A Study on Several Machine Learning Methods for Estimating Cabin Occupant Equivalent Temperature

Diana Hintea, James Brusey and Elena Gaura
Coventry University, Priory Lane, Coventry, CV1 5FB, U.K.

Keywords: Equivalent Temperature, HVAC Control, Machine Learning, Parameter Estimation.

Abstract: Occupant comfort oriented Heating, Ventilation and Air Conditioning (HVAC) control rises to the challenge of delivering comfort and reducing the energy budget. Equivalent temperature represents a more accurate predictor for thermal comfort than air temperature in the car cabin environment, as it integrates radiant heat and airflow. Several machine learning methods were investigated with the purpose of estimating cabin occupant equivalent temperature from sensors throughout the cabin, namely Multiple Linear Regression, MultiLayer Perceptron, Multivariate Adaptive Regression Splines, Radial Basis Function Network, REPTree, K-Nearest Neighbour and Random Forest. Experimental equivalent temperature and cabin data at 25 points was gathered in a variety of environmental conditions. A total of 30 experimental hours were used for training and evaluating the estimators' performance. Most machine learning tehniques provided a Root Mean Square Error (RMSE) between 1.51 °C and 1.85 °C, while the Radial Basis Function Network performed the worst, with an average RMSE of 3.37 °C. The Multiple Linear Regression had an average RMSE of 1.60 °C over the eight body part equivalent temperatures and also had the fastest processing time, enabling a straightforward real-time implementation in a car's engine control unit.

1 INTRODUCTION

Vehicle HVAC systems aim to ensure that passengers are thermally comfortable. However, thermal comfort is influenced by a large number of environmental variables and, furthermore, thermal preferences can vary greatly between individuals due to physiological, behavioural and cultural factors.

Nilsson's equivalent temperature-based model was shown to provide the highest correlation scores with subjective occupant comfort data in Hintea et. al (Hintea et al., 2014). Although equivalent temperature is shown to be necessary for estimating thermal comfort, it cannot feasibly be measured in real-time in a manufactured vehicle. A solution to this is virtual sensing. Virtual sensing is applied in a variety of domains (Wenzel et al., 2007; Way and Srivastava, 2006; Srivastava et al., 2005) and the idea behind the concept of vehicular virtual thermal comfort sensing is that, based on data from a set of cabin environmental sensors, readings from virtual sensors (equivalent temperature sensors in this case) are inferred. Therefore, a method that estimates occupant body part equivalent temperatures from a minimalistic set of inexpensive cabin environmental sensors is proposed, consisting of two stages. First, using a mutual information-based approach, the set of cabin environmental sensors that correlate well with the body part equivalent temperatures is selected. Second, a machine learning approach is applied to infer the occupant body part equivalent temperatures from the previously selected cabin environmental sensors.

The purpose of this paper is to establish, based on empirical data, which of seven different machine learning methods (Multiple Linear Regression (MLR), MultiLayer Perceptron (MLP), Multivariate Adaptive Regression Splines (MARS), Radial Basis Function Network (RBF), REPTree, K-Nearest Neighbor (KNN) and Random Forest (RF)) is the most suitable for cabin occupant equivalent temperature estimation based on the estimation accuracy provided and the processing time required for the estimation.

The paper is structured as follows: Section 2 reviews the machine learning techniques used for estimating occupant equivalent temperature. Section 3 presents the experimental data sets gathered for evaluation purposes, while Section 4 presents the results obtained through training and testing of the presented estimators and a comparison of their performance. Fi-

Hintea D., Brusey J. and Gaura E..
A Study on Several Machine Learning Methods for Estimating Cabin Occupant Equivalent Temperature.
DOI: 10.5220/0005573606290634
In *Proceedings of the 12th International Conference on Informatics in Control, Automation and Robotics* (ICINCO-2015), pages 629-634
ISBN: 978-989-758-122-9

nally, Section 5 concludes the paper.

