INCLASS: Incremental Classification Strategy for Self-Aware Epileptic Seizure Detection

Lorenzo Ferretti[1], Giovanni Ansaloni[2],Renaud Marquis[3], Tomas Teijeiro[2], Philippe Ryvlin[3], David Atienza[2], Laura Pozzi[1]

[1]Computer Systems Institute, Università della Svizzera italiana, Lugano, Switzerland,

[2]Embedded Systems Laboratory, EPFL, Lausanne, Switzerland

[3]Lausanne University Hospital, CHUV, Lausanne, Switzerland

***Abstract*—Wearable Health Companions allow the unobtrusive monitoring of patients affected by chronic conditions. In particular, by acquiring and interpreting bio-signals, they enable the detection of acute episodes in cardiac and neurological ailments. Nevertheless, the processing of bio-signals is computationally complex, especially when a large number of features are required to obtain reliable detection outcomes. Addressing this challenge, we present a novel methodology, named INCLASS, that iteratively extends employed feature sets at run-time, until a confidence condition is satisfied. INCLASS builds such sets based on code analysis and profiling information. When applied to the challenging scenario of detecting epileptic seizures based on ECG and SpO2 acquisitions, INCLASS obtains savings of up to 54%, while incurring in a negligible loss of detection performance (1.1% degradation of specificity and sensitivity) with respect to always computing and evaluating all features.**

***Index Terms*—self-aware systems, epileptic seizure detection, wearable health companions**

I. Introduction

Epilepsy is one of the most common neurological diseases, affecting more than 50 million people worldwide [1]. In one third of patients whose seizures are not well controlled by available therapies, the occurrence of seizures entails the risk of serious consequences, including sudden unexpected deaths (SUDEP) [2] [3]. The continuous monitoring of patients in ambulatory settings might help minimizing such risks by enabling timely interventions when epileptic attacks occur.

Key enablers for such scenario are Wearable Health Companions (WHCs), which monitor bio-signals such as ECG and SpO2 [4] [5] [6], whose acquisition is much less obtrusive when compared to traditional approaches based on EEG. Indeed, seizures are fequently responsible for changes in various cardiac features [7] [8] [9]. The calculation of a large number of computationally complex features, however, may strain the limited resources of WHCs. Moreover, each has a different discriminative power, which also significantly varies across patients.

Such scenario exposes a complex optimization problem: that of identifying the subset of features that maximizes the accuracy of seizure detection, while minimising the entailed workload. Addressing it, we propose a methodology, named INCLASS, which leverages self-aware mechanisms [10]. Self-awareness postulates that intelligent systems should continuously measure their own performance, and autonomously adapt in relation to it. Embodying such paradigm, our approach is based on iteratively performing seizure detection using classifiers with *increasing complexity*, while at each step measuring the classification confidence. In this way, the extraction of computationally-demanding features is only performed when required by the confidence score, and instead waived when not required.

We herein propose solutions to the two main challenges arising from such stance. First, we illustrate a **design-time** methodology to automatically construct classifiers based on increasingly complex feature sets. To this end, our approach relies on the traversing of call graph trees, annotated with execution time profiling and feature discriminance information. Then, at **run-time** we introduce a strategy for the (self-) assessment of the confidence in a detection outcome from a given feature set. In case of high confidence, the outcome is accepted, otherwise the process is repeated while increasing the feature set size, and hence the complexity.

The main contributions of our work are the following:

- We show an automated framework for the analysis of feature discriminance and cost, allowing the identification of classification models with increasing complexity.
- We propose a self-aware metric to assess the quality of classification outcomes, and a self-adaptive strategy which selectively invokes more complex classifiers based on it.
- We apply our INCLASS methodology to the concrete scenario of seizure detection based on ECG- and SpO2-derived features, validating it on extensive acquisitions annotated by medical experts. We demonstrate that our approach results in workload reduction of up to 54%, with only a 1.1% reduction in classification accuracy.

The rest of the paper is organized as follows: Section II presents the most relevant works in the context of seizure detection and self-aware systems. Section III describes the INCLASS framework proposed in this work. Then, Section IV describes the experimental setup and the used dataset, and Section V shows the results achieved by our methodology. Lastly, Section VI presents our conclusive remarks.

