

Fine-grained RNN with Transfer Learning for Energy Consumption Estimation on EVs

Yining Hua*, Michele Sevegnani, Dewei Yi, Andrew Birnie, and Steve McAslan

Abstract—Electric vehicles (EVs) are increasingly becoming an environmentally-friendly option in current transportation systems thanks to reduced fossil fuel consumption and carbon emission. However, the more widespread adoption of EVs has been hampered by two factors: the lack of charging infrastructure and the limited cruising range. Energy consumption estimation is crucial to address these challenges as it provides the foundations to enhance charging-station deployment, improve eco-driving behaviour, and extend the EV cruising range. We propose an EV energy consumption estimation method capable of achieving accurate estimation despite insufficient EV data and ragged driving trajectories. It consists of three distinct features: knowledge transfer from Internal Combustion Engine/Hybrid Electric Vehicles (ICE/HEV) to EVs, segmentation-aided trajectory granularity, time-series estimation based on bi-directional recurrent neural network. Experimental evaluation shows our method outperforms other machine learning benchmark methods in estimating energy consumption on a real-world vehicle energy dataset.

Index Terms—electric vehicle, energy consumption estimation, trajectory segmentation, transfer learning, recurrent neural network.

I. INTRODUCTION

The rapid depletion of conventional energy resources, massive increase of air pollution, and accelerating climate change are some of the main drivers for the ongoing transition to renewable, cleaner energy sources to replace traditional fossil fuels. Several studies [1], [2] have identified the transport sector as one of the primary contributors to both total energy consumption and carbon emissions, and, therefore, policymakers, industry, and researchers worldwide are motivated to develop new and more efficient transportation systems relying on green energy. A promising development in this direction has been the promotion of a gradual replacement of internal combustion engines vehicles (ICEs) and hybrid electric vehicles (HEVs) with electric vehicles (EVs) [3].

Although EVs have been heavily supported by government agencies in terms of both funding and policy, the popularisation of EVs is still challenged by the perceived limitations

of the all-electric range (AER) [4]. To extend AER, concerted research efforts are required in areas such as charging infrastructure deployments [5], eco-routing-planning before trips, and eco-driving-coaching during trips [6]. These research challenges have been identified to have energy consumption estimation as a common foundation study [7], [8]. Bozorgi et al. [9] showed that the accurate estimation of energy consumption can help locate charging stations and provide essential information under specific weather and traffic condition for the route-planning and driving-coach processes in each drive cycle.

In the past decade, various methods have been attempted to estimate the energy consumption for HEVs and EVs. As summarised in [10], the conventional energy estimation methods can be categorised into two paths. The first path estimates the energy consumption based on external factors, such as weather conditions, road grade, etc [11]. Instead of the environmental factors, the estimation methods in the second category are focusing on the internal factors, such as kinetic/EV power systems [12]. Based on the state-of-the-art research, the key factors of EV energy consumption estimation includes both internal and external factors. Thus, in more recent work such as [13], research proposed real-time energy estimation methods based on hybrid factors e.g., vehicle speed, tractive effort and road elevation.

However, there are still two open issues in the literature. First, compared with ICE/HEV data, the energy consumption data for pure EV is still limited [11], and without large-scale training data, it is difficult to train efficient deep learning models. Second, in practical scenarios, the length of driving trajectories may vary [14] and therefore it becomes difficult to define adaptive models capable of providing accurate estimation results based on ragged trajectory input. To solve these two problems, we propose a segmentation-aided transfer learning method to estimate the energy consumption of EVs. It consists of the following three features. First, to tackle the insufficient data problem in EV application, our method uses ICE and HEV data to train the model, and adapt the learnt knowledge to EV data by deep transfer learning method. Second, to achieve an adaptive model efficient at various trajectory lengths, our method segments driving trajectories into a fine-grained level depending on both GPS and dynamic information of EVs, where the segmentation process is operated in a purely unsupervised manner by Gaussian mixture models (GMMs). Third, to extract spatio-temporal information, a deep bi-directional recurrent neural network (BiRNN) is applied to the segmented fine-grained driving trajectories.

This work is supported by the Engineering and Physical Sciences Research Council, under PETRAS SRF grant MAGIC (EP/S035362/1) and the University of Glasgow Impact Acceleration Account. (*Corresponding author: Yining Hua)

Yining Hua is with School of Computer Science, University of Lincoln, Lincoln, U.K. (email: yhua@lincoln.ac.uk), and School of Computing Science, University of Glasgow, U.K. (email: yining.hua@glasgow.ac.uk). Michele Sevegnani is with School of Computing Science, University of Glasgow, Glasgow, U.K. (email: Michele.Sevegnani@glasgow.ac.uk) Dewei Yi is with Department of Computing Science, University of Aberdeen, Aberdeen, U.K. (email: dewei.yi@abdn.ac.uk) Andrew Birnie and Steve McAslan are with NXP Laboratories UK Ltd., Glasgow, U.K. (email: {andrew.birnie, steve.mcaslan}@nxp.com)

The key contributions of this paper are summarised as follows.

- To the authors' best knowledge, this is the first attempt to transfer the knowledge learnt by a deep BiRNN from ICE/HEV data (i.e., source domain) to EV data (i.e., target domain).
- A fine-grained trajectory segmentation method is proposed to achieve an adaptive estimation model for various lengths and shapes of driving trajectories.
- To evaluate the performance of our proposed method, comprehensive experiments are conducted on a recently collected real-world vehicle energy dataset. The experimental results demonstrate the superiority of our proposed method alongside 21 other competing benchmark methods.

II. METHODOLOGY

To achieve accurate estimation performance, our proposed method has three technical novelties, 1) trajectory segmentation, 2) Bi-directional RNN and 3) transfer learning. The framework design is as illustrated in Fig. 1.

