Energy-efficient routing based on vehicular consumption predictions of a mesoscopic learning model

Michail Masikos[a,*], Konstantinos Demestichas[a], Evgenia Adamopoulou[a], Michael Theologou[a]

[a] *National Technical University of Athens, Heroon Polytechneiou 9, Zografou, Athens, Greece*

Abstract

This paper proposes an alternative approach for determining the most energy efficient route towards a destination. An innovative mesoscopic vehicular consumption model that is based on machine learning functionality is introduced and its application in a case study involving Fully Electric Vehicles (FEVs) is examined. The integration of this model in a routing engine especially designed for FEVs is also analyzed and a software architecture for implementing the proposed routing methodology is defined. In order to verify the robustness and the energy efficiency of this methodology, a system prototype has been developed and a series of field tests have been performed. The results of these tests are reported and significant conclusions are derived regarding the generated energy efficient routes.

Keywords:
energy-efficient routing, mesoscopic learning model, FEV, context-aware routing, consumption factor analysis, machine learning, artificial neural networks

1. Introduction

Environmental impact and economic factors impose the need for reducing the amount of energy spent by a vehicle in order to travel from a source to a destination point. Minimizing the consumed fuel leads not only to financial

*Corresponding author
Email address: mmasik@telecom.ntua.gr (Michail Masikos)

savings but also to simultaneous reductions in the released emissions, as their volume is proportional to the vehicles consumption rate [1]. Even in the case of zero emission vehicles, like Fully Electric Vehicles (FEVs), reducing the energy consumption contributes into limiting both the travel cost as well as the environmental impact coming from the generation (in power stations) and transfer of the energy required for vehicle recharging.

Considering the current progress in the development of energy-efficient vehicles, which achieve low consumption rates by means of aerodynamic shapes, energy-efficient engines or the use of alternative fuel resources (e.g. FEVs), further improvements can be enabled through the utilization of information and communication technologies (ICT). Eco-driving and eco-routing techniques implemented by intelligent transportation systems have been proposed for enhancing the vehicles energy efficiency. In particular, eco-driving systems [2, 3] analyze the current status of the vehicle together with consumption-related parameters and provide valuable feedback to the user in order to modify his/her driving style and attitude in an energy-efficient manner. Thus, such systems are quite efficient in reducing a vehicles current energy consumption when following an energy-demanding route (e.g. a route characterized by steep upward slopes), but they do not inform the user beforehand to avoid (if possible) such routes and to follow better ones [4, 5]. This latter task, due to its inherent uncertainty, is not as straightforward and is performed by the so-called eco-routing systems [6, 7].

Eco-routing systems [8] try to identify the most energy-efficient route towards the desired destination based on their estimation about the energy required to travel along each one of all the possible routes and prevent the driver from making a bad choice (i.e. selecting an energy demanding route). The effectiveness, however, of such systems is limited due to the incertainty of predictions that introduces an amount of error into the calculations. In addition to this limitation the existing systems cannot be used in case of FEVs as the FEVs' peculiarities were not taken into consideration during their development. Existing eco-routing systems can be effectively applied only in case of vehicles powered by internal combustion engines. The proposed system aims at tackling both these limitations by implementing an innovative mesoscopic consumption model that minimizes prediction error and an advanced routing engine that is especially designed for FEVs.

Such an eco-routing system is proposed in the present paper, enabling the discovery of the most energy-efficient route towards the destination based on more accurate energy cost predictions. In order to improve the estimation accuracy achieved by existing eco-routing techniques, the proposed system implements a context-aware learning model. Implementing a learning model involves defining an appropriate model that comprises a set of parameters and then optimizing these parameters using past experience [9, 10]. A systematic description of the developed learning model and the introduced routing methodology is presented in the rest of this paper. In particular, Section 2 contains an overview of the existing eco-routing techniques and energy consumption models developed so far. Section 3 describes in detail the development process of the proposed routing methodology and elaborates thoroughly on the introduced learning model. A system architecture suitable for application in FEVs is presented in Section 4. Based on this architecture, a system prototype is implemented and installed in a FEV so as to perform a series of field trials. The prototype implementation as well as the field trials results are reported in Section 5. Finally, Section 6 summarizes the work described in the present paper and emphasizes on the degree of energy efficiency achieved by the proposed methodology.

2. Related work

Several studies have investigated the impact of route choices on the energy consumption and emission rates of vehicles [11, 12, 13, 14, 6, 15, 16]. The common finding of these studies is that following the fastest path towards the destination is not always the best choice from an environmental and energy consumption perspective. For example, comparing the results of a driving experiment performed in Japan [11], the fuel consumption of the ecological route is 9% lower than that of the time priority route, while its travel time is 9% longer. In another experiment performed between the Los Angeles Airport and the Los Angeles center [12], the least fuel consumption route is compared against the shortest duration route. According to this comparison, the least fuel consumption route is 25% more energy efficient and 8% slower than the shortest duration route. Likewise, the results of a field trial performed in the Northern Virginia area [13] demonstrate that significant improvements in energy consumption (18-23%) and air quality (4-5% reduction in NOx and 20% reduction in CO2) can be achieved when motorists utilize a slower and 30% longer arterial route instead of a faster highway route.

Hence, considering the existence of an eco-friendly route as a possible routing choice, several models have been proposed for finding the path that minimizes vehicular consumption. A classification of these models can be based on the type of the methodology employed for predicting the energy consumption along all possible paths towards the destination, which enables their categorization into macroscopic, mesoscopic and microscopic models.

A macroscopic, non-iterative algorithm for estimating vehicular fuel consumption is presented in [17]. The algorithm uses Willan's internal combustion engine model [18] and needs no instantaneous values of speed and acceleration. The efficiency of the proposed algorithm has been verified with measurement results for the following three cycles: motor vehicle expert group (MVEG-95), European driving cycle (ECE), and extra-urban driving cycle (EUDC). Another macroscopic emission estimation tool, called MOBILE6, is utilized in the study performed in [13], and its performance is compared against that of two microscopic tools, i.e. the VT-Micro model and the comprehensive modal emissions model (CMEM). The comparison results of the study, however, demonstrate that macroscopic tools can produce erroneous conclusions given that they ignore transient vehicular behavior along a route.

Transient vehicle states are captured by microscopic models like the one presented in [11]. Authors describe a fuel consumption factor analysis using Oguchis consumption model and identify five factors as major contributors in vehicular consumption, i.e. the base consumption, the friction loss, the altitude change loss, the air drag loss and the acceleration loss. The base fuel consumption factor refers to fuel used for the inertial resistance of the engine and the transmission, the air conditioner and some other electric components, while the other four factors express energy losses due to the vehicles movement. The consumption values estimated by the microscopic tool are, then, fed to a Dijkstra-based [19] routing engine and the most energy-efficient route is extracted. The driving experiments conducted in areas with different geographical features and in various traffic conditions identified base consumption and geographic morphology as the dominant determinants of vehicular fuel consumption.

Apart from deterministic models that are usually based on the laws of Physics [11], microscopic tools include also models that exploit artificial intelligence techniques. The fuel consumption predictive system described in [20, 21] uses a neural network in order to infer vehicular consumption from previously collected experience. The inputs of the network include the brand of the vehicle, the engine type, the vehicle weight, the vehicle class and the transmission system type, while the output corresponds to the vehicles fuel consumption rate indices for each test cycle (i.e. city, highway or combined).

Despite their acceptable performance, microscopic consumption estimation models are quite complex and detailed for application in dynamic route guidance systems. Building a system whose performance depends on the continuous retrieval of microscopic parameters (e.g. instantaneous speed, instantaneous acceleration, or road grade) is not practical, considering that acceptable accuracy can also be achieved by less complex mesoscopic models.

