Available online at www.sciencedirect.com
ScienceDirect
IFAC PapersOnLine 52-11 (2019) 116–121

A Machine Learning-based Soft Sensor for Laundry Load Fabric Typology Estimation in Household Washer-Dryers

Gian Antonio Susto * Leonardo Vettore * Giuliano Zambonin * Fabio Altinier ** Daniele Beninato ** Terenzio Girotto ** Mirco Rampazzo * Alessandro Beghi *

* Department of Information Engineering, University of Padova, Italy. (e-mail: gianantonio.susto@dei.unipd.it)
** Electrolux Italia S.p.A., Porcia (PN), Italy.

Abstract: Fabric care manufactures are striving to make more energy efficient and more user-friendly products. The aim of this work is to develop a Soft Sensor (SS) for a household Washer-Dryer (WD) that is able to distinguish between different fabrics loaded in the machine; the knowledge of load composition may lead to a more accurate drying, faster processed and lower energy consumption without increasing the production costs. Moreover, automatic classification of load fabric will lead to an enhanced user experience, since user will be required to provide less information to the WD to obtain optimal drying processes. The SS developed in this work exploits sensors already in place in a commercial WD and, on an algorithmic point of view, it exploits regularization methods and Random Forests for classification. The efficacy of the proposed approach has been tested on real data in heterogeneous conditions.

Keywords: Domestic Appliances, Fabric Care, Heat Pump Washer-Dryer, Human-centric Control Solution, Logistic Regression, Machine Learning, Random Forest, Regularization, Soft Sensors.

1. INTRODUCTION

In the context of household fabric care appliances development, manufacturers are constantly striving for making products more efficient and more user friendly. One of the main obstacle in this process of optimization can be related to the uncertainty in the laundry loaded in the appliance. The laundry characteristics (like weight, fabric, contained water) has a major impact in the drying and washing processes: being aware or estimating such characteristics can enable several process optimizations, in terms of both performances and consumptions. In particular, the availability of information about the laundry could be exploited in several apparent ways during a washing or drying cycle: for example, in washing processes, the knowledge of laundry composition can be exploited to load the proper amount of water while, in drying processes, such information can be used to set the drying process duration or to provide the user with an accurate estimation of the remaining time of the process.

Classically, in some laundry treating appliances, the user can or is requested to provide information (such as weight or fabric typology) regarding the laundry through a user interface. However, such practice is deplorable for some reasons: (i) users provide subjective and inaccurate information; (ii) the request of information from the machine may be considered as inconvenient and generating discomfort. The usage of dedicated physical sensors to characterize laundry is generally not possible or not costly-effective in household appliances. For these reasons, manufacturers have to resort to indirect information on the laundry inferred from other sensors or provided by the users. For example in [Zambonin et al. 2018, Susto et al. 2018b,a] Soft Sensors (SS) [Kadlec et al. 2009] based on Machine Learning approaches have been presented to estimate the laundry weight in Washing Machines, Washer-dryers and Tumble Dryers. A SS aims at providing through a model an estimate of a relevant quantity, called output, that may be unmeasurable or costly/time-consuming to measure based on more accessible 'cheap to measure' variables, called inputs. Such technology is generally based on *Machine Learning* (ML) supervised techniques [Hastie et al. 2009] that exploit the availability of historical data where the relationship between inputs and output is measured.

In this work we present a SS for the automatic estimation of laundry fabric type in household Heat Pump Washer-Dryers (WD-HP). Developing a SS for laundry household equipment is a challenging task for many reasons: (i) laboratory data, where the water content is accurately measured and laundry composition is precisely described, are costly; (ii) computational capacity in current appliances is limited, posing an hard constrain on the ML-based solutions to be embedded in the equipment. In this work we overcome these two issues by exploiting laboratory data already collected for other product development purposes and by exploiting *regularization* Scholkopf and Smola [2001], a Machine Learning framework that allows, in some cases, to provide effective sparse linear models that are easily implementable.

