i-ETC: ISEL Academic Journal of Electronics, Telecommunications and Computers
Vol. 9 , n. 1 (2023) ID-1

An Approach to Estimate Electric Vehicle Remaining Driving Distance

David A. S. G. Albuquerque[a] Artur J. Ferreira[a,b] David P. Coutinho[a,b]
[a]ISEL, Instituto Superior de Engenharia de Lisboa, Instituto Politécnico de Lisboa
[b]Instituto de Telecomunicações, Lisboa, PORTUGAL
A43566@alunos.isel.pt artur.ferreira@isel.pt david.coutinho@isel.pt

***Abstract*— The use of *electric vehicle* (EV) has grown rapidly over the past few years. With the increase of renewable energy mix share and the mature technology available on market, the EV is now accepted as a reliable and eco-friendly means of transportation. Considering that the charging time is high when compared to oil based and that the charging stations are still few, usually one of the key parameters of choice to EV to customer is its *driving range* (DR) capability. This is a decisive factor since it minimizes the driver's anxiety on a trip. The EV autonomy at a given point in time, denoted as the eRange, depends on many factors making its forecasting a difficult task. In this paper, we propose the use of *machine learning* (ML) techniques to compute the eRange. We use regression techniques on models trained with publicly available datasets, evaluated with standard metrics. The experimental results of ML techniques improve on the results by providing accurate and smoother estimates of the eRange, easing drivers anxiety.**

Keywords: dataset construction; driving range; electric vehicle; energy consumption; machine learning; regression; Python

I. INTRODUCTION

The global concern on climate change has been a major focus on recent international agreements, such as the Paris Agreement [9], leading many car manufacturers to introduce EV as the eco-friendly solution for sustainable transport for the future. Thus, EV has grown in popularity in recent years and as a result, car manufacturers have increased the competitiveness on the vehicle's performance [23], namely the DR capability (the distance such that the EV can travel with a single charge), since it is a decisive factor for customers [21].

On a trip, the EV autonomy at a given point in time, defined here as eRange, is an estimate of the remaining driving distance regarding multiple factors that might influence its range, such as the EV battery charge at that moment. The eRange is expressed in kilometers. This estimate eases the driver's anxiety on a trip to a charging station and allows the driver to do the best possible trip planning [36], [37]. The estimation of the eRange is a challenging task, since it depends on different parameters, such as:

- battery *state-of-charge* (SoC);
- commute type - city driving or highway driving;
- drivers behavior;
- road inclination;
- vehicle design;
- weather conditions.

These factors exhibit a wide range of variation, leading to a challenge when performing this estimation. Moreover, it also depends on the assumptions that one takes by using some priori knowledge about the problem. Figure 1 shows the main influencing forces on a vehicle, which are some of the key variables that lead to the actual battery energy consumption and makes the eRange computation a challenge task. The accurate estimation of the eRange allows customers to rely on its vehicle for longer travel time and efficient charging plans. The challenges and difficulties posed in the eRange estimation have lead to recent studies on this topic [20], [41].

A. The use of machine learning

In the past years, the rise in popularity of ML [11], [12], [31] has shown its effectiveness with a variety of fields such as big data [17], [48], pattern recognition analysis and data mining [14]. This is due to its nature of learning models from existing data to gradually achieve better results making it a widely recognized tool for complex problems [32]. As a result, some previous works have employed ML techniques to address the eRange estimation problem with supervised, unsupervised and reinforcement learning paradigms, making accurate predictions.

Some statistical-based eRange estimation techniques use the average consumption of energy in the past minutes on a trip. These simple estimations may produce inaccurate and noisy results, since they account only a small set of factors and the eRange estimation depends on multiple variables. It is expected that the use of ML methods will provide better

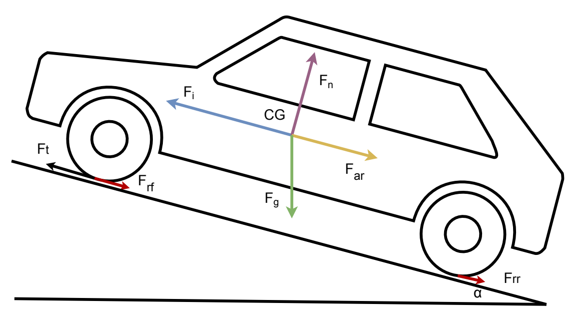

Fig. 1. The key influencing forces on a moving vehicle. (F_i, inertial force; F_t, tractive force; F_g, gravitational force; F_{rr}, rear rolling resistance force; F_{fr}, front rolling resistance force; F_{ar}, aerodynamic (air) drag; F_n, normal force; CG, center Figure; α, the road slope).

estimates. Although, the training of the models may take more time than the existing approaches, it is expected that their use will improve prediction accuracy. The challenge is to learn accurate models with adequate response time when placed on-board of the vehicles.

B. Our proposal

In this paper, we propose to address the eRange forecasting problem with regression ML techniques. We devise a three phase approach:

- the dataset construction and preprocessing;
- learning ML models;
- evaluation of the learned ML models, with standard metrics.

With this approach, we advance the state-of-the-art by using ML techniques different from the ones in previous studies. We only consider parameters related to the vehicle and we do not use historic data traffic.

The remainder of this paper is structured as follows. Section 2 refers to the state of the art on existing eRange estimation solutions and their use of the available datasets. In Section 3, we present the approach and methodologies adopted in this work. The experimental evaluation and discussion are reported in Section 4. The paper ends in Section 5 with concluding remarks and directions for future work.

II. State of the Art

Nowadays, EV have motivated multiple studies concerning related problems in this field, such as statistical measurement of charging [16], eRange computation [41], charging topologies [43], and regenerative braking [44]. A good eRange prediction is a key EV feature as it reduces driver's anxiety while driving and allows better trip planning.

When assessing a solution to the eRange prediction problem, real EV driving data in the form of a dataset is required to learn and evaluate the proposed model, and compare it to existing alternatives, making it indispensable in determining the effectiveness of the chosen solution.

In this section, we refer to the literature and resources for the eRange forecasting problem, the availability of public domain datasets, eRange estimation approaches without and with ML techniques.

A. Public domain EV datasets

The EV eRange estimation is a recent research topic. On the past years, some datasets regarding EV have become publicly available for researchers. These datasets are composed by vehicle data and trip data, which are useful to the eRange forecasting problem. These datasets are mainly composed by two types of feature:

- *time-series features*, where the data points vary as a function of time;
- *trip-invariant features*, in which a given value is kept for the entire trip.

Time-series features are usually the SoC, energy consumption, speed, acceleration, and elevation. The trip-invariant features refer to vehicle information such as battery capacity, *average energy consumption* (AEC), *full battery energy* (FBE), *full driving distance* (FDD) also known as *full battery distance* (FBD), vehicle weight, trip information such as commute type (city or highway), total energy consumption, and total distance. Table I summarizes the key publicly available EV datasets, namely:

- the *vehicle energy dataset (VED)* [33];
- the Emobpy dataset [24];
- the Classic EV X project dataset [19];
- the *Charge Car* project of the CREATE Lab at Carnegie Mellon University [1];
- the cloud based EV dataset provided by the *national big data alliance of new energy vehicles* (NDANEV) [8].