Before the training stage, time-series feature values extracted from the raw route data are clustered by Gaussian mixture model (GMM), where the number of clusters is determined by Bayesian information criterion (BIC) test. After the clustering, the data from various-length routes are segmented into fine-grained trajectories, and ready to be used for model training. At the training stage, an advanced recurrent neural network, Bi-directional long short-term memory (BiLSTM) is used as neural network architecture. By providing both forward and backward temporal dependencies, BiLSTM can fully exploit the time-series information of the data. Moreover, to overcome the insufficiency problem of EV data, the model is first pre-trained by ICE and HEV data, and then transfer learning is utilised to transfer the learnt knowledge from ICE and HEV data to the EV domain. After the model is well trained, with the new-coming route data, our estimation model is able to output the predicted energy consumption with low error.

The detailed design of trajectory segmentation, BiLSTM and transfer learning is introduced in the following subsections, respectively.

A. Trajectory Segmentation

In the energy consumption estimation scenario, the lengths of different trajectories vary a lot. When the trajectories are sampled every 100 milliseconds as in our experimental dataset, the longest trajectory consists of 4915 sampling points, while the shortest trajectory has only 151 sampling points. If these raw trajectories are directly used in estimate model, the wide variety of data may affect the efficiency of spatio-temporal feature extraction, which leads to adaptability reduction of the estimation model. Thus, to obtain an adaptive model for various trips input, trajectory segmentation is used in both the training and the estimation processes. With effective trajectory segmentation, the estimation model can better capture the

robust, domain-invariant spatio-temporal information extracted from the trajectory data.

Regarding each trip trajectory, the optimal number of segments depends on its characteristics in both spatial (e.g., road curve, speed limit, etc.) and temporal (e.g., vehicle dynamic information, etc.) aspects. Thus, the segmentation process of different trip trajectories needs to be discriminated. In our proposed method, the optimal number of segments for each trajectory is determined by Bayesian Information Criterion (BIC) [15], which will be further illustrated in Section III-B1.

Since segmenting the trajectories manually is a labour-intensive work, GMM [16] is used to segment trajectories in an automatic way. Let $T = \{s_1, s_2, \dots, s_k\}$ be an example of trip trajectory, where T is a D -dimensional matrix and k represents the number of segments for this trajectory determined by BIC. Since there are four features taken into account in our proposed method, here $D = 4$. The four dimensions of trajectory T respectively represent feature latitude X , longitude Y , velocity V , and Celsius temperature C . In this way, the segmentation of trajectory can simultaneously take the information from spatial, kinematic, and environmental aspects into account.

Each trajectory segment s_i ($i \in [1, k]$) can be treated as a Gaussian component (a.k.a. cluster), which can be mathematically represented by a parametric distribution. For our GMM-based trajectory segmentation, the parametric distribution is denoted as $g_i(s_i|\mu_i, \Sigma_i)$, which represents a four-variate Gaussian density function as defined in Equation (1),

$$g_i(s_i|\mu_i, \Sigma_i) = \frac{1}{\sqrt{2\pi}\Sigma} e^{-\frac{1}{2}(s_i - \mu_i)^T \Sigma_i^{-1} (s_i - \mu_i)} \quad (1)$$

where μ_i and Σ_i respectively represents the mean vector and co-variance matrix of the four features (i.e., latitude, longitude, velocity, and out-vehicle temperature). The entire trajectory of T can be modelled as a mixture of weighted Gaussian distributions (Equation (2)).

$$P(T|\Theta) = \sum_{i=1}^k f(\theta_i) = \sum_{i=1}^k w_i g_i(s_i|\mu_i, \Sigma_i) \quad (2)$$

Here w_i represents its mixture weight, which satisfies the constraint $\sum_{i=1}^k w_i = 1$. For each trajectory segment s_i , we have parameter-set $\theta_i = \{w_i, \mu_i, \Sigma_i\}$.

To maximise the likelihood P of GMM, the optimal parameter-set $\Theta = \{\theta_1, \theta_2, \dots, \theta_k\}$ is obtained according to Algorithm 1, in which Θ is firstly initialised by k-means++ clustering method [17] (line 1), and then optimised by expectation-maximisation (EM) method [18] (line 4-9).

In this way, the GMM can be finally established by parameterising Θ , which consist of the mean vectors μ , co-variance matrices Σ , and mixture weights w of all component densities $i \in [1, k]$.

B. Bi-directional Recurrent Neural Network

After the trajectory segmentation, the input of the neural network is well-prepared and ready to be used for the purpose of training. Since the sensor data of our EV application are of a sequential nature, recurrent neural networks (RNNs) are utilised to train the estimation model since they have the ability to capture temporal dependency.

