Tailoring SVM Inference for Resource-Efficient ECG-Based Epilepsy Monitors

Lorenzo Ferretti,
Giovanni Ansaloni, Laura Pozzi
Università della Svizzera Italiana
Lugano, Switzerland

Amir Aminifar, David Atienza
Embedded Systems Laboratory
EPFL
Lausanne, Switzerland

Leila Cammoun, Philippe Ryvlin
Département des Neurosciences Cliniques
Centre Hospitalier Universitaire Vaudois
Lausanne, Switzerland

***Abstract*—Event detection and classification algorithms are resilient towards aggressive resource-aware optimisations. In this paper, we leverage this characteristic in the context of smart health monitoring systems. In more detail, we study the attainable benefits resulting from tailoring Support Vector Machine (SVM) inference engines devoted to the detection of epileptic seizures from ECG-derived features. We conceive and explore multiple optimisations, each effectively reducing resource budgets while minimally impacting classification performance. These strategies can be seamlessly combined, which results in 12.5X and 16X gains in energy and area, respectively, with a negligible loss, 3.2% in classification performance.**

***Index Terms*—Wireless Body Sensor Nodes, Seizure detection, Ultra-low-power design, Algorithmic optimisation.**

I. Introduction

In this last years, smart health monitors (Wireless Body Sensor Nodes or WBSNs [1]) have been proposed that, instead of only sampling and wirelessly transmitting data, are also capable of autonomous interpretation of acquisitions, in order to derive features of clinical relevance [2]. When performing a long-term health assessment, features are usually processed remotely. However, when dynamically detecting acute episodes, the opportunity arises to perform both feature extraction and event detection (*inference*) on this wearable devices (see Figure 1). On-device event detection enables a greater degree of autonomy, as life-threatening conditions are identified locally. Moreover, it has the potential to significantly increases energy efficiency, because only the detection outcome has to be transmitted on the wireless link [3].

Support Vector Machines (SVMs) are particularly well suited for the identification of health-threatening episodes, as they are able to cope with noise in acquisitions and to expose non-obvious relationships between features and events, defining characteristics in WBSN scenarios. Of particular relevance to our contribution is the high performance demonstrated by SVMs when detecting of epileptic seizures based on electrocardiogram (ECG) acquisitions, either employing Heart Rate Variability (HRV) [4] and ones derived from Lorentz plots [5]. Recently, the authors of [6] reported detection rates exceeding 80% by considering features extracted from HRV, Lorentz plots, and parameters derived from auto-regressive analysis and from the computation of the spectral density in various bands.

These studies suggest that the accuracy of ECG-based seizure detection does benefit from rich models. Nonetheless, their processing requirements may impose workloads beyond the capabilities of energy-constrained WBSN platforms. We therefore herein take a complementary stance, exploring instead the achievable efficiency gains that result from simplifying the inference implementation. We attain energy reductions of up to 12.5X with respect to the strategy in [6], with only a 3.2% quality degradation.

We base our analysis on ECG data collected from a cohort of patients with refractory epilepsy, whose seizures were recorded in an epilepsy monitoring unit. The used dataset corresponds to 7 patients with 140 hours of recordings and including 34 focal epileptic seizures, annotated by medical experts based on video and electroencephalography signals.

II. SVM inference analysis

A trained SVM identifies a surface in an N_{feat}-dimensional space that maximally separates the data points of a training set that belong to two different classes. On the one hand, the training set is composed of feature vectors $\mathbf{x}_i = [x_{i,0}, x_{i,1}, ..., x_{i,N_{feat}-1}]$, in which each vector element records the value of a feature extracted from three-minutes signal windows. On the other hand, the vectors $\mathbf{F}_j = [x_{0,j}, x_{1,j}, ..., x_{N_{test},j}]$ record all values assigned to feature j across windows. Labels $y_0, ..., y_{N_{test}}$, with $y_j \in \{-1, +1\}$, indicate the class of the corresponding training vector ($+1$ for ECG excerpts during seizures, -1 otherwise).

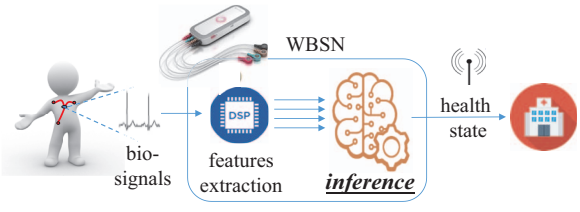


Fig. 1: ECG-based seizure detection on Wireless Body Sensor Nodes, which relies on a feature extraction stage followed by an event detection (inference) stage. This paper focuses on minimising the resource requirements of the latter stage.

This work has been partially supported MagicISEs (grant no. 200021-156397) project funded by the Swiss NSF, the MyPreHealth (grant no. 16073) project funded by Hasler Stiftung, and by the Human Brain Project (HBP) SGA2 (GA No. 785907).