Table I describes some features found in these datasets (the * symbol denotes that the feature has missing values).

The *VED* dataset [33] provides 54 different EV driving trip data records for estimation, but lack trip and vehicle information as well as enough EV model variety. It contains data from three distinct EV, all from the same model, the 2013 *Nissan Leaf*.

The *Emobpy* Python tool [24] focuses on EV trip and charge data generation through empirical mobility statistics and customizable assumptions. This tool provides an infinite supply of EV trips as well as proper vehicle information. This dataset lacks some features such as speed, elevation, trip, and commute type.

The *Charge Car* project of the CREATE Lab at Carnegie Mellon University [1] publicly supplies crowd-sourced data that has served previous eRange prediction models [47]. This dataset has a high vehicle diversity due to the open nature of the platform, allowing any user to upload combustion engine based vehicle information as well as the location data, speed, and weather, among other parameters. For instance, the battery information data could be supplied through the CREATE RAV4-recorder box [2]. The location, trip and vehicle information are then used to determine the simulated EV consumption for each trip. The key features of the dataset relate to speed, distance, traffic conditions, hills, and driving behavior. As of the time of writing, a total of 373 unique trips are publicly available.

A dataset collected through probe data from nearly 500 battery EV by the *Japan automobile research institute* (JARI) from February 2011 to January 2013 has the following features: time, location, vehicle state (driving, normal charging, or fast charging), speed, air-conditioner, heater state, and SoC. Although useful and featured in some papers [29], [30], [38], [39], for this paper we were unable to acquire this dataset from [6] perhaps due to the language barrier.

Previous studies in eRange prediction [20] were based on *EVteclab*'s *electric vehicles in action* (EVA) platform, a Flemish Living Labs project [5]. The platform supplies a dataset with monitoring data of 30 different models *Ford Connect* EV for a time window covering a whole year. This dataset although supplying a few meaningful parameters such as timestamp, latitude, longitude, and vehicle speed, was inaccessible at the time of this writing.

TABLE I
PUBLIC DOMAIN DATASETS WITH EV DATA AND THEIR KEY PROPERTIES.

	VED dataset [33]	**Emobpy** [24]	**Classic EV X project** [19]	**ChargeCar** [1]	**NDANEV** [8]
Trips	507	Unlimited	3	373	2372
EV Models	1	102	1	?	1
Number of EV	3	N/A	1	?	5
Features	timestamp, speed, location, battery SoC, battery voltage, battery current, AC power, heater power, outside air temperature (OAT)	timestamp, distance, instant energy consumption (IEC), consumption, average power, state	timestamp, IEC, remain battery energy (RBE), speed	timestamp, elevation, planar distance, adjusted distance, speed, acceleration, model power, actual power*, current*, voltage*	timestamp, speed, total voltage, total current, battery SoC, temp. range, motor voltage, motor current, mileage

The cloud based EV dataset supplied by the NDANEV [8] has been used in similar eRange prediction approaches [46]. The data was collected from *controller area network* (CAN) of five different EV of an undisclosed model through with T-BOX, later uploading it to NDANEV. This dataset distinguishes from the others by including battery cell temperature information, which measures the battery cell inconsistency.

As some datasets do not explicitly provide vehicle information, the EV-database [3] website supplies a database for existing EV, displaying AEC, DR, and usable battery energy. The presence of this data, enables that datasets lacking this feature, can be used in eRange forecasting models.

B. EV autonomy prediction approaches

The eRange forecasting problem has been an interesting topic in research in recent years, in part due to the increase in EV usage as they become more efficient. The forecasting difficulty is in part due to the fact that there are many factors to take into account when measuring it, such as battery and road information, previous vehicle trips, and vehicle weight. This has motivated researchers to seek for solutions for the problem, resorting to ML techniques.

Related work has shown the use of eRange computation on EV, stating the need for different types of accuracy on eRange forcasting as a function of the SoC state. In [45], the approach is to minimize the performance impact of minimum cost route searching from high accuracy eRange forecasting. Other studies have focused on eRange estimation accuracy, making use of more complex models.

The use of an adaptive *"history-based"* approach was proposed by [19], which relies on the past 10 minute AEC information gradually influenced by the vehicle instant consumption energy, as well as by the SoC. The approach in [18] computes the eRange through a *"basic"* algorithm which depends on the manufacturers invariant vehicle information such as FBE and AEC, as well as the instant SoC value. Once the first 10 minutes had passed, the *"history-based"* applies a previously configured energy step to the previous prediction, as functions of the computed AEC. This implementation yielded more optimistic eRange results, with slightly higher values than the *"basic"* approach and thus easing drivers anxiety.

In detail, the *"basic"* algorithm computes the eRange through the combination of the EV model's characteristics provided by the vehicle's manufacturer such as:

- the FBE, which is the maximum charge the EV battery can store;
- the AEC, which depends on the air-conditioner AcS and the commute type (highway or city driving).

It also requires the SoC value, at the time of the eRange computation. Thus, the eRange is computed by

$$\text{eRange}(AcS, AEC) = \frac{FBE}{AEC(AcS)} \times SoC \quad [km]. \tag{1}$$

Figure 2 shows the block diagram of this approach.

The *"history-based"* method for eRange computation [22] is depicted in Figure 3. An adaptive version of this method was introduced in [19]. It relies on the FBE, SoC, and *instant energy consumption* (IEC) parameters, to compute each minute an adaptive value for AEC. So, the eRange for the k minute is computed by

$$\text{eRange}(k) = \left\lfloor \frac{FBE}{\sum_{i=0}^{N-1} w_i \times AEC_A(k-i-1)} \times SoC(k) \right\rfloor, \tag{2}$$

where $\lfloor . \rfloor$ is the floor operator, N is the number of past minutes of an observation moving window and w_i are the predefined weights to the moving average computation of each minute's adaptive AEC (AEC_A). We assign exponentially lower weights to the least recent AEC values

$$w_i = \frac{1}{2^{(i+1)}}, \quad i \in \{0, \ldots, N-1\}. \tag{3}$$

This algorithm requires three additional parameters: the delta energy step, ΔS, the manufacturer constant AEC,

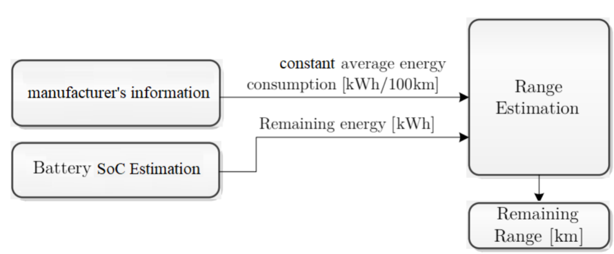

Fig. 2. *"Basic"* range estimation approach [22].

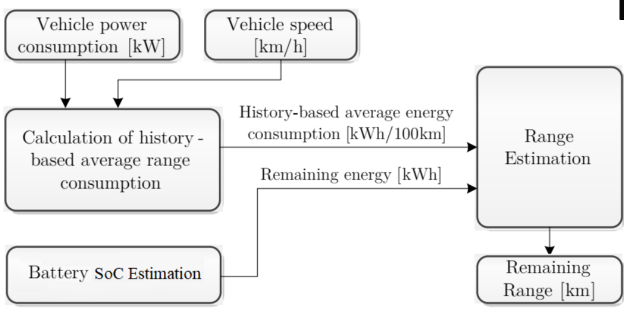

Fig. 3. *"History-based"* range estimation approach [22].