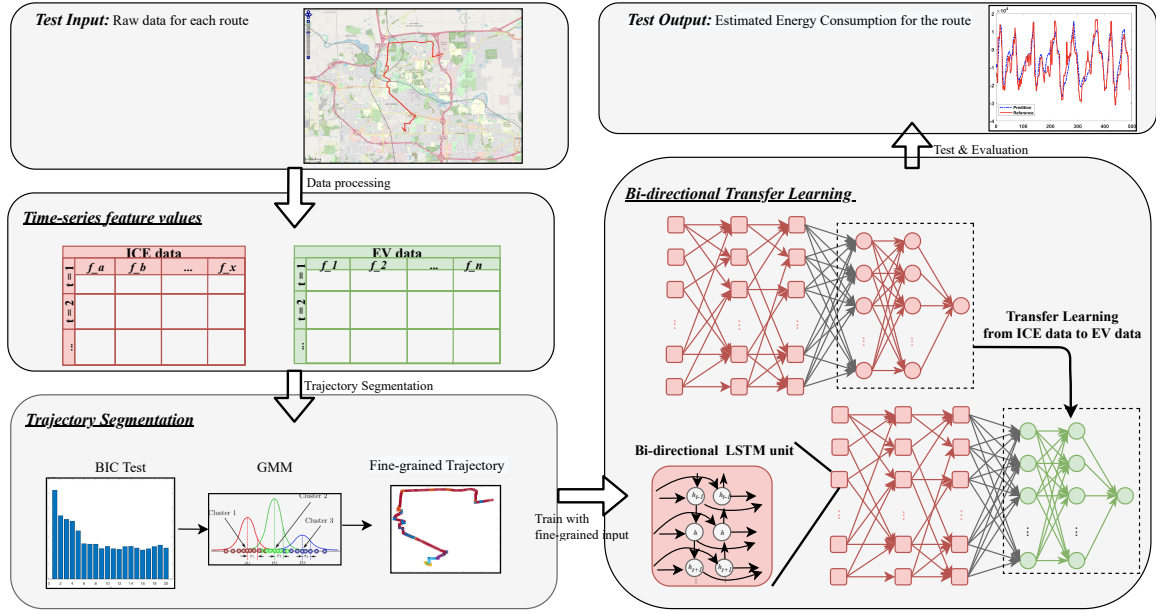


Fig. 1. The trip trajectory is input as the raw data, in which the values of four features (i.e longitude, latitude, temperature, and real-time speed) are collected with time stamped. After the data pre-processing, the training data is well prepared, consisting of the feature values and the corresponding measured energy consumption values. To deal with the trajectories with extremely uneven lengths, trajectory segmentation based on GMM is operated to splice the original trajectory into the optimal numbers of sub-trajectories depending on BIC test. After then, BiLSTM networks are initially trained by sufficient ICE data, and then the fully-connected layers are re-initialised to adjust the pre-trained model to fit in the EV data. Once the model is well trained, with the trajectory input, the energy consumption of EV can be estimated.

Algorithm 1 Trajectory Segmentation Algorithm

Require: Given the number of segments k , $k \in \{1, 2, \dots, N\}$, initial mixture parameters $\Theta^{(0)}$, and convergence threshold ϵ ;

- 1: k-means++ initialisation: $\Theta^{(t)} = \Theta^{(0)}$
- 2: **for** $k = 1 : N$ **do**
- 3: **for** $t = 1 : +\infty$ **do**
- 4: **E-step:** Compute the posteriori probability
- 5: $\hat{\gamma}_{ik} = p(z_i = k | b_i, \Theta^{(t)}) = \frac{w_i g(x_i | \mu_k, \Sigma_k)}{\sum_{j=1}^M w_j g(x_i | \mu_j, \Sigma_j)}$
- 6: **M-step:** Maximisation of the likelihood
- 7: $\hat{w}_k = \frac{1}{n} \sum_{i=1}^n \hat{\gamma}_{ik}$, $\hat{\mu}_k = \frac{\sum_{i=1}^n \hat{\gamma}_{ik} x_i}{\sum_{i=1}^n \hat{\gamma}_{ik}}$
- 8: $\hat{\Sigma}_k = \frac{\sum_{i=1}^n \hat{\gamma}_{ik} (x_i - \hat{\mu}_k)^T (x_i - \hat{\mu}_k)}{\sum_{i=1}^n \hat{\gamma}_{ik}}$
- 9: $\Theta^{(t)} = \{\hat{w}_k, \hat{\mu}_k, \hat{\Sigma}_k\}$
- 10: **if** $L(B|\Theta^{(t)}) - L(B|\Theta^{(t-1)}) \leq \epsilon$ **then**
- 11: Set mixture parameters $\Theta = \Theta^{(t)}$
- 12: break EM process
- 13: Calculate $BIC(k)$ of $GMM(k)$
- 14: **if** $BIC(k) < BIC(k^*)$ **then**
- 15: Set optimal component number $k^* = k$
- 16: **Ensure** obtained the optimal parameters for k^* and Θ

In RNN, through its cyclic connection feature, the output of each recurrent neuron y_t at time t are not only affected by input x_t , but also influenced by the previous hidden state h_{t-1} , where $h_t = \sigma(W_h h_{t-1} + W_x x_t + b)$. However, gradient vanishing is a frequently encountered issue in conventional RNN architectures, especially when the input has long sequences. It is because conventional RNN algorithms only focus on short-term memory, and long-term memory is easily forgotten. Thus, to avoid these issues, our proposed method trains the model with a special type of RNN, long short-term memory

(LSTM) [19].

In LSTM, hidden neurons in traditional RNNs are replaced by memory cells. For each LSTM memory cell at time t , there are three inputs (c_{t-1}, h_{t-1}, x_t) and three outputs (c_t, h_t, o_t). In our scenario, x_t refers to the values of EV features (i.e. GPS position, speed, and out-vehicle temperature) at time t , while h_{t-1} represents the prediction at time $t-1$, with has the same definition as in traditional RNNs. Besides hidden states and current input (output) values, each LSTM cell has a distinct element in both input and output, cell state c , which carries the cumulative information of the sequence data from each step, and jointly controlled by the three gates, i.e. an input gate i , a forget gate f , and an output gate o . The values of these three gates are in range 0 to 1. With these three gate functions, LSTM can well alleviate the gradient vanishing or exploding problems by regulating the flow of information into and out of the cell [20].

In our application scenario, after the real-time GPS coordinate, speeds, and out-vehicle temperatures at time index t are input as x_t , the forget gate value f_t is calculated by Equation (3)

$$f_t = \sigma(W_f \odot [h_{t-1}, x_t] + b_f) \quad (3)$$

where σ is the logistic function, h_{t-1} is the previous state of the hidden layers at time index $(t-1)$. W_f and b_f are the weight matrix and the bias of forget gate neurons, respectively. Similarly, the value of input gate is then computed by Equation (4)

$$i_t = \sigma(W_i \odot [h_{t-1}, x_t] + b_i) \quad (4)$$

where W_i and b_i are the weight matrix and the bias of input gate neurons, respectively. With the value of forget gate f_t

and input gate i_t , the estimated cell state \tilde{c}_t can be calculated by Equation (5), which is a vector of new candidate values created by a tanh layer and b_c is the bias of memory cell.

$$\tilde{c}_t = \tanh(W_c \odot [h_{t-1} + b_c]) \quad (5)$$

The final cell state c_t can then be derived from previous cell state c_{t-1} , current forget gate value f_t , input gate value i_t , and estimated cell state \tilde{c}_t , as shown in Equation (6).