Mesoscopic models estimate emissions and/or fuel consumption on a link basis. Their input parameters reflect average values of observable variables in the context of a time period, e.g. average speed, average acceleration or deceleration, etc. The mesoscopic research tool presented in [7] generates synthetic speed profiles based on historical link speed data, stores them as digital map attributes and uses them for calculating fuel costs per link. Link travel speeds are also employed in the mesoscopic energy consumption model of [22] together with the actual power needed to overcome the driving resistance for each link and with the volume over capacity traffic ratios. In [23] the authors introduce a dynamic eco-route planning system utilizing Dijkstras shortest path algorithm and consisting of a power-dependent consumption model and a dynamic traffic information database. A special mechanism for integrating the impact of dynamic changes of traffic conditions on route planning is described in [12]. In particular, the authors propose an eco-routing navigation system that consists of: a Dynamic Roadway Network database, which is a digital map of a roadway network that integrates historical and real-time traffic information from multiple data sources through an embedded data fusion algorithm; a multivariate regression model that estimates an energy/emissions operational parameter set [fuel, CO_2, CO, HC, NOx] based on vectors of vehicle characteristics, roadway characteristics, traffic characteristics and other explanatory variables; a routing engine, which contains shortest path algorithms used for optimal route calculation; and a user interface that allows the interaction with the user. The reported validation results suggest a reasonable estimation performance; nevertheless, researchers identify some system limitations that may result in errors in the estimated trip fuel consumption and emissions.

Considering the progress achieved in the area of eco-routing, the characteristics of the developed models and the results of the performed sensitivity and empirical analyses (e.g. [24, 25]), the authors of the present paper introduce an innovative mesoscopic approach for energy-efficient routing based on machine learning functionality. In particular, this paper differs from previous work in the following aspects:

- It proposes a context-aware routing methodology that applies a learning model for estimating the vehicular energy consumption. Due to this models retraining capability, the introduced methodology is characterized by increased robustness and continuous adaptability to any contextual change affecting consumption (e.g. degradation of engine performance due to aging or upgrade of a local road to arterial).

- Unlike any other deterministic model for estimating the vehicular energy consumption, the proposed learning model can replicate the non-linear consumption patterns that are hidden in the collected consumption measurements and cannot otherwise be identified.

- The set of contextual parameters selected for learning aggregates all the factors that have been individually identified by previous studies as contributors to vehicular consumption (e.g. [26]). Furthermore, different approaches are proposed for integrating the impact of these factors into vehicular consumption calculations, so as to render possible the on-board installation of the system and its autonomous functionality.

- A suitable hardware and software architecture for implementing the proposed methodology in an autonomous on-board system is described in detail. The benefit from using such an eco-friendly navigational system is of great importance, as according to the exploratory study performed in [11], it can spare an average of 8% fuel in 46% of trips. An on-board prototype system has been developed and installed in a FEV, in order to perform the verification and validation field trials.

- While in previous studies emphasis was given to internal combustion engine vehicles, the proposed methodology is carefully developed in order to address effectively the peculiarities of FEVs.

3. Energy-efficient routing methodology

The core functionality of the proposed energy-efficient routing methodology includes an innovative learning model used for accurately estimating the energy consumption that will incur if the vehicle follows a specific route under a particular contextual frame. In fact, this core functionality can easily be integrated in several widely accepted routing engines (e.g. Dijkstra [19], A* [27], Bellman-Ford [28, 29]). Thus, after thoroughly describing the proposed learning model in the first part of this section, we proceed with its integration in a routing engine especially designed for FEVs. The description of this routing engine and an overview of the entire proposed methodology are included in the second part of this section.

3.1. Vehicular consumption estimation

The proposed mesoscopic consumption model estimates the vehicular consumption on a road link basis. As a road link we consider an edge of a directed graph G=(V,E) (with V={1, 2, ..., n} representing the set of nodes (vertices) and E as the set of arcs defined between each pair of nodes) that resembles the road network.

In order to estimate the vehicular consumption incurred while travelling through a road link one could follow an analytical approach. More specifically, a mathematical formula could be generated after identifying all the factors affecting vehicular consumption and modeling their participation in the total vehicular consumption. Then, this formula could be applied in order to estimate the consumption cost of each link.

In this paper, authors follow a different approach in estimating the vehicular consumption while travelling through a road link. Considering that it is not yet entirely feasible to develop an analytical model for estimating vehicular consumption due to the complexity and nonlinearities that dominate the factors participating in vehicular consumption we propose the development of an appropriate learning model. This model will detect the numerous patterns or regularities that dominate the underlying process by learning from the previously collected experience. Then, assuming that the future, at least the near future, will not differ significantly from the past moments of sample data collection, the proposed model is expected to predict accurately the future vehicular consumptions.

First of all, the selection of the training experience, namely the instance and the target attributes, through which the system will learn has a major influence on the efficiency of the learning model. As instance attributes we consider a number of contextual variables that compose the Vehicular Context ($\vec{\mathcal{V}}$), the Driver Profile ($\vec{D}$), the Weather Context ($\vec{W}$) and the Traffic Conditions ($\vec{T}$), namely the contextual attributes that influence the vehicular consumption. Attributes that refer to the Roadway Context ($\vec{R}$), like the road slope or the road class, are intentionally omitted, because they are taken into account in an implicit manner, since the proposed system considers a separate machine learning engine for each road link. As target attribute, on the other hand, we consider the energy consumed for travelling through the road link under consideration.

The proposed routing engine should be regarded as part of an autonomous on-board navigation system. Such a system can monitor and retrieve the target attribute of the vehicular consumption through the vehicles Controller Area Network (CAN) bus port. Instance attributes should also be selected in such a way so that they can adequately describe the contextual status and can easily be monitored and retrieved. Having this in mind, the following contextual attributes of each group are selected as the instance attributes of the training dataset in case of a FEV:

$$\vec{\mathcal{V}} = (h_b, \ell_b, \vec{s}_{aux}, w_l) \tag{1}$$

$$\vec{D} = (\overline{c}_d) \tag{2}$$

$$\vec{W} = (\theta, RH) \tag{3}$$

$$\vec{T} = (t_d, t_{mo}, t_{hr}) \tag{4}$$

where h_b and ℓ_b are the batterys state-of-health and state-of-charge, respectively, $\vec{s}_{aux}$ is the vector describing the status of the vehicles electric auxiliaries, w_l is the load weight, $\overline{c}_d$ is the drivers average consumption rate calculated by the vehicles trip computer, θ is the temperature , RH is the relative humidity, t_d is the current day of the week, t_{mo} is the current month of the year, and t_{hr} is the current hour of the day.

The context of the driver profile and the context of the traffic conditions are the most challenging to describe. The metric selected to quantify the driver profile is the average consumption rate calculated by the vehicles trip computer. This metric represents adequately the drivers usual driving behavior (i.e. a more or less aggressive attitude) and it can relatively easily be retrieved through the CAN bus port. On the other hand, the existence of recurrent traffic conditions is captured by considering the time window within which the user travels through the road link.

Subsequently, after defining the instance and target attributes of the learning model, an appropriate target function must be selected. As already reported, the target function f of the learning model should estimate the amount of energy required to travel through a specific road segment under specific contextual conditions ($f : C \to \mathbb{R}$). Considering that the energy consumption calculated by the target function is a real value and that the relations among the context factors are non-linear, we proceed with the

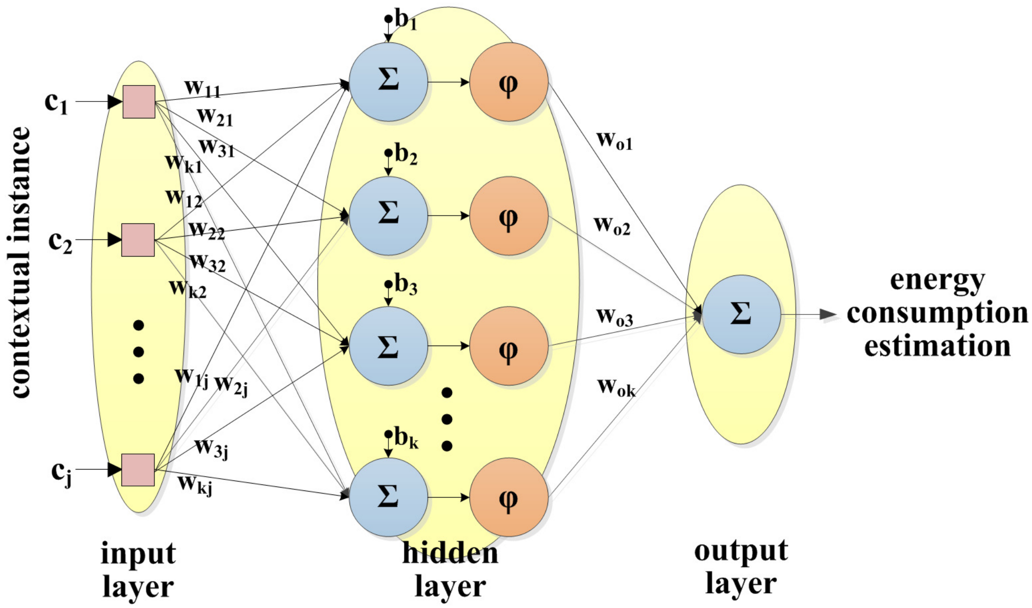

Figure 1: The structure of the MLP network used for estimating the vehicular consumption

selection of the target function based on the outcome of the Universal Approximation Theorem [30]. According to this, a multilayer perceptron (MLP) network with a single hidden layer, which contains finite number of hidden neurons, and a linear combination of the outputs of the hidden neurons as the network output constitutes a universal approximator of any m-dimensional, continuous and non-linear function. Therefore, the MLP network depicted in Figure 1 is proposed for the representation of the target function $\hat{f}$. Based on this network structure the form of the function $\hat{f}$ is:

$$\hat{f}(\vec{C}_0, \vec{w}) = \sum_k w_{ok}\varphi(\sum_j w_{kj}c_j + b_k) \tag{5}$$

