Probabilistic Deep Learning for Electric-Vehicle Energy-Use Prediction

Linas Petkevičius
Vilnius University, Lithuania
linas.petkevicius@mif.vu.lt

Simonas Šaltenis
Vilnius University, Lithuania
simonas.saltenis@mif.vu.lt

Alminas Čivilis
Vilnius University, Lithuania
alminas.civilis@mif.vu.lt

Kristian Torp
Aalborg University, Denmark
torp@cs.aau.dk

ABSTRACT

The continued spread of electric vehicles raises new challenges for the supporting digital infrastructure. For example, long-distance route planning for such vehicles relies on the prediction of both the expected travel time as well as energy use. We envision a two-tier architecture to produce such predictions. First, a routing and travel-time-prediction subsystem generates a suggested route and predicts how the speed will vary along the route. Next, the expected energy use is predicted from the speed profile and other contextual characteristics, such as weather information and slope.

To this end, the paper proposes deep-learning models that are built from EV tracking data. First, as the speed profile of a route is one of the main predictors for energy use, different simple ways to build speed profiles are explored. Next, eight different deep-learning models for energy-use prediction are proposed. Four of the models are probabilistic in that they predict not a single-point estimate but parameters of a probability distribution of energy use on the route. This is particularly relevant when predicting EV energy use, which is highly sensitive to many input characteristics and, thus, can hardly be predicted precisely. Extensive experiments with two real-world EV tracking datasets validate the proposed methods. The code for this research has been made available on GitHub.

CCS CONCEPTS

• **Computer systems organization → Neural networks**; • **Information systems → Spatial-temporal systems**; • **Applied computing → Forecasting**.

KEYWORDS

Spatio-temporal data, E-Vehicle energy consumption, Deep neural network, Probabilistic model, Sequential data

ACM Reference Format:
Linas Petkevičius, Simonas Šaltenis, Alminas Čivilis, and Kristian Torp. 2021. Probabilistic Deep Learning for Electric-Vehicle Energy-Use Prediction. In *17th International Symposium on Spatial and Temporal Databases (SSTD '21), August 23–25, 2021, virtual, USA.* ACM, New York, NY, USA, 11 pages. https://doi.org/10.1145/3469830.3470915

SSTD '21, August 23–25, 2021, virtual, USA

ACM ISBN 978-1-4503-8425-4/21/08.
https://doi.org/10.1145/3469830.3470915

1 INTRODUCTION

Transportation is currently undergoing a transformation driven by the emergence of new automotive technologies, such as electric and autonomous vehicles, as well as the continued digitalization of all aspects of transportation. For example, the efficiency of a fleet of autonomous electric vehicles will be highly dependent on effective routing and scheduling algorithms. Such algorithms will, in turn, depend on data-driven predictions of travel time and energy use.

Modern routing and mapping systems, such as Google Maps, use large amounts of geo-positioning data to maintain a dynamically updated time-dependent graphs as representations of road networks. Based on this data, the systems suggest optimized routes according to a set of criteria. Furthermore, travel time can be predicted. For electric vehicles, in addition to travel time, energy consumption prediction is equally important. This is essential for longer trips which involve planning of where and how much to recharge.

As the predictions are made based on historical data, the process usually benefits greatly from large amounts of such data. While travel-time predictions can be made on data from all kinds of vehicles, energy consumption predictions can benefit only from electric vehicle data. Further, only data concerning the specific model of the vehicle can be readily used. This further amplifies the scarcity of EV data due to the current limited spread of such vehicles.

Thus, we argue for a two-tier architecture to produce energy consumption predictions. First, routing and travel-time-prediction sub-system generates a suggested route and predicts how the speed will vary along the route. This can benefit from large amounts of data from all kinds of vehicles and other telematic sources. Next, the route, its predicted *speed profile* and other contextual characteristics, such as the current weather as well as the slope of the road, are used to predict the expected energy use along the route. The prediction is based on machine learning models built on relevant smaller EV datasets.

The focus of this paper is to explore deep-learning methods for such predictions. As the speed profile of a route is one of the main predictors for energy use of EV [11] and it itself is a prediction, the important question is how accurate the speed profile should be to predict energy use effectively.

The second, related challenge stems from the fact that EV energy use is highly sensitive to many input characteristics, not all of which may be known at the time of prediction and/or recorded

in historical data used to train the prediction models. Consider an example in Figure 1, which is taken from the dataset used in the experimental study. It shows how energy use (or recuperation due to regenerative braking) varies for different trips on the same nine consecutive segments of the road network. Each point in the figure corresponds to a traversal of a segment and horizontal spacing of the points is used to separate them visually.

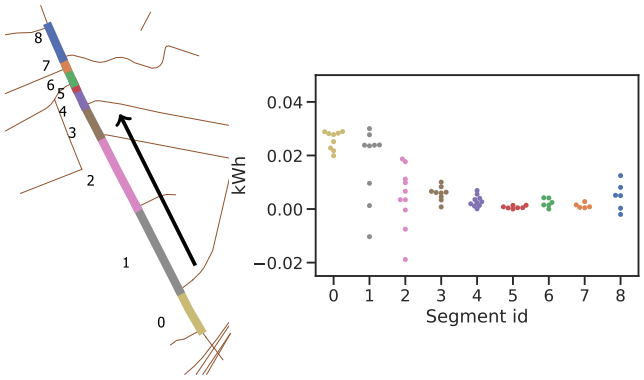

Figure 1: Variation of energy used/gained in consecutive segments under similar conditions of air temperature 12–16°C**, speed** 50–60 **km/h, and travel direction**

Motivated by such energy-use variability, we propose probabilistic prediction methods that, for a given route, predict not just a single-point estimate but a probability-distribution parameters, which can then be used, for example, to derive a confidence interval for a given confidence threshold. Such confidence intervals of energy can be used to calculate minimum and maximum charging times when planning longer trips.

The contributions of this paper are twofold. First, to the best of our knowledge, it presents the first in-depth exploration of a set of deep-learning models for EV energy use prediction that are learned from the GPS tracking data. As mentioned, some of the models predict not just point estimates but the parameters of predicted probability distributions. Next, the paper contributes with the insights from the extensive testing of the models on two different real-world EV tracking datasets.

The remainder of the paper is structured as follows. First, Section 2 briefly surveys related work. Section 3 describes different versions of machine-learning problems. Deep-learning models are covered in Section 4. Finally, Section 5 reports on the experimental study and Section 6 concludes the paper.

2 RELATED WORK

Using machine learning and related methods in the area of EV energy use has not yet gained a lot of attention. We distinguish two classes of studies: those related to EV charging infrastructure and those related to the energy use by the vehicles themselves.

The modeling of EV infrastructure starts from modeling the energy use of the charging stations [27]. Such studies are also extended to optimize scheduling of the charging times using actor-critic learning [32]. While the total energy use is proportional to the number of EVs within some region, the energy use prediction for individual charging stations is more complicated [25].

