Taylor & Francis
Taylor & Francis Group

Applied Artificial Intelligence
An International Journal

ISSN: (Print) (Online) Journal homepage: www.tandfonline.com/journals/uaai20

Few-Shot Learning for Triplet-Based EV Energy Consumption Estimation

Alminas Čivilis, Linas Petkevičius, Simonas Šaltenis, Kristian Torp & Ieva Markucevičiūtė-Vinckė

To cite this article: Alminas Čivilis, Linas Petkevičius, Simonas Šaltenis, Kristian Torp & Ieva Markucevičiūtė-Vinckė (2025) Few-Shot Learning for Triplet-Based EV Energy Consumption Estimation, Applied Artificial Intelligence, 39:1, 2474785, DOI: 10.1080/08839514.2025.2474785

To link to this article: https://doi.org/10.1080/08839514.2025.2474785

APPLIED ARTIFICIAL INTELLIGENCE
2025, VOL. 39, NO. 1, e2474785 (20 pages)
https://doi.org/10.1080/08839514.2025.2474785

Few-Shot Learning for Triplet-Based EV Energy Consumption Estimation

Alminas Čivilis[a], Linas Petkevičius[a], Simonas Šaltenis[a], Kristian Torp[b], and Ieva Markucevičiūtė-Vinckė[c]

[a]Institute of Computer Science, Vilnius University, Vilnius, Lithuania; [b]Department of Computer Science, Aalborg University, Aalborg, Denmark; [c]Normalis Tech, Vilnius, Lithuania

ABSTRACT
Predicting the energy consumption of an electric vehicle (EV) is often relevant when planning and managing electric mobility. The prediction is challenging as EV energy consumption is highly variable and dependent on *context*. First, this paper proposes an integrated framework for the collection of online telematic data, processing of this data, online maintenance of statistics, and machine-learning-based prediction of travel time and energy consumption. A key feature of the proposed framework is the preprocessing of the trajectory data into *triplets*, a convenient data unit that *captures the relevant context* necessary for effective energ y prediction. The second contribution of the paper addresses the effective *management of drastic change in context* through robust energy prediction models. In particular, using *few-shot* learning techniques, we tackle the problem of the need to create different energy prediction models for different EV types, from small EVs to electric buses. Experimental results on three different data sets demonstrate how energy prediction models adapt to different EV types.

ARTICLE HISTORY
Received 3 January 2024
Revised 10 February 2025
Accepted 18 February 2025

Introduction

The electrification and digitalization of transportation mark a transformative era. For example, the European Union is planning a ban on the sale of vehicles with internal combustion engines (ICEs) from 2035, Erbach (2023). This significant shift toward electric vehicles (EVs) raises new challenges. One such challenge is the prediction of the energy consumption of EV trips. As we argue below, except for short commute trips, effective use of electric vehicles requires an accurate prediction of energy consumption.

Consider long-distance trips that involve planning charging stops and predicting the required charging times (Popiolek et al. 2023; Šaltenis et al. 2023; Shi, Zeng, and Moura 2024; Subramanian et al. 2022). Without an

CONTACT Linas Petkevičius linas.petkevicius@mif.vu.lt Institute of Computer Science, Vilnius University, Didlaukio g. 47, Vilnius, 08303 Vilniaus m. sav., Lithuania

Supplemental data for this article can be accessed online at https://doi.org/10.1080/08839514.2025.2474785 accurate prediction of energy consumption, conservative, worst-case energy consumption would have to be assumed limiting the set of reachable charging stations when planning each leg of a trip.

The prediction challenge is amplified by the fact that the energy consumption of EVs varies a lot depending on contextual parameters, such as weather conditions, traffic, and road geometry. (Krogh, Andersen, and Torp 2015). Note that predicting EV energy consumption is more difficult than predicting the fuel consumption of ICE vehicles, which is much less variable, as the majority of fuel energy is expended as a mostly constant thermal loss. In contrast, electric vehicles are much more efficient in energy use, making the effect of environmental conditions more pronounced.

We consider the problem of predicting the EV energy consumption on a planned route in a road network. Ideally, the best energy consumption prediction would be achieved utilizing historical driving data collected on the same route under similar driving conditions. However, collecting such a large amount of data is extremely time-consuming and almost impossible due to the exponential number of possible routes in the road network. Therefore, statistics are typically calculated segment-wise. Each route is modeled as a sum of energy predictions for a sequence of road segments, making it easier to collect statistical data. However, this method loses the context of the trip as it is split into individual segments. Instead, *to capture the context*, we propose to split the trips into *overlapping segment triplets*, so that, for segment x in a trip, the attributes of the preceding segment and the attributes of the next segment are considered as the contextual attributes of the segment x. We do this for both statistics collection and model building, as well as when predicting the energy consumption on a given route.

Existing work on predicting EV energy consumption is scarce. We briefly review the most relevant research that employs machine learning methods. Recent studies use transformer neural networks (Shen et al. 2022), recurrent neural networks (Hua et al. 2022), probabilistic deep learning (Petkevičius et al. 2021), and classical machine learning methods like gradient-boosted regression tree and extreme gradient boosting (Chen, Lei, and Ukkusuri 2022; Roy et al. 2022), linear regression, random forest, and neural networks (Madziel 2024). However, these studies are mostly limited to experiments with a single EV type. In practice, there are many different EV types, from small EVs to large electric buses. This raises the question of whether it is possible to create an energy prediction model that could be effectively adapted when the context drastically changes with the introduction of new EV types.

The main problem of cross-EV-type energy prediction is the lack of data. Data collected from different EV types usually have different attributes, data is collected in various locations and under differing conditions. For the methods to work in real life, they must be robust in terms of varying parameters. To address this problem, we investigate the use of the *few-shot* learning approach.

Few-shot learning methods became very popular recently due to their ability to reuse existing knowledge for new tasks with little data (Wang et al. 2020). Such models were successfully applied for building energy predictions (Tang et al. 2023), wind power predictions (Meng et al. 2022), battery state-of-charge predictions (Zhang, Liu, and Su 2023), battery lifespan prediction (Meng et al. 2024), and forecasting of energy in smart grids (Xu, Li, and Li 2024). A recent study surveyed various approaches for few-shot learning from a supervised learning perspective using different loss functions (Li et al. 2023).

In this paper, we propose a few-shot learning approach for EV energy consumption prediction using triplet data, the transformed embeddings using the cumulative distribution of known domain features, and parameter-less Nadaraya-Watson regression (Nadaraya 1964). We demonstrate that the proposed approach is feasible to adapt to different EV types and does not require retraining of the model on new tasks.