The sigmoid activation function $\varphi(.)$ adopted in the proposed MLP network is:

$$\varphi_k\big(u_k(j)\big) = \frac{1}{1 + exp\big(-\alpha u_k(j)\big)} \qquad a > 0 \qquad \text{and} \qquad -\infty < u_k(j) < \infty$$

i.e. the logistic function where α is a constant setting the slope parameter of the sigmoid function and u_k is the weighted sum of the j synaptic input values.

The final step in building the learning model is to compute a uniform ε approximation ($\hat{f}_0$) to a given training set, i.e.

$$|\hat{f}_0(c_1, c_2, ..., c_j) - f(c_1, c_2, ..., c_j)| < \varepsilon \tag{6}$$

for all sets of instance attributes $(c_1, c_2, ..., c_j)$ that lie in the input space.

This is achieved by applying a learning algorithm in the selected target function, i.e. the selected MLP network. The most widespread choice in case of MLPs is the back-propagation algorithm [31], which searches the space of possible hypotheses using gradient descent to iteratively reduce the error in the network's fit to the training dataset. However, in order to accelerate the typically slow rate of convergence experienced with the method of gradient descent, the authors suggest the use of the scaled conjugate gradient descent method [32] that handles the supervised learning as a numerical optimization problem. The objective of the learning process is to adjust the weights of the MLP network so as to minimize the average squared error energy function E_{av} over all (N) examples of the training set:

$$E_{av} = \frac{1}{N}\sum_{n=1}^{N} E(n) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}e_0^2(n) = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{2}(d_0(n) - y_0(n))^2 \tag{7}$$

where e_0 is the error signal at the output neuron, d_0 is the desired response of the output neuron and y_0 is the function signal appearing at the output neuron.

Considering that the error surface of a MLP with supervised learning is a highly nonlinear function of the synaptic weight vector $\mathbf{w}$, the cost function $E_{av}(\mathbf{w})$ can be expanded using the Taylor series about the current point on the error surface $\mathbf{w}(n)$:

$$E_{av} = (w(n)+\Delta w(n)) \simeq E_{av} = \big(w(n)\big)+g^T(n)\Delta w(n)+\frac{1}{2}\Delta w^T(n)H(n)\Delta w(n)+... \tag{8}$$

$$g(n) = \frac{\partial E_{av}(w)}{\partial w}|_{w=w(n)} \tag{9}$$

$$H(n) = \frac{\partial E_{av}^2(w)}{\partial w^2}|_{w=w(n)} \tag{10}$$

where $\mathbf{g}(n)$ is the local gradient vector and $\mathbf{H}(n)$ is the local Hessian matrix.

The scaled conjugate gradient descent method tries to iteratively minimize the quadratic part of the Taylor series expansion of E_{av} in Eq. (8) and is

Table 1: A brief summary of the scaled conjugate gradient descent algorithm

Initialize
1. Select initial weight vector w_0 and scalars $0 < \lambda_0 < 10^{-6}, 0 < \sigma < 10^{-4}, \bar{\lambda}_0 = 0$. Set $r_0 = p_0 = -E'(w_0), k = 0$ and success = true
Iteration k (k=0,1,...)
2. If success = true, calculate the second-order information: $\sigma_k = \frac{\sigma}{\lvert p_k \rvert}$, $s_k = \frac{E'(w_k + \sigma_k p_k) - E'(w_k)}{\sigma_k}$, $\delta_k = p_k^T s_k$
3. Scale s_k: $s_k = s_k + (\lambda_k - \bar{\lambda}_k) p_k$, $\delta_k = \delta_k + (\lambda_k - \bar{\lambda}_k) \lvert p_k \rvert^2$
4. If $\delta_k \leq 0$, make the Hessian matrix positive definite: $s_k = s_k + (\lambda_k - 2\frac{\delta_k}{\lvert p_k \rvert^2}) p_k$, $\bar{\lambda}_k = 2(\lambda_k - \frac{\delta_k}{\lvert p_k \rvert^2})$; $\delta_k = -\delta_k + \lambda_k \lvert p_k \rvert^2$, $\lambda_k = \bar{\lambda}_k$
5. Calculate the step size: $\mu_k = p_k^T r_k$, $\alpha_k = \frac{\mu_k}{\delta_k}$
6. Calculate the comparison parameter: $\Delta_k = 2\delta_k \frac{[E(w_k) - E(w_k + \alpha_k p_k)]}{\mu_k^2}$
7. If $\Delta_k \geq 0$, error can be reduced. Set: $w_{k+1} = w_k + \alpha_k p_k$; $r_{k+1} = -E'(w_{k+1})$; $\bar{\lambda}_k = 0$, success = true. If (k mod N = 0), restart the algorithm: $p_{k+1} = r_{k+1}$ else create new conjucate direction: $\beta_k = \frac{\lvert r_{k+1} \rvert^2 - r_{k+1} r_k}{\mu_k}$, $p_{k+1} = r_{k+1} + \beta_k p_k$. If ($\Delta_k \geq 0.75$), reduce the scale parameter: $\lambda_k = \frac{1}{2}\lambda_k$; else error cannot be reduced. Set: $\bar{\lambda}_k = \lambda_k$, success = false
8. If $\Delta_k < 0.25$, increase the scale parameter: $\lambda_k = 4\lambda_k$
9. If success = true, set: $k = k + 1$, $\bar{\lambda}_{k+1} = \bar{\lambda}_k$, $\lambda_{k+1} = \lambda_k$; return to step 2
Stopping criterion
If the steepest descent direction $\lvert r_{k+1} \rvert < 10^{-6}$, return w_{k+1} and exit

Table 2: Scaled conjucate gradient algorithm parameters

Parameter	Value
Num. convergence tries	4
Maximum iterations	10000
Iterations without improvement	100
Convergence tolerance (ε)	10^{-5}
Min. gradient	10^{-6}

briefly described in Table 1. More specifically, the scaled conjugate gradient algorithm uses a numerical approximation for the second derivatives (Hessian matrix) and it avoids instability by combining the model-trust region approach from the Levenberg-Marquardt algorithm [33] with the conjugate gradient approach. This allows scaled conjugate gradient to compute the optimal step size in the search direction without having to perform the computationally expensive line search used by the traditional conjugate gradient algorithm. Table 2 summarizes the values selected for the parameters that define the stopping criteria of this iterative algorithm. Thus, the iterative process may terminate either successfully (i.e. the convergence tolerance or the minimum gradient is achieved) or without reaching convergence (i.e. the maximum number of either convergence tries or iterations without improvement or total iterations is reached). The scaled conjucate gradient descent method is usually much faster than the traditional steepest descent algorithms using either constant or variable learning rate. The learning rate is replaced with the constant β_k that is calculated by the Levenberg-Marquardt algorithm (Table 1).

According to the inductive learning hypothesis [9], the hypothesis identified by the learning process (due to fitting best the observed data) can reliably predict the vehicle's energy consumption. The results (Table 3) generated after applying the described learning process to the proposed vehicular consumption model justify the inductive learning hypothesis. In particular, according to the first set of results that correspond to the training dataset the proposed model is capable of learning the patterns underlying the vehicular consumption, while according to the second set of results that correspond to the validation dataset the trained model is capable of generalizing from the learned experience.