The contributions of this paper are the following:

- the one presented here is one of the first SS for home appliances and the first to estimate load fabric typology in fabric care home appliances;
- regularization approaches are used to enable on-board implementation of SS, while Random Forest are employed for ceiling analysis and for assessing classification performance in future hardware and architectures;

10.1016/j.ifacol.2019.09.127

- particular focus has been provided to the timing in which the SS provides the estimation; the availability of such result can be exploited in various ways in different phases of the drying process. In this perspective, two different solutions with different timing are illustrated in this work: a EARLY classification where the typology estimation is used for improving process handling and a LATE classification where such estimation is exploited for improving the end-cycle detection;
- results are validated on real experiments.

The remainder of the paper is organized as follows. Section 2 is dedicated to review SS solutions and estimation approaches in fabric care appliances and other contexts with limited computational resources; in Section 3, Machine Learning basic concepts are introduced and the methods employed in this work are illustrated. In Section 4 and in Section 5 the use case and the proposed Soft Sensor are respectively described. Section 6 is devoted to summarize the experimental results obtained on real industrial data. Remarks are then reported in Section 7.

2. RELATED LITERATURE REVIEW

The SS technologies have proliferated in the past years given the increased availability of data in many engineering scenarios and the growing attention in dedicated to extract valuable information from such data. SS technologies go under different names depending on the application area and/or background of the involved researchers; for example, 'Soft Sensor' is used in chemical and pharmaceutical manufacturing Bosca and Fissore [2011], 'Virtual Metrology' is used in semiconductor manufacturing Pampuri et al. [2012], Park and Kim [2016] while 'Virtual Sensor' is used in automotive Stephant et al. [2004]. SS technologies are also used for control purposes Tabbache et al. [2013]. As anticipated in the Introduction Section, only few works about the use of SS for household major appliances are available in scientific literature, in particular, for laundry load estimation problem: Zambonin et al. [2018], Susto et al. [2018b,a].

Some references could be found in literature for similar resource-constrained environments like wearable devices [1]. Wearable devices are constrained in terms of computational capability and memory, which calls for the design of algorithmic solutions that explicitly take into account these issues Belgioioso et al. [2014], Cenedese et al. [2015], Terzi et al. [2017].

Another example of SS development for a similar problem in industrial applications can be found in Paciello and Sommella [2013] where smart sensing issues concerning a challenging motorcycle application are studied. SS have been proposed here to solve problems such as measuring system backup and and fault diagnosis strategies. Vehicle suspensions represent one of the most interesting applications of dampening; developing damping technologies involves searching solutions for measurement challenges. Smart sensing is able both to improve noise filtering of the signals provided by low-cost sensors and detect specific motorcycle dynamics which most influence the road holding and comfort. A SS has been modeled and adopted as a benchmark (in terms of false alarms and correctly detected faults) in the development of fault detection strategies (i.e., threshold identification) directed to the sensor validation of the rear suspension stroke.

No other works to the best of our knowledge are available in literature for automatic laundry typology identification for major appliances and this is the main novelty of this work. When the aim is to quantify some essential information for the (washing/drying) process handling, all methods reported in patent literature for similar purposes are based or related to the knowledge of the process of interest and their implementations suffer from the adaptability to different operating conditions of washer and dryer machines.

3. MACHINE LEARNING FOR CLASSIFICATION

In this Section the methodologies employed for classification are briefly detailed; we refer the interested reader to Hastie et al. [2009] for a more detailed treatise of classification approaches. Two types of algorithms have been adopted: (i) regulation methods that can be easily implemented in present fabric care appliances and (ii) Random Forests, one of the most accurate approach to classification [Fernández-Delgado et al. 2014], that represent in this context a benchmark for regularization methods and allow us to have an assessment of load type fabric classification performance for future hardware or architectures.

3.1 Logistic Regression

Logistic Regression (LR) is one of the most widely-used classifiers Menard [2002]. In the binary case, given two possible response classes, the logistic regression describes the probability that the response belongs to one of the two classes. For example, let $C = \{c_1, c_2\}$ be the set with the two possible response classes, $\tilde{x}$ be an observation and $\tilde{y}$ its unknown response; then the logistic regression model is $Pr(\tilde{y} = c_1|\tilde{x})$.