AEC_C, and the minimum instance energy. The ΔS represents the amount of energy the AEC_A increment/decrement at each k minute. It is used to compute $AEC_A(k)$ according to

$$AEC_A(k) = \begin{cases} AEC_C, & k <= N \\ AEC_A(k-1) - \Delta S, & AEC_{ma}(k) < 0 \\ AEC_A(k-1) + \Delta S, & AEC_{ma}(k) > 0. \end{cases} \tag{4}$$

by adding or subtracting ΔS to the previous AEC_A calculation of the previous $k-1$ minute. Initially, AEC_A is equal to the pre-configured AEC_C, until it is possible to calculate the moving average with minimum number of N samples. In this equation, AEC_{ma} represents the moving average of the current k minute IEC values, where every non-zero IEC value is averaged for its calculation.

The minimum instance energy's role is to prevent the algorithm from performing an eRange calculation when the average IEC values for the current k minute are less than a predefined threshold value. In case the vehicle consumes negligible power, it would not cause an ΔS decrement or increment on the eRange, thus preventing inaccurate eRange values.

The *"history-based"* algorithm is an improvement yielding slightly optimistic values than the *"basic"* approach, because it is adapted to the vehicle current usage.

C. The use of machine learning techniques

The use of ML techniques for a multitude of cases [12] in fields such as big data [17], [48], and data mining [14] has proven its robustness on solving different problems.

As a result, some approaches for the eRange estimation problem have resort to supervised learning techniques. The use of *decision trees* (DT) [11], *random forest* (RF) [15], and *K-nearest-neighbor* (KNN) [10], [11] in *ensemble stacked generalization* (ESG) approach [40], through the *JARI* dataset [40] has shown a better prediction than its individual base models. Recent models using *gradient boosted regression tree* (GBRT) [25] have combined *extreme gradient boosting* (XGBoost), available at `https://github.com/dmlc/xgboost` and *light gradient boosted machine* (LightGBM), from `https://github.com/microsoft/LightGBM` to provide better predictive performance from these ensemble methods [46] with the NDANEV dataset. The approach classified four driving patterns from three parameters (speed, motor current, and change rate of motor current), through k-means clustering algorithm [13] and thus influencing the resulting eRange due to their different energy consumption rates [7].

Approaches using unsupervised clustering of *self-organizing maps* (SOM) [27] have been used for clustering big data into driving patterns, prior to range estimation [28]. The hybrid version of SOM with *regression tree* (RT) [25] has taken advantage of SOM neurons storage feature of nearing related neighbor information being kept closely together. Avoiding bushy trees and improving upon previous solutions by keeping meaningful knowledge extraction on bushy trees [47] both approaches used different datasets from undisclosed monitored data sources.

Reinforcement learning in the form of *neural network* (NN) [26] has also been used for external energies disturbances on the speed profile of a driving profile so that it could then be combined with *multiple linear regression* (MLR) for the estimation [20], using *EVteclab*'s dataset.

III. Machine Learning Approach

In this Section, we detail the proposed approach followed in this work. Figure 4 depicts the generic diagram of the approach in which we highlight that the ultimate goal is to compute the eRange from the data in the input dataset, using ML techniques.

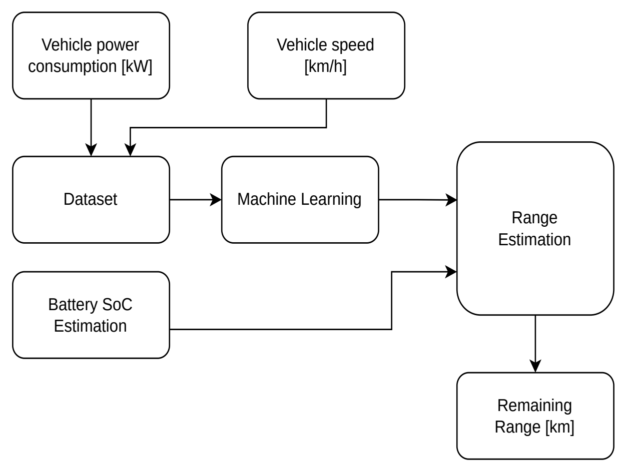

Fig. 4. Generic approach to compute the eRange value expressed in kilometers.

Figure 5 provides more details on the methodology steps that we have taken, dealing with datasets and the evaluation of the ML techniques. These steps are described in the following subsections.

A. Dataset construction and preparation

In some fields of ML use, we have many available ready-to-use datasets with full data. However, in this recent field of research about eRange computation, it is not straightforward to have such datasets. Thus, we need a dataset construction and preprocessing phase for this problem. In this work, a dataset was created from data with personally recorded

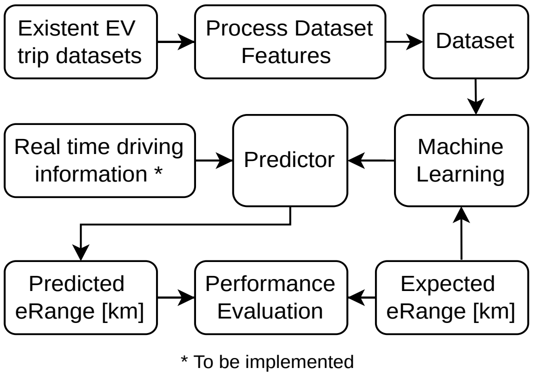

Fig. 5. The detailed steps of the proposed approach using ML to compute the eRange, from an EV dataset.

vehicle trips, as well as external existing and publicly available datasets from both the *VED* [33] and *ChargeCar* [1] datasets, integrated into the work dataset. The resulting dataset contains multiple trips with their respective vehicle power consumption (expressed in kW) and the vehicle speed (in km/h) in a time series format.

The composed dataset is used to train the selected eRange forecasting models on the learning phase through ML. Figure 6 depicts the dataset construction and composition for this work.

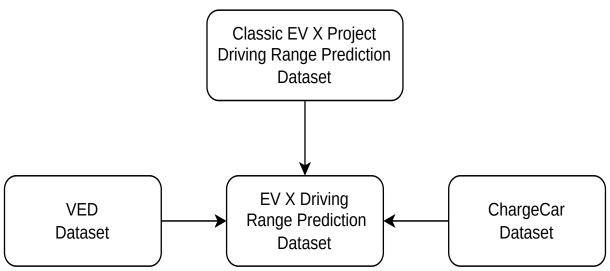

Fig. 6. The dataset construction and preparation for this work.

The key reasons for building such a dataset are as follows. When training a ML model for regression problems, the accuracy of the results on test data depends on the diversity of the data. To ensure model effectiveness on different vehicles and to avoid over-fitting, we opted for a diverse EV model dataset built from existing available datasets mentioned in Section II-A.

The algorithm integrates EV trip datasets for training, thus requiring EV trips time-series with the following features: SoC (in percentage), power consumption (in kW/h), distance (in km) and speed (in km/h). We also have vehicle information: AEC (kWh), FBE (kWh), and FBD (km). For this reason, both *VED* and *ChargeCar* datasets [1] were chosen.

