Available online at www.sciencedirect.com

ScienceDirect

Transportation Research Procedia 00 (2018) 000–000

Transportation Research Procedia

www.elsevier.com/locate/procedia

International Symposium of Transport Simulation (ISTS'18) and the International Workshop on Traffic Data Collection and its Standardization (IWTDCS'18)

Recurrent Neural Network Based Driving Cycle Development for Light Duty Vehicles in Beijing

Duoguan Qiu, Yuan Li*, Dapeng Qiao

School of Automation, Beijing institute of Technology, 5 South Zhongguancun Street, Haidian District, Beijing 100081, P. R. China

Abstract

This paper presents a data driven, Recurrent Neural Network (RNN) based technique that models real world driving patterns without domain knowledge to develop driving cycles for light duty vehicles (LDVs) in Beijing. In contrast to approaches that feature sub-cycle selection from the original data set through Markov process and data clustering, our method models the conditional probability distribution of vehicle speed with RNN, in which the driving cycles are generated step by step. As a consequence, the presented method excludes the necessity for domain knowledge based feature extraction and corresponding data vectorization during modelling stage. In the end of this paper, the proposed method is evaluated through comparisons between synthesized driving cycle and the original data set based on 14 metrics. As the final results suggest, both of the cycles obtained from the two models established are of relatively high power demand compared with the original data set, while one of the model is able to yield driving cycles that are more inclusive in terms of velocity levels with less candidates, which, to some extent, makes it more suitable for emission tests compared with the other one.

Keywords: Driving Cycle; Light Duty Vehicle; Recurrent Neural Network; Mean Tractive Force

1. Introduction

A driving cycle is a fixed schedule of vehicle operation which allows an emission test to be conducted under reproducible conditions, as Barlow et al. (2009) defined. Over the past decades, the industry has witnessed both proposals of various methods to develop driving cycles and further efforts adapting them to different regional traffic conditions. Typically, driving cycles are represented as sequences of vehicle speed versus time. A driving cycle, developed based on large amount of real world driving data, may serve as reference for emission & fuel consumption evaluation, vehicle design & certification and other related applications. For such purposes, it is crucial that the driving

* Corresponding author. Tel.: +8610-6891-4506.
E-mail address: liyuan@bit.edu.cn

2352-1465 © 2018 The Authors. Published by Elsevier Ltd.

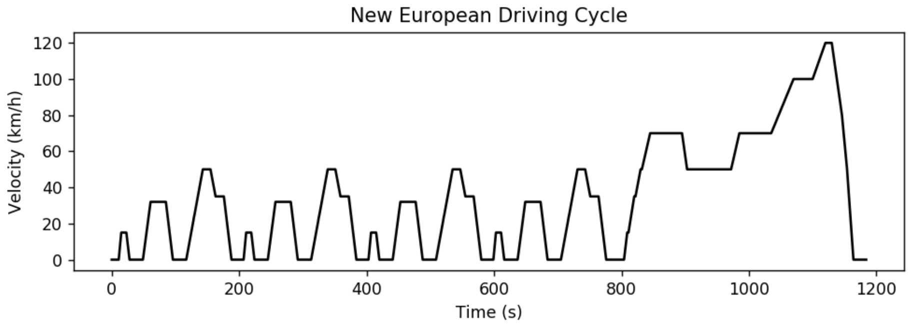

Fig. 1. The NEDC is composed of 4 Urban Driving Cycles (known as ECE-15 or UDC), from 0s to 780s and a Extra-Urban Driving Cycle (EUDC) from 780s to 1180s.

cycle developed should be representative enough in terms of the driving patterns, driver behaviors, regional traffic conditions etc., in real world domain. The New European Driving Cycle (NEDC), once adopted in China for vehicle certification, is a well-known driving cycle that features low level of acceleration, as illustrated in Fig. 1. However, given that such sequence is not representative enough in terms of real-world driving patterns (e.g. the constantly shifting speed in reality), and that it has been 20 years since the last update during which the driving condition has changed over time in China, the NEDC is no longer the optimal solution. Additionally, it has been criticized for underestimating the emissions on CO_2 (Fontaras and Dilara (2012)), compared to other driving cycles developed.