To model the relationship between $p(\tilde{x}) \equiv Pr(\tilde{y} = c_1|\tilde{x})$ and $\tilde{x}$ and obtain values between 0 and 1, we use the *logistic function* Hastie et al. [2009]:

$$p(\tilde{x}) = \frac{e^{\beta_0 + \sum_{j=1}^{m} \beta_j \tilde{x}_j}}{1 + e^{\beta_0 + \sum_{j=1}^{m} \beta_j \tilde{x}_j}} \tag{1}$$

where $\tilde{x_j}$ are the elements of the vector of measurements $\tilde{x}$ and β_j are some coefficients that needs to be estimated. After a bit of manipulation:

$$log\left(\frac{p(\tilde{x})}{1 - p(\tilde{x})}\right) = \beta_0 + \sum_{j=1}^{m} \beta_j \tilde{x_j} \tag{2}$$

where the left-hand side is called the *logit*; we can then see that the logistic regression model has a logit that is linear in $\tilde{x}$.
To estimate the unknown coefficients β_j we can use the method of *maximum likelihood*, for which the *likelihood function* is:

$$l(\beta_0, \ldots, \beta_m) = \prod_{i:y_i=c_1} p(x_i) \prod_{i':y_{i'}=c_2} (1 - p(x_{i'})). \tag{3}$$

The logistic regression classification is able to predict only a binary response, so it doesn't work for problems with more than two response classes, as is the case of the problem faced in this work. For this reason the logistic regression will be extended for multiclass (or multinomial) classification Aly [2005]. In order to reduce the overfitting problem, regularized LR will be adopted using weel-known shrinkage methods like Ridge Regression (RR) and Least Absolute Shrinkage and Selection Operator (LASSO).

[1] The rapid growth of IMUs (Inertial-Measurement Units) has allowed, in recent years, the development of compact sensor-equipped devices (e.g. smartwatches and smartphones) which lead efficient monitoring of human activities to be feasible.

3.2 Decision Trees and Random Forests

A classification problem can be solved by asking a series of carefully crafted questions about the features of the observations Larose and Larose [2014]. Each time an answer is given, a follow-up question is asked until is reached a conclusion about the class label of the observation. The series of questions and their possible outcomes can be organized in the form of a tree-structure where: (i) each interior node is a feature, and there are edges to children for each possible value of the feature; (ii) each leaf is a class. One of the algorithms that can be used to grown a Decision Tree (DT) is Hunt's Quinlan [1990], that recursively partitions the training dataset into successively purer subsets. Let $S_t = \{x_1, x_2, ..., x_c\}$ be the subset of the training dataset S associated with node t and $y = \{y_1, y_2, ..., y_c\}$ the class labels. The following is a recursive definition of Hunt's algorithm:

(1) if all observations in S_t belong to the same class y_t, then t is a leaf node labelled as y_t;
(2) if S_t contains observations that belong to more than one class an *attribute test condition* is selected to partition the records into smaller subsets. A child node is created from each outcome of the test condition and the records in S_t are distributed to the children based on the outcomes. The algorithm is then recursively applied to each child node.

This definition needs to be applied starting from the root r of the tree, where the starting conditions are: $t = r$, $S_r = S$ and $y = C$.
Using DTs brings many advantages, since they are easily interpretable, they require no data normalization, the computational expensive part is done off-line once and the classification is immediate; however they are high variance classifiers: they fully represent the dataset observations with the risk of overfitting.
To reduce this effect we can use Random Forests (RF), one of the most powerful and used classifier, still based on DTs. RF is an *ensemble learning method* [2] Dietterich [2000], Hastie et al. [2009], where a set of several different DTs is generated and the classification is done by a majority vote. To obtain a set of different DTs, RF uses *bagging* (or *bootstrap aggregating*) Breiman [1996], consisting on the creation of datasets by uniformly sampling with replacement from the original dataset.

4. DATA DESCRIPTION

Some of the tests contained a full *wash and dry* cycle (W&D), meaning that before the drying phase of the cycle there was the washing phase too. The signals provided include:

- data streams of sensors;
- signals that describe the work of several components, like compressor, motor, temperature probe, etc;
- additional signals constructed from the previous ones using filters or other data manipulations techniques;
- some external information, like laboratory conditions (e.g. humidity and temperature), energy consumption etc.