When configuring the algorithm's training, different datasets can be selected, as well as a minimum trip type and minimum driving time, as these variables have been tested and found to highly influence ML methods performance. On the preprocessing phase, some features such as AEC, FBE, and FBD are sometimes missing on certain datasets. These, however, can be computed from existing static EV datasets such as [3]. Other features such as power variation and distance, are trip dependent, being computed as follows. The ΔP metric measures the power consumption variation between the previous i trip instant and the next f, using the DC power formula of current A times voltage V,

$$\Delta P = P_f - P_i = V_f \times A_f - V_i \times A_i. \tag{5}$$

The acceleration a is needed to calculate the distance feature of the dataset. It is computed as the difference of the two trip instant velocity, v, values divided by the elapsed time Δt

$$a = \frac{v_f - v_i}{\Delta t}. \tag{6}$$

For the distance ΔD computation, we use the previously computed acceleration a between trip instants and apply it to the initial velocity for the time variation, yielding

$$\Delta D = v_i \times \Delta t + \frac{1}{2} \times a \times \Delta t^2. \tag{7}$$

The driver's driving patterns dataset feature are clustered from acceleration, battery current and change rate of motor current with SOM, as shown by [46].

B. Learning the models

The target (expected) eRange values are provided by an implementation of an eRange estimation with the adaptive *"history-based"* algorithm, decribed in Section II-B. From the features of the dataset, we apply this approach to compute the target eRange values $\mathbf{y_i}$. This approach addresses real-time AEC values, that relies on vehicle's past $N = 10$ minute window of the trip's energy consumption history as well as the real-time SoC value.

After training the ML algorithms with the dataset, the estimation phase performs the eRange forecasting on live SoC monitoring of a driving EV. The resulting prediction is then used for the computation of the evaluation metrics to compare against other algorithms. The application features training configurations such as dataset feature configuration, minimum trip time and trip minimum time step, also provide execution configurations, as prediction algorithms and evaluation methods.

C. Regression techniques

We have considered the following regression techniques:

- *linear regression* (LR) [25];
- *ensemble stacked generalization* (ESG) [40].

The ESG algorithm follows the Wolpert stacking technique [42], combining two models. The first one, named as base-model (Level-0) encompasses DT, RF, and KNN classifiers. The second model (Level-1) is AdaBoost, combining base model predictions to provide a single output. Figure 7 depicts the ESG model, which follows the original [40] implementation with some differences. The original application was the EV energy consumption prediction and not eRange. Moreover, the lack of availability of its *JARI* dataset could make this implementation's accuracy differ when training

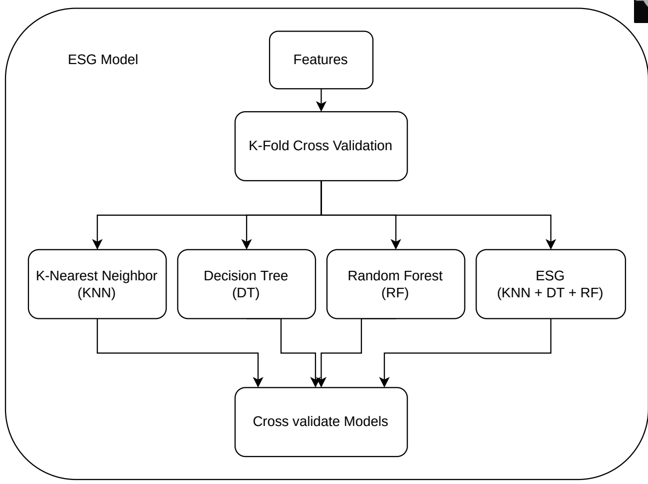

Fig. 7. Ensemble Stacked Generalization (ESG) model [40].

with our dataset. After the training of these ML algorithms, they are used for the prediction of eRange through real-time parameters. One of the predictors is then selected based on its performance with standard metrics. Then, it can be used for future execution in a real-time trip.

An additional ESG implementation named *ESG V2* was derived from the original ESG approach, with the same underlying ML algorithms, but changing some configurations to better fit the project's constructed dataset. The maximum number of features configured for DT and RF algorithms are 9 and 7, respectively. As for KNN, K is configured at 70, the distance metric is *minkowski* with parameter p set to 1.

D. Software platforms and programming languages

The software developed in this work was written in Python with the standard ML packages. The Python application was developed and published online [4], easing further academic research involving EV driving datasets and machine learning usage. The code is able to configure the EV X DRP dataset for training the ML algorithms, controlling the minimum time interval to reduce training bias, caused by trips with lower duration. The focus of the software is to allow the comparison of existing eRange prediction models with the project's implemented model. For this reason, two additional prediction models (besides "basic" and "history based" from [19]) were integrated into the application.

The software allows for algorithm training and trip execution customization settings. The selection of enabled eRange computation algorithms for training allows multi algorithm comparison for the same test trip. Some limitations exist when using the *Classic EV X Project Driving Range Prediction* eRange computation algorithms for ML training. Both algorithms require the EV models AEC and FBE provided by the manufacturer, limiting the training to datasets that do provide this data, effectively excluding datasets such as NDANEV.

Moreover, from this work, a Python application was developed, aiming for a future integration with the Classic EMini X [18] project which aims to transform a 1993 Rover Mini Cooper 1.3i (1300cc) into a fully EV. The vehicle could then request a new eRange estimation with real-time battery and road information through the existing eMini project software, as depicted in Figure 8.

IV. Experimental Evaluation

In this Section, we report on the experimental evaluation of our approach. First, we present the evaluation metrics considered in this work. Then, we report on experimental results on the built dataset.

A. Standard evaluation metrics

Since prediction accuracy must be assessed for each eRange forecasting algorithm, five standard evaluation metrics were chosen for this task. We describe these metrics in the following. The *mean absolute error* (MAE) is defined as

$$MAE(y_i, \hat{y}_i) = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i|, \tag{8}$$

where $\mathbf{y_i}$ represents the (real) observed value, $\hat{\mathbf{y}}_\mathbf{i}$ is the predicted value, and n is the target vector length. MAE is the average of the absolute difference between the actual and the predicted values.

The *mean squared error* (MSE) is the average of the square of the difference between the actual and predicted values,

$$MSE(y_i, \hat{y}_i) = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2. \tag{9}$$

It is sensitive to both bias and variance of the residuals, averaging the squared difference between the original and predicted values.

The *mean absolute percentage error* (MAPE) conveys roughly the same information as MAE. However, it makes more clear to compare between models due to the normalization by the $\mathbf{y_i}$ value,

$$MAPE(y_i, \hat{y}_i) = \frac{1}{n}\sum_{i=1}^{n}\left|\frac{y_i - \hat{y}_i}{y_i}\right| \times 100\%. \tag{10}$$

Unfortunately, this formula may lead to numeric problems due to the division by zero situation. One must take actions to prevent this problem, when using MAPE.