To address these problems, new methodologies for driving cycle development have been proposed, many of which feature sub-cycle (or modal sequence, micro-trip etc.) selection and concatenation (e.g. Grüner (2016); Gong et al. (2011)). The final driving cycle is developed, by selecting sub-cycles from different clusters (through Markov Chains or other techniques) and validating the concatenated sequence based on a set of metrics. Approaches of this kind demand robust clustering algorithms and domain knowledge based feature extraction for each sub-cycle from the original data, to achieve acceptable results.

In contrast to these approaches, methods that directly estimate the probability distributions on velocities, which opt out the necessity for knowledge based data clustering, by minimizing empirical errors of predictions on velocity, have been proposed by Bishop et al. (2012). Methods of this kind are able to generate driving cycles in which the sub-cycles are not included in the original data set.

While Markov Models are very popular in driving cycle development and the methods illustrated above, both sub-segment based and time stamp based, proved to be effective in terms of capturing some of the features related to driving patterns, the Markov process eventually discards the information from previous sequences as new predictions are made, given that each prediction at a specific time stamp takes into account only its previous hidden state, which corresponds to the a velocity value (or alternatively a preceding sequence, a set of extracted feature etc.). In contrast to commonly used the Markov Model based approaches, a Recurrent Neural Network (RNN) based model is established to capture the characteristics of real world driving patterns in this paper. There are a few researches that utilize neural network for capturing driving behaviors or traffic conditions, from data of the real world driving. Vaz et al. (2014) used neural network for driving cycle classification task. Wang et al. (2015) made use of neural network algorithm for driving cycle classification. But the performance of RNN for driving cycle development is yet to be well researched. We aimed to explore, in this paper, how RNN will contribute to driving cycle development for Light Duty Vehicles (LDVs) in Beijing.

2. Preliminaries

2.1. Problem Statement

The goal of this work is to obtain driving cycles that reflect driving pattern in Beijing as much as possible for fuel consumption and emission tests, given data (e.g. timestamps and velocities) collected in real world domain. The main problem is to consolidate a large amount of data as a single sequence at feasible length for such applications, while retaining the characteristics embedded in the original data.

The major contribution of this work is as follows: We obtained two final driving cycles from two separate sets of candidates, each of which was generated by specifying the first few steps and expanding the sequence step by step according to a conditional probability distribution of the vehicle speed at specific timestamp given the whole precedent velocity sequence. An RNN based model that is able to take a sequence of any length as precedent and yield predictions on $Pr(v_{t+1}|v_0, v_1, \dots, v_t)$ by learning from the data set, was established in this work.

2.2. Fundamentals of RNN

An RNN is formed by a variable number of identical RNN cells connected together, where each cell utilizes the state yielded by the previous one. In this paper, an RNN cell is defined as a unit that takes both an input vector x_t and a state vector s_t as its inputs, and yields $s_{t+1} = f(x_t, s_t; \Theta_s)$, the updated state, and an output vector $y_t = g(s_{t+1}; \Theta_y)$, where Θ_s, Θ_y are the set of pre-trained parameters of the cell that parameterize f, g, the functions that impart non-linearity. One of the RNN cell used in this paper, denoted as Vanilla RNN Cell, is defined as

$$s_{t+1} = \phi_1(W_{hh}s_t + W_{xh}x_t + b_h) \tag{1}$$

$$y_t = \phi_2(W_{ho}s_{t+1} + b_o) \tag{2}$$

where

- $t \in \mathbb{N}$ denotes the timestamp,
- $s_t, s_{t+1}, x_t, x_{t+1} \in \mathbb{R}^N$ are hidden states and inputs at corresponding timestamps,
- $y_t \in \mathbb{R}^N$ is the network output for time t,
- $W_{ho}, W_{hh}, W_{xh} \in \mathbb{R}^{N \times N}$ and $b_o, b_h \in \mathbb{R}^N$ are weights and biases that parameterize the model and,
- $\phi_1(\cdot)$ and $\phi_2(\cdot)$ are non-linear activation functions for updating hidden states and calculating output, respectively.

With s_t updated constantly, an RNN cell is able to

- allow the information in preceding input sequence to persist, and
- yield different outputs even for the same input vector x_t.

