



**HAL**  
open science

## Energy management for electric vehicles in smart cities: a deep learning approach

Mohammed Laroui, Aicha Dridi, Hossam Afifi, Hassine MOUNGLA, Michel  
Marot, Moussa Ali Cherif

### ► To cite this version:

Mohammed Laroui, Aicha Dridi, Hossam Afifi, Hassine MOUNGLA, Michel Marot, et al.. Energy management for electric vehicles in smart cities: a deep learning approach. IWCMC 2019: International Wireless Communications & Mobile Computing Conference, Jun 2019, Tanger, Morocco. pp.2080-2085, 10.1109/IWCMC.2019.8766580 . hal-02101524

**HAL Id: hal-02101524**

**<https://hal.science/hal-02101524>**

Submitted on 16 Apr 2019

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# Energy Management For Electric Vehicles in Smart Cities: A Deep Learning Approach

Mohammed Laroui<sup>\*†‡</sup>, Aicha Dridi<sup>†</sup>, Hossam Afifi<sup>†</sup>, Hassine Moun gla<sup>†‡</sup>, Michel Marot<sup>†</sup>, and Moussa Ali Cherif<sup>\*</sup>

<sup>\*</sup>Evolutionary Engineering & Distributed Information Systems Laboratory, Djillali Liabes University, Sidi Bel Abbes, Algeria

<sup>†</sup>LIPADE, Paris Descartes University, Sorbonne Paris Cite, Paris, France

<sup>‡</sup>UMR 5157, CNRS, Institute Mines Telecom, Telecom SudParis, Nano-Innov CEA Saclay, France

Emails: {mohammed.laroui,moussa.alicherif}@univ-sba.dz, {mohammed.laroui,hassine.moun gla}@parisdescartes.fr, {aicha\_dridi,hossam.afifi,michel.marot}@telecom-sudparis.eu

**Abstract**—We propose a solution for Electric Vehicle (EV) energy management in smart cities, where a deep learning approach is used to enhance the energy consumption of electric vehicles by trajectory and delay predictions. Two Recurrent Neural Networks are adapted and trained on 60 days of urban traffic. The trained networks show precise prediction of trajectory and delay, even for long prediction intervals. An algorithm is designed and applied on well known energy models for traction and air conditioning. We show how it can prevent from a battery exhaustion. Experimental results combining both RNN and energy models demonstrate the efficiency of the proposed solution in terms of route trajectory and delay prediction, enhancing the energy management.

**Index Terms**—Recurrent Deep Learning, Electric vehicles, Energy control

## I. INTRODUCTION

The extensive use of electric vehicles faces different challenges such as energy management [1], [2] and route selection [3], [4]. EVs use limited resource batteries that require an effective energy management. Battery is solicited for motor traction but also for other energy consumption sources that are more targeted to driver and passengers comfort. Simple energy estimation for the rest of the trip in an electric car is not efficient at all. It can result in a battery exhaustion when traffic jams are not taken in consideration. It can result also in an under usage of the battery if the delays/distances are over estimated.

In this paper, we propose an energy management solution for electric vehicles based on predicted trajectory and delay information using deep learning recurrent neural networks (RNN).

Recurrent neural network (RNN) [5] is a class of artificial neural networks where each neuron in the network is chained to its predecessor and successor. This neuron or group of neurons has its local memory. This sequential chaining and memory make it adapted to remember combinations of sequences. It is hence capable of predicting the future output based on a sequence of ordered input sequence. RNN are extensively used in natural language processing for word completion, translation or even composition.

In this work, we use Long Short Term Memory (LSTM) [6] and Gated Recurrent Units (GRU) as variants of RNNs. We

use them to model and learn the urban car trajectories and delays from a large real data set.

We apply this prediction to very accurate models for traction and air conditioning energy estimation.

