# A Comparison of Performance Metrics for Event Classification in Non-Intrusive Load Monitoring

Lucas Pereira
M-ITI / LARSyS
Funchal, Portugal
lucas.pereira@m-iti.org

Nuno Nunes
Tecnico U.Lisbon
Lisbon, Portugal,
M-ITI / LARSyS
njn@m-iti.org

***Abstract*—In this work, we analyse experimentally the behaviour of 18 different performance metrics when applied to classification algorithms in event-based Non-Intrusive Load Monitoring, identifying relationships and clusters between the measures.**

**Our results indicate that performance metrics have more in common than what was initially expected. Our results also suggest that in this multi-class classification problem, researchers should avoid micro-average and unweighted macro-average metrics in favor of their weighted macro-average counterparts. Finally, the results also suggest that probabilistic measures can provide important information that is not available when using more traditional performance metrics.**

## I. Introduction

Non-Intrusive Load Monitoring (NILM) is a technology to unobtrusively identify and monitor the energy consumption of individual appliances that co-exist in a building electric circuit [1].

As of today, NILM research is categorized into either event-based or event-less approaches. Event-based approaches are related to the early days of NILM and seek to disaggregate the total consumption by means of detecting and labeling the individual appliance transitions using previously trained event detection, classification and energy estimation algorithms (e.g., [2]. Event-less approaches, on the other hand, attempt to assign each sample of the aggregated power consumption to the total consumption of a specific combination of appliances, by means of statistical and probabilistic algorithms (e.g., [3]).

One of the most interesting challenges of NILM research (besides the disaggregation in itself) is how to evaluate and report the performance of the several proposed approaches, in part due to the lack of a consensus rgarding which performance metrics to use in each situation [4].

For example, only recently there seems to be an agreement upon grouping performance metrics according to two main categories: i) event detection performance metrics (ED) designed to evaluate the NILM's ability to track the consumption over time; and ii) energy estimation metrics (EE), designed to characterize and evaluate the NILM disaggregated data against the actual ground-truth [5]. Furthermore, most of the currently used metrics were "borrowed" from other application domains in machine-learning, making it unclear how such metrics behave when applied to different NILM algorithms. For example, the precision and recall metrics that are widely used in event classification problems have their origins in the information retrieval domain.

Consequently, it is a common practice to analyze how the different performance metrics behave when applied to a different machine learning problem, such that it is possible to ascertain to what extent and in which situations the results and conclusions obtained using such metrics can be extended to the new problem. For instance, in [6], [7] the authors study the performance of different classification algorithms across multiple problems (each problem was represented by its own dataset) using a variety of metrics with the goal of understanding which metrics were better suited in each situation.

In this paper we extend these two works to the domain of event-based NILM. More concretely, we analyze the behavior of 18 performance metrics when applied to evaluate six different classification algorithms across 11 datasets.

To this end, we first train and test the different algorithms across the 11 datasets by conducting a controlled parameters and features sweep. Then, for each resulting classification model we compute the respective performance metric values. We then investigate the existence of correlations between the results obtained from each performance metric. More particularly, we study the existence of linear ($Pearson$) and non-linear ($Spearman$) pairwise correlations. Finally, after the initial correlation analysis, we further explore the metrics correlation by means of hierarchical clustering.

The remaining of this paper is organized as follows. First we present the different algorithms, learning features, datasets and performance metrics that are used in this work. We then thoroughly describe the research method, and discuss our results. Finally we conclude this paper, outline some limitations and provide some directions for future work.

## II. Algorithms, Learning Features, Datasets and Performance Metrics

In this section, we describe the different algorithms, learning features, datasets and performance metrics that are used in this work.

### *A. Algorithms*

Literature reveals that a great number of classification algorithms have already been attempted in NILM research

[4]. Nevertheless, since in this work we are only interested in comparing the actual performance metrics, we have decided to implement only six classic supervised learning algorithms.

More concretely, we implemented discrete versions (i.e., the outputs are classes instead of scores) of the k-Nearest Neighbor (K-NN) [8], the K-Star (K*) [9], a Locally Weighted Naive Bayes (LWL-NB) [10], a Classification Support Vector Machine (SVM) with a Radial Basis Function (RBF) [11], a two-layer Artificial Neural Network (ANN) [12] and the J48 Decision Tree Classifier (DT) [13] using the Waikato Environment for Knowledge Analysis (Weka) software [14].

### *B. Learning Features*

Regarding the learning features, we evaluate each classification algorithm using 30 feature sets, composed from 13 different features. Next we briefly describe the individual features.