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \quad (6)$$

After obtaining the final cell state, the value of output gate is calculated as shown in Equation (7)

$$o_t = \sigma(W_o \odot [h_{t-1} + b_o]) \quad (7)$$

where W_o and b_o are the weight matrix and the bias of output gate neurons, respectively. Finally, the hidden state h_t (i.e., the predicted energy consumption of EV at current time index t) can be derived by output gate value o_t and memory cell state c_t .

$$h_t = o_t \odot \tanh(c_t) \quad (8)$$

Moreover, when the collected data is throughout a long period of time and the driving trajectories have overlaps, the driving pattern of similar routes may be close to each other. In this case, the forward and backward driving temporal dependencies may equally contribute to the prediction performance. Therefore, we adopt BiLSTM, which is an advanced LSTM architecture, to incorporate both directions of driving information. Similar to the LSTM network, each direction of BiLSTM network also contains three gates and one memory cell. The forward and the backward LSTM layer are propagated in parallel, and the temporal features of the time-series data are memorised from both directions. To avoid missing useful information when extracting temporal features, the output hidden states of BiLSTM $H = (h_f, h_b)$ are computed by concatenating the output hidden states of both forward h_f and backward LSTM h_b , as shown in Equation (9)

$$\begin{aligned} h_f &= o_f \odot \tanh(c_f) \\ h_b &= o_b \odot \tanh(c_b) \end{aligned} \quad (9)$$

C. Transfer Learning

To obtain a well-trained model, massive data should be learnt iteratively by the neural network. Its internal functioning mechanism is that a large number of weights are randomly initialised first, and then the weights are iteratively updated according to the labelled data. Deep neural network means the architecture has more than one hidden layers. The deeper a network is, the larger number of weights contained, which requires more iterations to let the weight-values converge. Thus, obtaining a well-trained model from scratch requires massive labelled data, and the training process is extremely time-consuming. If the data provided are insufficient, deep neural network may experience an overfitting problem, which will seriously affect the accuracy of the estimation model.

As mentioned in the introduction section, the data insufficiency issue is one of the key challenges in EV research. To avoid the overfitting problem and save training time, we use transfer learning (TL) at our training stage, which enables

the neural networks trained by the ICE/HEV data first, and then transfer the learnt knowledge to the EV domain. To be specific, the idea of TL is to transfer knowledge from one domain (source domain D_s) to another related domain (target domain D_t) [21].

In our energy estimation scenario, the source domain is the energy consumption estimation of ICE/HEV, and the target domain is the estimation of EV. The transferred knowledge is the in-direct features extracted from the given input (GPS position, speed and out-vehicle temperature). After pre-training the neural networks through the source domain data, the common inner layers on the input side can be directly used in the final model for the target EV domain, and the custom final layers on the output side need to be further trained by a relatively small amount of data from target EV domain. In this way, even though the EV data are not sufficient to train an estimation model from scratch, the weights in the pre-trained deep neural network model can converge after enough iterations without an overfitting problem.

III. EXPERIMENTAL EVALUATION

In this section, the experimental evaluation of our proposed method is elaborated from four aspects, i.e. 1) data pre-processing and dataset construction, 2) implementation details and evaluation metrics 3) performance evaluation compared with benchmarks, and 4) ablation study to show how each technical feature can improve the estimation accuracy.

A. Dataset

1) *Data Pre-processing*: In this paper, VED dataset [14] is used for experimental evaluation purpose. The data provide the real-time information such as GPS position, out-vehicle temperature, speed, which is collected from EV, ICE, HEV. However, the energy consumption rate of vehicles, i.e. mile-per-gallon (MPG) and power (Power = Voltage \times Current) are not directly provided in the raw data. Thus, to train the energy estimation model, the MPG and power information need to be calculated in advance from existing data.

For EV and plug-in HEVs, instantaneous electric energy consumption rate can be obtained by multiplying the instantaneous voltage and current values of their batteries. For ICE and HEVs, the rate of fuel consumption (FC) is obtained as in [14]. To be specific, when FC rate cannot be directly estimated according to the information from original equipment manufacturer, it can be estimated based on mass air flow (MAF) by introducing a correction factor *correction*. Under various driving conditions, engines require different air/fuel ratio (AFR). To maintain a balanced AFR value, fuel trims, including short-term and long-term ones, are used for adjusting the fuel metering based on the oxygen sensor. As shown in Equation (10), the correction factor *correction* is defined based on the instantaneous long-term fuel trim value FT_L (%), short-term fuel trim value FT_S (%), and AFR value AFR (%).

$$correction = \frac{(1 + FT_S/100 + FT_L/100)}{AFR} \quad (10)$$

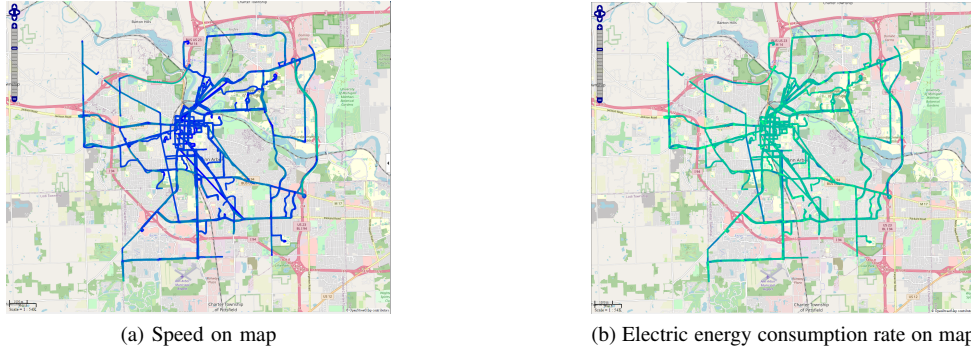


Fig. 2. The data collected by EV: (a) the collected driving trajectories along with corresponding speed are plotted on street map, where the darker colour of blue indicates positions with lower speed; (b) the collected driving trajectories along with corresponding instantaneous energy consumption are plotted on street map, where the darker colour of green indicates positions with lower energy consumption.