The *root mean squared error* (RMSE) measures the standard deviation of residuals with the square root of MSE, defined by

$$RMSE(y_i, \hat{y}_i) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}. \tag{11}$$

The R^2 metric is the coefficient of determination, with the best possible score being 1.0 and it can take negative values, as opposed to all previous metrics mentioned before in this section. It is defined by

$$R^2(y_i, \hat{y}_i) = 1 - \frac{\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}{\sum_{i=1}^{n}(y_i - \bar{y}_i)^2}, \tag{12}$$

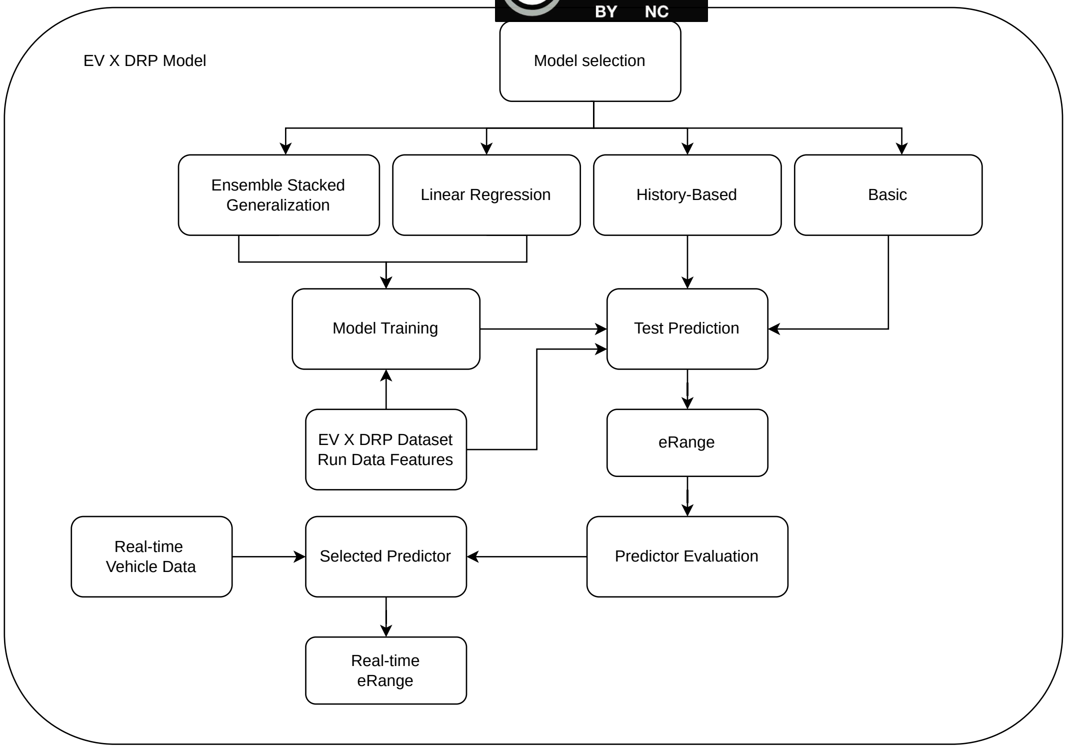

Fig. 8. EV X Driving Range project block diagram with future integration of the Python application.

with $\bar{y}_i$ denoting the average (expected) value of y_i. It represents the proportion of the variance in the dependent variable. As opposed to the other evaluation metrics, for which lower values are perceived as better, in the case of R^2 higher values are desirable. For this measure, lower values may indicate the prevalence of redundant or irrelevant variables.

B. *Experimental results*

The training and execution of the algorithms was done on a computer with the Manjaro Linux operating system, 5.18.19-3-MANJARO kernel, a AMD Ryzen 9 3900X CPU, and 48 Gb of RAM. The Python runtime is version 3.9 using Jetbrain's Pycharm as the *integrated development environment* (IDE).
The application displays different eRange computed results for the selected trip and prediction algorithms, allowing for an easy overview of the different dataset parameters, making the initial input dataset configuration to depend on multiple datasets.

A conventional 47 minutes trip from the VED dataset [33] for a 2013 Nissan Leaf model was run with a minimum trip time of 10 minutes and a minimum timestamp of 0 seconds.

1) Execution trip parameters: Figure 9 shows the trip parameters SoC, Speed, and IEC, respectively. In these graphs, we observe a connection between these indicators (SoC, Speed, and IEC). The SoC is globally a time-decreasing function, as expected. The rate of decrease depends on the speed of the vehicle. IEC is proportional to the speed of the EV and sometimes it takes negative values, which correspond to the regenerative braking instants. Moreover, when speed is zero, the value of IEC is also zero. We notice a clear

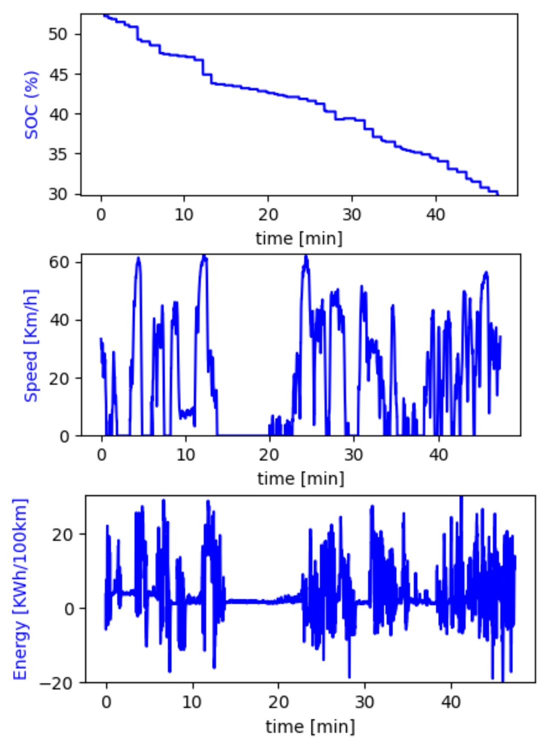

Fig. 9. Execution trip parameters. Evolution of the State of Charge (SoC, in %), speed (in km/h), and Instantaneous Energy Consumption (IEC, in kWh per 100 km), as a function of the trip time.

decrease on the IEC values on the time window from 12 to 20 minutes. After the 20 minutes time point, the EV moves with low speed and the value of IEC increases slowly assuming small positive values.

2) Evaluation with 10 minutes minimum trip time: The comparison of the four eRange prediction models is depicted in Figure 10. The training data is composed by trips that have at least 10 minutes of duration. The trips were filtered by the time it took for each trip to complete, before being selected for training. We report the experimental results of the "history-based" approach (in red), ESG (in green), ESG V2 (in light blue) and the LR approach (in purple). On the "history-based" approach, we use Equation (4) to compute the $AEC_A(k)$; on the first 10 minutes, we have $AEC_A(k) = AEC_C$.

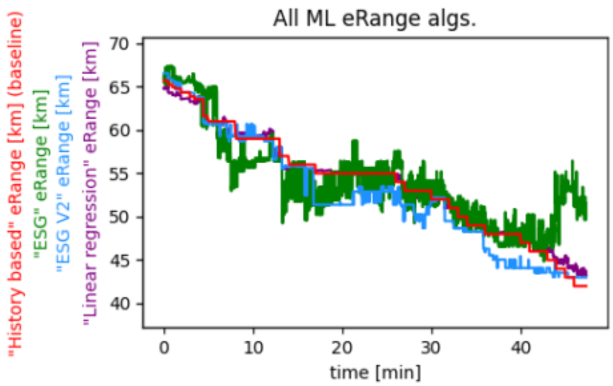

Fig. 10. eRange computation by the four algorithms (minimum 10 min trip time for training).