*1) Deltas:* Delta features measure the average amount of change of a particular power metric, and are extracted by computing the difference between the average values of the samples in a post- and a pre-event window. In this particular case, we are using the delta features for real power ($P$), reactive power ($Q$) and current RMS ($I$).

*2) Harmonics:* Despite the presence of harmonic powers in the grid is not the most desirable situation, as they can degrade the mains efficiency, they provide a very attractive method of characterizing the different electric loads. Here we are using current ($H_{I,n}$) and instantaneous power ($H_{IV,n}$) harmonics up to the $21^{st}$ component.

*3) Raw and Quantized Waveforms:* Raw waveform features consist of a number of measurements of a particular metric taken from within the vicinity of the power event. Quantized waveforms are down-sampled versions of the raw waveforms that are obtained by quantizing the raw data into n bins. In this work, we use measurements taken from one period of instantaneous current ($I_{WF}$ and $I_{QWF}$) as well as quantized measurements taken from one period of instantaneous current combined with one period of instantaneous voltage ($IV_{QWF}$). For the quantization procedure, we set $n = 20$ and each bin is represented by the respective median.

*4) Data-driven:* Data driven features are learned directly from the data without the necessity of incorporating any domain knowledge. Here we use the set of features that was identified and explored in [15]. More particularly we use VI binary images ($VI_{BIN}$) and a number of principal components extracted from the binary images ($VI_{BIN\_PCA}$) and the raw and quantized waveforms ($I_{WF\_PCA}$, $I_{QWF\_PCA}$ and $IV_{QWF\_PCA}$).

### *C. Datasets*

In this work we use the PLAID dataset [16], which contains current and voltage measurements for 11 appliance types measured across 55 houses.

Recent works with this dataset have considered each house in PLAID as if it was a different dataset [15], [17], in what can be considered a variation of the 1-fold cross validation technique. In other words, the data from one house is used as test data while the remaining 54 houses are used to train the learning algorithms.

Here instead we have decided to split PLAID into eleven different datasets where each one is constituted by the data of five houses. The reasons behind this decision are twofold: i) to compensate for the fact that the number of examples can be considerably different between houses (e.g., in the most extreme case we have one house with only two events and another with 36), and ii) to have a more manageable number of datasets when performing the different comparisons.

### *D. Metrics*

In NILM the classification task is a multi-class problem, i.e., each power event can be classified into more than two different appliances. As such, most of the performance metrics available for this kind of problems were adapted from their binary classification counterparts.

Multi-class classification metrics can be calculated over the entire class collection, which is called micro-averaging, or by averaging the performance of each individual class, which is called macro-averaging [18].

In micro-averaging, each class counts the same for the average, as such larger classes dominate the measure; In macro-averaging, first the average for each class is determined, and only then each class counts the same for the final average. This difference is particularly important when the collection is skewed, which is indeed the case of NILM, since in a household it is expect that some appliances will trigger much more power events than others.

Macro-average metrics are not without their own caveats. For instance, one evident issue with macro-averaging is that it does not consider the number of samples in each class. Hence if there are very few examples of one appliance, the metric values for that appliance will be unreliable since it will tend to have a large variance that will necessarily affect the statistical significance of the final per-class average. Consequently, it is common practice to weight the individual class metrics by the respective number of instances, thus making the final average less sensitive to smaller classes. This is known as weighted macro-average.

Next we briefly describe the performance metrics used in this work.

*1) Confusion Matrix Based Metrics:* As the name suggests, confusion matrix based metrics are derived from the values in the confusion matrix.

In this work we selected 13 metrics, namely: Accuracy ($A$), Error-rate ($E$), Precision ($P$), Recall ($R$), $F_{0.5}$, $F_1$, $F_2$, Standardized Mathews Correlation Coefficient ($SMCC$) [19], False Positive Rate ($FPR$), False Positive Percentage ($FPP$), Precision-Recall Distance to Perfect Score ($DPS_{PR}$), Recall-FPR Distance to Perfect Score ($DPS_{Rate}$), and Recall-FPP Distance to Perfect Score ($DPS_{Perc}$).

The $DPS_{Rate}$ and $DPS_{Perc}$ metrics were originally defined to evaluate event detection algorithms [20]. Here we adapt these two metrics and add the $DPS_{PR}$. Lastly,it is also important to remark that in the case of micro-average metrics, $P$, $R$ and $F_\beta$ all have the same value [18]. As such, we only consider the $F_1$ metric.