In summary, the main contributions of our work are

- the triplet-based trajectory data management strategy to capture context and significantly improve task learnability as well as to decrease the complexity of the task;
- an architecture for a framework that integrates online telematic data processing, maintenance of statistics, machine learning, and prediction of travel time and energy use;
- novel few-shot learning approach using the cumulative distribution function for feature transformation using embedding learning and parameter-less Nadaraya-Watson regression for new task learning when new types of EVs are introduced.

First, we present the motivation for triplets and the general architecture of the envisioned energy-prediction system in Section 2. Next, the proposed machine learning models are presented in Section 3. Section 4 presents the results of the extensive experimental study. Finally, Section 5 concludes the paper and points to possible future work.

Problem setting and architecture

As mentioned in the introduction, the goal of this work is to predict the EV energy consumption on a planned route in a road network. We assume that this prediction is done based on the prediction models built and maintained using a continuously replenished data warehouse of telematic traces – recordings of vehicle trips. First, we present the idea of splitting such traces into triplets. Then, we discuss the architecture for collecting the data, building prediction models on it, and using the models for prediction.

Triplet formulation

We assume the road network is broken into road segments, where a piece of road between two intersections is considered a segment. Further, long segments can be broken into shorter ones if needed. Then, we model a planned route as a sequence of road-segment traversals.

To estimate the EV energy consumption along a planned route, physical properties of the vehicle, road conditions such as surface and shape, and weather conditions (temperature, wind direction) should be used. The most significant parameters that influence energy consumption, independent of the vehicle model, are vehicle speed and acceleration, road shape (uphill, downhill, curvature), and road type (highway vs. service road, etc.). While road shape and road type are fixed for each segment, car speed and acceleration depend on the driver, time-varying traffic conditions, and the route being driven. The route the driver is going on affects how each segment is driven, as drivers might have different plans at the end of a road segment. For example, in Figure 1, a driver on a highway (segments a, b, c) might continue straight on the next segment (take path a to b) or exit the highway (take path a to $a1$), resulting in different speed profiles on segment a and, thus, different energy consumptions for the same segment. Similarly, how segment c is traversed depends on whether the driver has just merged from segment $b1$ or is simply continuing from segment b.

Previous work on EV energy prediction (Petkevičius et al. 2021) demonstrates the benefits of a two-step process. First, the speed on each segment of the route is predicted based on time-dependent historical speed statistics collected on the segment. Then, this speed prediction is used as one of the covariates in the machine-learning model for energy prediction.

We argue that capturing the next and previous segments on a route is important both for a more refined collection of historical speed statistics and as additional context when inferring energy consumption along the route. To address this, for each *segment* on a route, we define a *triplet* which is a sequence of three consecutive road segments: *pre-segment*, *segment*, and *post-segment*. We split all recorded trips into overlapping triplets and aggregate speed statistics triplet-wise. Then, a planned route is also transformed into a sequence of overlapping triplets, where we, first, estimate the average speed on the middle segment of each triplet and, then, perform machine learning inference using the attributes of the surrounding segments as additional covariates.

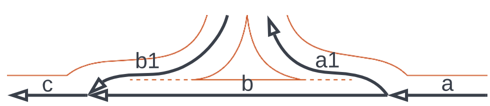


Figure 1. An example of a triplet. The triple consists of (a, b, c) segments. The cases (a, a1), (a, b), (b, c), and (b1, c) are the pairs or duplets.

Sometimes, when planning a route, we might not have historical data statistics for a particular triplet. However, we might have data for a pair of adjacent segments or only for a single segment. We term pairs of segments *duplets*. A triplet can then be constructed from two duplets: the *pre-duplet*, consisting of the pre-segment and the segment, and the *post-duplet*, consisting of the segment and the post-segment. When predicting speed using duplet-level statistics, two different speed values are available for a given segment: one from a pre-duplet and another from a post-duplet. Different strategies can be employed in such a scenario, such as calculating the average speed of the two duplet speeds, taking the minimum or maximum speed, or using single-segment data. If there is no statistical data for a given segment at all, the segment speed can be estimated to be equal to the average speed on this type of road or the speed limit on the particular segment.

Framework architecture

In the following, we describe the envisioned architecture of a framework that integrates online telematic data processing, maintenance of statistics, machine learning, and prediction of travel time and energy use.

Figure 2 shows the architecture that is divided into two components: *model building*, shown in the upper part of the figure, and *online prediction*, shown in the lower part of the figure.

Model building

The model-building component of the framework is responsible for processing the incoming telematic data from the vehicles that are being tracked by the system.

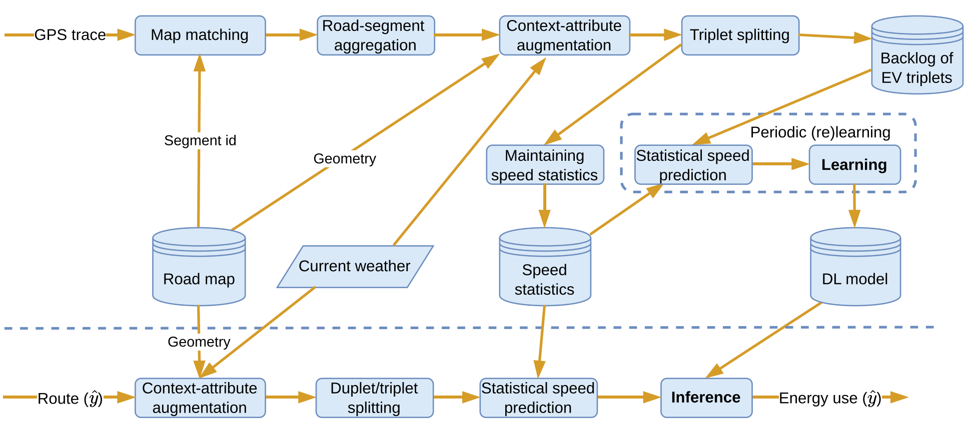


Figure 2. Information flow in the framework. The flowchart visualizes the data aggregation to fixed-dimension information, which could be used for statistical predictions.

We assume that each tracked vehicle periodically uploads its trips as *GPS traces*. Each GPS trace is a sequence of *GPS elements*. What kind of information is included in a GPS element may vary, but we require that it includes a GPS position, a timestamp, and, if this is an EV, energy used since the last GPS element:

$$GPSTrace = (vid, \langle \ldots, (pos_i, t_i, [\Delta e_i]), \ldots \rangle) \tag{1}$$

Here, *vid* is a vehicle identifier. The received GPS traces are first map-matched, assigning each GPS element a map-segment id (*sid*). Next, the elements are aggregated, so that for each traversal of a segment, all corresponding GPS elements are replaced with computed traversal time (*stt*) and energy used (*se*):

$$Segment\ Trace = (vid, \langle \ldots, (sid_i, stt_i, se_i), \ldots \rangle) \tag{2}$$

The aggregated segment traversals are then enriched with context information. In particular, a digital road map is used for segment road type (*srt*), segment length (*slen*), average heading of the segment (*shead*), elevation at segment start (*sels*), and at segment end (*sele*). The heading is defined as an angle relative to the true north. Depending on the average length of the segments, more detailed elevation information may be necessary for each segment or, segments may need to be split.