This work has been partially supported the ML-edge (Grant: 200020-182009) and ADApprox (Grant: 200020-188613) project funded by the Swiss NSF, and by the EC H2020 FVLLMONTI project (GA No. 101016776), and the ERC Consolidator grant COMPUSAPIEN (GA No. 725657).

II. RELATED WORKS

Seizure monitoring of epileptic patients with WHCs must provide high detection accuracy while performing the low-power processing of unobtrusive acquisitions. As an alternative to stigmatising EEG apparatuses, several approaches based on accelerometry and muscle contractions measurements have been proposed, targeting seizures with strong motor components [11] [12] [13].

Recently, several studies have proposed using cardiac sensors instead of (or complementing) accelerometers. Indeed, the authors of [14] [15] [16] base their analysis on Heart Rate (HR) measurements. Higher levels of sensitivity and specificity can be plausibly reached by combining multiple detection methods. Indeed, Cogan et al. [17] suggest that combining multiple extra-cerebral biological inputs acquired does increase performance. Their method considers measurements derived from a plethora of inputs including electrodermal activity (EDA), heart rate (HR), oxygen saturation (SpO2), temperature and accelerometry data. From these heterogeneous data sources, many different features can be derived. Limiting themselves to ECG and SpO2 acquisitions, the authors of [18] consider 17 different indicators, including time- and frequency-based ones.

The computation effort required for feature computation has a negative impact on the WHC efficiency, and negatively affects battery lifetime. In a recent work, Forooghifar et al. [19] have showcased that approaches based on self-awareness can address this challenge, leading at the same time to low energy consumption and high detection performance. In their work, the authors propose two different classifiers having different algorithmic complexities and performance, enabling the system to switch to low-power mode in order to reduce the energy consumption when possible.

We follow a similar approach, but, instead of manually designing a low-power and a high-performance model, we propose an end-to-end methodology to generate them *automatically*. We also devise a self-aware mechanism to switch among models at run-time based on a self-assessment of classification confidence.

III. THE INCLASS METHODOLOGY

To perform an interpretation of a monitored bio-signal, two steps are required, as shown in Figure 1a. First, a number of features must be extracted from the data acquired in a time window. Then, such features are given as input to a detector, which discriminates them between classes, e.g. seizure and non-seizure. By extracting and processing a smaller number of features (Figure 1b) less computation effort is incurred, and a different (lower) accuracy is obtained. The computation effort of feature extraction, in fact, changes dramatically with the number or features considered, while the cost of the detection phase is, in comparison, quite small.

Therefore, it is possible (and we propose to) perform multiple classifications/detections, starting from few features, and incrementally enlarging the feature set size, until a good-enough confidence is reached (Figure 1c). The overhead of performing multiple detections and testing the confidence level will then

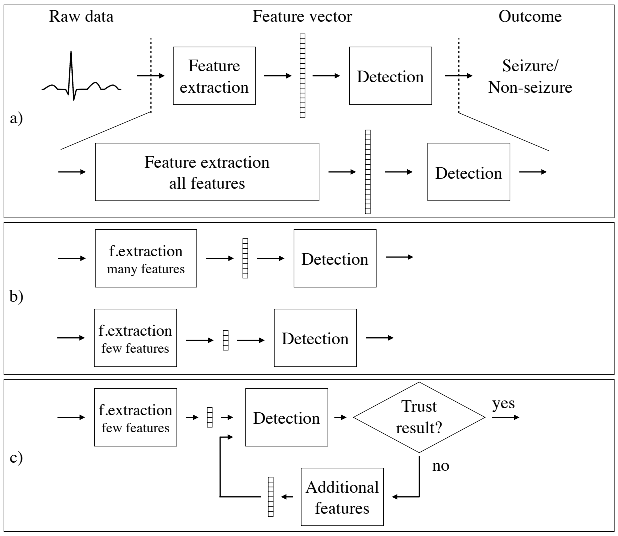

Fig. 1. a) Seizure detection flow and its different steps. b) Example of seizure detection models of decreasing complexity, which use reduced features sets. c) In a self-aware approach, features are incrementally acquired, and the resulting detection confidence is assessed.

be negligible with respect to the gains derived by only rarely employing more extensive feature sets.

But how and when to choose which features? We propose a system that can alternate at run-time among any number of higher and lower cost classifiers, built at design-time, depending on a desired cost-performance tradeoff.