The "history-based" approach shows an increase in eRange when regenerative breaking is charging the battery and "plateau" sections when the minimum instance energy is not enough to trigger a recalculation for the eRange. These sections have been smoothed by the LR algorithm, which shows a smooth evolution on the forecasted values. The LR algorithm improves on the "history-based" approach that exhibits a "staircase effect", which may cause anxiety on the driver, each time the estimated value drops in a step. On the one hand, LR seems to be the best ML approach for this problem. On the other hand, the ESG algorithm provides a more optimistic estimate, yielding larger eRange values. One possible cause for this performance may be the missing original dataset training features such as elevation, however the R^2 value also indicates poor fitting between the selected dataset features and the prediction value.

Figure 11 shows the behavior of the "basic" and "history-based" approaches, for the eRange estimation. These two approaches provide similar estimates of the eRange values. On the 12 minutes to 20 minutes time window, the "history-based" approach provides a stable and constant estimation. The "basic" counterpart provides a decreasing estimate on this time window.

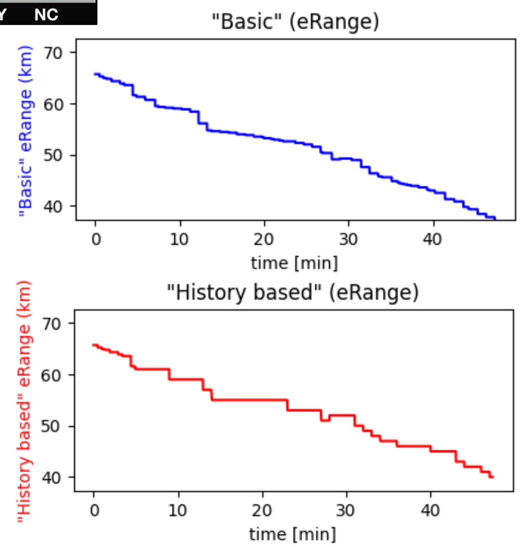

Fig. 11. Basic and History-Based approaches for eRange computation.

We now report on the experimental results of the LR, ESG and ESG V2 approaches in Figure 12. Again, we consider only trips with less than 10 minutes. Both algorithms exhibit the decreasing trend with LR providing a smoother prediction evolution. The ESG method, due to being a stacking technique, provides estimates with larger deviation. These estimates can be smoothed with a moving-average filter [34], [35]. The ESG shows less deviation from the baseline, however it still struggles with the selected testing trip eRange forecasting.

The evaluation metrics, described in Section IV-A, for this experiment with LR and ESG are reported in Table II.

TABLE II
LR AND ESG PERFORMANCE METRICS WITH MINIMUM 10 MINUTES TRIP TIME. THE BEST RESULT IS IN BOLD FACE.

ML approach	MAE	MSE	MAPE	RMSE	R^2
LR	**0.597**	**0.603**	**0.012**	**0.776**	**0.985**
ESG	2.317	12.135	0.047	3.483	0.689
ESG V2	1.519	3.650	0.029	1.910	0.906

3) Evaluation without minimum trip time: We now report the experimental results of the training with trips without minimum duration. Figure 13 shows the experimental results of the four algorithms herein considered. Figure 14 depicts the individual results for the LR, ESG, and ESG V2 algorithms, respectively.

The experimental results of the LR algorithm are a close approximation to the target ("history-based" approach), depicted in Figure 11. These results are in accordance with the evolution of the IEC value, as reported in Figure 9. The evaluation metrics for this experiment with LR and ESG are reported in Table III.

These experimental results show that eRange computation can be achieved with ML techniques, overcoming the existing "basic" and "history-based" approaches. By comparing the results in Table II and Table III, we conclude also that

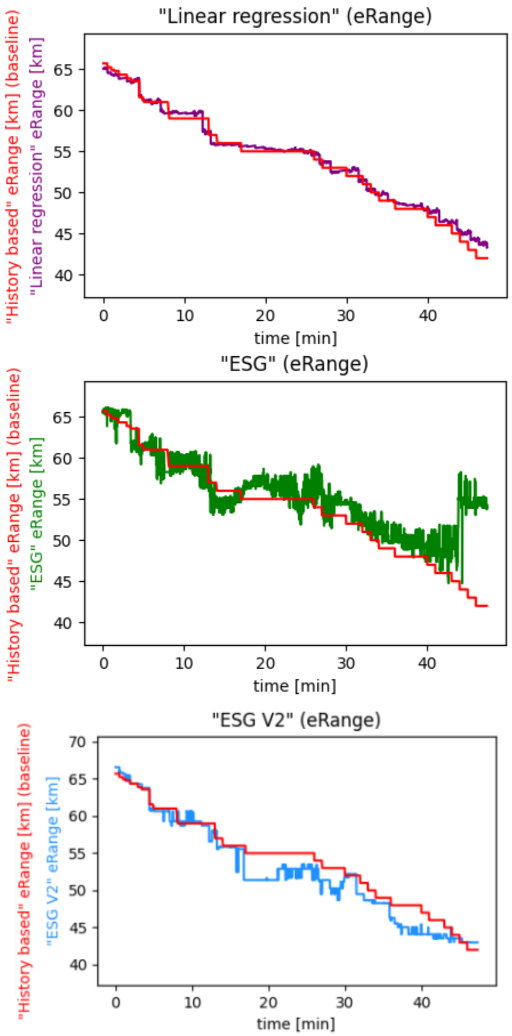

Fig. 12. LR, ESG, and ESG V2 approaches for eRange computation (minimum 10 min trip time for training).

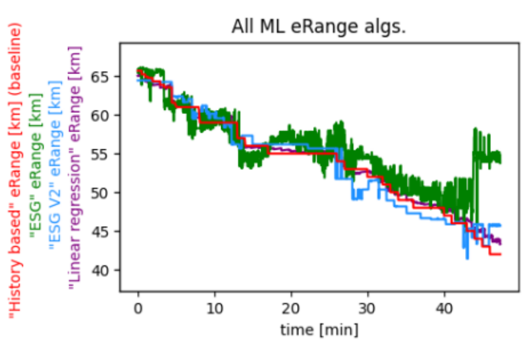

Fig. 13. eRange computation by the four algorithms (no minimum trip time for training).

the training with trips with no minimum duration improve on the forecasting, since we attain better results in this case, however training time on ESG models differ extensively, as reported in Tables V and VI.