While the digital road map provides static context information, current weather information is a dynamic part of the context. It includes temperature (*wt*), wind direction (*wwd*), and wind speed (*wws*).

As detailed in Section 2.1, an important part of the segment-traversal context is the triplet-related characteristics. In the triplet-splitting step, each segment traversal is augmented with such information so that it becomes a *triplet traversal*.

Such triplet traversals are used to maintain traffic speed statistics, the mean, and the standard deviation, at different aggregation levels. As Figure 3 shows, we propose to aggregate these speed statistics along two main dimensions: spatial and temporal. Considering the spatial dimension, a given triplet traversal updates the speed statistics for that specific triplet as well as for the corresponding pre-duplet, post-duplet, and center segments of the triplet. In the time dimension, we propose the aggregation hierarchy of three layers. At the most detailed level, a day is broken into three time-of-day periods (rush hour (7:00–9:00 and 15:00–17:00), night (22:00–6:00), and the rest of the day). Next, days are categorized either as weekdays or weekends. Finally, if there is not enough historical data for a given segment on the lower levels of the time dimension, global statistics are used. Note that we envision that the above, traffic-related, statistical data is collected for all types of vehicles. If the triplet traversal was from an EV, it is

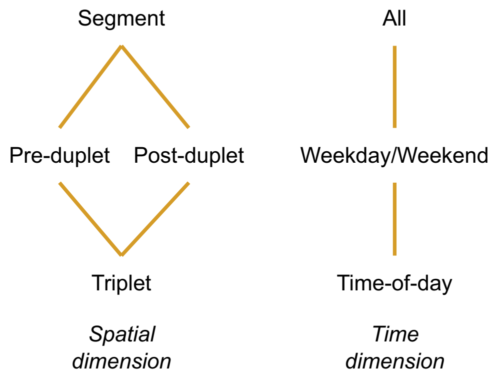


Figure 3. Aggregation hierarchies for speed statistics. The spatial aggregation via duplet presented on the left; the temporal aggregation of day type is presented on the right.

saved in the backlog of EV triplets – an accumulation of model training data. Periodically, this backlog is used to retrain the deep learning model.

Online prediction

The bottom of Figure 2 shows the workflow for the online prediction of energy consumption on a given route. As explained in Section 2.1, after the route segments are augmented with contextual attributes and the route is split into a sequence of overlapping triplets, the online prediction is a two-step process. First, an approximate speed estimate for each segment is derived as the mean of the collected historical speed values at the lowest levels of the aggregation hierarchies where historical data are available for this segment (see Figure 3).

Then, the machine learning model is fed this speed estimate as one of the covariates of the segment together with its other attributes. In addition, the attributes of its pre-segment and post-segment are used as covariates as well as triplet-specific covariates: the angle between the pre-segment and the segment, the angle between the segment and the post-segment, and the cardinalities of the pre-segment/segment and segment/post-segment intersections.

Machine learning models

Using the aggregated data and statistics maintained as described in the previous section, we proceed to develop machine learning models that are trained on that data and are used for energy prediction. First, we discuss different machine-learning problem formulations, and then we detail the workings of the proposed model.

Regression problem

We denote X as the input space and E as the target (consumed energy) space. Here, $x_{j,i} \in X \subset \mathbb{R}^{\frown}$ is an observation of trip j at segment i, and $e_{j,i} \in E \subset \mathbb{R}$ is the consumed/recuperated energy. Here, d_x is the number of all the segment- and triplet-related characteristics mentioned in Section 2. A simple regression model predicts energy consumption $\hat{e}_{j,i}$ for each segment $s_{r_j,i}$ independently: $\hat{e}_j$. The independent predictions can then be summed up to get the cumulative prediction of energy use for the whole route. The unknown parameters of a regression-like models are estimated by minimizing the mean square error loss (MSE):

$$\mathcal{L}_{Reg} = \frac{1}{M}\sum_{j=1}^{M}(e_j - \hat{e}_j)^2, \tag{3}$$

where M is the total number of trip segments in the training dataset.

In our experiment setup, we will use the series of parametric mathematical models $f_\theta(x)$. All used different deep neural networks (DNN) with different architectures are described in Supplementary file.

Deep probabilistic problem

An alternative way to look at the problem is to interpret the energy consumption (target) data as independent random variables $e_1, \ldots, e_M$. Denote by $F_i(e|\theta)$ the cumulative distribution function (cdf.) of e_i. Let

$$\mathcal{F}_0 = \{F(e|\theta), \theta \in \Theta \subset \mathbb{R}^2\}$$

be a parametric family of absolutely continuous cumulative distribution functions with continuous unimodal densities f.

Let us assume that distribution parameters $\hat{\theta} = (\theta_1, \theta_2)$ depend on explanatory variables x: $\hat{\theta} = g(x|\boldsymbol{\phi})$. We choose the operator $g(x|\boldsymbol{\phi})$ to be a deep neural network with unknown parameters $\boldsymbol{\phi}$. The model is trained on the training data $\{x_i, e_i\}_{i=1,..,N}$ and the output of the model is $\hat{\theta}$. Different from regression-like problem formulation, the e_i is just a realization of $F_i(y|\theta)$. Thus, a deep neural network is trained to predict the parameters of the distribution, i.e., the mean and deviation of energy consumption for a given segment:

$$\theta = g(x|\boldsymbol{\phi}) \tag{4}$$

$$s.t. \quad e \sim F(e|\theta)$$

Few-shot learning

Few-shot learning aims to address the problem of learning with only a few supervised data point instances. Formally, the training dataset $\mathcal{D} = \mathcal{D}_{Prior} \cup \mathcal{D}_{Adapted}$ is separated into two datasets $\mathcal{D}_{Prior}$ and $\mathcal{D}_{Adapted}$ containing different sets from different data domains, in our case, the domains of EV cars and buses. Each tuple $(X_i, \hat{y}_i) \in \mathcal{D}$ consists of data $X_i = \{x_{1,i}, \ldots, x_{M,i}\}$ containing M covariates and their corresponding labels $\hat{y}$ of energy consumption. The small dataset $\mathcal{D}_{Adapted}$ contains only a small number of supervised data pairs in contrast to $\mathcal{D}_{Prior}$. For the task of K-shot learning, there are exactly K annotated instances available in $\mathcal{D}_{Adapted}$. Training a regression model only on $\mathcal{D}_{Adapted}$ quickly leads to overfitting and poor generalization due to limited training (Chen et al. 2018; Yan et al. 2019).