Other type of problems concern modeling the individual energy use for a specific EV instance. Various management strategies are proposed using reinforcement learning to transfer efficient energy management from expected velocity to automated system control [6, 12]. The authors model the vehicle energy consumption in terms of the state-of-charge by minimizing the so-called Q-function using reinforcement learning. In contrast, we predict the energy consumption based on the data from many EVs. We view the optimization of driving as the next possible step that would benefit from the accurate energy-use predictions. Similar to our work, Liu et al. consider the speed profile as input to predict the state-of-charge (SOC) as output [13]. Deep learning applied on increased quantities of historical data enables modeling of EVs based on driver's history [2, 30], and on predetermined fixed routes [33].

The probabilistic energy prediction approach [22] was investigated by creating statistical models. We use probabilistic models on sequence data inspired by the works that concern with time-to-event churn prediction [15] or asset management [24]. We combine this approach with the recurrent models which are successfully used on spatio-temporal data in predicting extreme condition traffic [31] or origin-destination forecasting [8].

3 MODELING EV ENERGY USE

First, the notation and modeling of EV energy-use prediction is presented, then it is formalized as different machine learning problems.

3.1 Problem modeling

We assume the vehicles travel in a road network that is modeled as a directed weighted graph. A weight of an edge consists of a set of *attributes*, such as the length, the road type, and the slope (the difference of altitudes at the beginning and at the end of the edge). A *planned route*, r, is then a sequence of route *segments*, each of which corresponds to a planned traversal of an edge of the road network.

Similarly, we assume that there is a historical database of N map-matched trips. As a planned route, each trip $r_j, j = 1, \ldots, N$ from the database consists of trip segments $s_{r_j,k}$, $k = 1, \ldots, N_{r_j}$. Figure 2 illustrates such a trip and its segments as well as the speed and cumulative use of energy along the trip.

In addition to the attributes inherited from the road network, the segments of trips (and planned routes) inherit many other trip-specific attributes such as time of day, day of week, weather conditions, and most importantly travel time, which is either a recorded travel time (for a trip) or a time derived from a predicted *speed profile* (fro a planned route), as described in the next section. The totality of all such attributes, both road-specific and trip-specific is called the *characteristics* or *input covariates* of a segment.

An important attribute of a historical trip segment is the recorded energy use which can be both positive (a loss of energy) and negative (a gain of energy). Energy use is modeled as a function of all other input covariates.

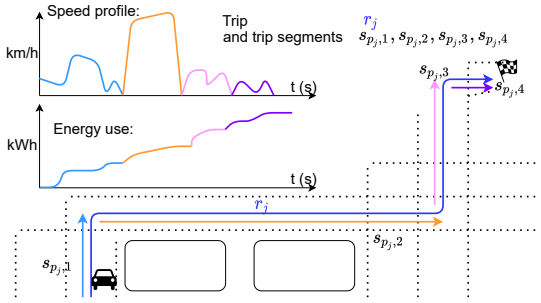

Figure 2: The route r_j, its segments $s_{r_j,k}$, speed profile, and energy prediction

We consider the EV energy-use prediction as a supervised learning problem to learn a mapping from above-mentioned characteristics x_j of an individual segment of a route to energy use on that segment $g_\phi : X \subset \mathbb{R}^{d_x} \rightarrow E \subset \mathbb{R}$, or, more generally, a mapping from the characteristics of all segments of a route to the total energy-use on a route $g_\phi : X \subset \mathbb{R}^{N_r \times d_x} \rightarrow C \subset \mathbb{R}$. The mapping should be learned from a given training dataset $\mathcal{D}_{tr} = \{(x_{j,i}, e_{j,i}) | 1 \leq i \leq N_{r_j}, 1 \leq j \leq N\}$, where $e_{j,i}$ is energy consumed on segment i of trip j and $x_{j,i}$ are its characteristics. In our main experimental study, the number of characteristics $d_x = 30$ and the maximum number of segments per trip $N_r = 550$. We also denote $M = \sum_{j=1}^{N} N_{r_j}$, the total number of segments in a training dataset. Finally, ϕ are the learned internal parameters of model g.

As suggested above, there are two different formulations of the learning problem. In the first version, called *segment-level* prediction, the model predicts energy use separately for each segment of a route. Thus, it is either applied on each segment, or, alternatively, it takes the characteristics of all segments as inputs and has N_r outputs. Finally, the predictions are summed to get the prediction of the total consumed energy on the route. For a trip r_j, $\hat{c}_{r_j}$ denotes the predicted total consumption of energy, while c_{r_j} denotes the corresponding ground-truth value in the training dataset:

$$c_{r_j} = \sum_{i=1}^{N_{r_j}} e_{r_j,i}, \quad \hat{c}_{r_j} = \sum_{i=1}^{N_{r_j}} \hat{e}_{r_j,i}. \tag{1}$$

In the second version of the problem, termed *accumulated* prediction, the model predicts $\hat{c}_{r_j}$ as its single output. Experimental study explores the benefits of the two approaches in terms of prediction accuracy.

Note that the granularity of the division of planned routes and recorded trips into segments depends on how the road network is subdivided into edges. The experimental study of this work benefits from Open Street Map (OSM) [23] and its rather fine-grained division of roads into multiple edges to capture differing topology of roads and variances of other attributes in the recorded trips and predicted routes. While out of scope of this work, the granularity can be adjusted by splitting/joining edges of the road network.

Figure 3 illustrates one trip from the main training dataset used in the experimental study. The average speed and energy use on each of the segments of the trip as well as the cumulative energy use are shown demonstrating the fine-grained subdivision of a trip into its segments.

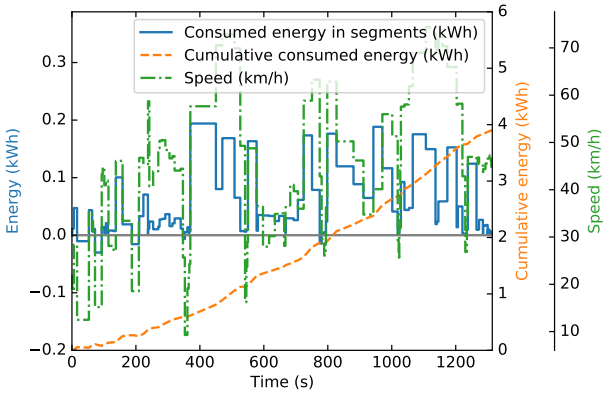

Figure 3: A speed profile and the corresponding energy use on a real trip

3.2 Speed profiles

As EVs convert the stored energy into kinetic energy very efficiently, and air resistance grows non-linearly with the increase in speed, average speed on a segment is one of the most important input charactersitics for predicting energy use. Further, the speed-energy dependency is further complicated by averaging over accelerations and decelerations inside the segments. Finally, in contrast to other charactersitics, it can not be known precisely at the time of prediction.

With this in mind, we propose four simple ways to generate the speed profile of a route. The speed profile is defined as a sequence of average-speed predictions—one for each segment of the route.