A subset of 23 signals among all the ones available has been selected, based on domain experts' advice and the study of several plots showing the signals behaviour in different conditions and with different load types.

[2] Ensemble methods use multiple learning algorithms to obtain better predictive performances than could be obtained from any of the constituent learning algorithms alone.

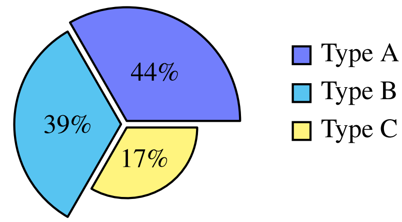

Fig. 1. Laundry load type distribution in the data, 3 classes.

A total of 211 tests is available and the goal is to use them to distinguish between 3 classes of load typology: *Type A* , *Type B* and *Type C* [3]. Table 1 shows the description of dataset in terms of laundry load size and machine temperature while Figure 1 represents the distribution of classes in the dataset. Tests can be *Cold* or *Hot* referring to the machine temperature

Table 1. Drying tests available. In brackets there is the number of drying tests extracted from a complete *wash and dry* cycle.

	Type A		*Type B*		*Type C*	
	Cold	*Hot*	*Cold*	*Hot*	*Cold*	*Hot*
load I	8 (2 W&D)	7	-	-	4	4
load II	-	-	-	-	6 (1 W&D)	4
load III	7	4	-	-	-	-
load IV	8 (2 W&D)	6	-	34	1	1
load V	-	-	-	-	6 (1 W&D)	1
load VI	3	3	-	-	8 (1 W&D)	-
load VII	3	3	7	6	-	-
load VIII	14 (5 W&D)	3	5	-	-	-
load IX	22 (7 W&D)	3	30	-	-	
Tot.	**65**	**29**	**42**	**40**	**25**	**10**

during process of interest. This distinction is made because the behaviour of several variables is strongly influenced by this characteristic, and many of these variables can be used to easily separate this two types of tests. There are just a few tests for every possible case, sometimes none; for this reason, it is difficult to study the differences between the three types of load in the same conditions [4]. Furthermore, for each of the load types there are many tests all with the same particular weight. For the reasons mentioned above, the available dataset can be considered imbalanced. Some common and simple strategies could be used to face the problem of imbalanced data (Guo et al. [2008]) such as undersampling the majority class or oversampling the minority class; however, these approaches have not been exploited here because of our small set of provided data [5].

5. SOFT SENSOR FOR FABRIC LOAD CLASSIFICATION

As anticipated before, one goal is to determine the LATE load estimation in useful time to improve the end-cycle detection. Therefore, experts suggested to consider only features [6] in the first "half" of the test, where the first "half" is computed as half of the time-to-end [7] estimated at the beginning of the cycle. After the first variable selection, for each variable one or more

[3] Details about laundry composition, machine temperature and signals available will be omitted here because of intellectual property rights.
[4] With equal load weight and *Cold/Hot* condition.
[5] It is preferred to use the dataset as it is avoiding undersampling here.
[6] The term "feature" is used here to describe elaborations of the raws signals that are statistically significant quantities extracted from such signals using the experts' knowledge and employed to summarize the informative content.
[7] Time-to-end is the expected end-time for the drying cycle.

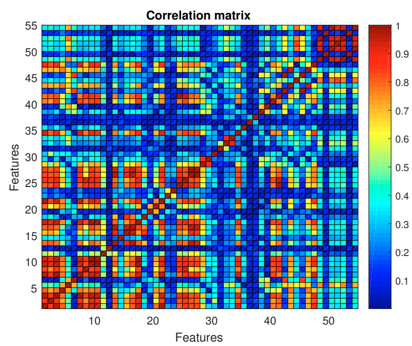

Fig. 2. Visual representation of the correlation matrix Φ of the extracted features. Features extracted from the same signals are in succession, that's why there is higher correlation between closer features.