With given MAF and correction factor, the estimated FC can be computed by multiplying MAF and correction factor (Fuel Consumption rate = $MAF \times correction$). While if MAF is also not available, a set of engine signals are required to estimate MAF first, including absolute engine load $|EL|$ (%), air density ρ_{air} (kg/m^3), engine cylinder displacement D_{eng} (L), and engine revolution per minute RPM_{eng} (rev/min). Equation (11) shows how MAF can be estimated.

$$MAF = \frac{|EL|}{100} \times \rho_{air} \times D_{eng} \times \frac{RPM_{eng}}{120} \quad (11)$$

2) *Dataset Construction*: After the data pre-processing, the VED dataset is divided into two domains, 1) source domain with ICE & HEV data, and 2) target domain with EV data. In the source domain, there are the data of 4504 trips collected from 264 ICE vehicles, 93 HEVs, and 24 plug-in hybrid electric vehicles (PHEVs), in which 2700 trips are used for training and 1804 ones are for testing. In the target domain, there are 495 driving trips collected from EV, in which 343 are used for training and 152 for testing. The collected trajectories are provided in Fig. 2 along with their corresponding speed and instantaneous energy consumption.

B. Implementation and Evaluation Metrics

1) *Implementation details*: To evaluate the estimation performance, the benchmark methods and our proposed method are implemented in MATLAB 2019a. All non-deep learning methods, including linear regression (LR), tree regression, ensemble learning, support vector machines (SVM), Gaussian process regression (GPR), and their variants, are implemented and evaluated as benchmarks on a CPU (Core i7 at 2.50 GHz with 16 GB of RAM). The deep learning methods, LSTM, BiLSTM, and our proposed method are implemented on a GPU (GeForce RTX2080ti). The baseline BiLSTM model has 15 layers, including one sequence input layer to reshape varying input sizes, 3 BiLSTM layers, 6 dropout layers, 4 fully connected layers, and one regression layer as output. Each BiLSTM layer has 100 hidden units, followed by a dropout layer with dropout rate set to 0.2. Each fully connected layer has 50 hidden units, followed by a dropout layer with dropout rate set to 0.5.

Moreover, to implement our proposed method, trajectory segmentation and transfer learning are integrated on the basis of baseline BiLSTM. For trajectory segmentation, since the length of driving trips are variable, the optimal number of segments for different trajectories may vary as well. To determine the optimal number of segments for every given trajectory, BIC is introduced as an effective measure to assess the quality of segmentation. According to Bayesian theory, the most appropriate model has the smallest BIC value. The BIC value is defined in Equation (12).

$$BIC = -2 \log L(\hat{\theta}) + P \log(N) \quad (12)$$

where $L(\hat{\theta})$ is the log-likelihood associated with parameter estimates of Gaussian mixture models. N is the number of samples and P is the number of parameters.

For each given driving trajectory, the BIC values of different cluster numbers k can be calculated, and the minimum BIC value refers to the optimal number of segments for this trajectory. To find the range of BIC test, the longest driving trajectory in the dataset has been tested from $k = 1$ to $k = 80$. In order words, we have calculated the BIC values when the trajectory is treated as a whole, or split into 2, 3, ..., 80 segments. The result refers to the optimal number of segments for the longest trajectory is 16. Thus, with the consideration of uncertainty, we set the BIC test range from 1 to 20. Then, for all the trips in the dataset, their BIC values with $k \in [1, 20]$ are calculated. The BIC test and segmentation results of some examples trips are provided in Fig. 3. The first column presents the trajectories in a street map, and the second column presents the values of BIC with $k \in [1, 20]$.

After the optimal cluster number k is determined, EM is iterated to optimise the likelihood of GMM. When the difference between the likelihoods of two successive iterations is within 10^{-8} , the model is treated as converged. At this stage, the raw trajectories are segmented into fine-grained ones and ready to be input into the BiLSTM model. The examples of segmentation results are illustrated in the third column of Fig. 3. At the transfer learning stage, after the BiLSTM model has been pre-trained by the source domain data, the last fully connected layer and the regression layer are re-initialised, and then trained by the EV data from the target domain.