2 MACHINE LEARNING ALGORITHMS

Several machine learning methods were implemented and evaluated for estimating equivalent temperature, namely MLR, MLP, MARS, RBF, REPTree, KNN and RF. They are presented in the following subsections.

2.1 Multiple Linear Regression

MLR (Draper and Smith, 1981) models the relationship between a response variable (the variable we want to provide an estimate for) and two or more explanatory variables (the variables from which the estimate is performed) by fitting a linear equation to the observed data. MLR was implemented in Python.

2.2 Multilayer Perceptron

MLP (Haykin, 1998) is a feed-forward artificial neural network model that consists of multiple layers of nodes and maps the input data onto an appropriate output. The back-propagation technique is used for training the network. The estimator was implemented in Python using WEKA libraries.

2.3 K-Nearest Neighbour

KNN (Cover and Hart, 1967) represents an instance-based lazy learning method considering the closest training examples in the feature space. The method relies in classifying an object by the majority vote of its neighbours. The object is assigned to the most common class amongst its k nearest neighbours, with k being a positive (typically small) integer. In the case of $k = 1$, the object is assigned to the class of the single nearest neighbour. The estimator was implemented in Python using the Orange software libraries.

2.4 Multivariate Adaptive Regression Splines

MARS (Friedman, 1991; Hastie et al., 2009) is a non-parametric regression technique that models non-linearities and interactions between variables. The basis functions, together with the model parameters (estimated with the least squares estimation method), are combined to predict the inputs. The estimator was implemented in Python using the Orange software libraries.

2.5 Radial Basis Function Network

RBF (Haykin, 1998) is an Artificial Neural Network that uses radial basis functions as activation functions. A RBF network consists of inputs, a hidden layer of basis functions and outputs. At the input of each neuron, the distance between the neuron centre and the input vector is calculated. The output of the neuron is then formed by applying the basis function to this distance. The RBF network output is formed by a weighted sum of the neuron outputs and the unity bias. Usually, RBF networks are complemented with a linear part. This corresponds to additional direct connections from the inputs to the output neuron. The estimator was implemented in Python using WEKA libraries.

2.6 Reduced Error Pruning Tree

REPTree (Witten and Frank, 2005; Quinlan, 1986) is a fast decision tree learner that builds a regression tree using information gain reduction and pruning. REPTree yields a sub-optimal tree under the restriction that a sub-tree can only be pruned if it does not contain a sub-tree with a lower classification error than itself. More accurate performance can be obtained at a higher computational cost. The estimator was implemented in Python using WEKA libraries. REPTree is usually used as a classifier, however it allows the selection of numerical outputs and, therefore, be used as an estimator.

2.7 Random Forest

RF (Breiman, 2001) is an ensemble classifier that consists of multiple decision trees and outputs the class produced by the largest number of individual trees. The estimator was implemented in Python using the Orange software libraries.

3 EXPERIMENTAL DATA SETS GATHERING

The test car used for the experimental data gathering was a Jaguar XJ (2010 model year). The INNOVA Flatman support manikin[1] was placed in the front passenger seat. Throughout the experimental trials, equivalent temperature was measured in real-time at eight locations (corresponding to head, chest,

[1]LumaSense Technologies The INNOVA "Flatman" Manikin: http://www.lumasenseinc.com/EN/products/thermal-comfort/flatman/the-manikin-innova-flatman.html

left lower arm, right lower arm, left upper arm, right upper arm, thigh and calf) using dry heat loss sensors attached to the Flatman and connected to an INNOVA thermal comfort data logger.

For the development and evaluation of the estimation methods, cabin environmental parameters were also measured, as follows:

1. Air temperature and relative humidity at six points (head, chest and feet level of the occupants, both on the left and right side) using type T thermocouples and Honeywell S&C HIH-5031 humidity sensors[2].
2. Solar loading at the driver sunroof using automotive solar sensors.
3. Cabin air and surface temperatures at 19 cabin points using type K thermocouples.
4. Driver centre and outboard face vent air temperatures using type K thermocouples.