The benefits of adopting the described learning model for approximating

Table 3: Analysis of MLP model's training and validation results

Metric	Training Set	Validation Set
Mean target value for input data	14.814707	17.150004
Mean target value for predicted values	14.815209	16.734498
Variance in input data	4.626004	5.4132010
Residual variance after model fit	0.0003825	0.0034700
Proportion of variance explained by model (R^2)	99.999%	99.653%
Coefficient of variation (CV)	0.001320	0.025410
Normalized mean square error ($NMSE$)	0.000009	0.000021
Correlation between actual and predicted	0.999995	0.992635
Maximum error (ME)	2.0410452	3.112311
Root Mean Squared Error ($RMSE$)	1.732103	2.000116
Mean Squared Error (MSE)	3.0001825	4.000465
Mean Absolute Error (MAE)	1.859544	2.252533
Mean Absolute Percentage Error ($MAPE$)	0.1255201	0.131343

the vehicular consumption function can be summarized in the following:

- The undefined nonlinearities that exist among the factors contributing into vehicular consumption can be detected.
- Once the training process is complete, the developed system is rendered capable of running in an autonomous fashion.
- The developed system is applicable to any part of the road network and to any contextual conditions prevailing at the time.
- The developed system is intelligent and able to adapt to any environmental change after repeating the learning procedure.
- The developed system is characterized by versatility, i.e. it can model different vehicles.

3.2. Energy-efficient routing

After defining the model for estimating the vehicular consumption we proceed with the formulation of the energy-efficient routing engine. The goal of this routing engine is to find the optimal route that minimizes the energy consumed by the vehicle while travelling from an origin to a destination point.

This route-planning problem can be transformed into the well-known shortest path problem after making the following assumptions:

- the road network is represented by a weighted directed graph G=(V,E,w), i.e. the set of vertices $V = \{u_1, u_2, \ldots, u_n\}$ denotes the nodes of the road network, the set of directed edges $E \subseteq VxV$ denotes the road links and the edge weight function w ($E \rightarrow \mathbb{R}$) denotes the function that calculates the cost associated to each road link (e.g. travel time, length, emissions, consumption, etc.);
- the vehicular consumption costs estimated by the developed model ($\hat{f}$: $C \rightarrow \mathbb{R}$ formulate the edge weight matrix of the graph;
- the vehicular consumption costs are assumed fixed for the time window under consideration. Thus, the dynamic shortest path problem examined is discretized in consecutive static instances that can be resolved by the classic shortest path algorithms for static networks (e.g. Dijkstra, A*);
- the vehicles origin O and destination D points are matched to the corresponding vertices of the graph (u_O and u_D respectively).

Following these assumptions, our goal to find the optimal route that involves the minimum energy consumption while travelling from O to D is translated into:

$$c_{OD} = min\{w(p')|p' \in P_{OD}\} \tag{11}$$

where c_{OD} is the minimum vehicular consumption cost for travelling from O to D, P_{OD} is the set of all the possible paths from O to D, p' is the optimal route and $w(p)$ is the computational function of the cost of any path p. In case of a static network the cost of a path p is:

$$w(p) = \sum_{i=1}^{k} w(u_{i-1}, u_i) \tag{12}$$

where k is the length of the path and $w(u_{i-1}, u_i)$ is the cost of the ith segment of the path.

The most famous algorithm that solves the shortest path problem for a graph with non-negative edge path costs is Dijkstra's algorithm. Assuming that the vehicular consumption incurred while travelling through a road link is nonnegative, this algorithm is applicable to the single pair shortest path problem described in Eq. (11)-(12). This assumption, however, is not always valid in the case of FEVs, which can recuperate energy through regenerative breaking (e.g. the vehicular consumption incurred when a FEV goes downhill on a long road link may be negative). In presence of negative edge path costs, a suitable algorithm for finding the shortest path is the Floyd-Warshall algorithm [34, 35]. This algorithm calculates the costs of the shortest paths between all pairs of a graphs vertices and it can easily reconstruct them after applying a minor modification. However, its cubic time complexity ($\mathrm{O}(|V|^3)$) has motivated the search for other alternatives. Another suitable algorithm for finding the shortest path from O to D is the Bellman-Ford algorithm that runs in $\mathrm{O}(|V|*|E|)$ time. Although the complexity of Bellman-Ford algorithm deteriorates also to $\mathrm{O}(|V|^3)$ in case of dense graphs (where $|E| \to |V|^2$), this probability is eliminated in case of road networks, which are represented by sparse graphs. Thus, for applications in road networks the Bellman-Ford algorithm runs faster than the Floyd-Warshall algorithm, but not as fast as the Dijkstra algorithm.

The fact that, in the present application, the shortest path algorithm might run multiple consecutive times renders the adoption of a fast algorithm more appealing. In order to overcome the limitation set by possible negative edge path costs, the shifting technique described by Johnson in [36] is applied on the networks edge path costs. The requirements for applying this technique on a network are the existence of fixed edge costs and the absence of negative cycles. The former can be tackled through the discretization into consecutive static instances, while the latter can be taken for granted. Otherwise, if, for example, such a round trip route existed, a FEV could end up with a higher battery charge level when returning back to its starting position.

According to Johnson, if the above requirements are satisfied, a suitable function h: $(V \to \mathbb{R})$, exists satisfying the following condition for all nodes u and v:

$$w'(u,v) = w(u,v) + h(u) - h(v) \geq 0 \tag{13}$$

where $w(u,v)$ is the initial edge cost and $w'(u,v)$ is the shifted edge cost. This shifting in edge costs does not affect the structure of shortest paths as the shifting values of the paths intermediate nodes counterpoise each other when calculating the total path cost (Eq. (12)). A potential representation for the function $h(.)$ could be the following:

$$h(u) := d_x(u) \tag{14}$$

where $d_x(u)$ denotes the shortest path distance from an arbitrary node x to node u computed by the Bellman-Ford algorithm.

After generating the matrix with the shifted edge costs, any shortest path algorithm suitable for networks with nonnegative edge path costs may be applied on the network under consideration. Thus, we choose to apply the Fibonacci heap implementation of the Dijkstra algorithm that runs in O($|E|$+$|V|$*log$|V|$) time and it is suitable for sparse graphs. Considering that the described technique includes both the subsequent execution of the Johnson technique as well as the subsequent execution of the Fibonacci heap implementation of the Dijkstra algorithm, it runs in O($|V|$*$|E|$+$|V|$*log$|V|$) time. Compared to the O($|V|$*$|E|$) time complexity of the plain Bellman-Ford algorithm the time complexity of the proposed shifting technique is might be worse, however, in case of multiple runs the first stage of computing the $h(u)$ function has to be executed only once (as long as the contextual frame remains unchanged). Following the Johnsons shifting approach the overall routing routine is briefly summarized in Table 4.

4. System Components and Architecture

The proposed energy-efficient methodology is implemented as part of an autonomous on-board navigation system that provides the driver with extra routing functionalities. The main architecture of this on-board system, i.e. its principal components and their interconnections, are presented in Figure 2. Following a bottom-up approach, the main components comprising the on-board system are: the Travelling Experience Handler, the Energy Efficient Routing Engine, the Navigation Engine, the User Interface and two storage units, i.e. the Travelling Experience Repository and the Machine-Learning (ML) Repository.

The Travelling Experience Handler is responsible for the collection, storage and retrieval of all historical data comprising the travelling experience. While the vehicle travels, this component monitors and retrieves (typically, through the CAN bus port, external sensors such as GPS, and the map

Table 4: A brief summary of the proposed routing routine

Initialize

```
  c_0 := current context ;
  for each edge e ∈ E :
    w_e := f̂_{0,e}(c_0) ; //f̂_{0,e}(c_0) represents the energy consumption learning
    model of the e edge
                    //w_e = w(u, v), each node is determined by a start node u
    and an end node v
  end for
  G' := G(V, E) ∪ ξ; //connect a ξ vertex through a zero distance edge
 with the graph G
  d_ξ(v) := BellmanFord(G', ξ) ; // calculate the distance of the vertex ξ
 to the rest of vertices
```

Routing

```
  if c_0 != current context:
    Initialize();
  end if

  for each vertex v ∈ V:
    shift[v] := d_ξ(v) ; //populate the shifting values array
    dist[v] := infinity ;
    predecessor[v] = undefined ;
    FibHeap.insertNode(infinity) ; //FibHeap is the implemented Fibonacci Heap
  end for
  FibHeap.changeNodeValue(startNode,0);

  while FibHeap.getNumberOfNodes() != 0 :
    currentValue = FibHeap.getMinimumNodeValue() ;
    currentNode = FibHeap.extractMinimumNode() ;
    dist[currentNode] = currentValue ;

    for each adjacentNode of currentNode :
    if dist[adjacentNode] > dist[currentNode] + w(currentNode, adjacentNode) :
      dist[adjacentNode] := dist[currentNode] + w(currentNode, adjacentNode) ;
      predecessor[adjacentNode] := currentNode ;
      FibHeap.changeNodeValue(adjacentNode,dist[currentNode]+
     +w(currentNode, adjacentNode)) ;
    end if
    end for
  end while

    //the predecessor[] contains the shortest paths
    //the dist[] contains the energy costs of the shortest paths
```

database) all the necessary contextual attributes in order to generate the corresponding training records. These records comprise the vehicles collected experience and are stored in the Travelling Experience Repository. Whenever this experience is needed by any other component of the system, it can be retrieved through the Travelling Experience Handler.