The rest of the paper is organized as follows: in section II, we present related work on EV energy management. In section III, we describe the recurrent neural network used for urban traffic learning. In Section IV, we describe the adaptation of RNNs to the Italian dataset and study effect of meta parameters on prediction precision. In section V, our energy management algorithm is presented and analysed. Finally in Section VI we conclude the paper and give perspectives.

## II. RELATED WORK

Interesting effort has been invested in the electric vehicle challenge including the energy management and route selection.

### A. EVs Energy Management Efforts:

The energy management for EVs is an important issue [1], where the optimal energy management allows to efficiently consume the energy which guarantees the use of the EVs for a long time. Tian et al. [7] presented a real-time charging station system for electric vehicle taxis by combining real-time GPS trajectories and historical charging of each electric vehicle taxi. They predicted the current state of each taxi and based on this information, when a taxi request for a recommendation, they recommend a charging station which offers the minimal total time. Experiment results showed that the proposed system reduces the cost time by 50%. Similarly, Manshadi et al. [8] proposed a decentralized optimization framework for wireless charging on transportation and electricity networks. It enables to evaluate the effects of the electricity demand and the effects of electricity pricing for electric vehicles. They considered that all the EVs paths have the same total cost for travel from source to destination. However, this work did not consider traffic jams and route delays, which could lead to more charging opportunities and more power usage for EVs. Gao et al [9] presented a speed optimization framework to minimize the energy consumption and the battery aging for intelligent EVs to optimize the real-life driving conditions. They derived

the battery life control-oriented model and formulated the speed optimization during the acceleration process. This was solved using the SQP algorithm. The simulation results show that the battery capacity can be reduced by 9.6% during acceleration from 0Km/h to 100Km/h within 14 seconds, with an increase of 1.6% of battery energy consumption.

### B. EVs route selection research efforts:

The route selection for EVs is a mechanism allowing to select the best route based on different information such as congestion, accident, etc. with an objective to find the short route with less energy consumption for EVs. The prediction [10] is generally used in vehicular traffic to find the optimal route from source to destination. De et al [11] presented a novel system called the intention-aware routing system (IARS) for EVs. It allows the electric vehicles to compute a routing policy in order to minimize their expected journey time while taking into account the intentions, or policies of the other vehicles. They demonstrated the efficiency of the proposed IARS using realistic settings with real data from the Netherlands. Experimental results show that the IARS leads to more than 50% reduction in overall journey times and 80% improvement in waiting times at charging stations. However, capacity distribution to every charging station is not considered in this work.

Li et al [12] proposed a multi-objective route optimization model to determine the optimal route for EVs, and coordinate wireless and plug-in charging strategies, to trade off the charging cost and total time consumption for a trip. To prove the efficiency of the proposed model they used two systems for testing, including real-world road for Xi'an city.

Yi et al [13] proposed an energy aware routing framework for EVs with the objective of enhancing the future electric transportation systems. They introduced a stochastic decision-making framework in order to handle the effects of different environmental factors on the energy cost of transportation. In addition, they formulated the optimal routing problem as a stochastic programming problem, and they applied the risk control of the total energy for objective to find the minimum energy route. The simulation results demonstrate the utility of this approach, However, the proposed approach is not always the best solution because sometimes the energy efficient route is not suitable for the customers.

## III. RECURRENT NEURAL NETWORK (RNN)

Various research works have been carried out dealing with real-time traffic flow predictions. These forecasts are more and more important if we consider the case of electric cars, to better control energy consumption.

These contributions generally demonstrate that the Recurrent Neural Network (RNN) [5] which is a class of artificial neural networks where each neuron in the network is chained to others, gives the best results to this day.

It is a class of neural networks that can predict the future sequence based on a learned pattern (sequence of normalized values).

In the presented work, we convert a sequence of vehicle hops corresponding to its trip to a time series. We train the model, tune it and use it for prediction. After analysis of both methods and their meta parameters, we will explain what was our selected one.