features were chosen and for every test, all the features were extracted and collected in a *Design Matrix* $X \in \mathbb{R}^{n\times m}$, where n is the number of tests in the dataset and m the number of the chosen features. Similarly, an *Output Matrix* $Y \in \mathbb{R}^{n\times 1}$ was built indicating the class of each test with the values y_1 to indicate *Type A* , y_2 to indicate *Type B* and y_3 to indicate *Type C* . As already anticipated in section 1, we need to implement estimation procedures that are deployable in firmware program, so the chosen features are simple to compute: they are for example maximum or minimum values and relative time positions, means, slopes.
In Figure 2, the correlation matrix $\Phi \in \mathbb{I}^{m\times m}$, with $\mathbb{I} \in [0, 1]$ and $[\Phi]_{i,j} = |\mathrm{corr}(x_i, x_j)|$, of the extracted features is depicted [8] . Every little square represents a degree of correlation between a couple of features. Basically in the diagonal of Φ are represented the correlations between the features and themselves, that are always equal to one. On the rest of the matrix we can see that some areas tends to have higher correlations (it happens because features extracted from the same signals are in succession).

The classification will also be distinguished based on the features used:

- EARLY *classification* where only the features in the first minutes of the drying cycle are used, for a total of 41 features;
- LATE *classification* where all the features in the first minutes of the dring cycle plus the features that comes later are used, for a total of 55 features;
- *trivial classification*, where one or at most two features computed for the EARLY case are used: *feature 1* and *feature 2* [9] ..

For each classification type we tested two different approaches, based on the considerations made in 4 about *Hot* and *Cold* tests:

(1) *Together*, without distinction between *Hot* and *Cold* tests: in this approach we train one single model on all the tests (Figure 3);
(2) *Divided*, with distinction between *Hot* and *Cold* tests: in this approach we first separate the tests in *Hot* and *Cold* based on a variable available from data and then training two models, one for *Hot* tests and one for *Cold* ones. This approach aims to take into account the different behaviour that many variables present in *Hot* or *Cold* situations (Figure 4).

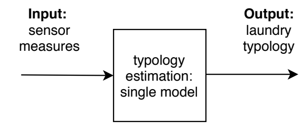

Fig. 3. Estimation algorithm using *Together approach*.

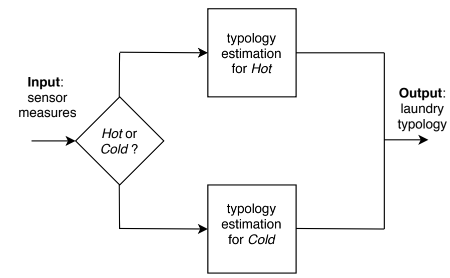

Fig. 4. Estimation algorithm using *Divided approach*.

To train the models we used the following approaches:

- *regularized Logistic Regression* (rLR), with 10 cross validation folds Hastie et al. [2009] for the lambda choice and two possible lambdas:
  - · λ_{min}, that is the value of λ that gives minimum mean cross-validated error; in the following we will indicate with LRλ_{min} the associated results;
  - · λ_{1se}, that is an heuristic choice producing a less complex model, which gives the most regularized model such that error is within one standard error of the minimum error.
- *Random Forest*, with 100 trees.

All the models were validated using 1000 Monte Carlo Cross-validations (CV) folds Hastie et al. [2009], with the dataset portioned in: 70% as training set and 30% as test set.
In particular, rLR is a preferable modelling approach for its simple implementability; however, we want to compare the performances of this model with the performances of RF to see how much better a more complex model can perform.
For each model the classification performances were calculated using the *confusion matrix*, that is a specific table layout that allows visualization of the performance of an algorithm; in this layout each column represents the instances in a predicted class, while each row represents the instances in an actual (real) class (or vice versa). This representation makes it easy to see if the system is confusing two classes (i.e. commonly mislabelling one as another) since in the main diagonal of the confusion matrix there is the percentage of exact matches between predicted and real classes, while other values represent the the mismatch between predicted and real classes. The classes of the confusion matrix are: *Type A Hot*, *Type B Hot*, *Type A Cold* and *Type B Cold*. An example of confusion matrix is depicted in Figure 5.