The energy-efficient routing functionality proposed in this paper is encompassed by the Energy Efficient Routing Engine. This engine consists of two subcomponents, i.e. the Routing Engine and the ML SubSystem. The ML SubSystem implements the Vehicular Consumption Estimation functionality and it consists of the ML Estimator, the Training Scheduler and the ML Trainer. The ML Trainer generates and trains the vehicular consumption MLP networks and stores them in the ML Repository, while the ML Estimator retrieves the structures of the trained MLP networks from the ML Repository, feeds them with the current contextual instance and estimates the corresponding vehicular consumption. The Training Scheduler, on the other hand, ensures the availability of any trained MLP network required for finding the most energy efficient path towards the destination. Furthermore, it renders the proposed routing technique robust to any contextual or structural change in the road network by monitoring the update status of the training datasets stored in the Travelling Experience Repository and retraining the corresponding MLP networks when appropriate.

The ML functionality provided by the ML SubSystem is utilized by the Routing Engine that implements the core routing functionality. In the current configuration of the Routing Engine component, Johnsons shifting technique is performed, in order to enable the application of the Fibonacci heap implementation of the Dijkstra algorithm. The road segment consumption costs required during the Dijkstra algorithm calculations are predicted by the ML Estimator after feeding it with the appropriate contextual instance. Thus, the Routing Engine component constructs step-by-step the most energy-efficient path towards the destination and provides the final outcome to the Navigation Engine.

The components described previously, namely the Travelling Experience Handler and the Energy Efficient Routing Engine, implement the energy efficient routing methodology proposed in this paper. The rest of the components of the On-board Navigation System depicted in Figure 2, i.e. the Navigation Engine and the User Interface, implement functionalities that are common to all modern navigation systems. The User Interface is a touch friendly interface that allows the interaction of the user with the system.

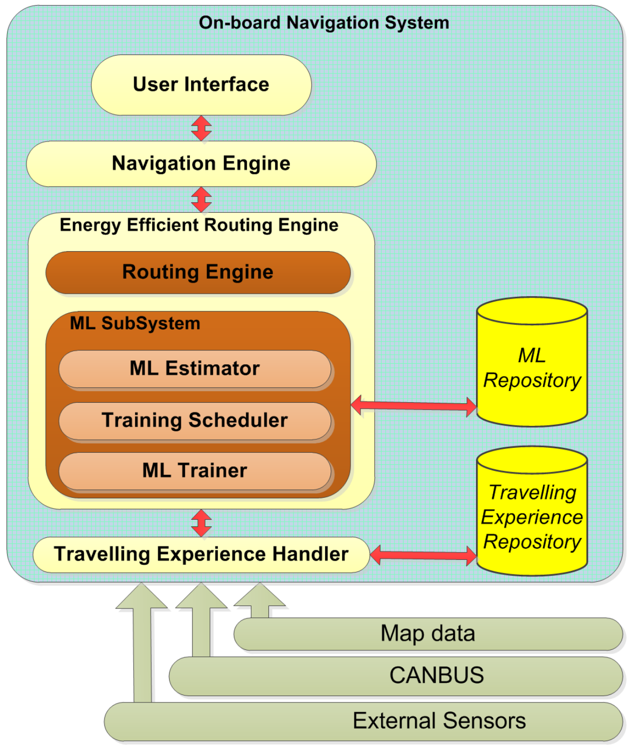

Figure 2: System architecture

This interaction involves changing the settings configuration, choosing a destination, initiating the routing process, initiating the navigation process and examining the map. The Navigation Engine forwards the routing requests to the Energy Efficient Routing Engine, provides vocal and graphical guidance for the calculated routes and monitors the path followed by the vehicle in order to detect any potential detour. In the latter case, it initiates the rerouting process, informing the user accordingly.

Implementing the proposed architecture ensures that the developed system can run in an autonomous fashion. The system is capable of collecting and storing locally, through its own monitoring mechanism, the experience required for training the vehicular consumption ML model. Therefore, there is no need for any external entity in order to render the on-board system functional, although the ability of exchanging travelling experience with other cars would improve the systems learning rate. Finally, concerning the storage capacity limits of the on-board system each repository has its own control mechanism that discards the most outdated records in case there is no space left for new entries, i.e. the oldest training records of the most frequently visited road segments in case of the Travelling Experience Repository, and the MLP structure records of the less frequently visited road segments in case of the ML Repository. Thus, the system is designed to be both robust and efficient.

5. Performance Evaluation

5.1. Prototype Implementation

To validate and evaluate the performance of the proposed routing methodology, a prototype system has been developed implementing the architecture described in the previous section. In particular, the system components and interfaces were implemented in C# .NET, while the system repositories were implemented in MS SQL database schemas. For the required core ML functionality the prototype implementation exploited the COM library provided by a commercial software package, namely the DTREG predictive modeling software [37].

The developed components were, afterwards, installed in a fan-less embedded computer with high connectivity capabilities especially designed for in-vehicle applications [38]. A dual-core mobile processor, a solid state storage disk, memory and storage stabilizers to withstand the challenges of high vibration, 3G connectivity and several external ports are the main characteristics that render this computer ideal for our prototype implementation. The user interaction with the system was enabled via a 7" LCD touch screen that was attached to the windshield and connected to the embedded computer.

The next step involved the installation of the prototype system in a probe vehicle. The vehicle selected for performing the evaluation campaign was a standard FEV with no special configuration. The characteristics of this vehicle can be summarized as follows: rear electric drive, permanent magneto synchronous engine, weight of 990kgr (sum of vehicle mass and battery weight), battery capacity of 22kWh, charge time of 8 hours, top speed of 120km/h and average range of 140km. Furthermore, another important characteristic of the selected vehicle was the availability of a CAN bus socket. The prototype system was attached to this port through the Vector Box VN1610 device [39], which is an adaptor with a USB interface for the CAN bus protocol. The integration of the embedded computer in the vehicle was finalized with the connection of the power supply and the connection of an external GPS receiver.

5.2. Field trial statistics

A field trial campaign for verifying the developed prototype and evaluating the performance of the proposed methodology was carefully planned and performed in the area of Chieri, a town located near Turin, Italy. This area features a mesh road network structure and variable traffic conditions. An overview of this area together with a representation of the spatial distribution of the measurements collected during the field trials are presented in Figure 3.