When testing for other trips through the 20-Fold Cross Validation method, we can observe that ESG and ESG V2

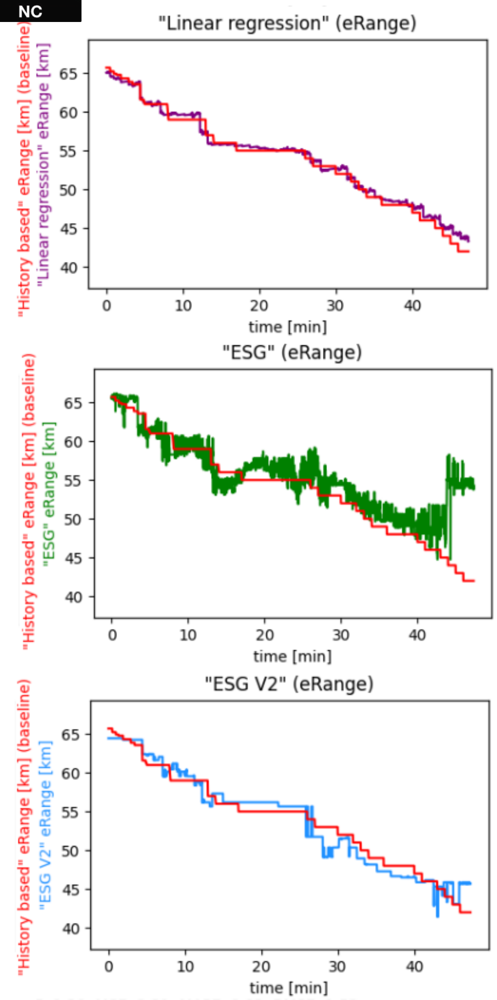

Fig. 14. LR, ESG, and ESG V2 approaches for eRange computation (no minimum trip time for training).

TABLE III

LR AND ESG PERFORMANCE METRICS WITH NO MINIMUM TRIP TIME. THE BEST RESULT IS IN BOLD FACE.

ML approach	MAE	MSE	MAPE	RMSE	R^2
LR	**0.696**	**1.077**	**0.483**	**0.984**	**0.997**
ESG	1.687	5.833	1.319	2.288	0.987
ESG V2	1.339	3.343	1.051	1.778	0.991

improve significantly their metrics, as depicted on Table IV. This highlights that ensemble methods that resort to DT significantly benefit from training and testing data similarities, as testing with smaller trips provides better results due to the higher count of smaller trips found in the dataset.

4) Running time analysis: We have performed a running time analysis of the training of the ML approaches. We have provided different versions of the training data with different minimum trip times, and then we train the ESG, ESG V2, and LR approaches. Table V reports the training time for every train trip that was used for testing the selected trip, while Table VI reports on the training for the 20-Fold Cross Validation.

TABLE IV

LR AND ESG CROSS VALIDATION METRICS FOR ALL MINIMUM TRIP TIMES (MTT). THE BEST RESULT IS IN BOLD FACE.

ML approach	MTT	MAE	MSE	MAPE	RMSE	R^2
LR	0m	**0.539**	**0.725**	**0.341**	**0.807**	**0.998**
ESG	0m	1.479	4.724	1.156	2.106	0.991
ESG V2	0m	1.279	2.996	1.106	1.677	0.994
LR	10m	**0.696**	**1.077**	**0.483**	**0.984**	**0.997**
ESG	10m	1.687	5.833	1.319	2.288	0.987
ESG V2	10m	1.339	3.343	1.051	1.778	0.991

TABLE V

THE EFFECT OF THE MTT CONSTRAINT ON ML TRAINING TIME WITH THE DATASET AND TESTING TIME WITH SELECTED TRIP FOR ESG ORIGINAL PAPER CONFIGURATION [40]; ESG V2 ADJUSTED FOR DATASET CONFIGURATION AND LR (PURPLE LINE).

ML	MTT	Trip count	Train time	Avg test time	All test time
LR	0m	503	138ms	0ms	4,223ms
ESG	0m	503	7m 25s 772ms	9ms	45s 020ms
ESG V2	0m	503	14m 10s 940ms	9ms	45s 300ms
LR	10m	159	72ms	0ms	4s 129ms
ESG	10m	159	4m 2s 146ms	9ms	44s 150ms
ESG V2	10m	159	7m 37s 998ms	9ms	44s 308ms

For both techniques, we observe as expected, the decrease in the training time as the minimum trip duration increases. The key reason is that the number of trips available for training decreases, as the minimum duration increases. The LR approach presents very fast training and achieves adequate results with a smooth varying curve on the prediction values. Although the ESG V2 technique performs relatively better than ESG, the approach needs to be further explored and its parameters need to be fine-tuned.

V. CONCLUSIONS

The electric vehicle remaining driving distance estimation is a relevant problem, since this estimate relieves the driver anxiety on a trip and allows for a better trip planning. There are some useful statistical approaches to perform this estimation. However, these techniques provide an estimate with some degree of error. The use of machine learning techniques to provide this estimation has been proven adequate, despite the fact that this a recent field of study. There are some public domain datasets with electric vehicle data, but their use is not straightforward, requiring a demanding construction and pre-processing stage to have a reliable dataset with accurate and complete trip data.

In this paper, we have composed such a dataset in which we have assessed the use of regression techniques to estimate the remaining driving range distance, based on different variables with vehicle data and trip data. We have compared the prediction accuracy of these techniques with standard metrics and found that linear regression shows promising prediction results as well as fast training. The ensemble stacked generalization V2 algorithm has shown better predictions than ensemble stacked generalization, but still both would benefit from the existence of long trip representation on the dataset. The performance of these algorithms seem to be very dependent on the dataset. The experimental results

TABLE VI

THE EFFECT OF THE MTT CONSTRAINT ON ML TRAINING WITH 20-FOLD CROSS VALIDATION FOR ESG ORIGINAL PAPER CONFIGURATION [40]; ESG V2 ADJUSTED FOR DATASET CONFIGURATION AND LR (PURPLE LINE).

ML	MTT	Trip count	All folds train time	Avg fold train time
LR	0m	503	30s 064ms	1s 503ms
ESG	0m	503	4h 3m 54s 919ms	12m 11s 745ms
ESG V2	0m	503	6h 50m 52s 398ms	20m 32s 619ms
LR	10m	159	14s 543ms	727ms
ESG	10m	159	1h 57m 29s 701ms	5m 52s 485ms
ESG V2	10m	159	3h 22m 59s 269ms	10m 08s 963ms

have also shown the impact of different datasets and training configurations, on existing machine learning models.

As future work, we plan to perform the integration of the developed application with the real-time data of the electric vehicle to continuously provide updated eRange estimations. We also plan to include more datasets and features, such as driving pattern, road elevation and traffic data. Moreover, additional machine learning techniques can be added to the established open source experimental setting.

REFERENCES

[1] Chargecar database. `http://www.chargecar.org/data`. Accessed: 2023-06-14.

[2] Chargecar's create rav4-recorder box. `http://chargecar.org/participate/lending`. Accessed: 2023-06-14.

[3] Electric vehicle database. `https://ev-database.org/car/1011/Nissan-Leaf`. Accessed: 2023-06-14.

[4] Electric vehicle x driving range prediction github repository. `https://github.com/davidalb97/TFM18-2122i`. Accessed: 2022-12-17.

[5] Evteclab's eva platform. `http://proeftuin-ev.be/content/eva-platform`. Accessed: 2023-06-14.

[6] JARI research database. `https://www.jari.or.jp/research-database/`. Accessed: 2023-06-14.

[7] Machine learning-based method for remaining range prediction of electric vehicles - source. `https://github.com/liangzhao123/range_prediction`. Accessed: 2023-06-14.

[8] National big data alliance of new energy vehicles. `http://www.ndanev.com`. Accessed: 2023-06-14.