The formulation of RNN could be of the following form: we consider an input sequence  $x = (x_1, \dots, x_t)$ , hidden vector sequence  $h = (h_1, \dots, h_t)$  and output vector sequence  $y = (y_1, \dots, y_t)$ .

$$h_t = H(W_{hx}x_t + W_{hh}h_{t-1} + b_h) \quad (1)$$

$$p_t = W_{hy}Y_{t-1} + b_y \quad (2)$$

Where:

$W_{hx}$  corresponds to the weight between the input and hidden layer parameters,  $W_{hy}$  corresponds to the weight between hidden and output layer parameters,  $W_{hh}$  corresponds to the weight between hidden layers,  $b_h$  and  $b_y$  symbolizes the bias vectors for the output and hidden layers.  $H$  is a nonlinear activation functions.

### A. Long Short-Term Memory (LSTM)

LSTM was firstly proposed in 1997 by Hochreiter and Schmidhuber for language modeling [6]. Indeed, one of the most popular problems that RNN suffers from is the vanishing gradient, and LSTM came to solve it.

The LSTM is composed of special blocks, called memory blocks. These memory blocks contain special multiplicative units called gates. A typical memory block is composed of three gates, The input gate is responsible for the addition of information by dealing with the upcoming data. The output gate is responsible for selecting useful information from the current cell state and showing it out as an output. The forget gate is responsible for removing information from the cell state, in fact information that is no longer required for the LSTM understanding (processing) is removed.

LSTMs have been used to advance state of the art of many difficult problems, including speech recognition and acoustic modeling. The first use of LSTM for traffic prediction was on 2015 [14]. Figure 1 display the structure of LSTM blocks.

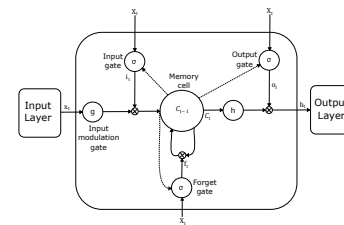


Figure 1: Structure of LSTM blocks

$$i_t = \sigma(W_{ix}X_t + W_{hh}h_{t-1} + W_{ic}c_{t-1} + b_i) \quad (3)$$

$$f_t = \sigma(W_{fx}x_t + W_{hh}h_{t-1} + W_{fc}c_{t-1} + b_f) \quad (4)$$

$$c_t = f_t * c_{t-1} + i_t * g(W_{cx}x_t + W_{hh}h_{t-1} + W_{cc}c_{t-1} + b_c) \quad (5)$$

$$o_t = \sigma(W_{ox}x_t + W_{hh}h_{t-1} + W_{oc}c_{t-1} + b_o) \quad (6)$$

$$h_t = o_t * h(c_t) \quad (7)$$

Where:

$$\sigma(x) = \frac{1}{1 + e^x} \quad (8)$$

The  $i_t$ ,  $f_t$ ,  $c_t$ ,  $o_t$ ,  $\sigma$  respectively correspond to the input gates, the forget gates, the memory cells, the output gates and the sigmoid activation function.

Today, LSTM is enriched with the 'Dense' property. In fact, we add at the end of this RNN one or several deep layers (fully connected neuron layers), resulting in a complex mathematical architecture that gives much better results than before [15]. Google has also improved Dense LSTM scalability with slight modifications (DLSTM-P).

### B. Gated Recurrent Units (GRU)

GRU was firstly proposed by Cho et al. in 2014 [16]. Similarly, to the LSTM blocks, the GRU has gates that modulate the flow of information inside the blocks. A typical GRU block is composed of a couple of gates. A reset gate like the forget gate in LSTM, it makes the block act as if it is reading the first symbol of an input sequence, allowing it to forget the previously computed state. And an update gate, that decides how much the block updates its activation, or content. Figure 2 shows the structure of the GRU blocks.

We decided to compare Dense GRU and Dense LSTM, for that we used the same optimization method. As it will be shown in the results, we observed a net improvement of predictions made with LSTM. That's why our choice will be focused on it.