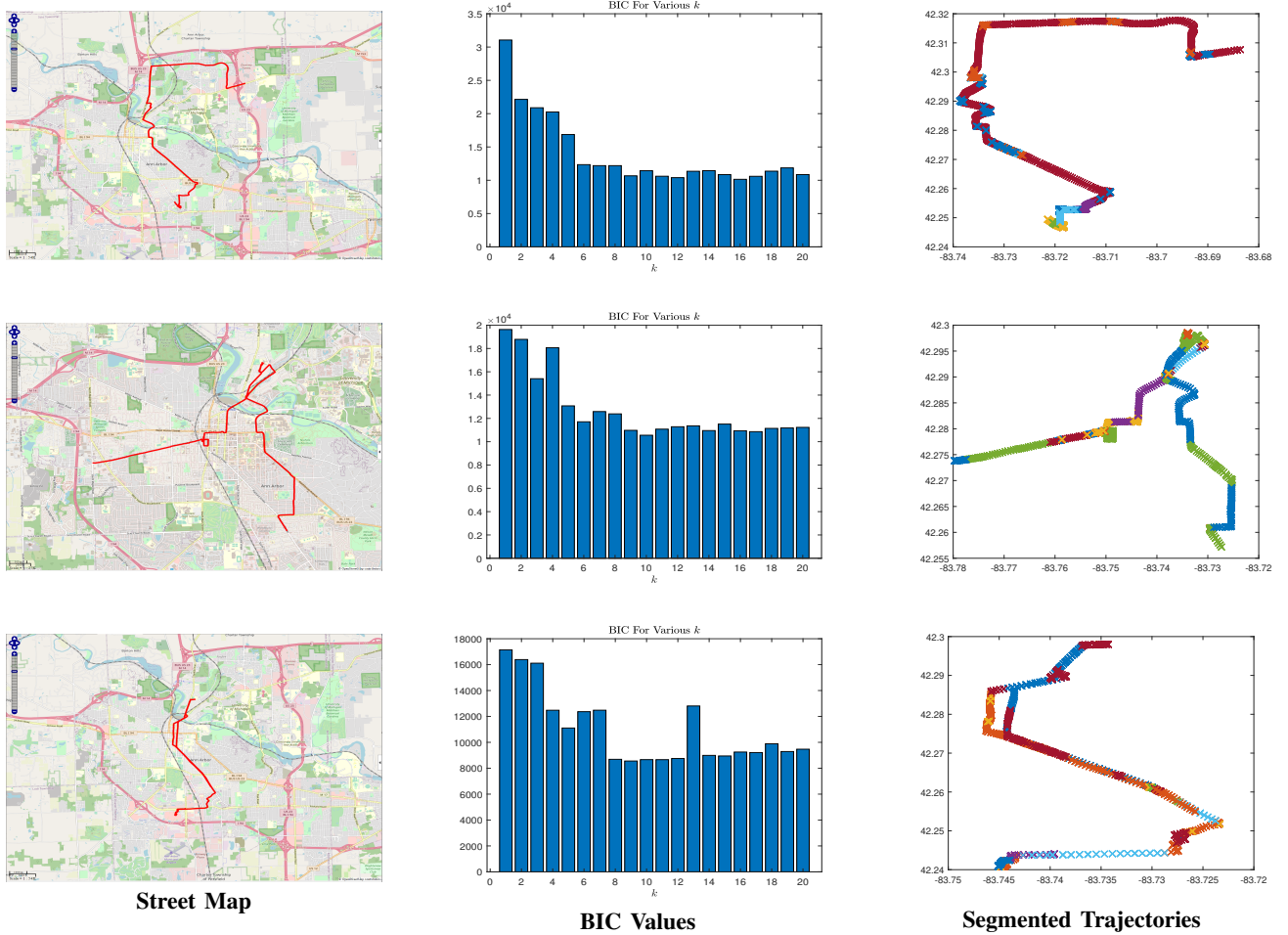


Fig. 3. For the example trajectories, the first column is their visualisation on the street map. The second column shows their corresponding BIC values when the trajectory is segmented into the range from 1 to 20 clusters. The third column illustrates the optimal segmentation results of the example trajectories.

2) *Evaluation metrics*: In this paper, four metrics are selected to evaluate the estimation accuracy of our proposed method, which are mean squared error (MSE), root mean squared error (RMSE), mean absolute error (MAE), and symmetric mean absolute percentage error (sMAPE). Their definitions are given as follows.

$$MSE = \frac{1}{n} \sum_{i=1}^n (E_m - E_i)^2 \quad (13)$$

$$RMSE = \sqrt{\frac{1}{n} \sum_{i=1}^n (E_m - E_i)^2} \quad (14)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |E_m - E_i| \quad (15)$$

$$sMAPE = \frac{1}{n} \sum_{i=1}^n \frac{2|E_m - E_i|}{|E_m| + |E_i|} \times 100\% \quad (16)$$

where n is the number of tested data, E_m and E_i are the measured and estimated energy consumption rate, respectively.

C. Performance Analysis

In this section, comparative experiments are conducted to evaluate the performance of various regression methods. Regression methods are used to build up the relationship between

features (i.e. latitude, longitude, velocity, and out-vehicle temperature) and instantaneous energy consumption so that instantaneous energy consumption can be worked out by using above the mentioned features. The performance of regression methods is evaluated by MSE, RMSE, MAE, and sMAP concurrently, which are defined in Section III-B2. Various regression methods are compared, including linear regression (LR) based methods, tree regression based methods, support vector machine based methods, ensemble learning methods, Gaussian process regression based methods, recurrent neural network based methods, and our proposed method. For LR-based methods [22], linear regression (LR), interactions linear regression (ILR), robust linear regression (RLR), and step-wise linear regression (SLR) are implemented and compared. For tree regression based methods [23], fine tree, medium tree, and coarse tree are implemented and compared. For SVM-based methods [24], linear SVM, Quadratic SVM, cubic SVM, fine Gaussian SVM, medium Gaussian SVM, and coarse Gaussian SVM are implemented and compared. For ensemble learning methods [25], boosted trees and bagged trees are implemented and compared. For GPR-based methods [26], squared exponential GPR, Matern 5/2 GPR, exponential GPR, and rational quadratic GPR are implemented and compared. For RNN-based methods [27], [28], LSTM and BiLSTM

