A, Dessì N, Pes B. Capturing heuristics and intelligent methods for improving micro-array data classification. In: Intelligent data engineering and automated learning—IDEAL. 2007. pp. 790–9. https://doi.org/10.1007/978-3-540-77226-2_79.
43. Khalid S, Khalil T, Nasreen S. A survey of feature selection and feature extraction techniques in machine learning. In: Proceedings of 2014 science and information conference—SAI, 2014. pp. 372–78. https://doi.org/10.1007/978-3-642-181213.
44. Rodriguez-Galiano VF, Luque-Espinar JA, Chica-Olmo M, Mendes MP. Feature selection approaches for predictive modelling of groundwater nitrate pollution: An evaluation of filters, embedded and wrapper methods. Sci Total Environ. 2018;624:661–72. https://doi.org/10.1016/j.scitotenv.2017.12.152.
45. El Aboudi N, Benhlima L. Review on wrapper feature selection approaches. In: Proceedings of 2016 international conference on engineering and MIS-ICEMIS 2016. https://doi.org/10.1016/978-0-77453666.
46. Shen HT. Dimensionality reduction. Encyclopedia of database systems, 2009. pp. 843–6. https://doi.org/10.1007/978-0-387-39940-9_551.
47. Marciano-Cedeño A, Quintanilla-Domínguez J, Cortina-Januchs MG, Andina D. Feature selection using sequential forward selection and classification applying artificial metaplasticity neural network. In: Proceedings of industrial electronics conference—IECON, 2010. pp. 2845–50. https://doi.org/10.1109/IECON.2010.5675075.
48. Le Thi HA, Nguyen VV, Ouchani S. Gene selection for cancer classification using DCA. Adv Data Min Appl. 2008. https://doi.org/10.1007/978-3-540-88192-6_8.
49. Lal TN, Chapelle O, Weston J, Elisseeff A. Embedded methods. Feature Extract Found Appl. 2006. https://doi.org/10.1007/978-3-540-35488-8_6.
50. L’Heureux A, Grolinger K, Elyamany HF, Capretz MAM. Machine learning with big data: challenges and approaches. IEEE Access. 2017;5:7776–97. https://doi.org/10.1109/ACCESS.2017.2696365.
51. Hsu H-H, Hsieh C-W, Lu M-D. Hybrid feature selection by combining filters and wrappers. Expert Syst Appl. 2011;38(7):8144–50. https://doi.org/10.1016/j.eswa.2010.12.156.
52. Prabhu P, Anbazhagan N. Improving the performance of k-means clustering for high dimensional data set. Int J Comput Sci Eng. 2011;3(6):2317–22.
53. Yong AG, Pearce S. A beginner’s guide to factor analysis: focusing on exploratory factor analysis. Tutor Quant Methods Psychol 2013;9(2):79–94. https://doi.org/10.20982/tqmp.09.2.p079.
54. Zebari R, Abdulazeez A, Zeebaree D, Zebari D, Saeed J. A comprehensive review of dimensionality reduction techniques for feature selection and feature extraction. J Appl Sci Technol Trends. 2020;1(2), pp. 56–70. https://doi.org/10.38094/jastt1224.
55. De Cauwer C, Verbeke W, Coosmans M, Fauts, Van Mierlo J. A data-driven method for energy consumption prediction and energy-efficient routing of electric vehicles in real-world conditions. Energies. 2017. https://doi.org/10.3390/en10050608.
56. Awad M, Khanna R. Support vector regression. Elifc Learn Mach. 2015–67–80.
57. Sagi O, Rokach L. Ensemble learning: a survey. WIREs Data Min Knowl Discov. 2018. https://doi.org/10.1002/widm.1249.
58. Chung YW, Khaki B, Li T, Chu C, Gadh R. Ensemble machine learning-based algorithm for electric vehicle user behavior prediction. Appl Energy. 2019;254(April):113732. https://doi.org/10.1016/j.apenergy.2019.113732.
59. Ullah I, Liu K, Yamamoto T, Zahid M, Jamal A. Electric vehicle energy consumption prediction using stacked generalization: an ensemble learning approach. Int J Green Energy. 2021;18(9):896–909. https://doi.org/10.1080/15435075.2021.1881902.
60. Friedman JH. Stochastic gradient boosting. Comput Stat Data Anal. 2002;38(4):367–78.
61. Cootes TF, Ionita MC, Lindner C, Sauer P. Robust and accurate shape model fitting using random forest regression voting. In: European conference on computer vision, 2012. pp. 278–291. Springer.
62. Anderson-Sprecher R. Model comparisons and R². 1994;48(2):113–17. https://doi.org/10.38094/jastt1224.
63. Mitchell HB. Data fusion: concepts and ideas. Springer, Berlin; 2012. https://doi.org/10.1007/978-3-642-27222-6.
64. Pedregosa F, Varoquaux G, Gramfort A, Michel V, Thirion B, Grisel O, Blondel M, Prettenhofer P, Weiss R, Dubourg V, Vanderplas J, Passos A, Cournapeau D, Brucher M, Perrot M, Duchesnay E. Scikit-learn: machine learning in python. J Mach Learn Res. 2011;12:2825–30.
65. Beavers AS, Lounsbury JW, Richards J, Huck SW, Skolits GJ, Esquivel SL. Practical considerations for using exploratory factor analysis in educational research. Pract Assess Res Eval. 2013;18:1–13.

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.