As mentioned before, among the features there is the *weight estimation*, that is a soft sensor prediction of the weight of the load Susto et al. [2018b], Zambonin et al. [2018], Susto et al. [2018a]. Being this feature an estimation, it's not always correct, thus we wanted to see how its precision affects the final

[8] The correlation coefficient of two random variables is a measure of their linear dependence. The MATLAB function *corrcoef* was used to compute such values.
[9] Details about feature computation and significance will be deliberately omitted because of the as agreed between authors and industrial partner.

load type estimation. In order to do so, we tried to train the LR model with λ_{min} and *Hot* and *Cold* divided for the EARLY and for the LATE classification with this feature.

6. RESULTS

As already explained, the goal for this work is to recognize the load type between the *Type A* , *Type B* and *Type C* , so a 3 classes classification problem. Because *Type C* loads have dedicated drying programs, a particular attention could be given to distinguish between the first two types. Thus the goal could be redrafted using only two possible classification classes and using 176 out of the 211 available tests. Therefore, In this work also a binary classification problem had been addressed but results reported here concern only multinomial classification problem since it was the main objective of this research.

6.1 Early classification

Table 2 visualizes the performances of the different models for the EARLY classification. The classifiers have good performances considering *Type A* and *Type A* classes, however, for the *Type C* class the models don't perform as well as for the other two classes: as we can see in the table, the *Type C Cold* have a Classification Rate (CR) -with LR- of 92%, but the *Type C Hot* have a CR [10] of 57% in the best case. This situation, as for the other cases, is invariant to the model or the features used to classify. Probably the problems with *Type C Hot* classification are caused by the low number of tests that belong to this category and by the distribution of these tests (see Table 1): there are only 10 tests that are *Type C Hot*, and they have all low weights (not represented in Table 1).
Regarding the sizes of the models, we have to consider that, since it's a multinomial classification with three classes, the LR method will generate three models (one for each class) in the *Together* case, and six models (one for *Hot* and one for *Cold* for each one of the three classes) in the *Divided* case. In Figure 5,

Table 2. Performances of the different models for the EARLY multinomial classification. Average results/statistics over 1000 Monte Carlo simulations.

Model	Approach	Classification rate [%]					
		Type A Hot	*Type B Hot*	*Type C Hot*	*Type A Cold*	*Type B Cold*	*Type C Cold*
LR λ_{min}	*Together*	83	90	57	91	91	**92**
	Divided	91	98	34	96	**99**	92
Random Forest	*Together*	86	**97**	38	**97**	93	85
	Divided	88	**98**	32	97	93	86

the confusion matrix of the models in Table 2 is shown as an example of final visualization; in particular Figure 5 shows results for LRλ_{min} models with *Divided* approach since it turned out to be an interesting result considering the trade off between performance and implementation constrains. From this matrix we can see that the *Type C* tests (in particular the *Hot* ones) that are not correctly classified are almost all confused with the *Type A* tests. For the *Hot* tests we can imagine that this problem arises because of the distribution and the number of the *Type C Hot* tests: as we have seen previously, the *Type C Hot* tests have all low weights and they are few in number, while the most of the low weights observations in the dataset belongs to the *Type A* class; for this reason we suppose that it is difficult for the model to distinguish between *Type A Hot* and *Type C Hot*.

[10]CR has been computed as $CR = \left\lceil \frac{\text{total matches}}{\text{total comparisons}} 100 \right\rceil$.

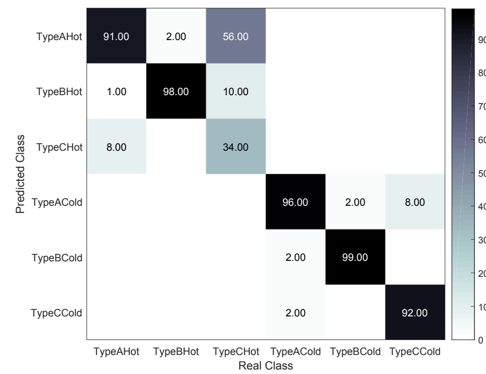

Fig. 5. Confusion matrix for the EARLY classification, values in [%] and correct classification in the main diagonal.