The typical approaches to few-shot learning are a) multi-task learning and b) embedding learning. The multi-task learning approach often suggests keeping the main tasks and using parameter sharing between existing and new tasks. In Zhang, Tang, and Jia (2018); Wang et al. (2020), the initial two task networks share the first few layers for the generic features and learn different final branches to deal with new tasks. In addition, there is the parameter-tying approach which regularizes the parameters, forcing the parameters of different tasks to be similar (Goodfellow, Bengio, and Courville 2016). The limitation of multi-task learning is the necessity to keep and process all of the original tasks and data for complement tasks. On the other hand, the embedding learning allows not to track all of the information about the previous tasks and reuse the high-level features for new tasks. Feature vectors could be calculated and saved separately, this saves computational costs.

The most common approach is task-invariant embedding when model $f_{\theta,fixed}(x)$ is used as a feature extractor. The model is parameterised with neural network parameters θ and fixed means. The model was trained on another task, thus it is used as a pre-trained model (Goodfellow, Bengio, and Courville 2016). Then, a new task is learned on top of the extracted features from model $f_{\theta,fixed}(x)$. At inference, the high-level embeddings are compared with the embedding of few-shot examples using some distance, e.g., cosine distance. Such an approach is used in (Snell, Swersky, and Zemel 2017; Tsimpoukelli et al. 2021; Vinyals et al. 2016). It has the benefits of a) the ability to work with the informative features learned before and b) the reduced number of parameters in use, since pre-trained model weights are fixed. Thus, we will use this embedding approach.

Proposed approach

We propose to use the statistical nearest neighbors method to leverage the high-level embedding distribution to encode second-order information from features cdf instead of using raw high-level feature values (See Figure 4). The changes in transformed embeddings aim to achieve the following goals:

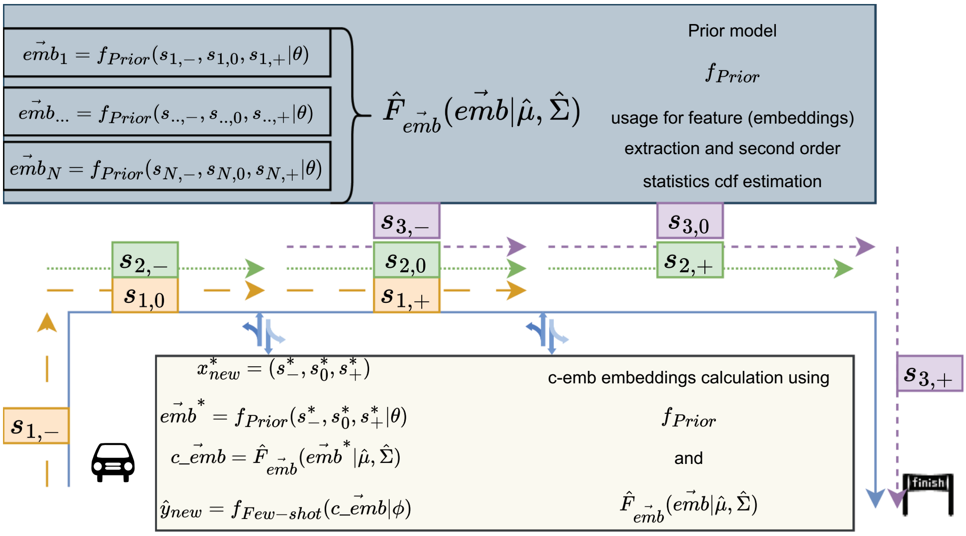


Figure 4. The cdf-embedding process. The trip is split into triplet segments s_-, s_0, s_+. The prior model f_{Prior} is used as a feature extractor. The $\hat{F}_{\vec{emb}}(\vec{emb}|\hat{\mu}, \hat{\Sigma})$ is estimated on the prior data $\mathcal{D}_{Prior}$ and is used to transform embeddings to the cdf-embeddings. During inference, the cdf-embeddings are calculated which represents second-order information of the data. The cdf-embeddings are then used to calculate the new task.

- to have normalized high-level features by applying the cdf transformation.
- to have a reduced number of samples for domain transfer by using the nearest neighbors technique.

Definition 3.1 (embeddings) Let $f_{fixed}(x|\theta) = f_{Prior}(x|\theta)$: $\boldsymbol{X} \rightarrow Y$ to be any pre-trained (Prior) deep neural network with fixed parameters θ. Deep neural networks consist of multiple nonlinear input transformations which transform the original input x into a new representation E which is informative for the final prediction:

$$Y = f(x|\theta) = f_L(f_{L-1}(\ldots f_2(f_1(x|\theta_1)|\theta_2)|\theta_{L-1})|\theta_L)$$

$$= [E_1 = f_1(x|\theta_1), E_2 = f_2(E_1|\theta_2), \ldots, E_L = f_L(E_{L-1}|\theta_L)]$$

where L is the number of layers (the transformations), and $\theta \subset \Theta^{d_\Theta}$ - the neural network parameters. The high-level representations or extracted features also known as embeddings are represented as a vector $\vec{emb} = E_{L-1}$.

The cdf by definition has the property that a data transformation passed to $F_\theta(\boldsymbol{X}) = U(0, 1)$ returns a uniform distribution. The uniform distribution maximizes the entropy of the distribution (Cover 1999) and improves the classification (Halbersberg, Wienreb, and Lerner 2020).

Some approaches work by training models $\mathcal{M}_{Prior}$ on data $\mathcal{D}_{finetune} \subseteq \mathcal{D}$ and include a novel domain $\mathcal{C}_{Adapted}$, resulting in the final model $\mathcal{M}_{Final}$:

$$\mathcal{M}_{Init} \xrightarrow{\mathcal{D}_{Prior}} \mathcal{M}_{Prior} \xrightarrow{\mathcal{D}_{finetune}} \mathcal{M}_{Final} \tag{5}$$

Such models have the crucial drawback in cases when the embeddings of the pre-trained model $\mathcal{M}_{Prior}$ are calculated on a data distribution that is different from the novel domain $\mathcal{D}_{Adapted}$. The embeddings mapped to a concentrated area of the embedding value space become uninformative.

Definition 3.2 (cdf-embeddings) The cumulative embeddings (cdf-embeddings) is a transformation of the embeddings. Under the dataset over which it is estimated, it produces the uniform distribution.

The main features of the cdf-embeddings are:

- The cdf-embeddings are calculated at inference time after $\mathcal{D}_{Prior}$ is fixed;
- The cdf-embeddings are calculated on the training data $\mathcal{D}_{Prior}$ and the few-shot data $\mathcal{D}_{Adapted}$;
- The cdf-embeddings require to know the distribution of the data $F_\theta(\boldsymbol{X})$. We make a law-of-large-numbers assumption on the normality of the embeddings data distribution, thus, the empirical estimates $\hat{\mu}$ and $\hat{\sigma}$ need to be saved under $\mathcal{N}(\hat{\mu}, \hat{\sigma})$ assumption, where $\hat{\mu}$ and $\hat{\sigma}$ are the mean and the standard deviation of the embeddings data distribution of the prior data $\mathcal{D}_{Prior}$.