Figure 3: The spatial distribution of measurements collected during the field trials

The field trial campaign consisted of two parts: the data collection campaign and the performance evaluation campaign. The purpose of the data collection campaign was to collect adequate measurements for training the MLP networks required by the proposed routing methodology. Due to the lack of past experience, the probe vehicle followed random routes in as many

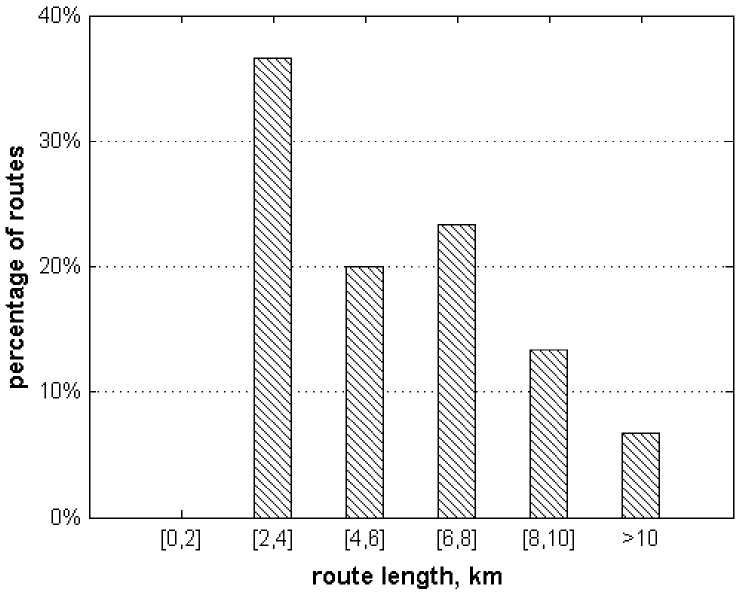

Figure 4: The frequency distribution of the length of the routes selected for the performance evaluation campaign

Table 5: MLP network parameters

Parameter	Value
No. of neurons (hidden layer)	20
Slope parameter	-0.8

different context instances as possible so as to enhance the quality of the training dataset. Of course, when applying the proposed methodology in a final product the required experience is accumulated through every day travelling. During a six-week period, the probe vehicle travelled approximately 4880 km and collected 16 MB of data corresponding to 168060 different contextual instances (93 bytes/record). A total of 2436 MLP networks corresponding to 2436 links of the road network depicted in Figure 3 were trained using the generated training dataset. Table 5 presents the values of the MLP networks' parameters.

On the other hand, the purpose of the performance evaluation campaign (which followed the training data collection campaign) was twofold, namely to verify the robustness and the stability of the developed prototype and to validate the efficiency of the proposed methodology. 40 different pairs of origin and destination points were selected in the greater area of Chieri depicted in Figure 3. Two routes were generated for each pair of points, i.e. the "energy efficient" route proposed by the developed methodology and the "fastest path" route proposed by a conventional navigation system, and their estimated metrics (energy consumption, travel time, route length) were recorded. Regarding the energy consumption metric, the estimations were performed by the proposed methodology when calculating the energy efficient route and by a conventional methodology, based on the routes length and the vehicles specified average consumption, when calculating the fastest path route. Then, the test vehicle was carefully driven to follow precisely both routes for each pair of origin and destination points and the corresponding actual metrics were recorded. In order for the metrics of each pair of routes to be comparable, both of the routes were travelled after ensuring the same context frame. Figure 4 presents the frequency distribution of the length of the generated routes. Considering the size of the testing area the length of the majority of the routes is less than 10 km.

Concerning the efficiency of the proposed methodology an analytical study on the measurements collected during this part of the field trials was performed and the results are presented in the following section.

5.3. Field trial results

The validation of the efficiency of the proposed methodology involves the validation of the implemented learning functionality and the validation of the developed routing engine.

In order to measure the accuracy of the implemented learning functionality, the energy consumption estimations of the generated energy efficient routes were compared to the corresponding values monitored by the testing vehicle while travelling these routes. Figure 5 presents both graphically and numerically the deviation of the energy consumption estimations from the corresponding actual values. The metric denoted as MPE stands for Mean Percentage Error and it is the computed average of percentage errors by which the forecasts (estimations) of a model differ from actual values of the quantity being forecasted. The calculated MPE of 0.85% indicates that our implemented learning functionality is quite accurate and unbiased. Furthermore, the slightly positive value of MPE indicates that the energy consumption is usually slightly overestimated. This conclusion is important

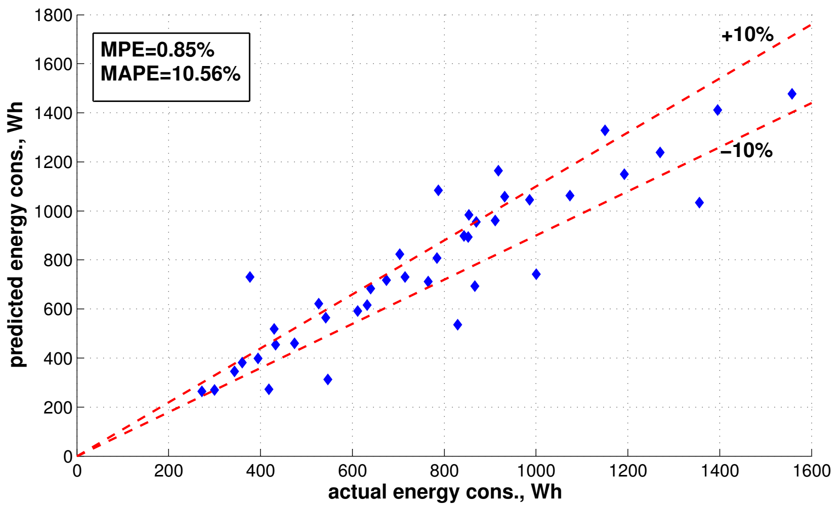

Figure 5: Performance metrics for the prediction accuracy of the proposed vehicular consumption model

as it helps ensure that the vehicle will reach its destination even in case of critical battery level. While calculating the MPE, the positive and the negative forecast errors can offset each other. For that reason, the MAPE metric, which stands for Mean Percentage Absolute Error, was also calculated. This metric is computed by summing the absolute differences between the forecasts and the corresponding actual measured values and by finally dividing the sum with the number of considered pairs (forecast-actual value). Considering the complexity of the consumption prediction task, the calculated MAPE value of 10.56% can be regarded as a successful result.

This conclusion is further established if we calculate and evaluate the corresponding metrics for the predictions performed by a reference vehicular consumption model. The reference model considered here is applied in several commercially available navigation systems and it estimates the vehicular consumption based on the vehicle's average consumption rate (Wh/km) and on the road segment's length (km). The results are presented in Figure 6.

The fact that the calculated MAPE (Figure 6) is two times greater than the one calculated previously (Figure 5) indicates that the improvement introduced in the energy consumption estimation by the proposed model is quite significant (improvement by a factor of two) (Table 6). The value of the MPE, on the other hand, is very close to the MAPE value. Therefore, the energy consumption estimation is almost always biased with a quite large

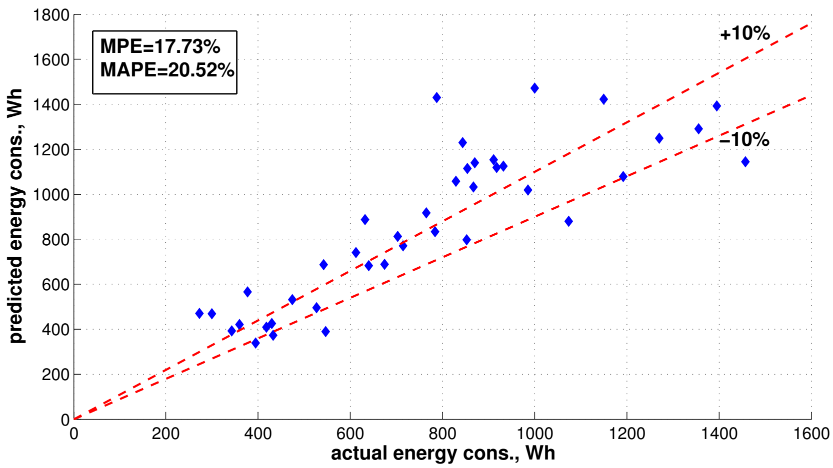

Figure 6: Performance metrics for the prediction accuracy of a reference vehicular consumption model

Table 6: Reference vehic. consumption model vs. Proposed vehic. consumption model

	Reference model	Proposed model
MPE	17.73%	0.85%
MAPE	20.52%	10.56%

percentage error.

After validating the accuracy of the energy consumption predictions performed by the proposed model against those performed by the reference model, we proceed with the validation of the developed routing engine. In particular, the efficiency of the developed routing engine is evaluated through the comparison of the characteristics (consumed energy, length, travel time) of the generated energy-efficient routes against the ones of the corresponding fastest path routes generated by a conventional navigation system. Several pairs of "energy-efficient" and "fastest path" routes are generated and travelled (as already explained in section 5.2) and the relative frequency distributions of the comparison results are presented in the following diagrams.