TABLE I
COMPARISON OF DIFFERENT METHODS

Method	MSE ($\times 10^7$)	RMSE ($\times 10^3$)	MAE ($\times 10^3$)	sMAPE
Linear Regression				
LR	11.036	10.505	7.7287	0.5519
ILR	11.034	10.504	7.7362	0.5522
RLR	11.034	10.504	7.7219	0.5524
SLR	11.033	10.504	7.7368	0.5521
Tree Regression				
Fine Tree	19.355	13.912	9.8293	0.5674
Medium Tree	16.179	12.720	9.0366	0.5626
Coarse Tree	13.213	11.495	8.2142	0.5550
Support Vector Machines				
Linear	11.043	10.509	7.7213	0.5418
Quadratic	11.012	10.494	7.6642	0.5305
Cubic	11.010	10.493	7.6768	0.5358
Fine Gaussian	12.915	11.364	8.2690	0.5676
Medium Gaussian	11.045	10.510	7.6723	0.5365
Coarse Gaussian	11.000	10.488	7.6690	0.5299
Ensemble Learning				
Boosted Trees	10.890	10.436	7.6594	0.5280
Bagged Trees	10.838	10.410	7.5274	0.5398
Gaussian Process Regression				
Squared Exponential	10.989	10.483	7.6996	0.5309
Matern 5/2	10.928	10.454	7.6658	0.5255
Exponential	10.881	10.431	7.6474	0.5278
Rational Quadratic	10.922	10.451	7.6711	0.5225
Recurrent Neural Network				
LSTM	5.879	7.668	5.492	0.4647
BiLSTM	4.892	6.994	5.010	0.4545
Fine-grained Recurrent Neural Network				
Our Method	4.228	6.503	4.557	0.3651

are implemented and compared. As shown in Table I, the estimation performance of our proposed method is evaluated with the comparison against 21 benchmarks from 6 machine learning categories, i.e. linear regression, tree regression, ensemble learning, support vector machines, Gaussian process regression, and recurrent neural network.

According to the quantitative results from the experiment, ensemble learning and Gaussian process regression performs the best among the traditional machine learning methods without using neural networks. Moreover, benefiting from capturing sequential information, recurrent neural networks, especially BiLSTM, can outperform the traditional methods. On basis of BiLSTM, our method can achieve a further improvement by trajectory segmentation and transfer learning, which can achieve 4.228 for MSE, 6.503 for RMSE, 4.557 for MAE, and 0.3651 for sMAPE. The following ablation study section provides an insight into how each technical component improves the estimation performance.

D. Ablation Study

As mentioned above, the estimation improvement of our proposed method is a cooperative effect of three components, which are bi-directional feedback, trajectory segmentation and transfer learning. In this section, the respective contribution of each component is separately identified. In Table II, the first row presents the estimation performance of baseline method, one-directional LSTM. The next three rows then present the performance after adding the three components on the basis on baseline, successively. Specifically, when purely relying on forward learning direction, the estimation performance can achieve 5.879 for MSE, 7.668 for RMSE, 5.492 for MAE, and 0.4647 for sMAPE. By adding the backward-learning feature to the traditional forward-only LSTM, BiLSTM is able to learn from reverse time sequence data, and extract sequential information from both forward and backward directions. In this way, the energy estimation error in our EV scenario can achieve reductions in all the four evaluated metrics.

Next, our proposed trajectory segmentation algorithm is added before the training stage, so as to improve the learning performance of neural networks by feeding them with more subtle inputs. Since the lengths of driving trip trajectories in the raw data are variable, a trajectory is divided into segments in a purely unsupervised manner. By considering the information of individual segments, we can achieve 4.543 for MSE, 6.740 for RMSE, 4.795 for MAE, and 0.3871 for sMAPE.

Finally, a transfer learning framework is introduced into our proposed fine-grained energy consumption estimate on EV. The knowledge learnt from ICE and HEV domains are transferred to the EV domain, where the pre-trained model of transfer learning on EV is obtained by training on ICE and HEV. After applying transfer learning, our full method can achieve the best performance in terms of MSE, RMSE, MAE, and sMAPE, which are 4.228, 6.503, 4.557, and 0.3651, respectively.

IV. CONCLUSION AND FUTURE WORK

We have proposed an estimation model for EV energy consumption based on both vehicle (e.g. speed) and environment data (GPS position, temperature). To achieve better estimation performance, three different techniques were employed: trajectory segmentation, BiRNN, and transfer learning. Our model computes the estimated energy consumption rate starting from trajectory data with collected features, i.e. real-time speed, out-vehicle temperature, GPS coordinate. To capture robust information from the trajectory data, we designed a trajectory segmentation algorithm based on GMM that is applied before the training stage. In addition, since the EV data is not sufficient to train the deep neural networks from scratch, transfer learning is used in our method to train the model which has been pre-trained by sufficient ICE and HEV data. After the pre-trained model is ready, the parameters of full-connected layers are re-initialised and then trained by EV data. In this way, the knowledge learnt from ICE and HEV data can be transferred to the EV scenario. Our experimental results show the proposed method outperforms other methods on four

TABLE II
ABLATION STUDY ON VARIOUS COMPONENTS OF FRAMEWORK

Forward	Backward	Segmentation	Transfer Learning	MSE ($\times 10^7$)	RMSE ($\times 10^3$)	MAE ($\times 10^3$)	sMAPE
✓				5.879	7.668	5.492	0.4647
✓	✓			4.892	6.994	5.010	0.4545
✓	✓	✓		4.543	6.740	4.795	0.3871
✓	✓	✓	✓	4.228	6.503	4.557	0.3651

standard evaluation metrics. In more detail, our method can achieve 4.228 for MSE, 6.503 for RMSE, 4.557 for MAE, and 0.3651 for sMAPE. Moreover, we have also carried out an ablation study to investigate the contributions of various components in our method and discover that the estimation performance can be improved progressively by adding bi-directional learning feature, trajectory segmentation process, and transfer learning feature.

In this paper, we focused on developing an adaptive model for the accurate estimation of EV instantaneous energy consumption based on ragged trajectory data. In future work, hyperparameter study on our proposed method will be conducted to explore the optimal LSTM structure and parameters. Moreover, our fine-grained estimation method can be integrated into the optimisation process for the deployment of charging stations and also combined with the existing anomaly detection pipeline to carry out a more accurate on-board battery health diagnostic for EVs.