Very often the speed limits of segments are known [14]. Thus the first type of the speed profile, the *speed-limit* profile, sets the speeds to be equal to the speed limits of the road-network segments traversed by the route. While trivial to compute such a prediction is useful as a fallback option when other type of data is not available. It has been successfully used to optimize fuel use [16].

When historical data is available, likely speeds on segments can be predicted [10]. In the simplest form, for each segment, the average speed of all historical traversals of that segment is used as a prediction. This is the second type of the speed profile, which we term the *average* profile.

The third type of the speed profile is similar to the second, but it averages only over the historical trips that happened during the same day-of-week type (weekend vs. workday) and the same time-of-day category. Figure 4a) shows the rationale behind this, demonstrating that, in the main experimental dataset, the number of trips is quite different on weekends vs. on workdays and when considering different times of day. Thus, the global average would, for example, drown the higher speeds of the uncongested weekend traffic in large amounts of workday traffic data. We suggest three time-of-day categories: rush hour (`7:00-9:00` and `15:00-17:00`),

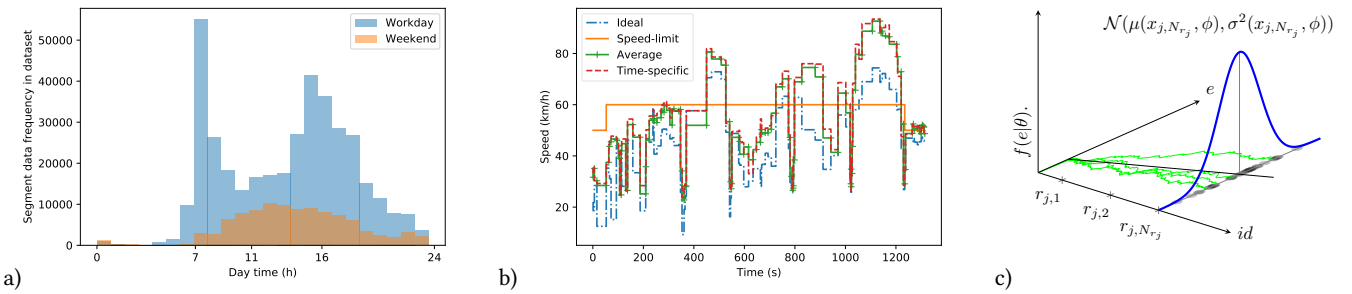

Figure 4: (a) The distribution of the number of segment traversals over different times of day and weekends vs. workdays; (b) Four types of speed profiles for an example trip; (c) Probabilistic prediction scheme: deep neural network predicts distribution parameters θ where energy e is modelled probabilisticly at each segment $r_{j,i}$.

night (`22:00-6:00`), and the rest of the day. The three categories combined with the binary day-of-week type gives a total of six different time-of-travel categories according to which average speeds on each segment are computed to get what we term a *time-specific average* profile.

The fourth type of the speed profile is the *ideal* speed profile that matches the real future traversal of the route perfectly. Such a profile is obviously not possible in a real system but can be used as an evaluation baseline. The four types of speed profiles for the trip from Figure 3 are illustrated in Figure 4b).

3.3 Machine learning problems

With the descriptions of data in place, the energy-use prediction problem is formalized as different variants of machine learning.

3.3.1 Regression problem. We denote X as the input space and E as the target (used energy) space. Here, $x_{j,i} \in X \subset \mathbb{R}^{d_x}$ is an observation of trip j at segment i, and $e_{j,i} \in E \subset \mathbb{R}$ is the used/gained energy. A simple regression model predicts energy consumption $\hat{e}_{j,i}$ for each segment $s_{r_j,i}$ independently: $\hat{e}_j = (\hat{e}_{j,1}, ..., \hat{e}_{j,N_{r_j}})$. The independent predictions can then be summed up to get the cumulative prediction of energy use for the whole route (see Equation (1)). The unknown parameters of a regression-like model are estimated by minimizing the mean square error loss (MSE):

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

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

3.3.2 Sequential regression problem. Since the observations in segments are spatially-dependent, it is natural to consider sequential models [28]. The problem can be modeled as a *sequence-to-sequence* (*seq-to-seq*) task and we employ the long short-term memory network (LSTM) [7] as a model. For each trip r_j, the model takes characteristics of all the segments $s_{r_j,k}, k = 1, .., N_{r_j}$ and predicts the energy use for each of them. The unknown internal parameters for LSTM can also be estimated by minimizing the MSE.

3.3.3 Deep probabilistic problem. As illustrated in Figure 1, the energy use of EVs is sensitive to many input characteristics such as the data granularity, various (unknown) external conditions, the driving style of the driver, the load of the car, or the equipment turned on. This makes precise predictions hard and point estimates can be misleading. In order to estimate and reduce the uncertainty in prediction we consider the probabilistic approach.

Suppose that target data are independent random variables $e_1, \ldots, e_M$. Denote by $F_i(e|\theta)$ the c.d.f. 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 $\theta_{j,i}$ depend on explanatory variables $x_{j,i}$: $\theta_{j,i} = g(x_{j,i}|\phi)$. We choose the operator $g(x|\phi)$ to be a deep neural network with unknown parameters ϕ. The model is trained on the training data $\{x_{j,i}, e_{j,i}\}_{i=1,..,N_{r_j},j=1,..,N}$ and the output of the model is $\theta_{j,i}$. Differently from regression-like problem formulation (see Section 3.3.1) the $e_{j,i}$ is just a realization of $F_i(y|\theta)$. Thus, a deep neural network is trained to predict the parameters of the distribution

$$\begin{aligned} &\theta_{j,i} = g(x_{j,i}|\phi) \\ &s.t. \quad e \sim F(e|\theta) \end{aligned} \tag{3}$$

Figure 4c) illustrates the probabilistic problem formulation, where ϕ are the parameters of a normal distribution.

Next, we discuss how to optimize probabilistic deep neural networks. In contrast to the point-estimate problem formulation where the loss functions like MSE, L_1, L_2 or similar can be used, these functions do not fit probabilistic problem formulation since the model predicts distribution parameters and not energy point estimates. Instead, we can use maximum likelihood estimation:

$$\mathcal{L}_{ML} = \prod_{i=1}^{M} f(e_i|\theta_i).$$

In practice $\mathcal{L}(\theta|e)$ is rarely applicable due to stability problems; thus, negative log-likelihood is used instead:

$$\mathcal{L}_{log-ML} = -\sum_{i=1}^{M} \log(f(e_i|\theta_i)).$$

Since parameters $\theta = \theta(\phi, x)$ have a functional form, we can also use this loss function and gradient-based methods to estimate model parameters ϕ. With given training data, we can learn both the network parameters $\hat{\phi}$ and the predicted distribution parameters $\hat{\theta}$.

3.3.4 Deep sequential probabilistic problem. Similarly to Section 3.3.2, the probabilistic energy-use prediction problem can be formulated as a seq-to-seq task, but instead of predicting the energy consumption sequence $e_1, \ldots, e_{N_{r_j}}$, the model predicts the distribution parameters $\theta_1, \ldots, \theta_{N_{r_j}}$ for all segments of the route r_j.