3. RNN Based Conditional Probability Distribution Modelling and Driving Cycle Development

3.1. Velocity Prediction

In this paper the velocities of vehicles are confined to a set of discrete values denoted as $v^{(i)} \in \mathcal{S}, i = 1, 2, \dots, |\mathcal{S}|$. The value of input for RNN at each timestamp, x_t, is given by encoding $v_t = v^{(i)}, v^{(i)} \in \mathcal{S}$ as a $|\mathcal{S}|$ dimensional vector, where all entries are zero except for the i_{th}, which is one (denoted as one-hot encoding). To make prediction on the conditional probability distribution on v_{t+1}, we made use of y_t to parameterize a multinomial distribution, yielding

$$Pr(v_{t+1}|v_0, v_1, \dots, v_t) \approx Pr(v_{t+1}|s_0, x_0, x_1, \dots, x_t) = Pr(v_{t+1}|y_t) \tag{3}$$

$$Pr(v_{t+1} = v^{(k)}|y_t) = softmax(y_t^k) \tag{4}$$

$$softmax(y_t^k) = \frac{e^{y_t^k}}{\sum_{i=1}^{|S|} e^{y_t^i}} \tag{5}$$

. Therefore the conditional probability of a specific output sequence $\boldsymbol{v}$ $(\hat{v}_1, \hat{v}_2, \dots, \hat{v}_T)$, given input sequence $\boldsymbol{x}$ $(x_0, x_1, \dots, x_{T-1})$ and the initial state s_0, is computed as

$$Pr(\boldsymbol{v}|\boldsymbol{x}, s_0) = \prod_{i=0}^{T-1} Pr(v_{t+1} = \hat{v}_{t+1}|y_t) \tag{6}$$

. Therefore the loss function of the model (I.e. the penalty for divergence in the conditional probability distributions between prediction and the reality) based on MLE (Maximum Likelihood Estimation) is defined as

$$L(\boldsymbol{x}, \boldsymbol{v}) = -\sum_{t=0}^{T-1} \log Pr(v_{t+1} = \hat{v}_{t+1} | y_t) \tag{7}$$

.

3.2. The DRNN Structure

In this paper a Deep Recurrent Neural Network (DRNN) structure, achieved by forming a large RNN cell composed of multiple inner RNN cells and some cross layer connections, is established. For cell in each layer, the input vector consists of

- the outputs from its previous layer (for the first layer this part of input is excluded) and,
- the input of outer cell[1].

And outputs of the outer cell is formed by the weighted outputs from RNN cells of each layer plus a bias vector, defined as

$$y_t = b_o + \sum_i W_{h^i o} y_t^{h^i} \tag{8}$$

where

- y_t is the output of the outer cell at time t and,
- $y_t^{h^i} \in \mathbb{R}^{N_{h^i}}$ denotes the output of the cell in i_{th} layer at time t.

With such structure, each inner cell is able to learn its own version of state vector given the input sequences. The cross-layer connections allow the outer cell to weight each state vector its inner cells has produced to make prediction.

3.3. Driving Cycle Development

For each sub-sequence the training set, the loss defined in (7) is iteratively minimized through adjusting parameters in $\Theta_s \cup \Theta_y$ according to the gradients of loss function with respect to the parameters. Once the model had been trained, each of candidates was generated step by step [2], as illustrated in algorithm 1. Each candidate was clipped by simply selecting its longest subsequence (that starts with $t = 0$) such that the last value of the subsequence equals to zero. In validation stage, all candidates in DC will be ranked based on its proximity to the original data set in terms of features that are related to emissions and energy consumptions of Light Duty Vehicles (LDVs). In this paper each candidate was evaluated through relative errors on features (specified in Table 1) with respect to the corresponding values calculated from the whole data set. To evaluate level of service for each driving cycle, the mean tractive force (MTF) introduced by Nyberg et al. (2015, 2016) and the corresponding features were calculated, based on the parameter configuration specified in Table 2.

4. Experiments

The data used in the experiments, starting from January 1_{st} to October 30_{th}, 2017, adding up to a total of 122.9 thousand kilometers, are collected from 9 LDVs at 1Hz. As a paradigm for driving cycles development, data are continuously collected through terminal devices installed on these registered vehicles that freely roamed in Beijing.

[1] Before being connected to each layer, the input of outer cell are transformed by pre-multiplying a weight matrix for dimensional compatibility.
[2] In case of driving cycle development, the relatively small probabilities of those non-achievable velocities are truncated for prediction while generating driving cycle candidates.