6.2 Late classification

Also in this case similar considerations made for *Type C* EARLY classification are still valid i.e, as represented in Table 3 the *Type C* , in the *Cold* case are recognized correctly the most of the times, while in the *Hot* case they still have a very low classification rate (57% at best).

Table 3. Performances of the different models for the LATE multinomial classification. Average results/statistics over 1000 Monte Carlo simulations.

Model	Approach	Classification rate [%]					
		Type A Hot	*Type B Hot*	*Type C Hot*	*Type A Cold*	*Type B Cold*	*Type C Cold*
LR λ_{min}	*Together*	81	92	57	92	**96**	90
	Divided	88	98	33	96	**99**	93
Random Forest	*Together*	86	**99**	48	96	96	86
	Divided	89	**100**	32	97	96	87

Considering EARLY and LATE classification (Table 2 and Table 3 respectively), in the latter 14 more features are used exploiting additional information from the drying cycle. CR in LATE case shows only a little improvement revealing the fact that the first part of the drying cycle seems to be more useful for the typology classification problem at hand.

6.3 Trivial classification

In Table 4 the results of the trivial classification for the multinomial problem are reported. As table shows, the performances of *Type C* are very bad in both *Hot* and *Cold* case and this is supposed to be due to the fact that the two features exploited to obtain classification results do not describe well *Type C* data which are characterized by a low sample size.

Table 4. Performances of the different models for the *trivial multinomial classification*. Average results/statistics over 1000 Monte Carlo simulations.

Model	Features used	Classification rate [%]					
		Type A Hot	*Type B Hot*	*Type C Hot*	*Type A Cold*	*Type B Cold*	*Type C Cold*
LR λ_{min}	*feature 1*	87	**95**	0.00	74	85	40
	feature 2	62	**86**	00	79	54	33
	Both	83	**94**	02	79	85	40

This kind of result (*trivial classification*) was studied with the aim of exploring possibilities for online implementation considering a very limited set of features. Observing Table 4 and Table 2 it is possible to quantify the loss of performance using only one/two features according to the case of interest.

7. CONCLUSION

This work dealt with the development of Soft Sensors to estimate load fabric type in household Heat Pump Washer-Dryers during the drying phase in order to improve drying performance, energy efficiency and user experience. To the best of our knowledge, this is the first work in literature to address this problem. The purpose was to obtain estimates taking advantage of signals and sensors already available in heat pump washer-dryers, in order to not increase hardware costs. These challenges have led to usage of statistical predictors; in particular, a regularized linear model, with limited number of inputs, was welcomed for its straightforward implementability. In order to have a benchmark, a second type of classifier has been employed, RF which generally outperforms linear solutions, and, while not being simple to be implemented on a fabric care product, it is here employed to provide a ceiling analysis.

Main results can be summarized as below:

- High difficulty to distinguish the *Type C* class in the *Hot* case, probably due to a very limited availability of tests. On the other hand, *Type B* shows good classification results in general;
- In all the studied cases, the performance of Random Forests were comparable to the one of regularized logistic regression, so a logistic regression solution will always be preferred for implementation given its simple form;
- The number and type of features influenced the performances of the classification only partially: the best results (almost identical) were obtained with EARLY and LATE classification, while slightly worse results were obtained with trivial classification (except for *Type C* case).

In addition to the goodness of the results obtained, two strength of this work needs to be underlined: (i) the novelty of the work: to the best of our knowledge there are no other works in literature that discuss virtual sensors development for laundry load type estimation; (ii) an effective soft approach: no impact on hardware and no dedicated Design of Experiments (DOE) was required.

Some potential development lines have emerged during this work such as further studies using a more homogeneous dataset in number of tests for laundry type and load size; in particular the use of a new richer amount of data could improve the performance of the multinomial classification.

REFERENCES

Aly, M., 2005. Survey on multiclass classification methods. Neural networks 19, 1–9.

Belgioioso, G., Cenedese, A., Cirillo, G. I., Fraccaroli, F., Susto, G. A., 2014. A machine learning based approach for gesture recognition from inertial measurements. In: 53rd IEEE Conference on Decision and Control. IEEE, pp. 4899–4904.