*2) Area Under Curve Metrics:* The Area Under the Receiver Operating Characteristic curve ($ROC-AUC$) is commonly used as a summary of two performance metrics ($R$ and $FPR$), and is traditionally calculated using the trapezoidal rule when evaluating scoring classifiers. However, since in our scenario we are using discrete algorithms the $AUC$ should not be measured by employing that rule given that the possible presence of outliers could lead to distorted results [21]. Instead, the nonparametric Wilcoxon statistic is used, as suggested by Hanley and Mcneil [22].

In this work, we selected three variations of the $ROC-AUC$, namely, the Wilcoxon based ROC-AUC ($WAUC$), the Wilcoxon based ROC-AUC Balanced ($WAUCB$), and the Geometric Mean AUC ($GAUC$). A more detailed explanation of each metric can be found in [19].

*3) Probabilistic Metrics:* In this work we also look at probabilistic metrics, that is, metrics that measure how far the predictions are from the true result. More precisely, we investigate the Mean Absolute Error ($MAE$), and the Root Mean Squared Error ($RMSE$) [21].

## III. Methods

In this section we thoroughly describe the methods used in this paper.

### A. Training and Testing

In order to gain deeper insights on the nature and structure of the data that is generated by the event classification algorithms we perform a sweep of the their parameters and learning features.

Regarding the parameter sweep, we decided to switch only one parameter of each algorithm while leaving the remaining parameters set to their default values. Using this strategy, we ensure that each classification algorithm is tested the same number of times, but more importantly, we assure that changes in the obtained results are fully justified by one single parameter and the set of learning features. In table I we list the parameters that are switched in each algorithm, and the respective values.

As for the feature sweep we have decided to evaluate each classification algorithm using 30 feature sets. The feature sets are presented in table II and were created from the 13 learning features presented above. We refer to the features sets from 1 to 12 as single-feature since they either contain a single feature (sets 3 to 5 and 7 to 11) or combine features of the same type (sets 1, 2, 6 and 12). The remaining 18 sets are referred to as multi-feature since they combine features from different types.

TABLE I
LIST OF SWEPT PARAMETERS PER ALGORITHM AND RESPECTIVE VALUES

| Algorithm | Parameter | Values |
|---|---|---|
| K-NN | Number of neighbors ($K$) | 1, 3, $\sqrt{n_{sc}}$, $\frac{n_{sc}}{2}$, $n_{sc}$ |
| KStar | Blending ($b$) | 1%, 20% 50%, 75%, 100% |
| LWL-NB | Number of neighbors ($K$) | $n_t$, 3, $\sqrt{n_{sc}}$, $\frac{n_{sc}}{2}$, $n_{sc}$ |
| DT | Min instances per leaf ($minNumObjs$) | 1, 2, 5, 10, 15 |
| ANN | Learning rate ($learningRate$) | 0.1, 0.2, 0.3, 0.4, 0.5 |
| SVM | Miss-classification cost ($C$) | 0.01, 0.1, 1, 10, 100 |

Naturally we did not attempt each possible combination of features, since: i) with 13 individual features there will be a combinatorial explosion of the possible features sets, hence making this task extremely time consuming, and, ii) some of the features contain similar information (e.g., raw and quantized waveforms), which could easily result in overfitting. Therefore, we decided to select feature combinations that are complementary. For example, the feature sets 13 to 16 combine delta features with harmonic and waveforms features. It is also important to remark that the multi-feature sets are scaled before being passed to the learning algorithms, hence avoiding that learning features with higher values become more preeminent in the final results.

Finally, regarding the evaluation procedure we decided to split the training and testing sets by individual dataset. In other words, all the measurements from one dataset are used as testing data while the remaining ones are used as training data. This process is repeated once for each of the 11 datasets. By following this approach the models are always tested with previously unseen data, thus reducing the chance of over-