Definition 3.3 (cdf-embeddings finetuning). Let us consider the few data points from the novel domain $\mathcal{D}_{Adapted}$, then the cdf-embeddings finetuning is a transformation of the cdf-embeddings of the prior model $\mathcal{M}_{Prior}$ to the cdf-embeddings of the final model $\mathcal{M}_{Final}$.

Definition 3.4 (cdf-embeddings Nadaraya-Watson regression). The cdf-embeddings Nadaraya-Watson regression is a transformation of the cdf-embeddings of the prior model $\mathcal{M}_{Prior}$ to the cdf-embeddings of the final model $\mathcal{M}_{Final}$ using the Nadaraya-Watson regression Nadaraya (1964); Watson (1964); Bierens (1996).

$$\hat{y} = g(\boldsymbol{X}) = \frac{\sum_{i=1}^{N} K(\boldsymbol{X}_i, \boldsymbol{X}) y_i}{\sum_{i=1}^{N} K(\boldsymbol{X}_i, \boldsymbol{X})},$$

where $K(\boldsymbol{X}_i, \boldsymbol{X})$ is the kernel function, y_i is the target value of the ith data point, $\boldsymbol{X}_i$ is the cdf-embedding of the ith data point, $\boldsymbol{X}$ is the cdf-embedding of the new data point, and N is the number of the few-shot observations. In our experiments, we use the sklearn library for Nadaraya-Watson regression.[1]

Experimental study

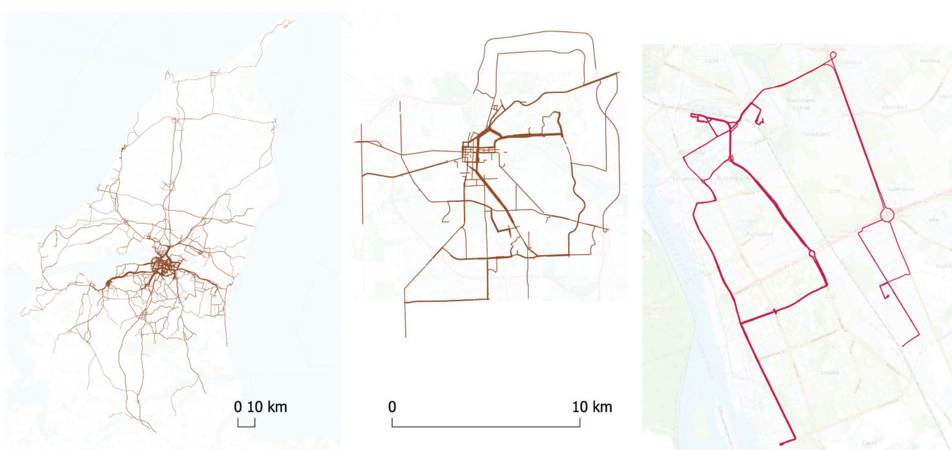
To validate the proposed models, experiments were performed on real telematic data. We present the data sets first, followed by the experimental results.

Data

Three datasets are investigated. In all datasets, the data is map matched to the OSM (OpenStreetMap) road network using the pgMapMatch package (Millard-Ball, Hampshire, and Weinberger 2019), and the characteristics are aggregated per segment. Figure 5 illustrates the geographical coverage of the three datasets.

Danish dataset

The largest EV dataset that was used contains data collected in Denmark throughout 2012 (Krogh, Andersen, and Torp 2015) and covers the trips of 164 almost identical EVs. The Denmark data is joined with the weather data from NOAA (Oceanic and Administration 2021), as well as digital elevation data (Observation and Center 2012). Finally, observations with missing values are removed. The dataset contains 442,963 segment traversals from 11,908 unique trips. We split the data 70/10/20% or 8335, 1191, 2382 trips for



(a) (b) (c)

Figure 5. The EV data set trajectories. Line thickness represents the number of trip segments on a road segment. (a) Denmark (North Jutland) dataset. (b) USA (Ann Arbor, Michigan) dataset. (c) Lithuania (Klaipėda) dataset.

training, validation and testing datasets, or 305,867, 47,315, 89,781 segments respectively as in Petkevičius et al. (2021).

USA dataset

The second EV dataset is extracted from the Extended vehicle energy dataset (eVED), a large-scale, open dataset of fuel and energy data collected in Ann Arbor, Michigan, USA (Zhang et al. 2022). The data captures GPS trajectories of vehicles along with the time-series data of their speed as well as fuel, energy, and auxiliary power use. In our case, only the EV data was used. Trips with more than 70% of successful map matchings are chosen. The data is augmented with the weather data from NOAA (Oceanic and Administration 2021). The trip data of eVED containes only the data from three vehicles. The train/test splitting was done by taking vehicle 10 data for training and using the data from the remaining two vehicles for testing.

Lithuanian dataset

The dataset of electric buses stems from Klaipėda, Lithuania. The data is collected from two buses during February–May 2023. The data is provided by electric bus manufacturer DANCER.[2] The data captures GPS trajectories of vehicles along with the time-series data of their speed as well as battery state, temperature, and weight.

Variables and data preprocessing

After the pre-processing, 30 trip-segment characteristics are used as the input for the models (see Table 1). The road type can be one of the following: living street, motorway, motorway link, primary, residential, secondary, secondary link, service, tertiary, track, trunk, trunk link, unclassified, unpaved; the road conditions cover: drifting, dry, fog, freezing, none, snow, thunder, wet.

Triplet data

The triplet data preparation from the segment-level data is done in two steps. First, the pre- and post-segments are identified for each segment in a trip. The

Table 1. The characteristics of a trip segment, in total 30 variables, and which variables arepresent in which of the three datasets.

Characteristic	# Var.	Denmark	USA	Klaipėda
Speed	1	+	+	+
Time (s)	1	+	+	+
Air temperature	1	+	+	+
Wind speed (ms)	1	+	+	-
Wind direction	1	+	-	-
Triplet angle diff	1	+	-	-
Time of day	1	+	+	+
Weekend	1	+	+	+
Road conditions	8	+	-	-
Road type	14	+	+	+

first and the last segments of a trip do not have both neighboring segments. In such cases, the pre/post segments are set to the same segment. Second, all 30 variables from each of the three segments are combined to a single triplet vector. The triplet data is used for training and testing of the models.