The sequential probabilistic model, in contrast with the simple probabilistic model, produces the sequence of N_r predictions of parameters for each route. Similarly the negative log-likelihood can be used for optimisation.

3.3.5 Confidence intervals. The main benefit of the probabilistic models is that they can estimate not only the most likely energy use but also provide information about the uncertainty of the decision, which can be expressed as a confidence interval for some given confidence threshold. To understand how such intervals are computed, consider the properties of the sum of normally distributed random variables. Let $e_i \sim F_i(e|\theta) = \mathcal{N}(e|\mu, \sigma^2)$. Then the sum c of the random variables at a given segment N_{r_j} (consumed energy of the trip) have the distribution:

$$c_{N_{r_j}} = \sum_{i=1}^{N_{r_j}} e_i \sim \mathcal{N}\Big(\sum_{i=1}^{N_{r_j}} \mu_i, \sum_{i=1}^{N_{r_j}} \sigma_i^2\Big).$$

Thus, the $(1-\alpha)\cdot 100\%$ predicted confidence interval at segment N_{r_j} can be expressed as:

$$CI_{1-\alpha, N_{r_j}} = \Big(\sum_{i=1}^{N_{r_j}} \hat{\mu}_i - z_{1-\alpha/2}\sqrt{\sum_{i=1}^{N_{r_j}} \hat{\sigma}_i^2}, \sum_{i=1}^{N_{r_j}} \hat{\mu}_i + z_{1-\alpha/2}\sqrt{\sum_{i=1}^{N_{r_j}} \hat{\sigma}_i^2}\Big),$$

where $\hat{\mu}$ and $\hat{\sigma}$ represent the estimated parameters of the normal distribution of the energy use in the segment, and $z_{1-\alpha/2}$ is the α-level critical value of the normal distribution.

Figure 5 summarizes the inputs and the outputs of the discussed problem formulations. The number of input characteristics for each vertex is shown as 30 which corresponds to our main experimental study. Considering the outputs, note that both point-estimate and probabilistic problem formulations come in two versions: the accumulated and the segment-level.

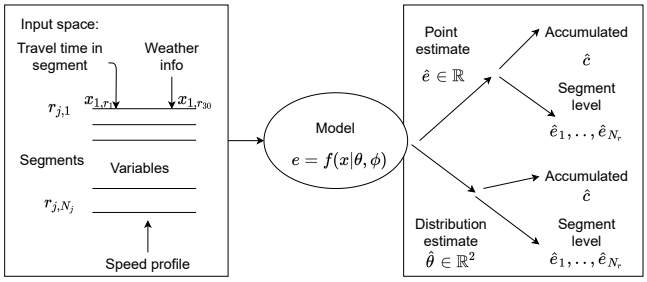

Figure 5: Inputs and outputs of the models.

4 DEEP-LEARNING MODELS

In the following, deep learning models are proposed to solve the different versions of ML problems formulated in the previous section. More specific technical implementation details are provided in Section 5.

4.1 Deep neural network

The deep neural network (DNN) implements an operator function $f_\phi : X \subset \mathbb{R}^{d_x} \to E \subset \mathbb{R}^{d_y}$ [5]. The composition of L non-linear transformations are applied in fully connected networks where $L-1$ are hidden layers and the last one is the output layer. The input can be defined as $X^0 = X \subset \mathbb{R}^{30}$. Then the l-th layer has n_l unknown parameters $\phi \subset \mathbb{R}^{n_{l-1}+1}$, receives an input vector $X^{l-1} \in \mathbb{R}^{n_{l-1}}$, and transforms it into vector $X^l \in \mathbb{R}^{n_l}$ by applying a linear transformation, followed by a component-wise (non-linear) activation or transformation function σ:

$$X^l = \sigma(\phi^l X^{l-1}),$$

where $\phi^l \in \mathbb{R}^{n_l \times n_{l-1}+1}$, $1 \le l \le M$ and X^l is the input for the $(n+1)$-th layer [5]. To construct DNNs, the batch normalization as non-linear transformation is employed [9].

4.2 Long short-term memory model

One of the most commonly used models for sequential-data problems is the LSTM [7] model which relies on the combination of the gate mechanism and the state updates. The state of LSTM is described as a pair of vectors (c_t, h_t) which is interpreted as long- and short-type memory, respectively. The time/sequence in our case is the segment order $t = 1, \ldots, N_r$. Getting the new input x_t, the LSTM updates h_{t-1} to new candidate memory $\tilde{c}_t$ and gate variables i_t, f_t, and o_t, which denote input, forget, and output gates, respectively. The input gate i_t controls the integration of candidate $\tilde{c}_t$ into c_t, allowing activation of certain gates. The forget gate f_t controls integration or previous memory c_{t-1}, and the output gate o_t transforms c_t into new hidden state h_t:

$$\begin{aligned}
i_t &= \sigma(\phi_{1,i} x_t + \phi_{2,i} h_{t-1} + b_i),\\
f_t &= \sigma(\phi_{1,f} x_t + \phi_{2,f} h_{t-1} + b_f),\\
o_t &= \sigma(\phi_{1,o} x_t + \phi_{2,o} h_{t-1} + b_o),\\
\tilde{c}_t &= \tanh(\phi_{1,c} x_t + \phi_{2,c} h_{t-1} + b_c),\\
c_t &= f_t \odot c_{t-1} + i_t \odot \tilde{c}_t,\\
h_t &= o_t \odot \tanh(c_t),
\end{aligned}$$

where $\odot$ denotes element-wise multiplication, σ denotes sigmoid, and tanh denotes hyperbolic tangent activations. We use two stacked LSTM modules, such that the inputs for the second LSTM model are the outputs of the first module: $x_t^{(2)} = h_t$. The output for the classification model is selected to be a fully connected layer $e_t = \phi_t o_t^{(2)} + b_t^{(2)}$. As in other models, the dimension of the output depends on whether we are predicting a point estimate or distribution parameters.

4.3 Other approaches

Apart from the models described above, we experimented with the Transformer NNs [29] which have become the state-of-the-art technique in natural language processing and, recently, in vision [4].

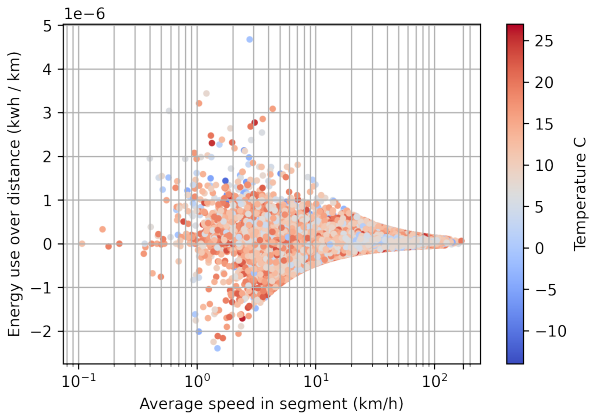

Figure 6: Normalised energy use for different average speeds and temperatures. Each point is a trip segment.