TABLE II
LIST WITH THE DIFFERENT FEATURE SETS

| Category | Features |
|---|---|
| Delta | 1: $[P,Q]$ 2: $[P,Q,I]$ |
| Harmonics | 3: $[H_{I.n}]$ 4: $[HIV,n]$ |
| Raw Waveforms | 5: $[I_{WF}]$ |
| Quantized Waveforms | 6: $[I_{QWF}]$ 7: $[IV_{QWF}]$ |
| | 8: $[VI_{BIN}]$ 9: $[VI_{BIN\_PCA}]$ |
| Data-driven | 10: $[I_{WF\_PCA}]$ 11: $[I_{QWF\_PCA}]$ |
| | 12: $[IV_{QWF\_PCA}]$ |
| Combined | 13: $[P,Q,H_{I,n}]$ 14: $[P,Q,H_{IV,n}]$ |
| | 15: $[P,Q,I_{WF}]$ 16: $[P,Q,IV_{WF}]$ |
| | 17: $[P,Q,VI_{BIN\_PCA}]$ |
| | 18: $[P,Q,H_{I,n},I_{QWF}]$ |
| | 19: $[P,Q,H_{I,n},IV_{QWF}]$ |
| | 20: $[P,Q,H_{IV,n},I_{QWF}]$ |
| | 21: $[P,Q,H_{IV,n},IV_{QWF}]$ |
| | 22: $[P,Q,H_{I,n},VI_{BIN\_PCA}]$ |
| | 23: $[P,Q,H_{IV,n},VI_{BIN\_PCA}]$ |
| | 24: $[H_{I,n},VI_{BIN\_PCA}]$ |
| | 25: $[H_{IV,n},VI_{BIN\_PCA}]$ |
| | 26: $[I_{QWF},VI_{BIN\_PCA}]$ |
| | 27: $[IV_{QWF},VI_{BIN\_PCA}]$ |
| | 28: $[I_{QWF\_PCA},VI_{BIN\_PCA}]$ |
| | 29: $[IV_{QWF\_PCA},VI_{BIN\_PCA}]$ |
| | 30: $[I_{WF},VI_{BIN\_PCA}]$ |

fitting during training. Furthermore, since all the models are trained with a large and diverse set of examples, the chance of classification bias is also reduced.

### B. Performance Metrics Calculation

In this step we compute the performance metrics for each of the models that result from the parameter and feature sweeps. To do this, we first count the true positives, false positives, true negatives and false negatives (i.e., the contingency matrix) for each of the tested models. The resulting contingency matrices are then used to calculate the different performance metrics.

This was done using the *one-vs-all* approach, where one binary confusion matrix is created for each class on the training data and later summed to form the final *one-vs-all* confusion matrix [23].

### C. Pairwise Correlations

In this step we compute the linear ($Pearson$) and non-linear ($Spearman$) pairwise correlations between the performance metrics that are used to evaluate the classification models. We then calculate a cross-dataset pairwise correlation matrix for each coefficient.

Considering that for one particular classifier there are $x$ metrics, this allows for $n = x \times (x-1)/2$ unique pairwise correlations. Thus, in our case there are, for each correlation coefficient, 171 and 231 unique pairwise correlations for the micro- and macro-average metrics, respectively. Furthermore, considering that each model is evaluated against 11 datasets, there is a total of 66 correlations matrices. These matrices are then averaged to form a cross-dataset correlation matrix for each correlation coefficient.

It is important to note that, under no circumstances we merge the evaluation results obtained from the different model-dataset pairs. Instead, we merge only the pairwise metric correlations. The reason for this is the fact that there is evidence that event detection and classification algorithms depend heavily on the datasets [24]. Thus, producing cross-dataset averages can lead to biased conclusions since it is possible that good results in one dataset compensate for poor results in other datasets and vice-versa.

### D. Hierarchical Clustering

In this step we build clusters from the resulting average pairwise correlation matrices using hierarchical clustering. To do so we first define the dissimilarity function, i.e., a function that defines the distance between two clusters (or metrics). Then, we define the linkage function that is used to join (i.e., cluster) the different pairs of metrics and clusters.

Regarding the former, in this work we use the dissimilarity function that is defined in equation 1, where $D$ is the distance and $|C|$ is the absolute value of the correlation between the clusters.

$$D = 1 - |C| \tag{1}$$

As for the linkage distance, we use the average-group distance, which joins an existing group to the element (or group) whose average distance to the group in minimum. This method is also known as Un-weighted Pair Group Method with Arithmetic Mean (UPGMA) and the distance between two groups $A$ and $B$ is given by equation 2, where $d$ is a distance function (in our case the Euclidean distance) and $|A|$ and $|B|$ are the size of groups $A$ and $B$, respectively.

$$D_{AB} = \frac{1}{|A||B|} \times \sum_{a \in A} \sum_{b \in B} d(a,b) \tag{2}$$

## IV. Results and Discussion

The average rank and linear correlations between all the performance metric is presented in figure 1. Metrics with average pairwise correlations closer to one ($\rho \geq 0.9$) appear highlighted as they are expected to behave more similarly than others.

A first general observation is that for any of the averaging techniques, the resulting correlations are very strong. This is particularly evident in the micro-average metrics ($\bar{\rho} = 0.96$). The very strong pairwise correlations are also expressed in the dendrogram show in figure 2, where it can be observed that only the probabilistic metrics appear outside the main cluster.

Ultimately, when using micro-average metrics that are based on the confusion-matrix or $AUC$, the larger classes will dominate the metric, which in the case of NILM can become problematic due to the unbalanced nature of the problem. For example, classifiers that do a great job with refrigerators but fail to classify appliances with less examples (e.g., a coffee machine or a toaster) will be ranked similarly to classifiers that happen to also correctly classify the less represented appliances.