In order to reach this capability, two main questions must be answered: 1) Out of all possible subsets of features, and hence out of all resulting classifiers, which are the most relevant to the problem? i.e. which provide a good tradeoff between cost and performance? 2) And given those subsets, at which moment in time should the system decide to employ one or the others? In the next section, we provide our answers to these questions.

A. Design Time: Model Generation / Classifier Construction

Addressing the first of the above-mentioned research questions, INCLASS identifies feature sets of increasing size. The starting point for such identification problem is a system computing and evaluating *all* available features (which we call "baseline model"). From it, INCLASS generates low-complexity models by assessing their importance (i.e.: their discriminative value) and computational cost obtained profiling the functions execution time (more details is Section V). In the following we describe how we can assign a value of importance and of cost to each model, and then we finally proceed to show the strategy to employ to generate models.

Feature cost: We calculate the cost of each feature by processing the source code of the application implementing the baseline model. We first build the application call graph, i.e. a graph where nodes represent the functions in the code, and edges represent calls between functions. We then profile it in order to extract the execution time for each node. Figure 2 shows the call graph of the feature extraction and detection implementation for the baseline model considered in this work, which, similarly to [18], extracts 17 different features. After profiling, we automatically annotate each node in the call

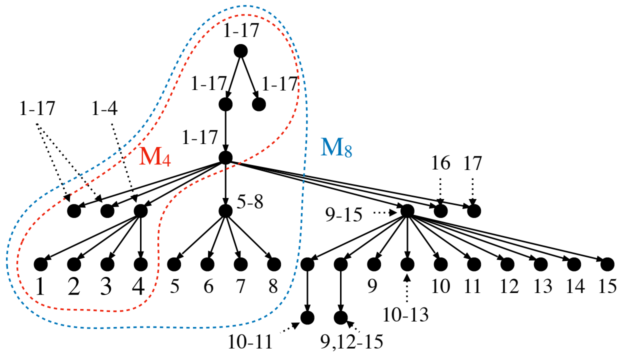

Fig. 2. Call graph of the seizure detection algorithm implementation. Nodes in the graph are function calls, and each node is labelled with the features that are computed in it. The dotted lines highlight the subgraphs of the call graph associated to feature extraction for two different sets of features (the first, in red dotted-line, for features 1 to 4; the second, blue dotted line, for features 1 to 8.)

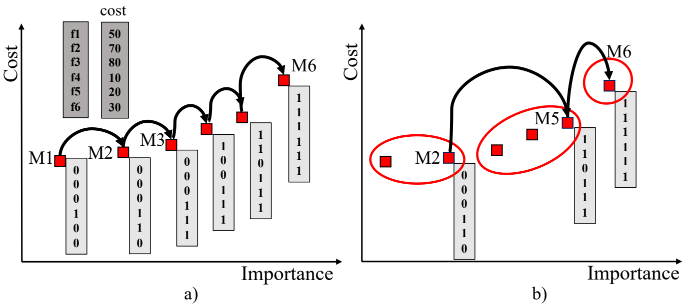

Fig. 3. a) Models depicted as points in the Cost/Importance search space. Each model is generated by incrementally adding a single feature to the previous one. In this example there are six features, indicated by the boolean vector of length 6 next to each point. The first model uses only feature 4 – the feature with lowest cost; the second uses features 5 (next in cost) in addition to feature 4, etc. b) Points are later clustered, in order to reduce the number of models used at run-time.

graph with the processing cost of the function. Nodes are further annotated, manually and via pragmas, with the ID of the features to which they contribute to. By traversing the call graph we can then calculate the total cost of a feature set (i.e. a model), by summing up the cost of the functions that are annotated with the IDs of the features in that set.

As an example, Figure 2 highlights the portion of the call graph associated to two different sets of features, which in turn generate two different models: one including features 1 to 4, and the other including features 1 to 8.

Feature importance: We measure the importance of the features on detection by evaluating the contribution that each feature has on the estimation outcome. While such measures can be obtained for a variety of detection strategies [20], we restrict ourselves to bagged Decision Trees (DTs) classifiers.

We measure the contribution of each feature on the DTs using the the Gini impurity or index [21]. The Gini impurity is a general indicator of the feature importance and is obtained measuring the contribution – in term of information gain – that each feature has on the DT splits. Once the feature importance for each tree has been identified, these are averaged among all the trees in the ensemble, returning a feature score for each feature [22].