However, as our data is non-homogeneous, the positional encoder becomes meaningless. After discarding the positional encoder and adapting a vision transformer [4] to our problem, the experiments seem to indicate that the model is inferior to other, simpler models, thus we do not report these results and leave further investigation of the potential of these types of models as an interesting future research direction.

5 EXPERIMENTAL STUDY

5.1 Data

The primary EV dataset was collected in Denmark throughout 2012 [11] and involved tracking a fleet of 164 almost identical EVs. This study uses a version of the original data that is map matched to the OSM road network, and where the characteristics are aggregated per segment. The EV data is joined with the weather data from NOAA [20], as well as digital elevation data with a resolution of 1 arc-second [19] is used to calculate the altitude differences of the ends of segments. Finally, observations with missing values are removed. The dataset contains 442963 segment trips from 11908 unique trips. We split the data to 70/10/20(%) or 8335, 1191, 2382 trips for training, validation and testing datasets, or 305867, 47315, 89781 segments respectively.

Table 1 provides the main descriptive statistics of the data and Figure 6 illustrates the volatility of the energy used/gained, especially at lower speeds. This is explained by the fact that a lot of the short and slow trip segments in the data correspond to inner city stop-and-go traffic with many accelerations and decelerations. It is interesting whether the proposed models can capture this volatility.

Special attention has to be taken to prepare the speed-limit data used to generate the speed-limit speed profiles of routes. Multiple speed-limit values are missing in the OSM road-network data (zero values). To compensate, we first compute the median speed values for each OSM road type in the dataset. Then, each missing speed limit is replaced by the median speed of the corresponding road type.

Table 1: Descriptive statistics of the main characteristics of segment traversals

	Speed (ms)	Time (s)	Driven (m)	Air temp. C	Wind speed	Altitude diff (m)	E (kwh)
mean	48.64	12.46	168.01	8.49	4.83	-0.06	0.03
std	21.63	18.10	278.54	7.50	2.57	3.69	0.07
min	0.11	0.03	0.02	-16.00	0.00	-56.00	-1.48
25%	32.92	3.00	35.99	3.00	3.00	-1.00	0.00
50%	47.75	7.01	84.27	8.00	5.00	0.00	0.01
75%	61.93	15.02	181.40	14.00	7.00	1.00	0.04
max	169.08	1400.64	9626.21	27.00	16.00	43.00	4.48

After pre-processing, 30 trip-segment characteristics are used as the input for the models (see Table 2).

The EV data covers a wide range of scenarios from long trips to city routes (see the left map in Figure 7). Most importantly, it covers wide range of weather conditions $[(-16, 27)]^{\circ}$C and winds up to 16 m/s.

The second EV dataset is extracted from the Vehicle Energy Dataset (VED), a large-scale, open dataset of fuel and energy data collected in Ann Arbor, Michigan, USA [21]. The data was collected from Nov, 2017 to Nov, 2018 and captures GPS trajectories of vehicles along with the time-series data of their speed as well as fuel, energy, and auxiliary power use. Only a small portion of the data was used as only three fully electric vehicles are present in it and just one vehicle data with most data was used. The data is map-matched to OSM road network using the pgMapMatch package [17]. Only tracks with successful map matchings are chosen (match score of more than 70%). Tracks are split into sub-tracks if the time gap between two consecutive GPS points is longer than 17sec. Each data point is augmented with the weather data from NOAA [20]. As for the main dataset, digital elevation data [19] is

Table 2: The characteristics of a trip segment

Characteristic	Comment	# Var.
Speed	Speed profile data	1
Time (s)	Time of traversal	1
Air temperature	Weather info	1
Wind speed (ms)	Weather info	1
Altitude diff	Segment start and end altitude difference (m)	1
Travel time	The day-time category	1
Weekend	The weekend indicator	1
Road conditions	Indicator of 'drifting', 'dry', 'fog', 'freezing', 'none', 'snow', 'thunder'	8
Road type	Indicator of 'living_street' ' 'primary', 'residential', 'secondary' 'secondary_link', 'service', 'tertiary' 'track', 'trunk', 'trunk_link', 'motorway_link', 'unclassified', 'unpaved', 'wet', 'motorway'	14
	Total characteristics (d_x)	30

Table 3: The selected model architectures and the total number of unknown model parameters. The number of last layer outputs O is one for regression models and two for probabilistic.

	DNN *segment-level*	DNN *accumulated*	LSTM *segment-level*	LSTM *accumulated*
Layers from input (top) to output (bottom)	Dense (B, 16)	Dense (B, N_r, 16)	LSTM (B, N_r, 40)	LSTM (B, N_r, 40)
	BatchNormalization (B, 16)	BatchNormalization (B, N_r, 16)	LSTM (B, N_r, 40)	LSTM (B, N_r, 40)
	Dense (B, 64)	Dense (B, N_r, 64)	LSTM (B, N_r, 40)	LSTM (B, 40)
	BatchNormalization (B, 64)	BatchNormalization (B, N_r, 64)	Dense (B, N_r, O)	Dense (B, O)
	Dense (B, 64)	Dense (B, N_r, 64)		
	BatchNormalization (B, 64)	BatchNormalization (B, N_r, 64)		
	Dense (B, 256)	Flatten (B, 35200)		
	BatchNormalization (B, 256)	Dense (B, 256)		
	Dense (B, 8)	BatchNormalization (B, 256)		
	Dense (B, O)	Dense (B, 8)		
		Dense (B, O)		
Num. of parameters (MSE)	26049	9020865	37321	37321
Num. of parameters (ll)	26058	9020874	37362	37362

used to calculate the slopes of segments. We split the data to 300 trips for training and 182 trips for testing.

5.2 Software and hardware

Experiments are carried out using the high-level deep-learning framework Keras [3] version 2.4 with Tensorflow [1] version 2.4.1 as a back-end. Computations are performed on a computer with NVIDIA GTX 1080 Ti. The parameter optimisation is carried out using the Adam optimiser with learning rate 0.01 and batch size of 128. [1]

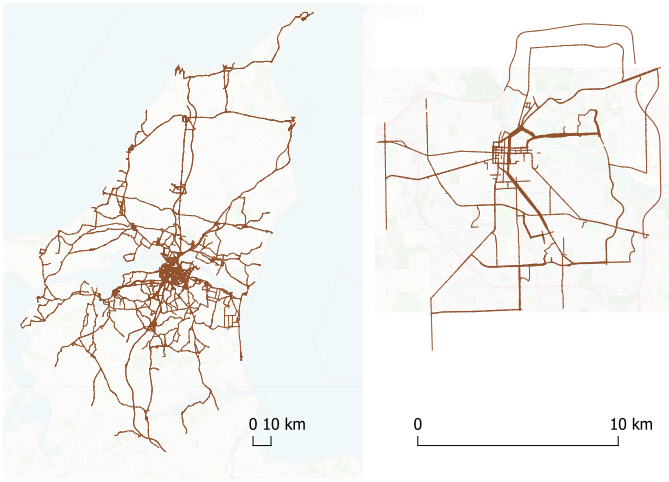

Figure 7: The EV data coverage. Line thickness represents a number of trip segments on a road segment

5.3 Models

Eight different neural networks are constructed combining the ML problem formulations from Section 3 and models from Section 4. The details of the architectures are provided in Table 3. The number of input covariates is relatively small $d_x = 30$. Thus, we try to avoid the dropout transformation since there are multiple critical covariates like speed or seconds traveled. Instead, we use the batch normalization layers [9], and set the batch size $B = 128$ in our experiments.