Results

Experiments

In our study, multiple configurations of deep neural networks were considered. Experimentally, the best regression models for energy consumption were identified. The architectures of those models are presented in Table 2. For all models, training was carried out for up to 50 epochs with a batch size of 128, with early stopping by waiting 20 epochs to track testing loss. The binary mean square error loss function and Adam optimizer Kingma (2014), Goodfellow, Bengio and Courville (2016) deep with a learning rate of 0.001 were used for neural network parameter estimation.

Each of the configurations listed in Table 2 are considered in two versions: 1) as a regression model, with MSE as a loss function and the output dimension of one; 2) as a probabilistic model with the negative log-likelihood (ll).

Nadaraya-Watson regression of c-embeddedings

First, we investigated the c-embeddings Nadaraya-Watson regression. The results (Klaipėda dataset) are presented in Table 3.

Table 2. Configurations of regression neural network models.

Model	Architecture/hyper-parameters
M1	FL-D-FC(4*H)-BN-Relu-D-FC(H)-BN-Relu-D-FC(1)
M2	FL-BN-FC(H)-Relu-BN-Res(1,2,1)-FL-FC(1)
M3	FL-BN-MHA(FC(H), 4)-Res(2)-FL-FC(1)
M4	VS(H)-FC(1)
M5	VS(H)-BN-FC(H)-MHA(FC(H), 8)-FC(1)
M6	FL-MHA(FC(H), 8)-FL-FC(1)
M7	FL-BN-Wide-Deep(4*H,H)-MHA-FC(1)
M8	FL-Cross(128,16)-Deep(4*H,H)-FL-FC(1)
M9	FL-MHA(FC(32), 8)-Cross(4*H,H)-Deep(4*H,H)-FL- MHA(FC(16), 8) -FC(1)
M10	FL-BN-Wide-Deep(4*H,H)-FC(1)

The model architectural representations. FL stands for Flattening, D – for Dropout ($p = .2$), BN – for batch normalization, Res – for residual block, MHA – Multi-head attention, VS – continuous variable selection layer, Wide – wide connection, Deep – deep connection, neural network layers, and Relu is a non-linear point-wise activation. H is the hidden state parameter, $H = 16$.

Table 3. The mean absolute value (MAE) statistics of c-embeddings of the investigated models on unseen test-data segments (Klaipėda dataset). Columns are the models used as prior models (feature extractors) for c-embeddings.

M1	M9	M8	M6	M7	M3	M2
7.3592	30.7558	26.0246	26.7407	29.2027	30.6492	14.5868

The table demonstrates that the c-embeddings of Nadaraya-Watson regression are not sufficient for transfer learning under a small embedding dimension (e.g. $H = 16$). The results are not better than the M1 model. The benefits of parameterless models are not sufficient for practical use in this particular case.

Triplet data model

Next, we present the results of the investigated models on the Danish dataset. Table 4 shows that the larger parameter models outperform the smaller neural networks. However, it should be noted that the M9 model, while 10× larger than most of the models, does not outperform the M4 model. The results show that the triplet-level predictions, compared to the trip-level predictions, are more challenging (Petkevičius et al. 2021). The M2 and M8 models were overfitted, thus they failed significantly according to the tested performance measures. The best results (Petkevičius et al. 2021) demonstrated on long sequence data reach MAE 0.22 and MSE 0.32. This demonstrates that the results in Table 4 using zero-shot approach on individual triplets fall shortly below the supervised learning results.

The USA dataset results are presented in Table 5. The results show that the models are not performing well onunseen data. The data split was arranged by selecting vehicle 10 as a training dataset (1235 records), and the remaining two cars (total 12,143 records) as the testing subset. One can see that the model became overfitted and did not generalize well on unseen data.

Table 4. The statistics of investigated models on unseen test-data segments (Denmark dataset).

Model	MAE	MSE	RMSE	R2	# params
M4	**0.0232**	**0.0026**	**0.0510**	0.5480	87387
M3	0.0246	0.0036	0.0604	**0.3659**	83657
M5	0.0254	0.0029	0.0542	0.4884	29107
M9	0.0269	0.0039	0.0623	0.3246	233793
M1	0.0333	0.0045	0.0673	0.2115	16449
M10	0.0354	0.0057	0.0757	0.0019	14755
M6	0.1004	0.0124	0.1116	.	52929
M7	0.1045	0.1207	0.3475	.	98265
M2	2.5250	75.5819	8.6938	.	41833
M8	44.0542	1049110.1642	1024.2608	.	31135

Table 5. The statistics of investigated models on unseen test-data segments (USA dataset).

Model	MAE	MSE	RMSE	R2	# params
M1	**0.3873**	**1.9940**	**1.4121**	**0.0394**	**16449**
M9	0.4429	2.0484	1.4312	0.0132	233793
M4	0.4718	2.1310	1.4598	.	87387
M5	0.5474	2.2365	1.4955	.	29107
M8	0.6107	2.9297	1.7116	.	31135
M7	1.5798	4.3663	2.0896	.	98265
M6	5.3829	86.0566	9.2767	.	52929
M3	6.5579	46.1958	6.7968	.	83657
M10	10.8146	118.8738	10.9029	.	14755
M2	66.1188	4384.3006	66.2141	.	41833

The Klaipėda dataset results are presented in Table 6. The results show that the models are performing relatively well on unseen data. The data split was arranged by selecting one bus to the training dataset (40603 records), and a second bus (total 65,516 records) to the testing subset. We will demonstrate the improvement over the M1 model creation in the following sections.

Probabilistic models

The deep probabilistic models have the benefit of the uncertainty estimation which could aid the route planning and the energy consumption estimation (Petkevičius et al. 2021). The deep probabilistic models are trained on the same data as regression models, but predicting the distribution parameters instead of the point prediction. The distribution parameters are estimated by minimizing the negative log-likelihood loss function. We trained the models with the same setup as regression models. The best models are presented in Table 7.

Table 7 shows that the probabilistic models benefit from the uncertainty estimation with 97.28% of true energy consumption values falling into $\hat{y} \pm 3\hat{SD}$ interval. However, most of the models, due to stability issues, at some point in the training process produced the NaN values, which prevented obtaining meaningful results for these models. Due to the lack of convergence and stability of the models, the probabilistic models were not investigated further in this study.

Transfer learning of c-embeddedings

The transfer-learning models are widely known in deep learning, especially in finetuning the backbone features of ImageNet pre-trained models (Goodfellow, Bengio, and Courville 2016; Zhuang et al. 2020). The features from the prior knowledge enable more robust representations under a limited amount of data.

Table 6. The statistics of investigated models on unseen test-data segments (Klaipėda dataset).