Design space of possible Models, and its exploration. Given n features there exist 2^n detection models in the design space. For the 17 features considered in our work, 131072 possible models hence exist.

This exponential growth of the design space with the number of features does not allow the generation and evaluation of all possible models. Furthermore, some of these models (subset of features) might be *dominated* by others, for example they might have lower importance for the same cost, and hence are of no interest. To explore the design space effectively, approximating the Pareto-front of best performing feature sets while avoiding exhaustive explorations, INCLASS adopts a heuristic that starts from the simpler possible classifier and incrementally builds on it. The first step consists in *ordering* available features according to a suitable criterium: we choose to order them in increasing cost[1]. Starting then with the classifier having a single feature – the one appearing first in the sequence, i.e. the one with the lowest cost – INCLASS then incrementally generates additional classifiers by adding each feature in order of increasing cost. Hence, given n features, the algorithm generates n different classifiers built incrementally. This is depicted in Figure 3a for a toy example consisting of six features.

The second step of this first, design-time phase is to further limit the number of considered classifiers at run-rime. Indeed, switching too many times among classifiers would cause the total execution time to grow rapidly, due to repeated execution of the detection phase. To avoid this issue, we propose to further limit the number of classifiers available at run-time by performing *hierarchical clustering* on the n generated points. For each cluster, we then retain only the model with highest importance score. This is depicted in Figure 3b: in this example, if the first 2 points are clustered together then a single model is generated (called M_2) which uses the first two features of the list. If the next 3 points are clustered, then a second classifier (called M_5) is generated which uses the first 5 features. etc. At run-time the system will then alternate among these three classifiers only (M_2, M_5, M_6). In general: in a model M_j all and only the first j features in the sequence (pre-ordered according to the chosen criterium; increasing cost in our case) are extracted.

This strategy is evaluated in Section V, where we show that the classifiers thus built well approximate the Pareto curve of the design space (Figure 5).

B. Run-time: Self-adaptation strategy

In order to take advantage of the INCLASS approach, the system, starting from the simpler trained model, performs a detection and then self-assesses its confidence level. According to the confidence score, the classification outcome (seizure/non-seizure) is either accepted, or a more complex model triggering the computation of further features is invoked.

In our methodology the confidence metric is defined as a score obtained by the tree detection of the model ensemble. This score is defined as the weighted average of the class posterior probabilities. For each decision tree, the confidence

[1] In case of features having the same cost, the feature with the higher importance is selected.

score is the probability of classifying a seizure (or non-seizure) given that input. This is the probability of the observation originating from the class (seizure or non-seizure), computed as the fraction of observations of the class in a tree leaf [23]. Then, scores for each decision tree are averaged to obtain a unique score value for the entire ensemble.

The main advantage of the INCLASS approach is to use simpler (from a computational viewpoint) classification strategies when detection is easy to perform, and use more complex ones only when required. Hence, run-time and the associated energy cost can be greatly reduced. Given i detection models ordered by complexity INCLASS enables to save energy when:

$$C_{ext\,i} + (C_{det} \cdot i) < C_{all} \tag{1}$$

where $C_{ext\,i}$ is the cost of extraction of all features used by model i, where the cost of detection (C_{det}) is payed i times instead of just once, and where C_{all} is the cost of using the baseline classifier, i.e. the one relying on the complete set of features. This condition is often satisfied in practice because, as discussed earlier in this section, C_{det} is small with respect to the cost of feature extraction, and because the step of hierarchical clustering reduces the number of classifiers used, in practice limiting term i above.

This run-time strategy is evaluated in the Experimental Section, where we consider various clustering factors and we show how we can tangibly save computing time for a negligible degradation in specificity and sensitivity.

IV. Experimental setup

A. Dataset

We have evaluated the INCLASS methodology using a dataset acquired at the CHUV Hospital in Lausanne, Switzerland. The dataset includes ECG and SPO2 records originating from four patients, as well as reference seizure annotations performed by medical experts. The signals have been split in 212 seizure and 1060 non-seizure windows (a $1/5$ ratio), and each window has a duration of 2 minutes.

We have divided the dataset in 70% training, 15% test and 15% validation data. Training data was used to train the models identified by INCLASS. Test data was employed to fine-tune the confidence level required to trust a classification outcome. Validation data was only employed to evaluate the performance of different configuration after training and fine-tuning. We repeated all experiments 5 times, with different (random) splitting of the training/test/validation datasets.