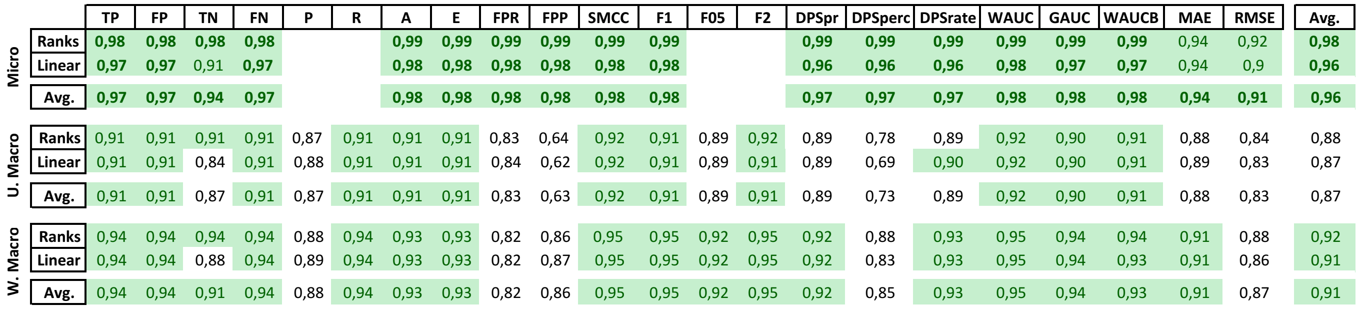
With regard to the macro-average metrics it is possible to observe strong pairwise correlations between all the $AUC$ metrics, with both coefficients above 0.95. However, since they form different clusters they should not be used interchangeably. For example, it can be seen from the dendrograms in figures 3 and 4 that $GAUC$ is closely correlated to $DPS_{Rate}$, whereas $WAUC$ is more correlated with $F_2$. Ultimately, this confirms early finding from [6], [7] that for classification problems $AUC$ metrics tend to correlate well between themselves, and with most performance metrics ($\rho > 0.85$).

Other metrics that evidence very strong pairwise correlations are the $SMCC$ and the three $F_\beta$-measures, in both weighted and unweighted versions. This confirms the theoretical guarantees that $SMCC$ is a metric that can be safely applied to both balanced and unbalanced problems [25].

From a more individual perspective, it is also possible to observe that some metrics do not correlate well with any metrics. This is the case of $FPR$, $FPP$ and $DTP_{Perc}$, which as it can be seen from the dendrograms in figures 3 and 4, will not join other cluster until a cut-off distance of at least 0.2.

Likewise, it is possible to observe that the weighted macro-average Precision is also isolated (it only joins other metrics at a cut-off distance around 0.17), which may be an indicator that this metric tends to be more sensitive to unbalanced datasets than the others.

Finally, we should also mention the fact that the probabilistic metrics have lower average correlation values when