Each of the configurations listed in Table 3 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) as a loss function and the output dimension of two, corresponding to the two distribution parameters of the normal distribution.

For probabilistic models, the parameters of the distribution are $\theta = (\mu, \sigma)^T \subset \mathbb{R} \times (0, \infty) \subset \mathbb{R}^2$. Thus, variance σ must be positive at all times. To achieve this, we employ the Softplus activation [34] in connection with our loss function. This results in the positive but unbounded range for the variance parameter.

In order to have a unified data structure for all the different models, all of the trips with the length $N_{r_j} < 550$ are converted to a fixed size of $N_r = 550$ using zero padding as is usually done for LSTM models [18].

The LSTM models are constructed by stacking of the recurrent neural networks. Two stacking layers are used (see Table 3).

[1] The repository of the experiments can be found online at https://github.com/linas-p/EVDPEP (accessed on 01 July 2021).

Table 4: The statistics of investigated models on unseen test-data trips. The last three columns present the normalised statistics where energy use is divided by the traveled distance (kWh/km).

Model	Loss	Speed profile	*Acc.*	RMSE	MAPE	MAE	MSE	EVS	RMSE_n $\cdot 10^{-8}$	MAE_n $\cdot 10^{-8}$	MSE_n $\cdot 10^{-15}$
dnn	mse	*ideal*	0	0.50	0.57	0.17	0.25	0.87	3.47	1.61	1.20
dnn	mse	*ideal*	1	0.90	1.76	0.45	0.81	0.73	21.48	4.16	46.14
dnn	mse	*speed-limit*	0	0.55	0.52	0.17	0.31	0.83	3.12	1.59	0.97
dnn	mse	*speed-limit*	1	1.03	3.67	0.65	1.06	0.48	50.14	4.86	251.40
dnn	mse	*time-specific*	0	0.72	0.68	0.19	0.51	0.73	5.26	1.83	2.77
dnn	mse	*time-specific*	1	0.65	0.68	0.21	0.42	0.77	8.56	1.97	7.32
dnn	mse	*average*	0	0.56	0.62	0.18	0.31	0.84	3.84	1.77	1.48
dnn	mse	*average*	1	0.89	1.70	0.33	0.80	0.58	22.65	2.76	51.32
dnn	ll	*ideal*	0	1.10	1.45	0.45	1.21	0.51	8.70	4.83	7.57
dnn	ll	*time-specific*	0	0.96	1.26	0.38	0.92	0.58	7.64	3.96	5.83
dnn	ll	*average*	0	0.90	1.18	0.35	0.80	0.62	7.22	3.62	5.21
lstm	mse	*ideal*	0	0.76	1.41	0.26	0.57	0.68	15.90	2.79	25.28
lstm	mse	*ideal*	1	**0.54**	**0.52**	**0.16**	**0.29**	**0.84**	4.26	1.57	1.81
lstm	mse	*speed-limit*	0	0.77	1.80	0.29	0.59	0.67	23.02	2.67	52.98
lstm	mse	*speed-limit*	1	0.65	0.64	0.17	0.42	0.77	4.26	1.73	1.81
lstm	mse	*time-specific*	0	0.81	3.27	0.52	0.65	0.70	40.97	4.40	167.83
lstm	mse	*time-specific*	1	0.55	0.65	0.17	0.31	0.83	4.50	1.61	2.02
lstm	mse	*average*	0	0.77	0.68	0.20	0.59	0.69	6.12	2.02	3.75
lstm	mse	*average*	1	0.56	0.75	0.18	0.32	0.82	5.72	1.74	3.27
lstm	ll	*ideal*	0	0.50	0.83	0.18	0.25	0.86	6.45	1.68	4.15
lstm	ll	*ideal*	1	1.34	5.76	0.83	1.80	0.01	77.17	4.51	595.46
lstm	ll	*speed-limit*	0	**0.58**	0.98	**0.22**	**0.34**	0.82	13.25	2.10	17.55
lstm	ll	*speed-limit*	1	1.34	5.60	0.82	1.80	0.01	74.62	4.48	556.80
lstm	ll	*time-specific*	0	0.75	**0.91**	0.26	0.56	0.72	6.08	2.52	3.70
lstm	ll	*time-specific*	1	1.34	5.58	0.83	1.79	0.02	73.95	4.48	546.81
lstm	ll	*average*	0	0.78	2.94	0.53	0.60	0.77	41.38	3.76	171.20
lstm	ll	*average*	1	1.35	5.24	0.77	1.82	0.02	70.89	4.35	502.52

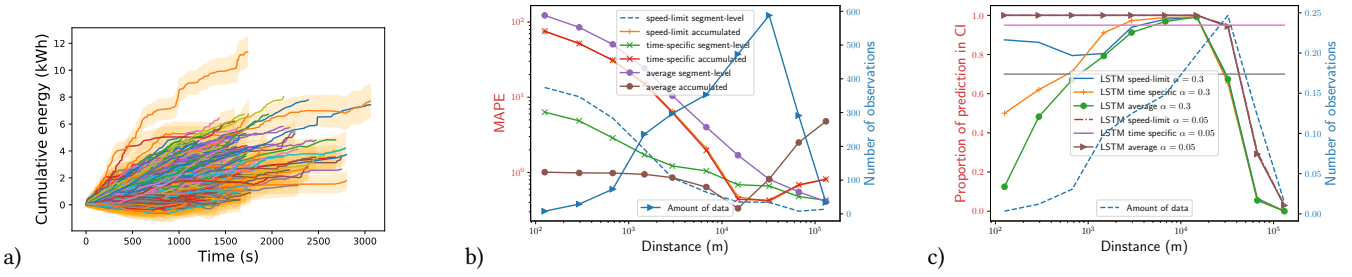

Figure 8: (a) Visualization pf 95%-confidence intervals ($\alpha = 0.05$) for various trips of the test set using the LSTM probabilistic model; (b) Error (in terms of MAPE) of predictions against the length of the trips; (c) Number of trips in the test set which fall in the predicted CIs at significance level $\alpha = 0.05$ and $\alpha = 0.3$ respectively.

Table 3 shows that a simple DNN *accumulated* model is rather complicated if the input data covers the hole trip information.

All of the models are trained for 50 epochs, as the learning curves reach stability at around 25 epochs.

5.4 Model comparison metrics

To have a broader view, a number of various metrics are used to compare the accuracy of the investigated models:

MSE: The mean square error, which is used as a loss function and is defined in Equation (2).

RMSE: The root mean square error or residuals.