Three types of trials (a total of 70 individual trials) were performed, as described below.

3.1 Variable Cabin Temperatures with Steady State External Conditions

The trials were performed within an enclosed space, characterised by stable ambient air temperature, in order to avoid the effects of wind and sun. The subjects were pre-conditioned to 22 °C in a separate room for 20 minutes. The test car cabin was also pre-conditioned to 22 °C. The subject entered the car and remained in static conditions (HVAC set-point of 22 °C and air flow set on medium or high as per trial) for 10 minutes. The HVAC set-point temperature was then increased by 1 °C every 3 minutes until it reached 28 °C.

The subject then left the car and was again preconditioned to 22 °C, as was the car cabin. The subject entered the car, again remaining in static conditions (HVAC set-point of 22 °C) for 10 minutes. The HVAC set-point temperature was decreased by 1 °C every three minutes until it reached 16 °C. This procedure was performed four times per subject, with each combination of medium and high air flow and with and without solar loading on the driver side of the car.

3.2 User Control with Steady State External Conditions

The purpose of these trials was to gain knowledge of the HVAC inputs performed by the subjects in order to reach a comfortable temperature, starting with several pre-conditioning temperatures. These trials were also performed with the vehicle in an enclosed space, characterised by stable ambient air temperature and shielded from the wind and sun. The car cabin and the subjects were pre-conditioned to a neutral (22 °C), hot (28 °C), or cold (16 °C) temperature for 20 minutes prior to the trial. The subject entered the car and remained inside for 15 minutes, during which they were permitted to adjust the air conditioning at will in order to make themselves comfortable. The control adjustments performed were logged by the observer. These trials were performed both with and without simulated solar loading on the driver side of the car, with each condition tested once per subject.

3.3 User Control During Short Journeys

The trials consisted of subjects driving the test car on private roads. The car and the subjects were preconditioned to a neutral (22 °C), hot (28 °C), or cold (16 °C) temperature. The subjects entered the car and drove for 15 minutes, during which they were permitted to adjust the air conditioning at will in order to make themselves comfortable. The subjects were required to turn and change speed at frequent intervals in order to simulate daily driving routines. The hot and cold tests were performed twice per subject, while the neutral tests were performed once. The adjustments made to the HVAC inputs were also logged by the observer.

4 EQUIVALENT TEMPERATURE ESTIMATION PERFORMANCE EVALUATION

To implement the machine learning methods the author used Python (van Rossum and Drake, 2001), the WEKA software libraries (Hall et al., 2009) and the open-source Orange software libraries (Demsar et al., 2013). As a result of an empirical investigation, the parameters corresponding to each machine learning method were set as follows: for MLP – the number of hidden layers is 2, the learning rate is 0.2, the momentum is 0.2 and the training time corresponds to 500 epochs; for KNN – k is 5; for MARS – the maximum degree of the terms in the model is 2 and the maximum number of terms in the forward pass is 10; for RBF – the minimum standard deviation for the clusters is 0.1, the learning rate is 0.2 and the number of clusters for K-Means corresponds to 2 epochs;

[2]Sensirion SHT75 - Digital Humidity Sensor (RH&T): http://www.sensirion.com/en/01_humidity_sensors/06_humidity_sensor_sht75.html

for REPTree there is no restriction on the maximum depth, the minimum total weight of the instances in a leaf is 2 and the number of data folds used for pruning is 3 and for RF – the number of trees in the forest is 100.

Cross Validation (CV) was used to evaluate each estimator's performance on the full set of experimental data, indicating how well the algorithm generalised to unseen data. Both K-fold CV (presented in Algorithm 1) (with $k = 10$) and Leave One Out Cross Validation (LOTOCV) (presented in Algorithm 2) were applied. The author also used LOTOCV, not just 10-fold CV, to better cope with the tendency of autocorrelation for time series data and, also, with the existing trial-to-trial variation. The outputs of the estimators were compared to the original measured equivalent temperature and Root Mean Square Error (RMSE) was used as an accuracy measure. The estimation was performed using the best two cabin sensors selected as described by Hintea et. al (Hintea et al., 2011)

Algorithm 1: K-fold Cross-Validation process.