Bosca, S., Fissore, D., 2011. Design and validation of an innovative soft-sensor for pharmaceuticals freeze-drying monitoring. Chemical Engineering Science 66 (21), 5127 – 5136.

Breiman, L., 1996. Bagging predictors. Machine learning 24 (2), 123–140.

Cenedese, A., Susto, G. A., Belgioioso, G., Cirillo, G. I., Fraccaroli, F., 2015. Home automation oriented gesture classification from inertial measurements. IEEE Transactions on automation science and engineering 12 (4), 1200–1210.

Dietterich, T. G., 2000. Ensemble methods in machine learning. In: Multiple Classifier Systems. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 1–15.

Fernández-Delgado, M., Cernadas, E., Barro, S., Amorim, D., 2014. Do we need hundreds of classifiers to solve real world classification problems. J. Mach. Learn. Res 15 (1), 3133–3181.

Guo, X., Yin, Y., Dong, C., Yang, G., Zhou, G., 2008. On the class imbalance problem. In: 2008 Fourth international conference on natural computation. Vol. 4. IEEE, pp. 192–201.

Hastie, T., Tibshirani, R., Friedman, J., 2009. The Elements of Statistical Learning: data mining, inference and prediction. Springer.

Kadlec, P., Gabrys, B., Strandt, S., 2009. Data-driven soft sensors in the process industry. Computers & chemical engineering 33 (4), 795–814.

Larose, D. T., Larose, C. D., 2014. Discovering knowledge in data: an introduction to data mining. John Wiley & Sons.

Menard, S., 2002. Applied logistic regression analysis. Vol. 106. Sage.

Paciello, V., Sommella, P., Oct 2013. Smart sensing and smart material for smart automotive damping. IEEE Instrumentation Measurement Magazine 16 (5), 24–30.

Pampuri, S., Schirru, A., Susto, G. A., De Luca, C., Beghi, A., De Nicolao, G., 2012. Multistep virtual metrology approaches for semiconductor manufacturing processes. In: 2012 IEEE International Conference on Automation Science and Engineering (CASE). IEEE, pp. 91–96.

Park, C., Kim, S. B., 6 2016. Virtual metrology modeling of time-dependent spectroscopic signals by a fused lasso algorithm. Journal of Process Control 42, 51–58.

Quinlan, J. R., March 1990. Decision trees and decision-making. IEEE Transactions on Systems, Man, and Cybernetics 20 (2), 339–346.

Scholkopf, B., Smola, A. J., 2001. Learning with kernels: support vector machines, regularization, optimization, and beyond. MIT press.

Stephant, J., Charara, A., Meizel, D., 2004. Virtual sensor: Application to vehicle sideslip angle and transversal forces. IEEE Transactions on industrial electronics 51 (2), 278–289.

Susto, G. A., Maggipinto, M., Zannon, G., Altinier, F., Pesavento, E., Beghi, A., 2018a. Machine learning-based laundry weight estimation for vertical axis washing machines. In: 2018 European Control Conference (ECC). IEEE, pp. 3179–3184.

Susto, G. A., Zambonin, G., Altinier, F., Pesavento, E., Beghi, A., 08 2018b. A soft sensing approach for clothes load estimation in consumer washing machines. pp. 1252–1257.

Tabbache, B., Benbouzid, M. E. H., Kheloui, A., Bourgeot, J.-M., 2013. Virtual-sensor-based maximum-likelihood voting approach for fault-tolerant control of electric vehicle powertrains. IEEE Transactions on Vehicular Technology 62 (3), 1075–1083.

Terzi, M., Cenedese, A., Susto, G., 2017. A multivariate symbolic approach to activity recognition for wearable applications. IFAC-PapersOnLine 50 (1), 15865 – 15870, 20th IFAC World Congress.

Zambonin, G., Altinier, F., Corso, L., Sessolo, M., Beghi, A., Susto, G. A., 08 2018. Soft sensors for estimating laundry weight in household heat pump tumble dryers. pp. 774–779.