MAPE: The mean absolute percentage error:

$$\mathcal{L}_{mape} = \frac{1}{M}\sum_{j=1}^{M}\frac{|e_j - \hat{e}_j|}{e_j},$$

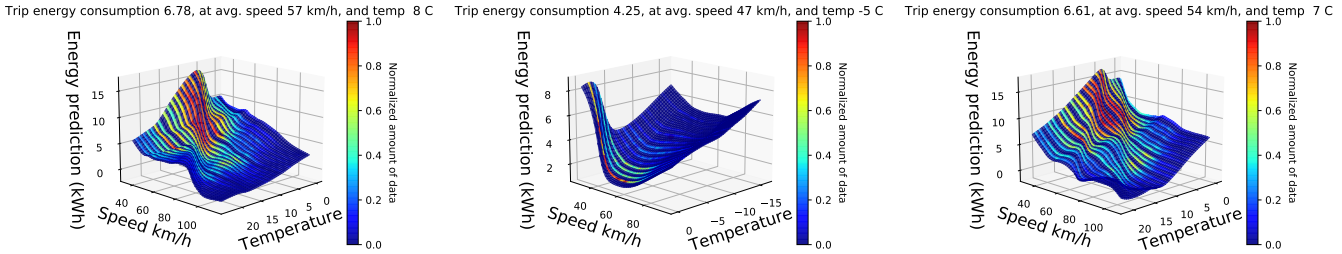

Figure 9: Predictions of LSTM probabilistic models when changing average speed in segments and air temperature, around the fixed value from real trip. Color indicates the amount of training trip segments with similar speed and temperature values.

MAE: The median absolute error.
EVS: The explained variance score:

$$\mathcal{L}_{evs} = 1 - \frac{Var(e_j - \hat{e}_j)}{Var(e_j)}$$

where Var is variance. All of the metrics are calculated using the scikit-learn library [26].

5.5 Model comparison results

First, we present the results of the investigated models on the primary dataset. The four probabilistic models and one DNN model trained with MSE is not included in the comparison to avoid distortion of the aggregated results. These models were not able to learn reasonable neural-network parameters and returned NaN (not-a-number) values or large errors similarly to the investigated Transformer models.

Table 4 shows that threre are some LSTM models that perform very well (for example, the accumulated model with the *ideal* speed profile), but on average simple DNN models perform up to 10% better than sequential LSTM models. However, it should be noted that DNN models are 240x larger than LSTM models (see Table 3).

The table also confirms that the *ideal* speed profiles provide the best models as expected see, for example, the thirteen line of the Table 3. On the other hand, we do not see the significant difference between the *time-specific* profiles and the *speed-limit* or *average* profiles based on RMSE, but *speed-limit* profile is significantly worse on MAPE metric. This can be best seen in the aggregated results shown in Table 5.

To summarize, the experiments show that the best results on realistic (not *ideal*) speed profiles are provided by the DNN model trained with point estimates using *time-specific* speed profiles and *segment-level* outputs. On the other hand, the best probabilistic models are learned using the LSTM-based architecture (see the highlighted lines at the bottom of Table 4).

The next set of experiments investigates the scalability of the models in terms of the length of the routes. Figure 8b) shows that the accuracy of predictions behaves as a linear dependency on the log-log scale. Surprisingly, the prediction error decreases as the length of the route increases. This can be explained by the fact that the training data includes more of such relatively long trips (the blue solid line in the graph). In addition, shorter trips mainly correspond to inner city driving with stop-and-go traffic, the energy consumption of which is harder to predict.

Table 5: Model accuracy aggregated according to the speed profile types

speed_type	RMSE	MAPE	MAE	MSE
ideal	0.755	1.809	0.342	0.660
average	0.830	1.874	0.363	0.750
time-specific	0.824	1.862	0.366	0.736
speed-limit	0.821	2.201	0.386	0.753

5.6 The benefits of probabilistic models

The results show that the models predicting point estimates provide slightly better accuracy than probabilistic models. Nevertheless, as mentioned in the introduction, the predicted level of uncertainty of prediction can be useful especially when planning charging on longer EV trips. In the following we explore the confidence intervals computed from the parameters returned by the probabilistic models (see Section 3.3.5). Figure 8a) visualizes such confidence intervals for the evaluation routes showing both the actual consumed energy and the predicted confidence interval as it changes along each of the routes.

To evaluate the generated confidence intervals, we investigate how often the recorded consumption falls in the predicted confidence interval (CI) varying the length of the route (see Figure 8c). The graph shows that the amount of training data has a significant impact on the CI predictions. Further, if a trip is short, small variations may lead to falling out of the CI. On the other end of the spectrum, the small number of long trips does not provide sufficient variability for the deep learning model to see enough cases during training.

5.7 Exploring the learned models

One of the benefits of having mathematical prediction models is the ability to model various unseen situations. Figure 9 shows how one of the learned models behaves around three of the trips from the dataset. For each graph, we start from a single trip (its characteristics and energy consumption are given at the top of

Table 6: The accuracy of the investigated models on the unseen test trips of the VED dataset

	Method	loss	Speed profile	RMSE	MAE
0	lstm	ll	*ideal*	10.35	2.11
1	lstm	ll	*speed-limit*	11.99	5.34
2	lstm	ll	*time-specific*	9.85	1.76
3	lstm	ll	*average*	10.44	1.97
4	lstm	mse	*ideal*	7.78	1.47
5	lstm	mse	*speed-limit*	11.46	3.13
6	lstm	mse	*time-specific*	9.72	2.75
7	lstm	mse	*average*	11.29	2.27
8	dnn	ll	*ideal*	49.37	26.64
9	dnn	ll	*speed-limit*	13.72	3.59
10	dnn	ll	*time-specific*	25.93	12.66
11	dnn	ll	*average*	11.92	1.52
12	dnn	mse	*ideal*	9.91	2.33
13	dnn	mse	*speed-limit*	13.20	5.13
14	dnn	mse	*time-specific*	11.51	4.82
15	dnn	mse	*average*	10.66	3.77

the graphs). Then, we change the speed characteristic of all the segments by scaling it in interval $[0.6, 2]$. Similarly, the temperature characteristic is scaled in interval $[0, 3]$). For visualisation of the amounts of trips in the dataset the speed and temperature space was divided into a uniform grid and the number of trips in each grid cell was normalised to unit interval dividing by the maximal value in the grid. The coloring of the graphs show that the dataset has only limited coverage in the generated speed and temperature ranges, but the model predicts non-linear change of energy consumption with respect to the decreasing temperature and high/low speeds. Rather smooth predictions indicate that the model is not overfitted.

5.8 Modeling the state of charge

To stress test the models we train them and evaluate on the very small second EV dataset (VED). In contrast to the Denmark dataset, this data does not explicitly contain information about the energy consumption. Instead, the data records the *state of charge* as an output (similarly to the experiments reported by T.Liu [13]).

Thus, in our models we replace the target e_j with $s_j = SOC_j - SOC_{j-1}$ as a difference of SOC on a segment of a trip. Table 6 reports the results of the experiments with the *segment-level* models.