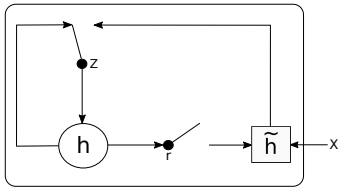


Figure 2: Structure of GRU blocks

$$r_t = \sigma(W_{xr}x_t + W_{hr}h_{t-1} + b_r) \quad (9)$$

$$z_t = \sigma(W_{xz}x_t + W_{hz}h_{t-1} + b_z) \quad (10)$$

$$\tilde{h}_t = f(W_{xh}x_t + W_{hh}(r_t \odot h_{t-1}) + b_h) \quad (11)$$

$$h_t = (1 - z_t) \odot h_{t-1} + z_t \odot \tilde{h}_t \quad (12)$$

The  $r_t$ ,  $z_t$ ,  $\tilde{h}_t$ ,  $h_t$ , respectively correspond to the reset, update gates, candidate hidden layer values, and the hidden layer values at time  $t$ .

## IV. DATA SET PREPARATION, RNN TRAINING AND PREDICTION

### A. System Description

From the dataset, we could analyze the trip lengths. Trips are periodically sampled by GPS but the recorded time stamps are not exactly periodic, which is probably due to some phenomena like jitter. Nevertheless, it gives a good approximation of the trip description. We call a "hop" each time a user is recorded by a base station. Figure 3 shows three cumulative distribution functions (CDF): the CDF of the number of hops per trip, the CDF of the trip distance in kilometers and the one of the trip duration in seconds. The y-axis corresponds to the cumulative probability density, the x-axis is either the trip duration in seconds, or the trip length in number of hops or in kilometers. It can be noted that 85% of the trips are at most 20 hops large. That is why in our learning method we describe trips by a 20x3 dimensional array, each position being described by the latitude, longitude and time of the user.

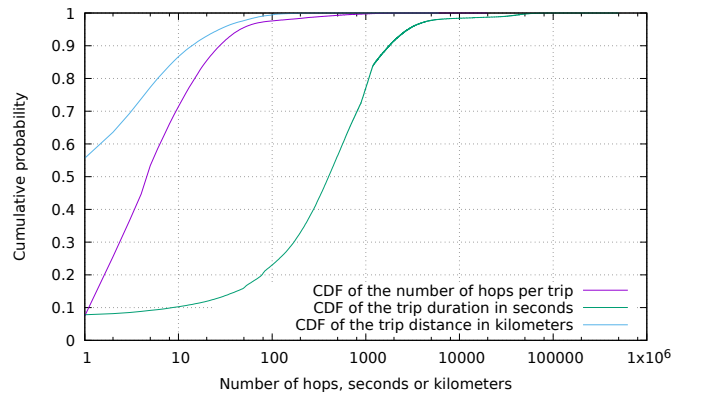


Figure 3: Cumulative distribution functions (CDF)

For each car trajectory, we keep 20 hops, that define the trip. Each hop is identified by 3 values: time, latitude and longitude. In the presented work, we decided to shift our data by four steps. The ML algorithm given the current position must predict the next four positions of the car during the trip. Figure 4 below schematizes how the label is shifted compared to the input. In the training phase, if our input contains the 15 first values of the trip of a specific car, the label will contain the 15 values shifted by this shift (4 in our case). In the test step, given an input, the ML algorithm will predict the 4 next values of the trip. If we increase the shift step, the performance of the algorithm will decrease or it will need more training. Next step is to scale the data, since NNs work better on values between -1 and 1.

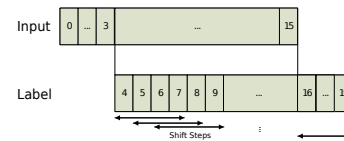


Figure 4: Necessary shifts for target data prediction

## B. System Setup

The performance of the proposed solutions has been evaluated after extensive training. We have used real mobility traces of taxis in Rome Italy [17] to train and predict routes. 60 days of traffic, where each day contains approximately 70000 different vehicle trajectories, were used. The code is based on keras <sup>1</sup> with a tensorflow <sup>2</sup> backend for DLSTM and DGRU which are implemented in python <sup>3</sup> environment. Table I outlines the different configurations used in the training step <sup>4</sup>.