Model	MAE	MSE	RMSE	R2	# params
M4	**0.0990**	**0.0177**	**0.1332**	**0.4041**	**87387**
M5	0.0991	0.0181	0.1345	0.3921	29107
M8	0.1016	0.0220	0.1485	0.2598	31135
M9	0.1030	0.0216	0.1470	0.2741	233793
M7	0.1047	0.0204	0.1429	0.3143	98265
M2	0.1091	0.0234	0.1529	0.2149	41833
M3	0.1105	0.0214	0.1464	0.2805	83657
M1	0.1255	0.0238	0.1544	0.1991	16449
M10	0.1858	0.0553	0.2351	.	14755
M6	1.9136	6.6088	2.5708	.	52929

Table 7. The statistics of investigated deep probabilistic models on unseen test-data segments (Denmark dataset). Rows are models investigated as prior models for c-embeddings. Columns show the performance statistics.

Model	MAE	MSE	RMSE	R2	CI	Number of params
M1	0.0339	0.0063	0.0063	.	0.9728	16482
M10	**0.0355**	**0.0061**	**0.0061**	.	**0.9705**	**14862**

Table 8. The mean absolute value (MAE) statistics of c-embeddings in investigated models on unseen test-data segments (USA dataset). Rows are the models investigated as the prior models for c-embeddings. Columns represent the models investigated.

Model	M1	M8	M9	M6	M2	M3	M10	M7
M8	0.449	1.255	0.949	0.408	0.463	0.502	0.423	0.444
M9	0.459	0.864	0.580	0.400	0.462	0.520	0.502	0.446
M4	0.447	1.125	0.906	0.400	0.515	0.511	0.413	0.443
M6	0.466	0.588	0.448	0.409	0.471	0.577	0.492	0.445
M2	0.471	0.614	0.735	0.409	0.480	0.514	0.487	0.445
M3	0.473	0.523	0.571	0.390	0.448	0.597	0.475	0.452
M10	0.455	0.497	0.541	**0.372**	**0.465**	**0.471**	**0.462**	**0.445**
M7	0.464	0.688	1.504	0.396	0.453	0.501	0.415	0.442

Table 9. The mean absolute value (MAE) statistics of c-embeddings in investigated models on unseen test-data segments (Klaipėda dataset).

Model	M1	M8	M9	M6	M2	M3	M10	M7
M1	0.138	0.123	0.100	0.117	99.664	17.986	0.100	0.111
M8	0.134	0.100	0.104	0.145	0.117	0.128	**0.098**	**0.116**
M9	-	0.111	0.103	0.186	-	0.147	0.105	0.103
M6	0.133	0.104	0.106	0.135	0.171	0.110	0.100	0.145
M10	0.139	0.104	0.100	0.217	0.124	0.166	0.099	0.123
M7	0.133	0.101	0.104	0.158	303.788	0.198	0.100	0.109

The results of transfer learning using c-embeddings of the USA dataset are presented in Table 8. The table shows that the c-embeddings demonstrate better accuracy (M6 model) than using the original data. It is worth mentioning that the majority of the feature extractor models using the Prior model trained on the Denmark data produce robust results for the new models. The results show that the c-embeddings are sufficient for transfer learning.

The results of the experiments on transfer learning using c-embeddings on the Klaipėda dataset are presented in Table 9. One can see in the table that the M1 deep neural network is not a sufficient feature extractor and multiple models cannot learn from those features. The best results (MAE = 0.1) are equivalent to the original model training results. The majority of results are within 10% error margin. The results show that the c-embeddings are sufficient for transfer learning.

Conclusions

Motivated by the necessity to address different EV vehicle types when predicting energy consumption, we propose a suite of few-shot learning models for EV energy-use prediction. In particular, we propose not only efficient trip decomposition to the triplet level, but also an approach for adapting models for different EV vehicle models. We investigate the learnable and parameter-free approaches using few-shot learning. The experiments show that the prior model embeddings can be successfully used to model different domains of EV vehicles/buses. The datasets used for this study, including the bus data, are mainly from urban areas. Further work and experiments are needed to improve the domain adaptation across the different EV vehicle domains.

Notes

1. SciKit Kernel Density.
2. Similar, but less detailed data is also publicly available at https://dancerbus.com/techniniai-duomenys-gyvai/?lang=enhttps://dancerbus.com/techniniai-duomenys-gyvai/?lang=en.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Funding

This work was supported by the European Regional Development Fund [project No 01.2.2-LMT-K-718-02-0018] under a grant agreement with the Research Council of Lithuania (LMTLT).

ORCID

Linas Petkevičius http://orcid.org/0000-0003-2416-0431

Code availability statement

The code used in the experimental study is openly available at Github repository. https://github.com/linas-p/Few-Shot-Learning-for-EV-Energy-Consumption.

Data availability statement

Some of the experimental data used in this study is openly available. In particular, the eVED dataset can be found at https://github.com/zhangsl2013/eVED https://github.com/zhangsl2013/eVED., reference number (Zhang et al. 2022), the DANCER bus dataset is proprietary and not publicly available, but similar data can be obtained at https://dancerbus.com/technical-data-live/ https://dancerbus.com/technical-data-live/. The processed Danish EV data are accesible on Github repo https://github.com/linas-p/Few-Shot-Learning-for-EV-Energy-Consumption.

References

Bierens, H. J. 1996. *Topics in advanced econometrics: estimation, testing, and specification of cross-section and time series models*. Cambridge, UK: Cambridge university press.

Chen, H., Y. Wang, G. Wang, and Y. Qiao. 2018. LSTD: A low-shot transfer detector for object detection. In *AAAI Conference on Artificial Intelligence (AAAI)*, New Orleans, Louisiana, USA, 2836–43.

Chen, X., Z. Lei, and S. V. Ukkusuri. 2022. Prediction of road level energy consumption of battery electric vehicles. In *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Macau, China, 2550–55. IEEE.

Cover, T. M. 1999. *Elements of information theory*. Hoboken, NJ, USA: John Wiley & Sons.

Erbach, G. 2023. CO2 emission standards for new cars and vans. European Union. Accessed May 30, 2023. https://www.europarl.europa.eu/RegData/etudes/BRIE/2022/698920/EPRSBRI(2022)698920EN.pdf/.

Goodfellow, I., Y. Bengio, and A. Courville. 2016. *Deep learning*. Cambridge, MA, USA: MIT press.

Halbersberg, D., M. Wienreb, and B. Lerner. 2020. Joint maximization of accuracy and information for learning the structure of a Bayesian network classifier. *Machine Learning* 109 (5):1039–99. doi: 10.1007/s10994-020-05869-5.

Hua, Y., M. Sevegnani, D. Yi, A. Birnie, and S. McAslan. 2022. Fine-grained RNN with transfer learning for energy consumption estimation on EVs. *IEEE Transactions on Industrial Informatics* 18 (11):8182–90. doi: 10.1109/TII.2022.3143155.

Kingma, D. P., and J. Ba. 2014. Adam: A method for stochastic optimization. arXiv preprint arXiv:1412.6980.