[9] Paris Agreement, UN Treaty No. XXVII-7-d. 12 2015.

[10] D. Aha, D. Kibler, and M. Albert. Instance-based learning algorithms. 6:37–66, 1991.

[11] E. Alpaydin. *Introduction to machine learning*. The MIT Press, fourth edition, 2020.

[12] S. Amershi, A. Begel, C. Bird, R. DeLine, H. Gall, E. Kamar, N. Nagappan, B. Nushi, and T. Zimmermann. Software engineering for machine learning: A case study. In *IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 291–300, 2019.

[13] H. Bock. Clustering Methods: A History of k-Means Algorithms. In *Selected Contributions in Data Analysis and Classification*, Studies in Classification, Data Analysis, and Knowledge Organization, chapter 15, pages 161–172. Springer Berlin Heidelberg, 2007.

[14] I. Bose and R. Mahapatra. Business data mining - a machine learning perspective. *Information & Management*, 39(3):211–225, 2001.

[15] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, 2001.

[16] A. Brighente, M. Conti, D. Donadel, and F. Turrin. Evscout2.0: Electric vehicle profiling through charging profile. *CoRR*, abs/2106.16016, 2021.

[17] T. Condie, P. Mineiro, N. Polyzotis, and M. Weimer. Machine learning on big data. In *IEEE 29th International Conference on Data Engineering (ICDE)*, pages 1242–1244, 2013.

[18] D. Coutinho. Classic eMini project: Electrification of a classic mini, technical report. Draft version, July 2021.

[19] D. Coutinho. Classic EV X project driving range prediction. Technical report, July 2021. Draft version.

[20] C. De Cauwer, W. Verbeke, T. Coosemans, S. Faid, and J. Van Mierlo. A data-driven method for energy consumption prediction and energy-efficient routing of electric vehicles in real-world conditions. *Energies*, 10(5), 2017.

[21] O. Egbue and S. Long. Barriers to widespread adoption of electric vehicles: An analysis of consumer attitudes and perceptions. *Energy Policy*, 48:717–729, 2012. Special Section: Frontiers of Sustainability.

[22] A. Enthaler and F. Gauterin. Method for reducing uncertainties of predictive range estimation algorithms in electric vehicles. In *2015 IEEE 82nd Vehicular Technology Conference (VTC2015-Fall)*, pages 1–5, 2015.

[23] E. Figenbaum, N. Fearnley, P. Pfaffenbichler, R. Hjorthol, M. Kolbenstvedt, R. Jellinek, B. Emmerling, G. Bonnema, F. Ramjerdi, L. Vågane, and L. Iversen. Increasing the competitiveness of e-vehicles in Europe. *European Transport Research Review*, 7(3):28, Aug 2015.

[24] C. Gaete-Morales, H. Kramer, W.-P. Schill, and A. Zerrahn. An open tool for creating battery-electric vehicle time series from empirical data, emobpy. *Scientific Data*, 8(1):152, Jun 2021.

[25] T. Hastie, R. Tibshirani, and J. Friedman. *The elements of statistical learning*. Springer, 2nd edition, 2009.

[26] S. Haykin. *Neural networks: a comprehensive foundation*. Upper Saddle River, N.J.: Prentice Hall, 2nd edition edition, 1999.

[27] T. Kohonen. *Self-organizing maps*. Springer series in information sciences, 30. Springer, Berlin, 3rd edition, Dec. 2001.

[28] C.-H. Lee and C.-H. Wu. A novel big data modeling method for improving driving range estimation of EVs. *IEEE Access*, 3:1980–1993, 2015.

[29] K. Liu, J. Wang, T. Yamamoto, and T. Morikawa. Exploring the interactive effects of ambient temperature and vehicle auxiliary loads on electric vehicle energy consumption. *Applied Energy*, 227:324–331, 2018. Transformative Innovations for a Sustainable Future Part III.

[30] K. Liu, T. Yamamoto, and T. Morikawa. Impact of road gradient on energy consumption of electric vehicles. *Transportation Research Part D: Transport and Environment*, 54:74–81, 2017.

[31] T. Mitchell. *Machine learning*. McGraw Hill, 1997.

[32] T. Mitchell. The discipline of machine learning. Technical Report CMU ML-06 108, 2006.

[33] G. Oh, D. Leblanc, and H. Peng. Vehicle energy dataset (VED), a large-scale dataset for vehicle energy consumption research, 2019.

[34] A. Oppenheim and R. Schafer. *Discrete-Time Signal Processing*. Pearson, third edition, 2010.

[35] S. W. Smith. *The Scientist and Engineer's Guide to Digital Signal Processing*. California Technical Publishing, 1997.

[36] M. Smuts, B. Scholtz, and J. Wesson. A critical review of factors influencing the remaining driving range of electric vehicles. In *1st International Conference on Next Generation Computing Applications (NextComp)*, pages 196–201, 2017.

[37] Y. Song and X. Hu. Learning electric vehicle driver range anxiety with an initial state of charge-oriented gradient boosting approach. *Journal of Intelligent Transportation Systems*, 0(0):1–19, 2021.

[38] X.-H. Sun, T. Yamamoto, and T. Morikawa. Stochastic frontier analysis of excess access to mid-trip battery electric vehicle fast charging. *Transportation Research Part D: Transport and Environment*, 34:83–94, 2015.

[39] X.-H. Sun, T. Yamamoto, and T. Morikawa. Fast-charging station choice behavior among battery electric vehicle users. *Transportation Research Part D: Transport and Environment*, 46:26–39, 2016.

[40] I. Ullah, K. Liu, T. Yamamoto, M. Zahid, and A. Jamal. Electric vehicle energy consumption prediction using stacked generalization: an ensemble learning approach. *International Journal of Green Energy*, 18(9):896–909, 2021.

[41] B. Varga, A. Sagoian, and F. Mariasiu. Prediction of electric vehicle range: A comprehensive review of current issues and challenges. *Energies*, 12(5), 2019.

[42] D. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.

[43] M. Yilmaz and P. Krein. Review of battery charger topologies, charging power levels, and infrastructure for plug-in electric and hybrid vehicles. *IEEE Transactions on Power Electronics*, 28(5):2151–2169, 2013.

[44] M. Yoong, Y. Gan, G. Gan, C. Leong, Z. Phuan, B. Cheah, and K. Chew. Studies of regenerative braking in electric vehicle. In *2010 IEEE Conference on Sustainable Utilization and Development in Engineering and Technology*, pages 40–45, 2010.

[45] Y. Zhang, W. Wang, Y. Kobayashi, and K. Shirai. Remaining driving range estimation of electric vehicle. In *IEEE International Electric Vehicle Conference*, pages 1–7, 2012.

[46] L. Zhao, W. Yao, Y. Wang, and J. Hu. Machine learning-based method for remaining range prediction of electric vehicles. *IEEE Access*, 8:212423–212441, 2020.

[47] B. Zheng, P. He, L. Zhao, and H. Li. A hybrid machine learning model for range estimation of electric vehicles. In *IEEE Global Communications Conference (GLOBECOM)*, pages 1–6, 2016.

[48] L. Zhou, S. Pan, J. Wang, and A. Vasilakos. Machine learning on big data: Opportunities and challenges. *Neurocomputing*, 237:350–361, 2017.