| | | TP | FP | TN | FN | P | R | A | E | FPR | FPP | SMCC | F1 | F05 | F2 | DPSpr | DPSperc | DPSrate | WAUC | GAUC | WAUCB | MAE | RMSE | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Micro | Ranks | **0,98** | **0,98** | **0,98** | **0,98** | | | **0,99** | **0,99** | **0,99** | **0,99** | **0,99** | **0,99** | | | **0,99** | **0,99** | **0,99** | **0,99** | **0,99** | **0,99** | 0,94 | 0,92 | **0,98** |
| | Linear | **0,97** | **0,97** | 0,91 | **0,97** | | | **0,98** | **0,98** | **0,98** | **0,98** | **0,98** | **0,98** | | | **0,96** | **0,96** | **0,96** | **0,98** | **0,97** | **0,97** | 0,94 | 0,9 | **0,96** |
| | Avg. | **0,97** | **0,97** | **0,94** | **0,97** | | | **0,98** | **0,98** | **0,98** | **0,98** | **0,98** | **0,98** | | | **0,97** | **0,97** | **0,97** | **0,98** | **0,98** | **0,98** | **0,94** | **0,91** | **0,96** |
| U. Macro | Ranks | 0,91 | 0,91 | 0,91 | 0,91 | 0,87 | 0,91 | 0,91 | 0,91 | 0,83 | 0,64 | 0,92 | 0,91 | 0,89 | 0,92 | 0,89 | 0,78 | 0,89 | 0,92 | 0,90 | 0,91 | 0,88 | 0,84 | 0,88 |
| | Linear | 0,91 | 0,91 | 0,84 | 0,91 | 0,88 | 0,91 | 0,91 | 0,91 | 0,84 | 0,62 | 0,92 | 0,91 | 0,89 | 0,91 | 0,89 | 0,69 | 0,90 | 0,92 | 0,90 | 0,91 | 0,89 | 0,83 | 0,87 |
| | Avg. | 0,91 | 0,91 | 0,87 | 0,91 | 0,87 | 0,91 | 0,91 | 0,91 | 0,83 | 0,63 | 0,92 | 0,91 | 0,89 | 0,91 | 0,89 | 0,73 | 0,89 | 0,92 | 0,90 | 0,91 | 0,88 | 0,83 | 0,87 |
| W. Macro | Ranks | 0,94 | 0,94 | 0,94 | 0,94 | 0,88 | 0,94 | 0,93 | 0,93 | 0,82 | 0,86 | 0,95 | 0,95 | 0,92 | 0,95 | 0,92 | 0,88 | 0,93 | 0,95 | 0,94 | 0,94 | 0,91 | 0,88 | 0,92 |
| | Linear | 0,94 | 0,94 | 0,88 | 0,94 | 0,89 | 0,94 | 0,93 | 0,93 | 0,82 | 0,87 | 0,95 | 0,95 | 0,92 | 0,95 | 0,92 | 0,83 | 0,93 | 0,95 | 0,94 | 0,93 | 0,91 | 0,86 | 0,91 |
| | Avg. | 0,94 | 0,94 | 0,91 | 0,94 | 0,88 | 0,94 | 0,93 | 0,93 | 0,82 | 0,86 | 0,95 | 0,95 | 0,92 | 0,95 | 0,92 | 0,85 | 0,93 | 0,95 | 0,94 | 0,93 | 0,91 | 0,87 | 0,91 |

Fig. 1. Non-linear and linear pairwise correlations averaged by metric for all the datasets

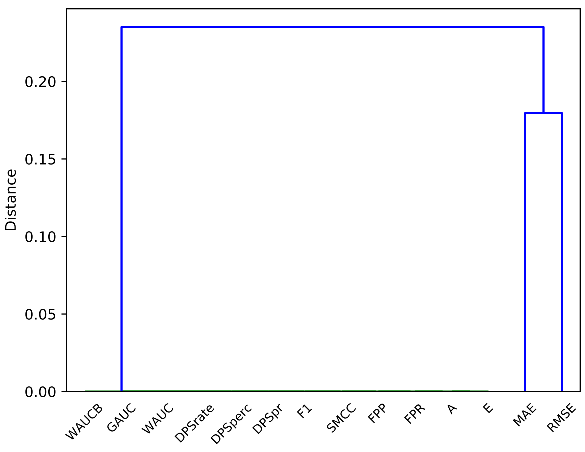


Fig. 2. Dendrogram for the non-linear correlations between the micro-average metrics

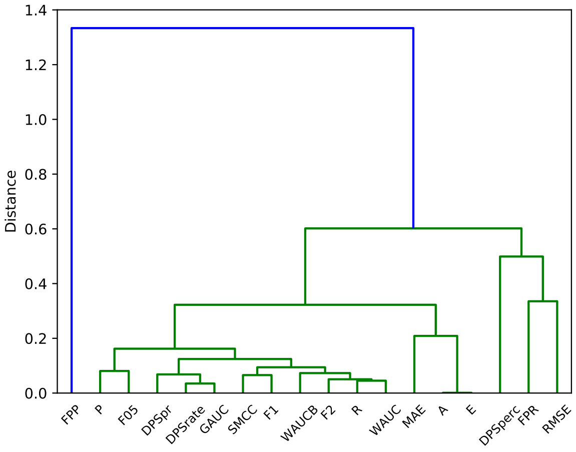


Fig. 3. Dendrogram for the non-linear correlations between the unweighted macro-average metrics

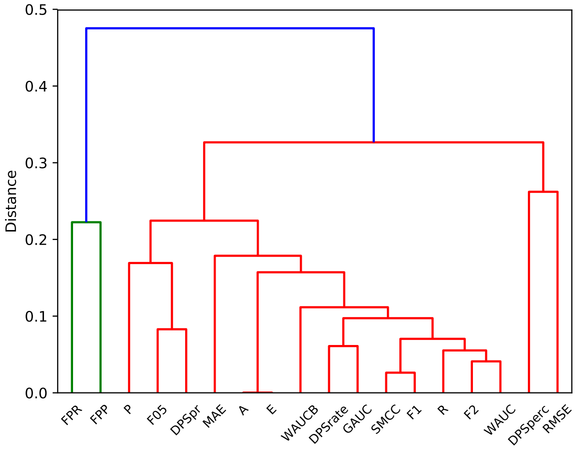


Fig. 4. Dendrogram for the non-linear correlations between the weighted macro-average metrics