Krogh, B., O. Andersen, and K. Torp. 2015. Analyzing electric vehicle energy consumption using very large data sets. In *International Conference on Database Systems for Advanced Applications*, Hanoi, Vietnam.

Li, X., X. Yang, Z. Ma, and J.-H. Xue. 2023. Deep metric learning for few-shot image classification: a review of recent developments. *Pattern Recognition* 138:109381. doi: 10.1016/j.patcog.2023.109381.

Madziel, M. 2024. Energy modeling for electric vehicles based on real driving cycles: an artificial intelligence approach for microscale analyses. *Energies (Basel)* 17 (5):1148. doi: 10.3390/en17051148.

Meng, A., S. Chen, Z. Ou, J. Xiao, J. Zhang, S. Chen, Z. Zhang, R. Liang, Z. Zhang, Z. Xian, et al. 2022. A novel few-shot learning approach for wind power prediction applying secondary evolutionary generative adversarial network. *Energy* 261:125276. doi: 10.1016/j.energy.2022.125276.

Meng, J., Y. You, M. Lin, J. Wu, and Z. Song. 2024. Multi-scenarios transferable learning framework with few-shot for early lithium ion battery lifespan trajectory prediction. *Energy* 286:129682. doi: 10.1016/j.energy.2023.129682.

Millard-Ball, A., R. C. Hampshire, and R. R. Weinberger. 2019. Map-matching poor quality GPS data in urban environments: The pgMapmatch package. *Transportation Planning and Technology* 42 (6):539–53. doi: 10.1080/03081060.2019.1622249.

Nadaraya, E. A. 1964. On estimating regression. *Theory of Probability & Its Applications* 9 (1):141–42. doi: 10.1137/1109020.

Observation, Earth Resources, and Science (EROS) Center. 2012. Shuttle radar topography mission 1 arc and 3 arc-second digital terrain elevation data - void filled.

Oceanic, National Centers For Environmental Information. National, and Atmospheric Administration. 2021. Local climatological data. Accessed April 5, 2021. www.ncdc.noaa.gov .

Petkevičius, L., S. Šaltenis, A. Čivilis, and T. Kristian. 2021. Probabilistic deep learning for electric-vehicle energy-use prediction. In *17th International Symposium on Spatial and Temporal Databas*, Virtual event, USA.

Popiolek, A., Z. Dimitrova, J. Hassler, M. Petit, and P. Dessante. 2023. Comparison of decentralised fast-charging strategies for long distance trips with electric vehicles. *Transportation Research Part D: Transport & Environment* 124:103953. doi: 10.1016/j.trd.2023.103953.

Roy, M., A. Nambi, A. Sobti, T. Ganu, S. Kalyanaraman, S. Akella, J. Subha Devi, and S. A. Sundaresan. 2022. Reliable energy consumption modeling for an electric vehicle fleet. In *ACM SIGCAS/SIGCHI Conference on Computing and Sustainable Societies (COMPASS)*, Seattle, Washington, USA, 29–44.

Šaltenis, S., A. Barauskas, A. Brilingaitė, L. Bukauskas, V. Čeikutė, and A. Čivilis. 2023. Charge-arrival-time profiles for long ev routes. *Institute of Electrical and Electronics Engineers Access* 11:133004–20. doi: 10.1109/ACCESS.2023.3334392.

Shen, H., Z. Wang, X. Zhou, M. Lamantia, K. Yang, P. Chen, and J. Wang. 2022. Electric vehicle velocity and energy consumption predictions using transformer and Markov-Chain Monte Carlo. *IEEE Transactions on Transportation Electrification* 8 (3):3836–47. doi: 10.1109/TTE.2022.3157652.

Shi, J., T. Zeng, and S. Moura. 2024. *The nature and strategy of minimizing the total travel time for long-distance driving of an EV*, 10. USA: IEEE Transactions on Transportation Electrification.

Snell, J., K. Swersky, and R. S. Zemel. 2017. Prototypical networks for few-shot learning. *Advances in Neural Information Processing Systems*, 30: 4077–87.

Subramanian, V., F. Feijoo, S. Sankaranarayanan, K. Melendez, and T. K. Das. 2022. A bilevel conic optimization model for routing and charging of EV fleets serving long distance delivery networks. *Energy* 251:123808. doi: 10.1016/j.energy.2022.123808.

Tang, L., H. Xie, X. Wang, and Z. Bie. 2023. Privacy preserving knowledge sharing for few-shot building energy prediction: A federated learning approach. *Applied Energy* 337:120860. doi: 10.1016/j.apenergy.2023.120860.

Tsimpoukelli, M., J. L. Menick, S. Cabi, S. M. Eslami, O. Vinyals, and F. Hill. 2021. Multimodal few-shot learning with frozen language models. *Advances in Neural Information Processing Systems* 34:200–12.

Vinyals, O., C. Blundell, T. Lillicrap, D. Wierstra. 2016. Matching networks for one shot learning. *Advances in Neural Information Processing Systems* 29:3630–38.

Wang, Y., Q. Yao, J. T. Kwok, and L. M. Ni. 2020. Generalizing from a few examples: A survey on few-shot learning. *ACM Computing Surveys (CSUR)* 53 (3):1–34. doi: 10.1145/3386252.

Watson, G. S. 1964. Smooth regression analysis. *Sankhyā: The Indian Journal of Statistics, Series A*, 26:359–72.

Xu, J., K. Li, and D. Li. 2024. An automated few-shot learning for time series forecasting in smart grid under data scarcity. *IEEE Transactions on Artificial Intelligence*, 5(6).

Yan, X., Z. Chen, A. Xu, X. Wang, X. Liang, and L. Lin. 2019. Meta R-CNN: Towards general solver for instance-level low-shot learning. In *International Conference on Computer Vision (ICCV)*, Seoul, South Korea, 9576–85.

Zhang, S., D. Fatih, F. Abdulqadir, T. Schwarz, and X. Ma. 2022. Extended vehicle energy dataset (eVED): an enhanced large-scale dataset for deep learning on vehicle trip energy consumption. arXiv preprint arXiv:2203.08630.

Zhang, S., Z. Liu, and H. Su. 2023. State of health estimation for lithium-ion batteries on few-shot learning. *Energy* 268:126726. doi: 10.1016/j.energy.2023.126726.

Zhang, Y., H. Tang, and K. Jia. 2018. Fine-grained visual categorization using metalearning optimization with sample selection of auxiliary data. In *European Conference on Computer Vision*, Munich, Germany, 233–48.

Zhuang, F., Z. Qi, K. Duan, D. Xi, Y. Zhu, H. Zhu, H. Xiong, and Q. He. 2020. A comprehensive survey on transfer learning. *Proceedings of the IEEE* 109 (1):43–76. doi: 10.1109/JPROC.2020.3004555.