1. The whole dataset is randomly partitioned into k samples of equal size.
2. One sample out of the k is selected as the validation data.
3. The remaining $k-1$ samples are used as training data.
4. The process is repeated k times, with each sample set used once as validation data.
5. The k results are averaged to provide a mean error over the individual cross-validation processes.

Algorithm 2: Leave-One-Trial-Out-Cross-Validation process.

1. The whole dataset is partitioned into n individual experimental trials.
2. One trial out of the n is selected as the validation data.
3. The remaining $n-1$ trials are used as training data.
4. The process is repeated n times, with each trial set used once as validation data.
5. The n results are averaged to provide a mean error over the individual cross-validation processes.

Tables 1 and 2 present the estimation errors (RMSE) using both LOTOCV and 10-fold CV for all seven different machine learning methods.

For KNN, the results of the evaluation show that the RMSE varied between 1.52 °C and 2.15 °C (for chest and head, respectively) when LOTOCV was applied and between 0.42 °C and 0.71 °C (for thigh and head, respectively) when 10-fold CV was applied. The results of the 10-fold CV are more accurate than the ones corresponding to LOTOCV due to overfitting (results are not representative for unseen data) the model.

For MARS, the RMSE varied between 1.27 °C and 1.70 °C (for thigh and head, respectively) when LOTOCV was applied and between 1.07 °C and 1.55 °C (for thigh and head, respectively) when 10-fold CV was applied.

For MLP, the results of the evaluation show that the RMSE varied between 1.40 °C and 1.87 °C (for chest and head, respectively) when LOTOCV was applied and between 1.76 °C and 2.98 °C (for thigh and head, respectively) when 10-fold CV was applied.

For MLR, the RMSE varied between 1.30 °C and 1.91 °C (for thigh and head, respectively) when LOTOCV was applied and between 1.33 °C and 1.81 °C (for thigh and head, respectively) when 10-fold CV was applied.

For RBF, the results of the evaluation show that the RMSE varied between 2.93 °C and 4.47 °C (for chest and head, respectively) when LOTOCV was applied and between 3.09 °C and 4.54 °C (for thigh and head, respectively) when 10-fold CV was applied. These results are worse than the ones produced by all other learning-based approaches.

For REPTree, the RMSE varied between 1.47 °C and 2.09 °C (for chest and head, respectively) when LOTOCV was applied and between 0.99 °C and 1.91 °C (for calf and head, respectively) when 10-fold CV was applied. The results of the 10-fold CV are significantly better than the ones corresponding to LOTOCV because of over-fitting (results are not representative for unseen data).

For RF, the results of the evaluation show that the RMSE varied between 0.95 °C and 1.45 °C (for thigh and head, respectively) when LOTOCV was applied and between 1.40 °C and 1.96 °C (for thigh and head, respectively) when 10-fold CV was applied. The results of the 10-fold CV are significantly better than the ones corresponding to LOTOCV because of overfitting (results are not representative for unseen data).

P-values were generated through paired t-tests for each combination of models in order to establish the significance of these results. MLR is significantly better than KNN (p-value of 3.12e-023), RBF (p-value of 5.10e-009), REPTree (p-value of 6.82e-012) and RF (p-value of 4.89e-078). The models MLP and MARS perform better than MLR with low confidence

Table 1: Equivalent temperature estimation results (RMSE) using Leave One Trial Out Cross Validation (LOTOCV) using different machine learning methods.