The baseline seizure detection application is inspired by [18]. It comprises a pre-processing phase which filters low-frequency baseline wandering and high-frequency (> 20 Hz) noise components, and a main feature extraction phase calculating 15 ECG and 2 SpO2 features. These features are listed in Table V-A, along with their cost and importance. ECG features include Lorentz Features [24] (7 features) and heart-rate variability features, (4 time-based and 4 frequency-based ones). SpO2 features are instead the mean and the standard deviation of the signal. All feature values are normalized with respect to the ones computed in a non-seizure window at the start of each data acquisition.

TABLE I
List of features considered in this work.
For each feature we report the feature name, its importance (normalized over total) and its cost.

ID	Feature Name	Importance	Cost
1	Mean RR	0.2594	0.1520
2	Std RR	0.0844	0.1525
3	RMSSD	0.1046	0.1511
4	pNN50	0.0468	0.1503
5	Total power	0.1444	0.4775
6	LF	0.0212	0.4924
7	HF	0.0309	0.4874
8	LF_HF	0.0227	0.5025
9	CSI filt. $\times$ slope	0.0108	0.1671
10	ModCSI filt. $\times$ slope	0.0213	0.1703
11	CSI $\times$ slope	0.0373	0.1703
12	ModCSI $\times$ slope	0.0106	0.1701
13	CSI	0.0150	0.1701
14	ModCSI	0.0153	0.1672
15	HR diff.	0.0246	0.1671
16	SPO2 mean	0.0682	0.1566
17	SPO2 std	0.0827	0.1582

We used ensembles of 30 Decision Trees as a classification strategy, employing bagging to improve generalization and reduce overfitting. We have implemented and evaluated the baseline model and the INCLASS framework using Matlab. Timing information about the profiled functions have been collected using the Matlab profiler over 100 different runs. For each function of the call graph the time spent executing them has been extracted and used to annotate the call graph nodes. Results are reported in terms of execution time savings with respect to the baseline model.

We measured the performance of seizure detection as the geometric mean (*GM*) of the achieved sensitivity (*Sen*) and specificity (*Spe*).

$$Sen = \frac{TP}{TP + FN} \quad Spe = \frac{TN}{TN + FP} \tag{2}$$

$$GM = \sqrt{Spe \cdot Sen} \tag{3}$$

where *TP* (True Positive) and *TN* (True Negative) represent correct classification of seizure and non-seizure respectively, while *FP* (False Positive) and *FN* (False Negative) are incorrect classification of seizure and non-seizures respectively. Sensitivity highlights the percentage of seizures that are classified correctly, while specificity shows the ability to properly classify non-seizure events.

V. Experimental Evaluation

In this section we describe the quality of results obtained by INCLASS. We analyze the classification performance and energy requirements of the proposed system, and we explore the different elements affecting the quality of the results, such as the selection of detection models and confidence level thresholds.

A. Results

1) Confidence threshold evaluation: Each time a feature window is processed, INCLASS generates a confidence score related to the detection outcome. This score is used to decide if the result can be trusted or if the framework should fall back

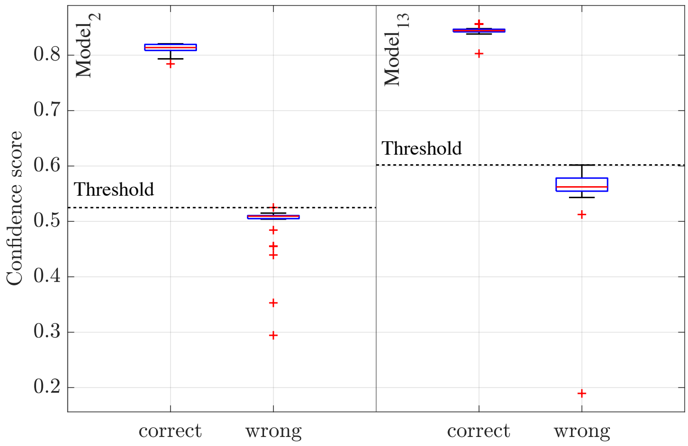

Fig. 4. Confidence score for correct and wrong classification for the M_2 and M_{13} models on test data. The confidence threshold (shown with dotted lines for the two cases) is set with an a-posteriori analysis of the confidence scores, based on the highest value that cause mis-classifications on the test data set.

to a more complex model and hence extract additional features. The level of confidence used to decide when to switch among models is identified by analysing a-posteriori the performance of the models on the test data.