Table I: Neural Network Configurations

Config	Epoch	Step per Epoch
1	4	300
2	4	3000
3	8	3000
4	8	5000
5	12	3000

For each configuration, we calculate the mean square error (MSE) in order to choose the best Epoch/Step Per Epoch giving to the optimal MSE values.

Figure 5 shows the mean square error with different configurations of the neural network listed in table I. We can notice that the use of 4 neurons (4 layers, 100 hidden neurons and two dense layers) enhances the efficiency of the prediction compared to the other values of neural. In addition, the use of 4 neural with the second configuration gives the best MSE value that equal to 0.011 with short execution time.

$$MSE = \frac{1}{n} \sum_{i=1}^n (P_i - \hat{P}_i)^2 \quad (13)$$

$$MAE = \frac{1}{n} \sum_{i=1}^n |(P_i - \hat{P}_i)| \quad (14)$$

$$RMSE = \left( \frac{1}{n} \sum_{i=1}^n (P_i - \hat{P}_i)^2 \right)^{1/2} \quad (15)$$

## C. Result analysis

Figure 6a displays the obtained results using the DLSTM and DGRU for route prediction, in fact, we show that the LSTM family gives better results compared with GRU one, where the MSE of the LSTM is 0.014, better than the MSE of GRU, that is equal to 0.015.

Figure 6b shows the results of route prediction using LSTM and GRU. We show that the LSTM is more close to real traces in all the routes, where the GRU stays alongside the real traces but very far compared to LSTM. In addition, the MSE is equal for LSTM and GRU where its value is 0.00004.

<sup>1</sup>Keras: <https://keras.io/>

<sup>2</sup>Tensorflow: <https://www.tensorflow.org/>

<sup>3</sup>Python: <https://www.python.org/>

<sup>4</sup>Python code and trained network will be published with article

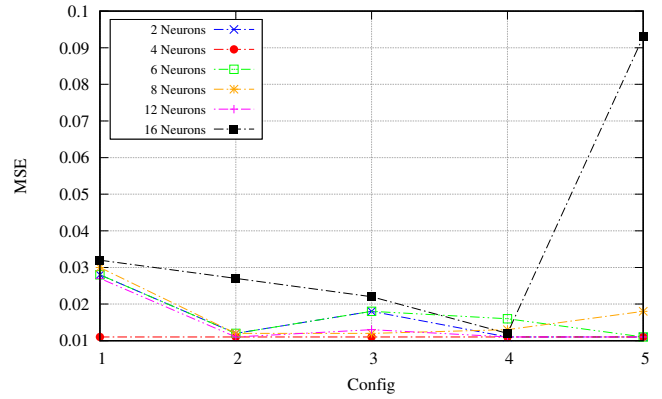


Figure 5: Mean square error while training (MSE)

Similarly, Figure 6c shows the predicted route using DLSTM and DGRU, from this figure we show that the GRU stays farther to real traces in all the routes. On the other side, LSTM is very close to real traces in all the routes except the final hop of the route where the LSTM is far from the real traces. In addition, the MSE is equal for both GRU and LSTM with a value equal to 0.00032.

Table II outlines the different values of MSE, MAE and RMSE for LSTM and GRU after the test step.

Route	MSE		MAE		RMSE	
	LSTM	GRU	LSTM	GRU	LSTM	GRU
1	0.0014	0.0015	0.00775	0.0138	0.038	0.038
2	0.00004	0.00004	0.00337	0.00382	0.006	0.006
3	0.00032	0.00032	0.0119	0.0136	0.041	0.040

Table II: Obtained values for mean square error (MSE), mean absolute error (MAE) and root-mean-square error (RMSE), in the test phase.