Algorithm 1 Driving Cycle Synthesis

Input:

S: the initial state for RNN cell

P: the first K velocity values of the sequence to be generated ($\hat{v}_0 = 0, \hat{v}_1, \ldots, \hat{v}_{K-1}$)

T: an integer that determines the maximum length of the sequence to be generated

M: the pre-trained DRNN model parametrized by Θ_s,Θ_y

E: the encoding function for discretized velocities

C: the function for clipping the driving cycle

Output:

DC: A set of driving cycles candidates yielded by the model.

```
1: while within computational cost do
2:     s_0 = S
3:     for t in 0, 1, ..., T − 1 do
4:         x_t = E(v̂_t)
5:         s_{t+1} = f(s_t, x_t; Θ_s)
6:         y_t = g(s_{t+1}; Θ_y)
7:         Sample v̂_{t+1} through Pr(v_{t+1}|y_t)
8:     DC_raw = v̂_0, ..., v̂_{K−1}, v̂_K, ..., v̂_T
9:     Add C(DC_raw) to DC
   return DC
```

Table 1. Features for Evaluation

Feature	Description	Metric
η_{idle}	% of time standing	-
η_{accel}	% of time accelerating	-
η_{cruise}	% of time cruising	-
η_{decel}	% of time decelerating	-
v_{avg}	average speed	km/h
v_{avgm}	average speed during motion	km/h
v_{max}	maximum speed	km/h
Δv^{+}_{bound}	max speed increment	$km/(h \cdot s)$
Δv^{-}_{bound}	max speed decrement	$km/(h \cdot s)$
$\|\dddot{v}\|_{max}$	jerkiness index	m/s^3
$\bar{F}_{trac}$	mean tractive force (MTF)	N
α	MTF component	m^2/s^2
β	MTF component	-
γ	MTF component	m/s^2

Table 2. Parameter Configurations for calculation of MTF components

Parameter	Description	Value	Metric
A_f	frontal area	2.15	m^2
c_d	drag coefficient	0.4	-
c_r	rolling resistance coefficient	0.013	-
m	vehicle mass (no cargo)	1250	kg
ρ_{air}	air density	1.29	kg/m^3
g	gravitational constant	9.80151	m/s^2

In these experiments, the original data, ordered by the registration code of the vehicles), were split into 95020 micro trips. The first 85% of data (70% for training set and 15% for validation set) are used for training the DRNN based model.

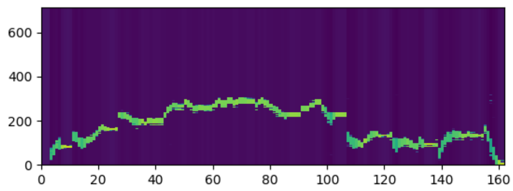

Fig. 2. Visualization of prediction on conditional probability distribution of velocity at each timestamp, given a real driving cycle as the input of the DRNN model. The brightness of pixel at coordinate (t, i) indicates the conditional probability of $v^{(i)}$ at time t, given its corresponding input sequence $(v_0, v_1, \ldots, v_{t-1})$.

4.1. The Velocity to Velocity Approach

In this experiment, the model directly predicts the probability of velocities, which is denoted as "V2V Approach" in this paper. We used GRUs (Gated Recurrent Units introduced by Cho et al. (2014)), to form the inner cells of the DRNN model. Parameters that define the structure of the network are specified in Table 3.

Table 3. Configuration of the Network in "V2V Approach"

Item	Value
Dimension of x_t	713
Dimension of y_t	713
Dimension of s_t	100×6
Number of Layers	6
Type of Cell	GRU

The dimension of x_t and y_t corresponds to a resolution of 0.2 km/h in velocity. After training the network eventually captures the conditional probability distribution of velocity at each timestamp, as visualized in Fig. 2.

4.2. The Velocity to Velocity Increment Approach

This experiment, denoted as "V2DV Approach", aims to reduce the computational cost by limiting the complexity of the model. In this experiment, the DRNN model predicts the increment of velocity instead, by changing equation (4) to $Pr\left((v_{t+1} - v_t) = v^{(k)}|y_t\right) = softmax(y_t^k)$, which reduces the dimension of output vectors. We further simplified the computations by using the "Vanilla RNN Cell" described in the previous sections as inner cells of the model. The whole configuration is specified in Table 4.