When processing the validation dataset, each time the confidence level of the model is lower than the confidence threshold, INCLASS does not trust the result and switches to a more complex model. Figure 4 shows the a-posteriori confidence scores obtained by M_2 and M_{13} on the test data in case of correct and wrong seizures detection. The boxplot shows the data distribution for the two models and the dotted lines highlights the thresholds. For each model we have set the confidence threshold as the maximum confidence score obtained in case of mis-classification.

2) Automatically generated models: Section III-A describes how INCLASS, starting from an empty set of n features, automatically generates n different models by incrementally adding an additional feature to the starting one (Figure 3a). The importance and cost of the 17 classifiers generated by INCLASS for this work is plotted in Figure 5, left, and shown in the highlighted 17 large square points. In order to see where these generated classifiers stood with respect to the rest of the design space, and hence to see if the proposed strategy did indeed approximate the Pareto curve, we have subsequently plotted *a*ll the remaining 131055 points (small blue squares), i.e. the points that our methodology leaves unexplored. As can be seen, our chosen classifiers lay very close to the Pareto curve. For comparison, in Figure 5, right, we show an alternative heuristic in which features are ordered in terms of decreasing importance, as opposed to increasing cost. Such choice results in a poorer approximation of the Pareto front.

Section III-A also describes how INCLASS, after generating n classifiers, performs hierarchical clustering in order to reduce the number of models effectively used at run time (Figure 3b). Indeed, as can be observed in Figure 5, many of the 17 generated models are very close in terms of importance and cost, and if all the generated models were used, the INCLASS framework may end up switching to models which do not differ significantly from the already tried ones. As shown in

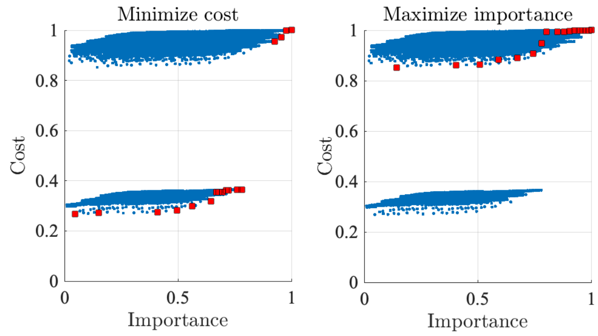

Fig. 5. Design space of all possible 2^{17} classifiers. Our strategy explores and generates only the 17 red square points, and leaves the remaining 131055 (small, blue) points unexplored. As it can be seen, the Pareto curve is well approximated by our strategy, which prioritizes features with low cost and is shown on the left. On the right, the 17 classifiers generated using a different priority (high importance) is also shown.

TABLE II
PERFORMANCE OF THE INCLASS FRAMEWORK ON THE TEST AND VALIDATION DATASETS VARYING THE NUMBER OF MODELS.

Test set			
# Models	GM(Sen,Spe)	Savings	Speedup
Baseline	91.8%	0%	1X
2	92.1%	47.23%	1.90X
3	93.4%	48.68%	1.95X
4	93.7%	33.69%	1.51X
5	94.0%	31.55%	1.47X
Validation set			
# Models	GM(Sen,Spe)	Savings	Speedup
Baseline	92.4%	0%	1X
2	92.4%	47.31%	1.90X
3	93.3%	48.72%	1.95X
4	93.3%	33.68%	1.51X
5	93.6%	31.75%	1.47X

Equation 1, INCLASS leads to benefit only if the inequality is satisfied. Switching many times among models causes the left term to grow rapidly, due to repeated execution of the detection phase, and therefore the inequality is unsatisfied soon while processing a window. As mentioned, to avoid this issue we perform hierarchical clustering on the identified models and we select, for each cluster, only the model having the highest importance score.

We have run a number of experiments addressing the use case of seizure detection to evaluate the performance of INCLASS when varying clustering granularity, and listed it in Table II. In our scenario, the configuration having three clusters, and hence switching among three models at run-time, maximise savings both on the test and validation sets. When using a higher number of clusters, savings instead decrease. This is due to the presence of many similar classifiers, causing INCLASS to perform a large number of detection and confidence evaluations, with the entailed overhead. We can also observe how the detection performance keeps increasing with the number of clusters, as more (and hence, more effective, when considered in conjunction) models are available, increasing the chance of a confident and correct classification.