The field trial results depicted in the diagram of Figure 7 validate the energy efficiency of the routes generated by the developed prototype system for all pairs of origin and destination points tested. This diagram presents both the energy savings estimated initially and the ones realized when following the generated routes instead of the corresponding fastest path routes with

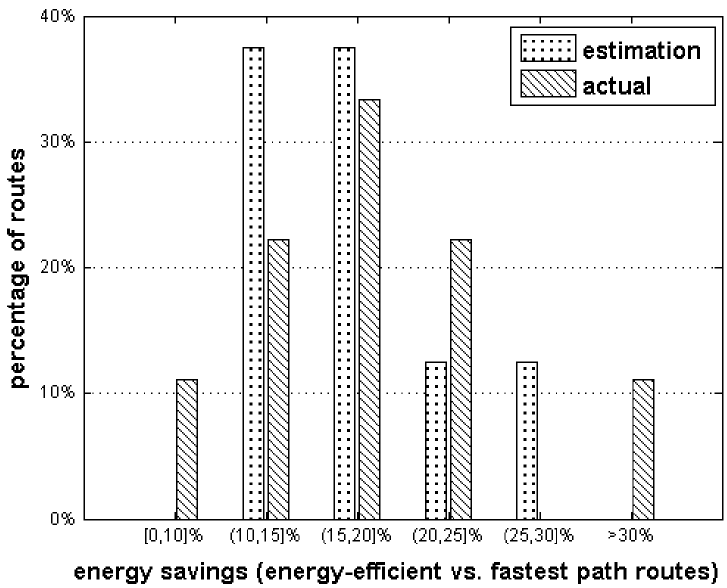

Figure 7: Relative frequency diagram for the estimated and the realized energy savings incurred when following the energy efficient routes instead of the corresponding fastest path routes

the testing vehicle. Based on the energy savings frequency distribution of the initial estimations, the energy efficient route spares on average 18.34%, and at least 15% energy in 65% of the tests. Furthermore, in the rest 35% of the tests, the calculated energy savings are at least 10% of the total energy required in case of the conventional fastest routing.

From the comparison of the estimated energy savings with the realized ones it is deduced that the latter are slightly higher than the former with a mean value of 20.69%. This observation further justifies the outcome extracted while validating the learning functionality efficiency that the proposed system usually overestimates slightly the energy consumption. Overstimating slightly the energy consumption along a route means that the actual energy required to travel along the particular route is less. Therefore, when the driver selects a route proposed by the developed system, he will actually save more energy than the amount initially estimated for that route.

Studying the length of the energy efficient routes against the length of the corresponding fastest path routes, we reach the results depicted in Figure 8. Based on this diagram, the energy efficient routes are usually (in around 40%

of the cases) up to 5% longer than the corresponding fastest path routes. However, there are a few cases referring to energy efficient routes, which follow quite longer paths (up to 15%) consisting of road segments with better energy related characteristics (e.g. slope, traffic conditions). The major outcome is that shorter routes are not necessarily more energy efficient, which proves that it is important to take into account the identified contextual parameters in the routing process.

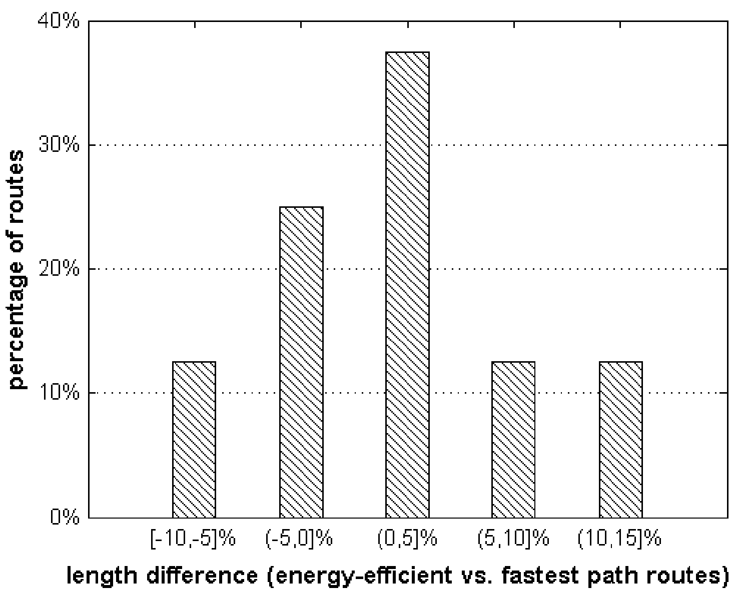

Figure 8: Relative frequency diagram for the comparison of the length of the energy efficient routes against the corresponding fastest path routes

Another important study involves the comparison of the time needed to travel through an energy efficient route instead of the corresponding fastest path route. During the field trials, the realized travel time was recorded for both the energy-efficient and the corresponding fastest path routes and the comparison results are presented in Figure 9. According to the depicted distribution diagram, in approximately 16% of the cases the route generated by the proposed methodology was not only more energy-efficient but also faster than the corresponding fastest path route generated by a conventional navigation system. However, it is not safe to conclude with certainty that in these cases the most energy-efficient route is identical to the fastest path route. There might be an even faster route, for the pair of origin and destination points under consideration, that has not been identified due to errors in travel time predictions. On the other hand, in the rest of the cases, a travel time loss occurred when the FEV followed the energy-efficient route, indicating possibly that these routes consisted of road segments with better energy related characteristics (e.g. slope) but lower speed limits. However, after considering the average realized travel time overhead of 10.26% against the average realized energy savings of 20.69%, the user might have a second thought on selecting the fastest path route, depending of course on the circumstances. As far as FEVs are concerned, which currently present heavy limitations regarding their battery capacity and their recharging process duration, selecting the generated energy-efficient route seems to be beneficial based on the extracted results.

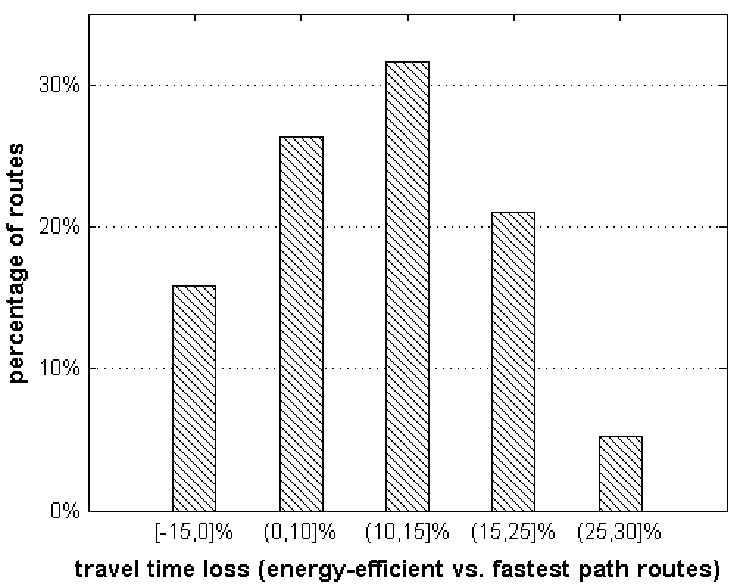

Figure 9: Relative frequency diagram for the time loss realized when following the energy efficient routes instead of the corresponding fastest path routes

6. Conclusions

After providing an overview for the current status in the area of eco-friendly systems (i.e. eco-driving and eco-routing mechanisms), this paper proceeded with the development of an on-board energy-efficient routing system especially designed for FEVs. Instead of validating the efficiency of the proposed methodology through simulations, a prototype was developed and a series of field tests were performed after installing the developed prototype in a testing FEV.

The results for these tests were quite promising for the proposed context-aware routing methodology. In particular, the introduced mesoscopic vehicular consumption model is quite accurate and unbiased as indicated by the MPE value of 0.85% that was measured during the conducted tests. The results concerning the functionality of the developed routing engine were also positive. Although the generated energy efficient routes were on average 1.45% longer in distance and 10.26% longer in travel time than the corresponding fastest path routes, we believe that the achieved average energy savings of 20.69% compensate adequately for this overhead in route distance and travel time. In order to further strengthen this argument, we present a numerical example. Firstly, we choose the median of the fastest path routes in our field tests, which is a route with a length of 5944 m, travel time of 13 minutes and required energy of 816 Wh. Based on the average values reported before, the expected characteristics of the corresponding energy efficient route generated by the proposed methodology are: length of 6030 m, travel time of 14.3 minutes and required energy of 647.17 Wh. This example reveals that the travel time loss (1.3 minutes) and the distance overhead (86m) are less important compared to the realized energy savings (168.83Wh), which are adequate for moving a FEV (with an average consumption of 154Wh/km) for approximately 1096 m more (i.e., the range of the testing FEV can be extended on average by 18%). Hence, we believe that these findings render the proposed energy routing methodology valuable for everyday use.