Table 4. Configuration of the Network in "V2DV Approach"

Item	Value
Dimension of x_t	713
Dimension of y_t	154
Dimension of s_t	100×6
Number of Layers	6
Type of Cell	Vanilla RNN Cell

4.3. Driving Cycle Generation and Validation

For both approaches the maximum length of each candidate is set to 1000. Candidates generated from both approaches (100 for "V2V Approach" and 2000 for "V2DV Approach") were separately ranked once the models are trained. The final driving cycles obtained are illustrated in Fig. 3 and Fig. 4.

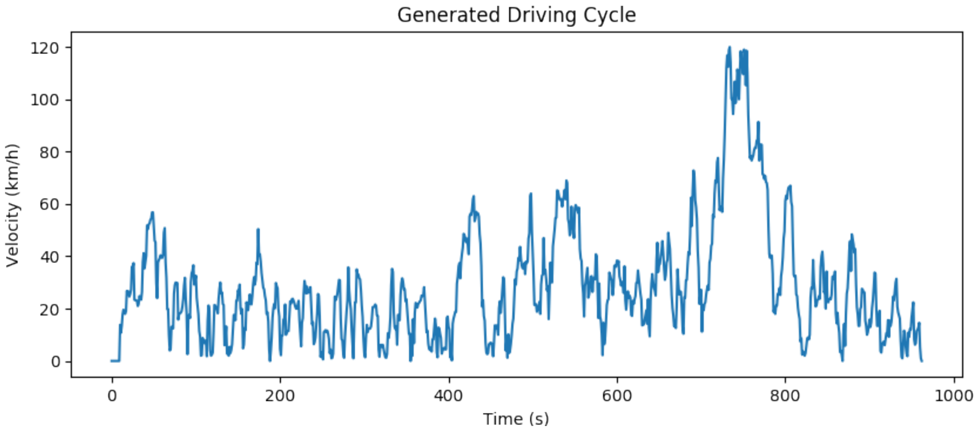

Fig. 3. The Generated Driving Cycle from "V2V Approach"

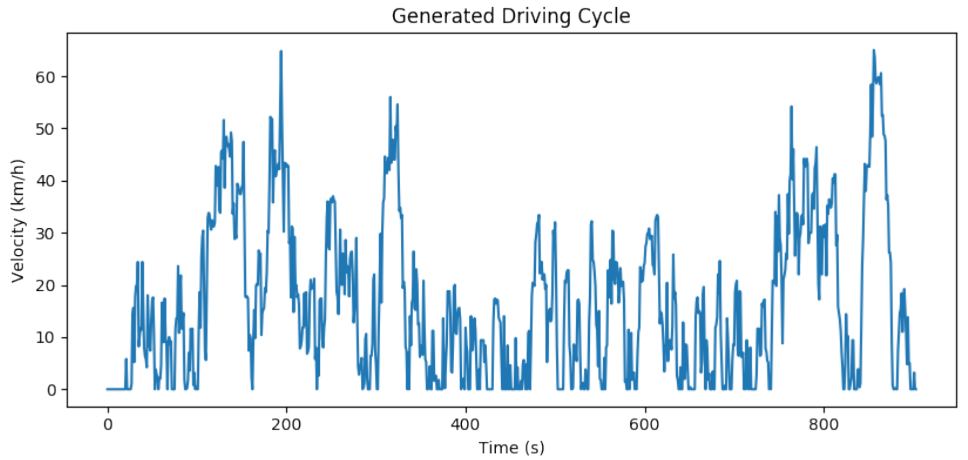

Fig. 4. The Generated Driving Cycle from "V2DV Approach"

And the values of features listed in Table 1 and corresponding relative errors (in parentheses) with respect to the whole data set are shown in Table 5.

4.4. Remarks

As is shown in Fig. 3 and Fig. 4 that while vehicle speed constantly changes over time, both driving cycles contain jerky sub-sequences. As Table 5 suggests, both our candidates had underestimated the fractions in which velocity stays still (e.g. η_{idle}, η_{cruise}), which results in more accelerations and decelerations in the driving cycle compared to the original data set. In terms of consolidating data for LDVs operating at high speeds, the "V2V Approach" outperforms the other with less candidates, as v_{max} of its best candidate reached 120.0 km/h, compared with 65.0 km/h, the counterpart, which makes the former cycle more suitable for evaluating emissions in highway driving.