Finally, we plot in Figure 7 both real and predicted traces from the DLSTM network. The red line (real) is quite close to prediction (yellow). Note that we can easily map the prediction on real streets with specific navigation algorithms but this is out of the scope of our work.

## V. ENERGY MANAGEMENT FOR EVS

We propose to manage EV energy based on our knowledge and predictions extracted from the previous sections. We assume that EVs have regular access to the prediction system. As explained before, we can adjust the number of predicted future steps and the management system in the car can hence know in advance the predicted trajectory along with the estimated times along that trajectory.

The main idea in our contribution on energy management is that time predictions when there are additional traffic jams, provide very precious information to battery management. Depending on the actual State of Charge of the battery (SoC), there may be serious risks that autonomy cannot lead the car to its final destination. Since it is known that comfort energy in a car can go up to 30% of the total power, cutting such non-critical accessories can provide this extra energy.



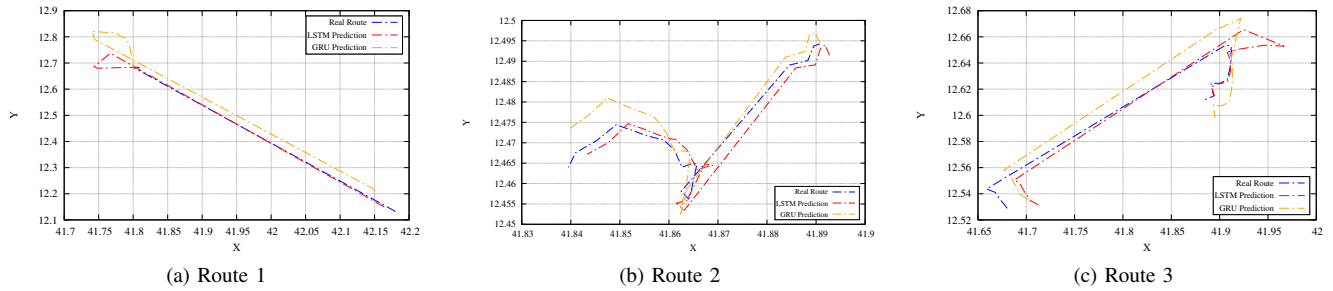


Figure 6: Predicted routes using DLSTM & DGRU

Deep learning provides an information that is difficult to obtain elsewhere. This information is regularly updated by the inherent sliding time series prediction system.

Before we explain the proposed algorithm, we need to present the energy management of a typical electric vehicle.

#### A. Traction Power Model

EV energy consumption goes mainly for the traction power. Secondary energy consumption is spent on air conditioning (cold or hot), and other onboard systems. The sum of those auxiliary sources can easily reach one third of the total consumption. They are hence not negligible.

First, we use a traction motor model presented in [18]. Although, it can be found in several other contributions, this early model has the advantage of being quite exhaustive. The EV is characterized by its weight, its batteries and other characteristics such as motor inverters and auxiliary loads. The consumed traction power looks like a fluctuating positive function. It depends on the road type (that can be extracted from our deep learning prediction), on the number of stops on the trajectory and the driver mood (also speed limitations!).

#### B. Air Conditioning model

Another study in [19] has provided a simple air conditioning power consumption model (validated in real conditions). It takes in consideration temperatures/humidity outside cabin, inside and required by the driver. It is said simple because it does not take more advanced parameters such as driver ethnic origin, number of passengers, etc... From this model, we calculate energy based on different required comfort profiles. For simplicity, we restrict the study on two models. We assume a summer temperature of 30 degrees outside the cabin with 80% humidity. The first model is called *FullComfort*. Its target is to decrease cabin temperature to 21 Celsius degrees and humidity to 50%. The second is called *HalfComfort* and targets a cabin higher temperature of 25 degrees.

#### C. Energy Management Algorithm

Based on those three elements: deep learning continuous prediction, required traction energy and auxiliary energy, we propose our management system. The algorithm is described in 1.