Future studies include the investigation of other machine learning algorithms (e.g. Generalized Rregression Neural Networks) for estimating the vehicular consumption. The adoption of more complex algorithms such as GRNNs might improve the achieved estimation accuracy. However, the extra delay introduced due to their increased complexity might not allow their application in on-board navigation systems performing real-time calculations.

Acknowledgment

This work is performed under project EMERALD, which has received research funding from the EU Seventh Framework Programme. This paper reflects only the authors views, and the Community is not liable for any use that may be made of the information contained therein.

References

[1] X. Ma, W. Lei, I. Andréasson, H. Chen, An Evaluation of Microscopic Emission Models for Traffic Pollution Simulation Using On-board Measurement, Environmental Modeling & Assessment 17 (2012) 375–387.

[2] M. A. S. Kamal, M. Mukai, J. Murata, T. Kawabe, Ecological Vehicle Control on Roads With Up-Down Slopes, Intelligent Transportation Systems, IEEE Transactions on 12 (2011) 783–794.

[3] L. Nouveliere, S. Mammar, H.-T. Luu, Energy saving and safe driving assistance system for light vehicles: Experimentation and analysis, in: Networking, Sensing and Control (ICNSC), 2012 9th IEEE International Conference on, 2012, pp. 346–351. doi:10.1109/ICNSC.2012.6204942.

[4] M. S. Alam, A. McNabola, A critical review and assessment of Eco-Driving policy & technology: Benefits & limitations, Transport Policy 35 (2014) 42–49.

[5] F. Mensing, E. Bideaux, R. Trigui, J. Ribet, B. Jeanneret, Eco-driving: An economic or ecologic driving style?, Transportation Research Part C: Emerging Technologies 38 (2014) 110–121.

[6] E. Ericsson, H. Larsson, K. Brundell-Freij, Optimizing route choice for lowest fuel consumption Potential effects of a new driver support tool, Transportation Research Part C: Emerging Technologies 14 (2006) 369–383.

[7] C. F. Minett, A. M. Salomons, W. Daamen, B. Van Arem, S. Kuijpers, Eco-routing: Comparing the fuel consumption of different routes between an origin and destination using field test speed profiles and synthetic speed profiles, in: Integrated and Sustainable Transportation System (FISTS), 2011 IEEE Forum on, 2011, pp. 32–39. doi:10.1109/FISTS.2011.5973621.

[8] Y. M. Nie, Q. Li, An eco-routing model considering microscopic vehicle operating conditions, Transportation Research Part B: Methodological 55 (2013) 154–170.

[9] T. M. Mitchell, Machine Learning, 1 ed., McGraw-Hill, Inc., New York, NY, USA, 1997.

[10] V. N. Vapnik, The nature of statistical learning theory, Springer-Verlag New York, Inc., New York, NY, USA, 1995.

[11] T. Kono, T. Fushiki, K. Asada, K. Nakano, Fuel Consumption Analysis and Prediction Model for "Eco" Route Search, in: 15th World Congress on Intelligent Transport Systems and ITS America's 2008 Annual Meeting, New York, NY, USA, 2008.

[12] K. Boriboonsomsin, M. J. Barth, W. Zhu, A. Vu, Eco-Routing Navigation System Based on Multisource Historical and Real-Time Traffic Information, Intelligent Transportation Systems, IEEE Transactions on 13 (2012) 1694–1704.

[13] K. Ahn, H. Rakha, The effects of route choice decisions on vehicle energy consumption and emissions, Transportation Research Part D: Transport and Environment 13 (2008) 151–167.

[14] J. Kang, T. Ma, F. Ma, J. Huang, Link-based emission model for eco routing, in: ITS Telecommunications (ITST), 2011 11th International Conference on, 2011, pp. 207–212. doi:10.1109/ITST.2011.6060054.

[15] J. Bandeira, T. G. Almeida, A. J. Khattak, N. M. Rouphail, M. C. Coelho, Generating Emissions Information for Route Selection: Experimental Monitoring and Routes Characterization, Journal of Intelligent Transportation Systems 17 (2012) 3–17.

[16] K. Ahn, H. Rakha, Field Evaluation of Energy and Environmental Impacts of Driver Route Choice Decisions, 2007. doi:10.1109/ITSC.2007.4357661.

[17] I. Ben Dhaou, Fuel estimation model for ECO-driving and ECO-routing, in: Intelligent Vehicles Symposium (IV), 2011 IEEE, 2011, pp. 37–42. doi:10.1109/IVS.2011.5940399.

[18] S. Pachernegg, A Closer Look at the Willans-Line, SAE Technical Paper (1969).

[19] E. W. Dijkstra, A note on two problems in connexion with graphs, Numerische Mathematik 1 (1959) 269–271.

[20] J.-D. Wu, J.-C. Liu, Development of a predictive system for car fuel consumption using an artificial neural network, Expert Systems with Applications 38 (2011) 4967–4971.

[21] J.-D. Wu, J.-C. Liu, A forecasting system for car fuel consumption using a radial basis function neural network, Expert Systems with Applications 39 (2012) 1883–1888.

[22] K. Kraschl-Hirschmann, M. Fellendorf, Estimating energy consumption for routing algorithms, Intelligent Vehicles Symposium (IV), 2012 IEEE (2012) 258–263.

[23] E. Yao, Y. Song, Study on Eco-Route Planning Algorithm and Environmental Impact Assessment, Journal of Intelligent Transportation Systems 17 (2013) 42–53.

[24] H. C. Frey, K. Zhang, N. M. Rouphail, Fuel Use and Emissions Comparisons for Alternative Routes, Time of Day, Road Grade, and Vehicles Based on In-Use Measurements, Environmental Science & Technology 42 (2008) 2483–2489.

[25] J. Bandeira, D. O. Carvalho, A. J. Khattak, N. M. Rouphail, M. C. Coelho, A Comparative Empirical Analysis of Eco-friendly Routes during Peak and Off-peak Hours, in: Transportation Research Board 91st Annual Meeting, Transportation Research Board, Washington DC, 2012.

[26] S. Carrese, A. Gemma, S. La Spada, Impacts of Driving Behaviours, Slope and Vehicle Load Factor on Bus Fuel Consumption and Emissions: A Real Case Study in the City of Rome, Procedia - Social and Behavioral Sciences 87 (2013) 211–221.

[27] P. E. Hart, N. J. Nilsson, B. Raphael, A Formal Basis for the Heuristic Determination of Minimum Cost Paths, Systems Science and Cybernetics, IEEE Transactions on 4 (1968) 100–107.

[28] R. Bellman, On a Routing Problem, Quarterly of Applied Mathematics 16 (1958) 87–90.

[29] L. R. Ford, Network Flow Theory, Technical Report, RAND Corporation, Santa Monica, CA, 1956. URL: http://www.rand.org/pubs/papers/P923.

[30] S. Haykin, Neural Networks: A Comprehensive Foundation, 2nd ed., Prentice Hall PTR, Upper Saddle River, NJ, USA, 1998.

[31] D. E. Rumelhart, G. E. Hinton, R. J. Williams, Learning representations by back-propagating errors, Nature 323 (1986) 533–536.

[32] M. F. Mø ller, A scaled conjugate gradient algorithm for fast supervised learning, Neural Networks 6 (1993) 525–533.

[33] R. Fletcher, Practical Methods of Optimization; (2Nd Ed.), Wiley-Interscience, New York, NY, USA, 1987.

[34] R. W. Floyd, Algorithm 97: Shortest path, Commun. ACM 5 (1962) 345—-.

[35] S. Warshall, A Theorem on Boolean Matrices, J. ACM 9 (1962) 11–12.

[36] D. B. Johnson, Efficient Algorithms for Shortest Paths in Sparse Networks, J. ACM 24 (1977) 1–13.

[37] DTREG, DTREG-Software For Predictive Modeling and Forecasting, 2013. URL: http://www.dtreg.com/.

[38] ADLINK, ADLINK MXE-5302 device, 2013. URL: http://www.adlinktech.com/PD/web/PD_detail.php?cKind=&pid=1103.

[39] Vector, Vector Box VN1610 device, 2013. URL: http://vector.com/vi_vn1600_en.html.