The error on $\bar{F}_{trac}$ indicates that both candidates are very power demanding, which poses tendencies of overestimation on fuel consumption in vehicle certification. While the α component of MTF, which reflects driving pattern in terms of air drag applied to the vehicle was successfully captured by candidate from "V2V Approach" with an error of 1.5%, results on component β and γ, which indicates the cost for overcoming rolling resistance and inertia, respectively, are of less satisfaction. Further adjustment on both models should be expected to address this issue.

Table 5. Evaluation of Driving Cycle Generated

Feature	V2V Approach	V2DV Approach	Metric
η_{idle}	1.45%(−94.0%)	18.81%(−22.8%)	-
η_{accel}	44.76%(+132.7%)	34.73%(+80.6%)	-
η_{cruise}	14.12%(−62.6%)	12.28%(−67.5%)	-
η_{decel}	39.67%(+113.2%)	34.18%(+83.7%)	-
v_{avg}	28.6(+19.5%)	16.52(−31.0%)	km/h
v_{avgm}	29.0(−8.2%)	20.35(−35.7%)	km/h
v_{max}	120.0(−15.7%)	65.0(−54.4%)	km/h
Δv^{+}_{bound}	14.4(−40.0%)	14.4(−40.0%)	$km/(h \cdot s)$
Δv^{-}_{bound}	16.0(−33.3%)	16.2(−32.5%)	$km/(h \cdot s)$
$\lvert\ddot{v}\rvert_{max}$	8.1(−21.7%)	8.4(−18.5%)	$km/(h \cdot s^2)$
$\bar{F}_{trac}$	600.35(+83.6%)	453.11(+38.6%)	N
α	151.92(+1.5%)	61.04(−59.2%)	m^2/s^2
β	0.63(−21.1%)	0.67(−15.9%)	-
γ	0.33(+257.5%)	0.25(+168.4%)	m/s^2
t_{total}	962	903	s

5. Future Work

In the future, different configurations for the RNN model should be tested to better balance the tradeoff between model complexity and the quality of the final driving cycle. The current jerkiness and high power demand of the candidates resulted from the model should be addressed to make the final candidate more convincing and feasible for vehicle certification related purposes. In addition, the ranking for candidates should be improved by taking into account more features and applying weights for each of them with further analyses to obtain more realistic results.

Acknowledgements

The authors gratefully acknowledge the supports from the project of China Automotive Test Cycles (CATC, 20150339093), National Natural Science Foundation of China (NSFC No. 61472037), and the helpful discussions with Dr. Yu Liu with Institute of Automation, Chinese Academy of Sciences.

References

Barlow, T.J., Latham, S., Mccrae, I.S., Boulter, P.G., 2009. A reference book of driving cycles for use in the measurement of road vehicle emissions.

Bishop, J.D.K., Axon, C.J., Mcculloch, M.D., 2012. A robust, data-driven methodology for real-world driving cycle development. Transportation Research Part D Transport & Environment 17, 389–397.

Cho, K., Merrienboer, B.V., Bahdanau, D., Bengio, Y., 2014. On the properties of neural machine translation: Encoder-decoder approaches. Computer Science .

Fontaras, G., Dilara, P., 2012. The evolution of european passenger car characteristics 2000-2010 and its effects on real-world CO_2 emissions and CO_2 reduction policy. Energy Policy 49, 719–730.

Gong, Q., Midlam-Mohler, S., Marano, V., Rizzoni, G., 2011. An iterative markov chain approach for generating vehicle driving cycles. Sae International Journal of Engines , 1035–1045.

Grüner, J., 2016. A tool for generating individual driving cycles - idcb. Sae International Journal of Commercial Vehicles 9.

Nyberg, P., Frisk, E., Nielsen, L., 2015. Driving cycle equivalence and transformation. IEEE Transactions on Vehicular Technology PP, 1–1.

Nyberg, P., Frisk, E., Nielsen, L., 2016. Using real-world driving databases to generate driving cycles with equivalence properties. IEEE Transactions on Vehicular Technology 65, 4095–4105.

Vaz, W., Landers, R.G., Koylu, U.O., 2014. Neural network strategy for driving behavior and driving cycle classification. International Journal of Electric & Hybrid Vehicles 6, 255–275.

Wang, J., Wang, Q.N., Zeng, X.H., Wang, P.Y., Wang, J.N., 2015. Driving cycle recognition neural network algorithm based on the sliding time window for hybrid electric vehicles. International Journal of Automotive Technology 16, 685–695.