TABLE III
CLUSTERS FORMED BY THE UNWEIGHTED AND WEIGHTED MACRO-AVERAGE NON-LINEAR PAIRWISE CORRELATIONS

| Dist. | Unweighted | Weighted |
|---|---|---|
| 0.05 | 1: [$DPS_{Rate}$, $GAUC$]<br>2: [$R$, $WAUC$]<br>3: [$A$, $E$] | 1: [$F_1$, $SMCC$]<br>2: [$F_2$, $WAUC$]<br>3: [$A$, $E$] |
| 0.1 | 1: [$F_1$, $F_2$, $SMCC$, $WAUC$, $GAUC$, $DPS_{Rate}$, $R$]<br>2: [$DPS_{PR}$, $F_{0.5}$]<br>3: [$A$, $E$] | 1: [$F_1$, $F_2$, $SMCC$, $WAUC$, $WAUCB$, $R$]<br>2: [$DPS_{Rate}$, $GAUC$]<br>3: [$DPS_{PR}$, $F_{0.5}$] 4: [$A$, $E$] |

compared to all the other metrics. This result is also in line with previous research [7], that argues in favor of using probabilistic metrics in the evaluation process, since they report performance taking into consideration not just the number of errors but also the distance to the true values. To summarize, in table III we list the clusters obtained after cutting the above dendrograms at cutoff distances of 0.05 and 0.1.

## V. Conclusion, Limitations and Future Work

In this paper we performed an experimental comparison of performance metrics for event classification algorithms in NILM. On the contrary to what one would expect, our results indicate that when applied to the NILM problem the behavior of the performance metrics is very similar to that exhibited when applied to other domains like medical diagnosis and face recognition.

There are however a few differences resulting from the unbalanced nature of the NILM problem, that are important to remark. For example, it is clear from this work that micro-average metrics are of very little use in this problem, since all the metrics end-up reporting exactly the same thing.

Likewise, it is also possible to observe that weighted and unweighted macro-average metrics have a slightly different behavior, especially the weighted macro-average Precision. This suggest that the weighted macro-averaging technique is more sensitive to unbalanced problem and therefore more suitable to the NILM case.

Nevertheless, at this stage it is not possible to totally conclude this, in particular due to some limitations in the selected datasets. For instance, the 11 datasets that we have used in this work do not totally represent what is actually happening in a real household since the number of instances for the different appliances does not follow the real distribution of the power events. Furthermore, we should mention that the datasets only contain positive transitions (i.e., loads going from the OFF to the ON state), and that all the examples were carefully commissioned such that the extracted features were the best possible, which is naturally far from the conditions that NILM algorithm will face when deployed in real houses.

Consequently, future analyses of performance metrics for event classification in NILM should be conducted using scenarios that are closer to those that will occur in real world situations (e.g., erroneous detections and previously unseen appliances).

A simple method to achieve this would be by deliberately introducing examples with erroneous and previously unseen classes, hence mimicking the presence of false positives and previously unseen appliances. Another possibility would be by introducing the concept of ceiling analysis [26], where event detection, feature extraction and event classification are executed in sequence and the output of one algorithm is the input of the other.

## Acknowledgment

The authors would like to thank professor Mario Bergés, for his valuable insights during the elaboration of this work.

## References

[1] G. Hart, “Prototype Nonintrusive Appliance Load Monitor,” MIT Energy Laboratory Technical Report, and Electric Power Research Institute Technical Report, Tech. Rep., Sep. 1985.

[2] M. Berges, E. Goldman, H. Matthews, L. Soibelman, and K. Anderson, “User-Centered Nonintrusive Electricity Load Monitoring for Residential Buildings,” *Journal of Computing in Civil Engineering*, vol. 25, no. 6, pp. 471–480, 2011.

[3] Z. Kolter and T. Jaakkola, “Approximate Inference in Additive Factorial HMMs with Application to Energy Disaggregation,” in *JMLR: W&CP 22*, vol. 22, La Palma, Canary Islands, Spain, 2012, pp. 1472–1482.

[4] M. Zeifman and K. Roth, “Nonintrusive appliance load monitoring: Review and outlook,” *IEEE Transactions on Consumer Electronics*, vol. 57, no. 1, pp. 76–84, 2011.