3) INCLASS performance evaluation: Lastly, we have evaluated different strategies to set the confidence threshold, still considering multiple clustering alternatives. In particular, in

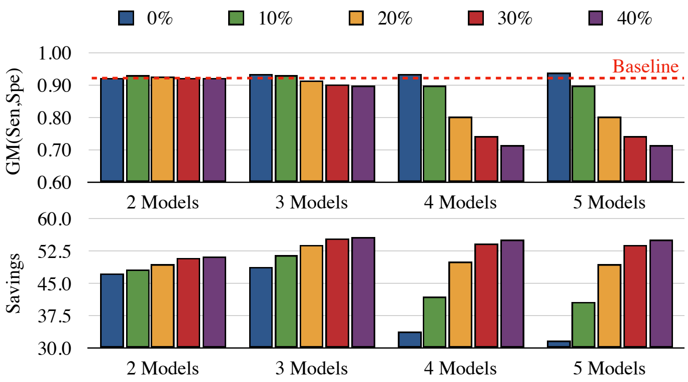

Fig. 6. Detection performance (GM) and savings obtained when decreasing confidence thresholds by 0%, 10%, 20%, 30% and 40%.

addition to setting the threshold to the highest value that causes mis-classifications as before, we also explored less stringent requirements, introducing tolerance levels of 10, 20, 30 and 40%.

Figure 6 shows the effect of applying these different tolerance values on the validation results. Lowering the thresholds required to be confident in classification outcomes allows to increase energy savings, as more classifications are accepted. Nonetheless, aggressive threshold settings impact detection performance, especially when using a large number of models, because detections are often performed by only evaluating very limited feature sets. In the 3 clusters scenario, using a 20% threshold allows to obtain a 54% saving with a detection performance loss of 1.1% with respect to the original baseline. Alternatively, using a 10% tolerance and 3 clusters allows to obtain a 51% workload saving with a detection performance increase of 0.6%.

VI. Conclusion

In this work, we have proposed an automated framework for the analysis of feature discriminant power and cost, allowing the generation of classification models with increasing complexity. These models can be used within the INCLASS self-assessment strategy to effectively decrease the workload of a seizure detection monitoring application. We showcased that our approach allows up to 54% workload savings with a negligible loss of 1.1% in the sensitivity and specificity of detections. We believe that such an approach can lead to the effective automation of self-aware applications design, including but not limited to the ones devoted to wearable health monitoring.

This work has been partially supported the ML-edge (Grant: 200020-182009) and ADApprox (Grant: 200020-188613) project funded by the Swiss NSF, and by the EC H2020 FVLLMONTI project (GA No. 101016776), and the ERC Consolidator grant COMPUSAPIEN (GA No. 725657).

References

[1] "World health organization, WHO (2019) Epilepsy." https://www.who.int/health-topics/epilepsy, accessed: 24-08-2021.

[2] T. Tomson, E. Beghi, A. Sundqvist, and S. I. Johannessen, "Medical risks in epilepsy: a review with focus on physical injuries, mortality, traffic accidents and their prevention," *Epilepsy research*, vol. 60, no. 1, pp. 1–16, 2004.

[3] D. J. Thurman, D. C. Hesdorffer, and J. A. French, "Sudden unexpected death in epilepsy: assessing the public health burden," *Epilepsia*, vol. 55, no. 10, pp. 1479–1485, 2014.

[4] J. van Andel, R. D. Thijs, A. de Weerd, J. Arends, and F. Leijten, "Non-eeg based ambulatory seizure detection designed for home use: what is available and how will it influence epilepsy care?" *Epilepsy & Behavior*, vol. 57, pp. 82–89, 2016.

[5] P. Ryvlin, C. Ciumas, I. Wisniewski, and S. Beniczky, "Wearable devices for sudden unexpected death in epilepsy prevention," *Epilepsia*, vol. 59, pp. 61–66, 2018.

[6] K. Vandecasteele, T. De Cooman, Y. Gu, E. Cleeren, K. Claes, W. V. Paesschen, S. V. Huffel, and B. Hunyadi, "Automated epileptic seizure detection based on wearable ecg and ppg in a hospital environment," *Sensors*, vol. 17, no. 10, p. 2338, 2017.