	Equivalent Temperature Estimation						
Target	KNN	MARS	MLP	MLR	RBF	REPTree	RF
Head	2.15 °C	1.70 °C	1.87 °C	1.91 °C	4.47 °C	2.09 °C	1.96 °C
Chest	1.52 °C	1.37 °C	1.40 °C	1.41 °C	2.93 °C	1.47 °C	1.42 °C
Lower arm left	2.08 °C	1.57 °C	1.56 °C	1.85 °C	3.15 °C	1.97 °C	1.79 °C
Lower arm right	1.87 °C	1.54 °C	1.59 °C	1.59 °C	3.31 °C	2.01 °C	1.78 °C
Upper arm left	1.76 °C	1.56 °C	1.48 °C	1.65 °C	3.26 °C	1.73 °C	1.67 °C
Upper arm right	2.00 °C	1.56 °C	1.74 °C	1.77 °C	3.76 °C	1.67 °C	1.78 °C
Thigh	1.58 °C	1.27 °C	1.18 °C	1.30 °C	3.01 °C	1.48 °C	1.40 °C
Calf	1.88 °C	1.49 °C	1.48 °C	1.81 °C	3.08 °C	1.82 °C	1.39 °C
Average	1.85 °C	1.51 °C	1.53 °C	1.66 °C	3.37 °C	1.78 °C	1.64 °C

Table 2: Equivalent temperature estimation results (RMSE) using 10-fold Cross Validation (10-fold CV) using different machine learning methods.

	Equivalent Temperature Estimation						
Target	KNN	MARS	MLP	MLR	RBF	REPTree	RF
Head	0.71 °C	1.55 °C	2.98 °C	1.81 °C	4.54 °C	1.91 °C	1.45 °C
Chest	0.46 °C	1.20 °C	1.87 °C	1.42 °C	3.13 °C	1.03 °C	1.06 °C
Lower arm left	0.59 °C	1.48 °C	2.23 °C	1.78 °C	3.36 °C	1.07 °C	1.20 °C
Lower arm right	0.64 °C	1.41 °C	2.25 °C	1.49 °C	3.36 °C	1.43 °C	1.23 °C
Upper arm left	0.46 °C	1.31 °C	2.08 °C	1.61 °C	3.22 °C	1.07 °C	1.12 °C
Upper arm right	0.63 °C	1.47 °C	2.62 °C	1.70 °C	3.74 °C	1.61 °C	1.25 °C
Thigh	0.42 °C	1.07 °C	1.76 °C	1.33 °C	3.09 °C	1.00 °C	0.95 °C
Calf	0.58 °C	1.31 °C	2.48 °C	1.91 °C	3.48 °C	0.99 °C	1.20 °C
Average	0.56 °C	1.35 °C	2.28 °C	1.63 °C	3.49 °C	1.26 °C	1.18 °C

Table 3: Classification time for all equivalent temperature estimators using LOCOTV.

Method	MARS	MLP	MLR	REPTree	KNN	RF	RBF
Classification time (seconds)	3.06	18.16	0.23	7.45	64.34	59.11	4.12

(p-values of 0.08 and 0.09, respectively). However, the improvement of these models over MLR is of only 0.13 °C average RMSE.

At this stage, another factor to be taken into consideration is the processing time, the time required to perform the estimation on an unseen data set. This is an important factor to consider prior to integrating the estimation method within a control unit. Table 3 provides a summary of the processing time required by the machine learning approaches.

MLR provided the fastest processing time, of 0.23 seconds, outperforming all other methods (significance: p-value of 2e-021 when combined with any of the other models). The MLR processing time was, therefore, the fastest for each individual trial. The classification time for MLR was lower than for all other estimation techniques, while the estimation error obtained was outperformed by two models, the MLP and MARS. The improvement in accuracy of the latter two methods is not significant, therefore the MLR approach is concluded to be the most suitable estimation approach.