#### Algorithm 1 Energy Prediction and Management Algorithm

```

1: Input: State of Charge, list of previous points triplets
2: Output: Use Full Air Conditioning, Half,Traction or Alert
3:
4: while Not at Dst do
5:   Calculate next predicted sequence triplets (x,y and time)
6:   Calculate Predicted Time to Dst
7:   Calculate FullComfort, HalfComfort and Traction energies to Dst
8:
9:   if TractionEnergy  $\geq$  SoC then
10:     Raise Severe Alert (Not Enough Energy)
11:
12:   else if HalfConfort  $\geq$  SoC then
13:     Raise Traction Only
14:
15:   else if FullConfort  $\geq$  SoC then
16:     Raise HalfConfort Only
17:   end if
18: end while

```



Figure 7: DLSTM predicted route versus real trajectory on Google Earth

#### D. Results

We evaluate the energy models for traction and comfort. We test two scenarios: full comfort means that the driver puts the air conditioning at its maximum power (21 degrees/50 % humidity). The half comfort is a compromise that offers a better temperature/humidity (26 degrees) condition over external conditions (30 degrees/80% humidity). Figure 8 shows the energy consumption for full, half comfort and

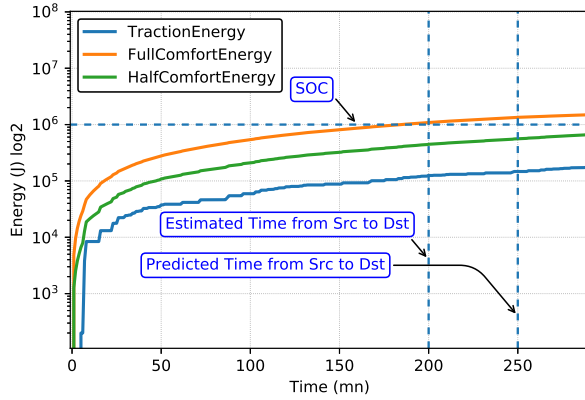


Figure 8: Energy Consumption For Different Driving Situations

traction only driving situations measured during estimated and predicted times to reach the destination. We use information from Route 2, to calculate distances, estimated and predicted times. We show that in the comfort situation, the energy is totally consumed where its value reaches the SoC value during exactly the estimated time. This means that for a simple average complexity car calculator, the estimated energy is sufficient to reach the destination with comfort driving. But in the reality, and through our predicted time where different criteria are considered, the energy is not sufficient to reach the destination with a comfortable driving. Its value reaches the SoC after 200 mn which is not the sufficient time to reach the destination. Consequently, the energy will be totally consumed before reaching the destination and the battery charge will be exhausted.

For the half comfort driving, we show that the predicted energy is sufficient to reach the destination in the estimated and predicted times. However, its exact value is totally different in the two trips. During the predicted time calculation, more energy is logically needed compared to the estimated time. Similarly, during the normal driving mode (traction only), the energy is sufficient to reach the destination for both trip duration. Again, more energy is consumed during the predicted trip because it is closer to the real traffic jam conditions.

## VI. CONCLUSION & FUTURE WORK

This paper presents a deep learning approach for EVs energy management, where Recurrent neural networks (RNN) are used for route and delay prediction with both LSTM and GRU models. We used real mobility traces of Rome, Italy to train and predict trajectories and delays with the two models. We succeed to tune our neural networks to reach a very good prediction rate. Experimental results show that the prediction using LSTM overpasses GRU. We then studied the energy consumption in three different driving situations, full comfort, half comfort and normal (traction only) based on estimated and predicted arrival times. An algorithm is proposed to periodically evaluate those energies and compare

to the State of Charge of the battery. The algorithm proposes to the driver the possible choices to safely reach destination. The code for RNN and energy calculation will be published. Next, we plan to develop a reinforcement learning algorithm that selects adequate actions on vehicle parameters in real time to optimize both driving and comfort.