[5] E. T. Mayhorn, G. P. Sullivan, J. M. Petersen, R. S. Butner, and E. M. Johnson, “Load Disaggregation Technologies: Real World and Laboratory Performance,” Pacific Northwest National Laboratory (PNNL), Richland, WA (US), Tech. Rep. PNNL-SA-116560, Sep. 2016.

[6] R. Caruana and A. Niculescu-Mizil, “Data Mining in Metric Space: An Empirical Analysis of Supervised Learning Performance Criteria,” in *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD ’04. New York, NY, USA: ACM, 2004, pp. 69–78.

[7] C. Ferri, J. Hernández-Orallo, and R. Modroiu, “An experimental comparison of performance measures for classification,” *Pattern Recognition Letters*, vol. 30, no. 1, pp. 27–38, Jan. 2009.

[8] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, Jan. 1967.

[9] J. G. Cleary and L. E. Trigg, “K*: An Instance-based Learner Using an Entropic Distance Measure,” in *In Proceedings of the 12th International Conference on Machine Learning*. Morgan Kaufmann, 1995, pp. 108–114.

[10] E. Frank, M. Hall, and B. Pfahringer, “Locally Weighted Naive Bayes,” in *Proceedings of the Nineteenth Conference on Uncertainty in Artificial Intelligence*, ser. UAI’03. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2003, pp. 249–256.

[11] C.-C. Chang and C.-J. Lin, “LIBSVM: A Library for Support Vector Machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 27:1–27:27, May 2011.

[12] C. M. Bishop, *Neural Networks for Pattern Recognition*. New York, NY, USA: Oxford University Press, Inc., 1995.

[13] Y. Zhao and Y. Zhang, “Comparison of decision tree methods for finding active objects,” *Advances in Space Research*, vol. 41, no. 12, pp. 1955–1959, Jan. 2008, arXiv: 0708.4274.

[14] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, “The WEKA Data Mining Software: An Update,” *SIGKDD Explor. Newsl.*, vol. 11, no. 1, pp. 10–18, Nov. 2009.

[15] J. Gao, E. Can Kara, S. Giri, and M. Bergés, “A feasibility study of automated plug-load identification from high-frequency measurements,” in *2015 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2015.

[16] J. Gao, S. Giri, E. C. Kara, and M. Bergés, “PLAID: A Public Dataset of High-resoultion Electrical Appliance Measurements for Load Identification Research: Demo Abstract,” in *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*, ser. BuildSys ’14. New York, NY, USA: ACM, 2014, pp. 198–199.

[17] K. Barsim, L. Mauch, and B. Yang, “Neural Network Ensembles to Real-time Identification of Plug-level Appliance Measurements,” in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Vancouver, BC, Canada, 2016.

[18] V. Van Asch, “Macro-and micro-averaged evaluation measures [[basic draft]],” 2013.

[19] H. Iba, Y. Hasegawa, and T. K. Paul, *Applied Genetic Programming and Machine Learning*, 1st ed. Boca Raton, FL, USA: CRC Press, Inc., 2009.

[20] K. D. Anderson, M. E. Bergés, A. Ocneanu, D. Benitez, and J. M. F. Moura, “Event detection for Non Intrusive load monitoring,” in *IECON 2012 - 38th Annual Conference on IEEE Industrial Electronics Society*, Oct. 2012, pp. 3312–3317.

[21] C. J. Willmott and K. Matsuura, “Advantages of the mean absolute error (MAE) over the root mean square error (RMSE) in assessing average model performance,” *Climate Research*, vol. 30, no. 1, pp. 79–82, Dec. 2005.

[22] J. A. Hanley and B. J. McNeil, “The meaning and use of the area under a receiver operating characteristic (ROC) curve,” *Radiology*, vol. 143, no. 1, pp. 29–36, Apr. 1982.

[23] R. Rifkin and A. Klautau, “In Defense of One-Vs-All Classification,” *Journal of Machine Learning Research*, vol. 5, no. Jan, pp. 101–141, 2004.

[24] N. Czarnek, K. Morton, L. Collins, R. Newell, and K. Bradbury, “Performance comparison framework for energy disaggregation systems,” in *2015 IEEE International Conference on Smart Grid Communications (SmartGridComm)*, Nov. 2015, pp. 446–452.

[25] B. W. Matthews, “Comparison of the predicted and observed secondary structure of T4 phage lysozyme,” *Biochimica et Biophysica Acta (BBA) - Protein Structure*, vol. 405, no. 2, pp. 442–451, Oct. 1975.

[26] H. Roncancio, A. Hernandes, and M. Becker, “Ceiling analysis of pedestrian recognition pipeline for an autonomous car application,” in *2013 IEEE Workshop on Robot Vision (WORV)*, Jan. 2013, pp. 215–220.