In contrast to the experiments with the primary dataset, the LSTM models provide the best results. The variation in the accuracies of the methods is due to the very small size of the dataset covering a relatively large geographical area (see Figure 7). This, as well as the results in Figures 8b) and 8c) demonstrate the importance of the amount of training data.

6 CONCLUSION

Motivated by the spread of electrical vehicles, we propose a suite of deep-learning models for EV energy-use prediction. In particular, we propose not only models that predict a point estimate but also models that predict a probability distribution. As the speed is a very important energy-use predictor, we investigate different speed-profile types. The experiments show that the *time-specific* speed profile or even the data-agnostic *average* speed profile are preferable to the *speed-limit* speed profile. Further, doing forecasting at the segment level and summing later seems to be preferable to the so-called accumulated models.

The primary dataset used for this study is chiefly from urban driving and concerns a single model EV. It would be interesting to train the proposed models on wider datasets. While the transformer models seem to be promising, its current adaptation is not performing better than the simpler models. Further work and experiments are needed to understand the suitability of these types of models for the problem of EV energy-use forecasting.

ACKNOWLEDGMENTS

This work was supported by 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).

REFERENCES

[1] M. Abadi, A. Agarwal, P. Barham, E. Brevdo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[2] S. Ayyadi, M. Maaroufi, and S. M. Arif. 2020. EVs charging and discharging model consisted of EV users behaviour. In *REDEC*.

[3] François Chollet et al. 2015. Keras. https://keras.io.

[4] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, et al. 2020. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929* (2020).

[5] I. Goodfellow, Y. Bengio, A. Courville, and Y. Bengio. 2016. *Deep learning*. Vol. 1. MIT press Cambridge.

[6] X. Guo, T. Liu, B. Tang, X. Tang, J. Zhang, W. Tan, and S. Jin. 2020. Transfer Deep Reinforcement Learning-Enabled Energy Management Strategy for Hybrid Tracked Vehicle. *IEEE Access* 8 (2020).

[7] S. Hochreiter and J. Schmidhuber. 1997. Long short-term memory. *Neural computation* 9, 8 (1997).

[8] Jilin Hu, Bin Yang, Chenjuan Guo, Christian S Jensen, and Hui Xiong. 2020. Stochastic origin-destination matrix forecasting using dual-stage graph convolutional, recurrent neural networks. In *ICDE*. IEEE.

[9] S. Ioffe and C. Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*. PMLR.

[10] S. Keller, R. Gabriel, and J. Guth. 2020. Machine learning framework for the estimation of average speed in rural road networks with openstreetmap data. *ISPRS International Journal of Geo-Information* 9, 11 (2020).

[11] B. Krogh, 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*.

[12] J. Liao, T. Liu, W. Tan, S. Lu, and Y. Yang. 2020. Data-Driven Transferred Energy Management Strategy for Hybrid Electric Vehicles via Deep Reinforcement Learning. *arXiv preprint arXiv:2009.03289* (2020).

[13] T. Liu, X. Tang, X. Hu, W. Tan, and J. Zhang. 2020. Human-like Energy Management Based on Deep Reinforcement Learning and Historical Driving Experiences. *arXiv preprint arXiv:2007.10126* (2020).

[14] D. Luxen and C. Vetter. 2011. Real-time routing with OpenStreetMap data. In *ACM SIGSPATIAL international conference on advances in geographic information systems*.

[15] E. Martinsson. 2016. *Wtte-rnn: Weibull time to event recurrent neural network*. Ph.D. Dissertation. Chalmers University of Technology & University of Gothenburg.

[16] C. Miao, H. Liu, G. G. Zhu, and H. Chen. 2018. Connectivity-based optimization of vehicle route and speed for improved fuel economy. *Transportation Research Part C: Emerging Technologies* 91 (2018).

[17] A. Millard-Ball, 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 (2019).
[18] N. T. Nam and P. D. Hung. 2019. Padding methods in convolutional sequence model: an application in Japanese handwriting recognition. In *International Conference on Machine Learning and Soft Computing*.
[19] Earth Resources Observation and Science (EROS) Center. 2012. Shuttle Radar Topography Mission 1 Arc and 3 Arc-Second Digital Terrain Elevation Data - Void Filled. (2012).
[20] National Centers For Environmental Information. National Oceanic and Atmospheric Administration. 2021. Local Climatological Data. (2021). www.ncdc.noaa.gov Accessed: April 5, 2021.
[21] G. Oh, D. J. Leblanc, and H. Peng. 2020. Vehicle Energy Dataset (VED), A Large-Scale Dataset for Vehicle Energy Consumption Research. *IEEE Transactions on Intelligent Transportation Systems* (2020).
[22] Peter Ondruska and Ingmar Posner. 2014. Probabilistic attainability maps: Efficiently predicting driver-specific electric vehicle range. In *IEEE ITSS*. IEEE.
[23] OpenStreetMap Foundation. 2021. OpenStreetMap. (2021). \https://www.openstreetmap.org/ Accessed: April 5, 2021.
[24] A.S. Palau, K. Bakliwal, M. H. Dhada, T. Pearce, and A. K. Parlikad. 2018. Recurrent neural networks for real-time distributed collaborative prognostics. In *ICPHM*.
[25] K. Park, S. Yoon, and E. Hwang. 2019. Flexible charging coordination for plug-in electric vehicles based on uniform stochastic charging demand and time-of-use tariff. In *ITEC*.
[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research* 12 (2011), 2825–2830.
[27] Y. M. Saputra, D. T. Hoang, D. N. Nguyen, E. Dutkiewicz, M. D. Mueck, and S. Srikanteswara. 2019. Energy demand prediction with federated learning for electric vehicle networks. In *GLOBECOM*.
[28] I. Sutskever, O. Vinyals, and Q. V. Le. 2014. Sequence to sequence learning with neural networks. *arXiv preprint arXiv:1409.3215* (2014).
[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin. 2017. Attention is All you Need. In *NIPS*.
[30] Y. Xiong, B. Wang, C. Chu, and R. Gadh. 2018. Electric vehicle driver clustering using statistical model and machine learning. In *PESGM*.
[31] Rose Yu, Yaguang Li, Cyrus Shahabi, Ugur Demiryurek, and Yan Liu. 2017. Deep learning: A generic approach for extreme condition traffic forecasting. In *SIAM*. SIAM.
[32] Y. Yuan, L. Lei, T. X. Vu, S. Chatzinotas, S. Sun, and B. Ottersten. 2020. Energy minimization in UAV-aided networks: actor-critic learning for constrained scheduling optimization. *arXiv preprint arXiv:2006.13610* (2020).
[33] L. Zhang, W. Liu, and B. Qi. 2020. Combined Prediction for Vehicle Speed with Fixed Route. *Chinese Journal of Mechanical Engineering* 33, 1 (2020).
[34] H. Zheng, Z. Yang, W. Liu, J. Liang, and Y. Li. 2015. Improving deep neural networks using softplus units. In *IJCNN*.