[7] C. Sevcencu and J. J. Struijk, "Autonomic alterations and cardiac changes in epilepsy," *Epilepsia*, vol. 51, no. 5, pp. 725–737, 2010.

[8] M. Zijlmans, D. Flanagan, and J. Gotman, "Heart rate changes and ecg abnormalities during epileptic seizures: prevalence and definition of an objective clinical sign," *Epilepsia*, vol. 43, no. 8, pp. 847–854, 2002.

[9] O. Devinsky, "Effects of seizures on autonomic and cardiovascular function," *Epilepsy currents*, vol. 4, no. 2, pp. 43–46, 2004.

[10] J. S. Preden, K. Tammemäe, A. Jantsch, M. Leier, A. Riid, and E. Calis, "The benefits of self-awareness and attention in fog and mist computing," *Computer*, vol. 48, no. 7, pp. 37–45, 2015.

[11] G. Becq, S. Bonnet, L. Minotti, M. Antonakios, R. Guillemaud, and P. Kahane, "Classification of epileptic motor manifestations using inertial and magnetic sensors," *Computers in biology and medicine*, vol. 41, no. 1, pp. 46–55, 2011.

[12] I. Conradsen, S. Beniczky, P. Wolf, T. W. Kjaer, T. Sams, and H. B. Sorensen, "Automatic multi-modal intelligent seizure acquisition (misa) system for detection of motor seizures from electromyographic data and motion data," *computer methods and programs in biomedicine*, vol. 107, no. 2, pp. 97–110, 2012.

[13] P. Jallon, "A bayesian approach for epileptic seizures detection with 3d accelerometers sensors," in *2010 Annual International Conference of the IEEE Engineering in Medicine and Biology*. IEEE, 2010, pp. 6325–6328.

[14] M. van Bussel, J. Arends, F. Massé, A. Serteyn, I. Tan, J. Penders, and P. Griep, "Detection of major epileptic seizures with heart rate changes: feasibility test of a state-of-the-art wearable sensor," in *conference; 9th ECE*, 2010, p. 56.

[15] F. Massé, R. R. Gonzenbach, A. Arami, A. Paraschiv-Ionescu, A. R. Luft, and K. Aminian, "Improving activity recognition using a wearable barometric pressure sensor in mobility-impaired stroke patients," *Journal of neuroengineering and rehabilitation*, vol. 12, no. 1, pp. 1–15, 2015.

[16] L. Ferretti, G. Ansaloni, L. Pozzi, A. Aminifar, D. Atienza, L. Cammoun, and P. Ryvlin, "Tailoring svm inference for resource-efficient ecg-based epilepsy monitors," in *2019 Design, Automation & Test in Europe Conference & Exhibition (DATE)*. IEEE, 2019, pp. 948–951.

[17] D. Cogan, M. Nourani, J. Harvey, and V. Nagaraddi, "Epileptic seizure detection using wristworn biosensors," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 5086–5089.

[18] F. Forooghifar, A. Aminifar, and D. A. Alonso, "Self-aware wearable systems in epileptic seizure detection," in *2018 21st Euromicro Conference on Digital System Design (DSD)*. IEEE, 2018, pp. 426–432.

[19] F. Forooghifar, A. Aminifar, L. Cammoun, I. Wisniewski, C. Ciumas, P. Ryvlin, and D. Atienza, "A self-aware epilepsy monitoring system for real-time epileptic seizure detection," *Mobile Networks and Applications*, pp. 1–14, 2019.

[20] F. K. Došilović, M. Brčić, and N. Hlupić, "Explainable artificial intelligence: A survey," in *2018 41st International convention on information and communication technology, electronics and microelectronics (MIPRO)*. IEEE, 2018, pp. 0210–0215.

[21] Y. Qi, "Random forest for bioinformatics," in *Ensemble machine learning*. Springer, 2012, pp. 307–323.

[22] "Classification trees in matlab," https://it.mathworks.com/help/stats/classificationtree-class.html, accessed: 14-09-2021.

[23] "Tree bag prediction," https://it.mathworks.com/help/stats/ compacttree-bagger.predict.html, accessed: 14-09-2021.

[24] J. Jeppesen, S. Beniczky, P. Johansen, P. Sidenius, and A. Fuglsang-Frederiksen, "Detection of epileptic seizures with a modified heart rate variability algorithm based on lorenz plot," *Seizure*, vol. 24, pp. 1–7, 2015.