## REFERENCES

- [1] B.-H. Nguyen, R. German, J. P. F. Trovão, and A. Bouscayrol, "Real-Time Energy Management of Battery/Supercapacitor Electric Vehicles Based on an Adaptation of Pontryagin's Minimum Principle," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 1, pp. 203–212, 2019.
- [2] H. Wu, G. K. H. Pang, K. L. Choy, and H. Y. Lam, "An Optimization Model for Electric Vehicle Battery Charging at a Battery Swapping Station," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 2, pp. 881–895, 2018.
- [3] J. Shi, Y. Gao, and N. Yu, "Routing Electric Vehicle Fleet for Ride-Sharing," in *2018 2nd IEEE Conference on Energy Internet and Energy System Integration (EI2)*. IEEE, 2018, pp. 1–6.
- [4] S. Pourazarm and C. G. Cassandras, "Optimal Routing of Energy-Aware Vehicles in Transportation Networks With Inhomogeneous Charging Nodes," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 8, pp. 2515–2527, 2018.
- [5] M. Salvaris, D. Dean, and W. H. Tok, "Recurrent Neural Networks," in *Deep Learning with Azure*. Springer, 2018, pp. 161–186.
- [6] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
- [7] Z. Tian, T. Jung, Y. Wang, F. Zhang, L. Tu, C. Xu, C. Tian, and X.-Y. Li, "Real-time charging station recommendation system for electric-vehicle taxis," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 11, pp. 3098–3109, 2016.
- [8] S. D. Manshadi, M. E. Khodayar, K. Abdelghany, and H. Üster, "Wireless charging of electric vehicles in electricity and transportation networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 5, pp. 4503–4512, 2018.
- [9] B. Gao, L. Guo, Q. Zheng, B. Huang, and H. Chen, "Acceleration Speed Optimization of Intelligent EVs in Consideration of Battery Aging," *IEEE Transactions on Vehicular Technology*, 2018.
- [10] M. Laroui, A. Sellami, B. Nour, H. Mounghla, H. Afifi, and S. Boukli-Hacène, "Driving path stability in VANETs," in *IEEE Global Communications Conference (GLOBECOM)*, 2018, pp. 1–6.
- [11] M. M. de Weerd, S. Stein, E. H. Gerding, V. Robu, and N. R. Jennings, "Intention-aware routing of electric vehicles," *IEEE Transactions on Intelligent Transportation Systems*, vol. 17, no. 5, pp. 1472–1482, 2016.
- [12] C. Li, T. Ding, X. Liu, and C. Huang, "An Electric Vehicle Routing Optimization Model with Hybrid Plug-in and Wireless Charging Systems," *IEEE Access*, 2018.
- [13] Z. Yi and P. H. Bauer, "Optimal Stochastic Eco-Routing Solutions for Electric Vehicles," *IEEE Transactions on Intelligent Transportation Systems*, no. 99, pp. 1–11, 2018.
- [14] R. Fu, Z. Zhang, and L. Li, "Using LSTM and GRU neural network methods for traffic flow prediction," in *31st IEEE Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328.
- [15] H. Sak, A. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Fifteenth annual conference of the international speech communication association*, 2014.
- [16] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder-decoder for statistical machine translation," *arXiv preprint arXiv:1406.1078*, 2014.
- [17] P. L. G. B. R. A. A. R. Lorenzo Bracciale, Marco Bonola, "CRAWDAD dataset roma/taxi (v. 2015-03)," in *CRAWDAD dataset roma/taxi (v. 2015)*, 2014.
- [18] F. L. Mapelli and D. Tarsitano, "Modeling of full electric and hybrid electric vehicles," in *new generation of electric vehicles*. IntechOpen, 2012.
- [19] A. Lahlou, F. Ossart, E. Boudard, F. Roy, and M. Bakhouya, "A dynamic programming approach for thermal comfort control in electric vehicles," in *2018 IEEE Vehicle Power and Propulsion Conference (VPPC)*. IEEE, 2018, pp. 1–6.