REFERENCES

- [1] "Bureau of Transportation Statistics (BTS), U.S. Consumption of Energy from Primary Sources by Sector," <https://www.bts.gov/us-consumption-energy-primary-sources-sector-metric>, accessed: 2021-11.
- [2] D. Yi, J. Su, L. Hu, C. Liu, M. Qudus, M. Dianati, and W.-H. Chen, "Implicit personalization in driving assistance: State-of-the-art and open issues," *IEEE Transactions on Intelligent Vehicles*, vol. 5, no. 3, pp. 397–413, 2019.
- [3] Z. A. Needell, J. McNERney, M. T. Chang, and J. E. Trancik, "Potential for widespread electrification of personal vehicle travel in the united states," *Nature Energy*, vol. 1, no. 9, pp. 1–7, 2016.
- [4] H. A. Bonges III and A. C. Lusk, "Addressing electric vehicle (EV) sales and range anxiety through parking layout, policy and regulation," *Transportation Research Part A: Policy and Practice*, vol. 83, pp. 63–73, 2016.
- [5] F. Guo, J. Yang, and J. Lu, "The battery charging station location problem: Impact of users' range anxiety and distance convenience," *Transportation Research Part E: Logistics and Transportation Review*, vol. 114, pp. 1–18, 2018.
- [6] L. Thibault, G. De Nunzio, and A. Sciarretta, "A unified approach for electric vehicles range maximization via eco-routing, eco-driving, and energy consumption prediction," *IEEE Transactions on Intelligent Vehicles*, vol. 3, no. 4, pp. 463–475, 2018.
- [7] X. Qi, G. Wu, K. Boriboonsomsin, and M. J. Barth, "Data-driven decomposition analysis and estimation of link-level electric vehicle energy consumption under real-world traffic conditions," *Transportation Research Part D: Transport and Environment*, vol. 64, pp. 36–52, 2018.
- [8] R. Zhang and E. Yao, "Electric vehicles' energy consumption estimation with real driving condition data," *Transportation Research Part D: Transport and Environment*, vol. 41, pp. 177–187, 2015.
- [9] A. M. Bozorgi, M. Farasat, and A. Mahmoud, "A time and energy efficient routing algorithm for electric vehicles based on historical driving data," *IEEE Transactions on Intelligent Vehicles*, vol. 2, no. 4, pp. 308–320, 2017.
- [10] H. Wang, D. Zhao, Y. Cai, Q. Meng, and G. P. Ong, "A trajectory-based energy consumption estimation method considering battery degradation for an urban electric vehicle network," *Transportation Research Part D: Transport and Environment*, vol. 74, pp. 142–153, 2019.
- [11] X. Wu, D. Freese, A. Cabrera, and W. A. Kitch, "Electric vehicles' energy consumption measurement and estimation," *Transportation Research Part D: Transport and Environment*, vol. 34, pp. 52–67, 2015.
- [12] C. Fiori, K. Ahn, and H. A. Rakha, "Power-based electric vehicle energy consumption model: Model development and validation," *Applied Energy*, vol. 168, pp. 257–268, 2016.
- [13] S. Modi, J. Bhattacharya, and P. Basak, "Estimation of energy consumption of electric vehicles using deep convolutional neural network to reduce driver's range anxiety," *ISA transactions*, vol. 98, pp. 454–470, 2020.
- [14] G. Oh, D. J. Leblanc, and H. Peng, "Vehicle energy dataset (VED), a large-scale dataset for vehicle energy consumption research," *IEEE Transactions on Intelligent Transportation Systems*, 2020.
- [15] D. Yi, J. Su, and W.-H. Chen, "Probabilistic faster r-cnn with stochastic region proposing: Towards object detection and recognition in remote sensing imagery," *Neurocomputing*, vol. 459, pp. 290–301, 2021.
- [16] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Personalized driver workload inference by learning from vehicle related measurements," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 49, no. 1, pp. 159–168, 2017.
- [17] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," Stanford, Tech. Rep., 2006.
- [18] D. Yi, J. Su, C. Liu, and W.-H. Chen, "Trajectory clustering aided personalized driver intention prediction for intelligent vehicles," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 6, pp. 3693–3702, 2018.
- [19] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [20] Y. Yu, X. Si, C. Hu, and J. Zhang, "A review of recurrent neural networks: LSTM cells and network architectures," *Neural computation*, vol. 31, no. 7, pp. 1235–1270, 2019.
- [21] L. Torrey and J. Shavlik, "Transfer learning," in *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.
- [22] J. A. Khan, S. Van Aelst, and R. H. Zamar, "Building a robust linear model with forward selection and stepwise procedures," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 239–248, 2007.
- [23] S. Gupta and A. Chug, "Assessing cross-project technique for software maintainability prediction," *Procedia Computer Science*, vol. 167, pp. 656–665, 2020.
- [24] H.-T. Bang, S. Yoon, and H. Jeon, "Application of machine learning methods to predict a thermal conductivity model for compacted bentonite," *Annals of Nuclear Energy*, vol. 142, p. 107395, 2020.
- [25] C. D. Sutton, "Classification and regression trees, bagging, and boosting," *Handbook of statistics*, vol. 24, pp. 303–329, 2005.
- [26] S. A. Mazari, A. R. Siyal, N. H. Solangi, S. Ahmed, G. Griffin, R. Abro, N. M. Mubarak, M. Ahmed, and N. Sabzoi, "Prediction of thermo-physical properties of 1-butyl-3-methylimidazolium hexafluorophosphate for co2 capture using machine learning models," *Journal of Molecular Liquids*, vol. 327, p. 114785, 2021.
- [27] J. Park, H. Son, J. Lee, and J. Choi, "Driving assistant companion with voice interface using long short-term memory networks," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 1, pp. 582–590, 2018.
- [28] T. Thireou and M. Reczko, "Bidirectional long short-term memory networks for predicting the subcellular localization of eukaryotic proteins," *IEEE/ACM Transactions on computational biology and bioinformatics*, vol. 4, no. 3, pp. 441–446, 2007.