5 CONCLUSIONS

This paper studied several machine learning methods for estimating cabin occupant equivalent temperature from a minimalistic set of inexpensive cabin environmental sensors. Seven different machine learning approaches were implemented and evaluated: Multiple Linear Regression (MLR), MultipleLayer Perceptron (MLP), K-Nearest Nighbour (KNN), Multivariate Adaptive Regression Splines (MARS), Radial Basis Function Network (RBF), REPTree and Random Forest (RF).

Most learning tehniques provided a RMSE between 1.51 °C (for MARS) and 1.85 °C (for KNN). RBF performed the worst, with an average RMSE of 3.37 °C. MLR had an average RMSE of 1.60 °C over the eight body part equivalent temperatures. MLR outperformed all other estimation techniques with regard to fast processing time (of 0.23 seconds). These two factors combined would enable a straightforward real-time implementation in a car's engine control unit in comparison to the other machine learning techniques evaluated.

ACKNOWLEDGEMENTS

The Low Carbon Vehicle Technology Project (LCVTP) was a collaborative research project between leading automotive companies and research partners, revolutionising the way vehicles are powered and manufactured. The project partners included Jaguar Land Rover, Tata Motors European Technical Centre, Ricardo, MIRA LTD., Zytek, WMG and Coventry University. The project included 15 automotive technology development work-streams that will deliver technological and socio-economic outputs that will benefit the West Midlands Region. The £19 million project was funded by Advantage West Midlands (AWM) and the European Regional Development Fund (ERDF).

The authors would like to thank the anonymous reviewers for their insightful comments.

REFERENCES

Breiman, L. (2001). Random forests. Technical report, University of California Berkeley.

Cover, T. and Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, IT-13.

Demsar, J., Curk, T., Erjavec, A., Gorup, C., Hocevar, T., Milutinovic, M., Mozina, M., Polajnar, M., Toplak, M., Staric, A., Stajdohar, M., Umek, L., Zagar, L., Zbontar, J., Zitnik, M., and Zupan, B. (2013). Orange: Data mining toolbox in python. *Journal of Machine Learning Research*, 14:2349–2353.

Draper, N. and Smith, H. (1981). *Applied Regression Analysis*. Wiley.

Friedman, J. (1991). Multivariate adaptive regression splines. *Annals of Statistics*, 19:1–67.

Hall, M., Frank, E., Holmes, G., Pfahringer, B., Reutemann, P., and Witten, I. (2009). The weka data mining software: An update. *SIGKDD Explorations*, 11.

Hastie, T., Tibshirani, R., and Friedman, J. (2009). *The Elements of Statistical Learning*. Springer.

Haykin, S. (1998). *Neural Networks: A Comprehensive Foundation*. Prentice Hall.

Hintea, D., Brusey, J., Gaura, E., Beloe, N., and Bridge, D. (2011). Mutual information-based sensor positioning for car cabin comfort control. In *Proceedings of the 15th international conference on Knowledge-based and intelligent information and engineering systems - Volume Part III*, KES'11, pages 483–492.

Hintea, D., Kemp, J., Brusey, J., Gaura, E., and Beloe, N. (2014). Applicability of thermal comfort models to car cabin environments. In *International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, volume 1, pages 769–776.

Quinlan, J. (1986). Introduction of decision trees. *Machine Learning*, 1:81–106.

Srivastava, A., Oza, N., and Stroeve, J. (2005). Virtual sensors: Using data mining techniques to efficiently estimate remote sensing spectra. *IEEE Transactions on Geoscience and Remote Sensing*, 43.

van Rossum, G. and Drake, F. (2001). *Python Reference Manual*. PythonLabs.

Way, M. and Srivastava, A. (2006). Novel methods for predicting photometric redshifts from broadband photometry using virtual sensors. *The Astrophysical Journal*, 647:102–115.

Wenzel, T., Burnham, K., Blundell, M., and Williams, R. (2007). Kalman filter as a virtual sensor: Applied to automotive stability systems. *Transactions of the Institute of Measurement and Control*, 29.

Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques*. Morgan Kaufmann.