The separating surface is expressed as a function of a subset of the training samples, termed Support Vectors (SVs), which allows to assign a label Y to new samples according to the following formula:

$$Y = sgn(\sum_{i \in SVs} (\alpha_i y_i \ k(\mathbf{x}_T, \mathbf{x}_i)) + b) \quad (1)$$

where $\mathbf{x}_T$ is the feature vector of the sample under test, $\mathbf{x}_i$ are the SVs, $\alpha_i \in (0, 1]$ are weights assigned to each SV during the training phase, y_i are the SVs labels, b is a scalar bias term, and a kernel function $k()$ determines the complexity of the separating surface.

The performance of different kernel functions are compared in Table I, adopting, as figures of merit, Sensitivity (Se), Specificity (Sp) and their Geometric Mean GM, defined as follows [7]:

$$Se = \frac{TP}{TP+FN} \quad Sp = \frac{TN}{TN+FP} \quad GM = \sqrt{(Se \times Sp)} \quad (2)$$

where TP is the number of true positives, TN the number of true negatives, FP and FN the number of false positives and false negatives, respectively. GM is high only if a high number of both seizure *and* non-seizure ECG excerpts are correctly classified, and is used as a measure of classification performance throughout the paper. Reported results refer to the average Se, Sp and GM over 24 folds, where for each fold the ECG windows originating from a recording session are used as the test set and all others as the training set.

Table I shows that, when applied to our target scenario, Gaussian and (especially) polynomial SVMs are much better than the linear model, with quadratic and cubic kernels having a similar average GM. We therefore focus the exploration presented in the rest of the paper on the simpler of these formulations, i.e., the quadratic SVM, whose inference formula is:

$$Y = sgn(\sum_{i \in SVs} (\alpha_i y_i \ (\mathbf{x}_T \cdot \mathbf{x}_i + 1)^2) + b) \quad (3)$$

The workload entailed by Equation 3 can be reduced by decreasing a) the number of features in each feature vector $\mathbf{x}_i$, b) the number of support vectors SVs and c) the number of bits to represent the feature values $x_{i,j}$ and the $\alpha_i y_i$ parameters.

While related works have also explored resource-constrained SVMs (mainly in the context of image processing applications), they usually deal with only one of the above-mentioned dimensions, such as the selection of the most representative features [8] [9], the reduction of the cardinality of the SV set [10] [11] or limited-bitwidths data representations [12] [13]. In the following, we instead explore the effectiveness of each of these strategies, as well as the attainable efficiency when all are synergistically employed.

TABLE I: Classification performance of different floating point SVM implementations.

SVM Kernel	Sp %	Se %	GM
Linear	75.6	82.3	72.9
Quadratic	92.3	86.6	86.8
Cubic	95.3	86.6	88.0
Gaussian	97.0	79.6	82.6

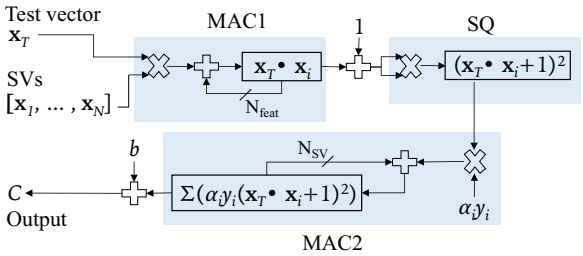


Fig. 2: Inference pipeline block scheme.

We explore the efficiency gains in terms of the area resources and energy footprint required by a hardware accelerator performing inference on a test sample. The implementation of such accelerator follows the scheme in Figure 2. At its input, it embeds an internal memory to store the SVs required by the SVM. The dot products $(\mathbf{x}_T \cdot \mathbf{x}_i)$ in Equation 3 are performed by a first multiplier-accumulator (MAC) unit, and results are then squared to compute the kernel function. A further MAC block accumulates the output of the kernel functions for each SV, multiplied by the $\alpha_i y_i$ coefficients. The accelerator output, and therefore the computed class of the test sample, is then the sign (i.e., most significant bit) of this second accumulator, taking into account the bias parameter b. Faster and more resource-hungry choices are possible, e.g., by computing multiple kernel functions in parallel, which would also benefit from the proposed optimisations with similar efficiency gains. Area and energy values of different configurations were retrieved via hardware synthesis targeting a 40nm technology.

III. Approximation techniques

Reducing the features set. The inference dimensionality is related to the size of the memory required to store the SVs, as well as to the number of multiply-accumulate operations necessary for the computation of the dot product operation embedded in the kernel function (the MAC1 block in Figure 2), and therefore its energy cost.

The set employed in [6], which we consider as a starting point for the exploration, is composed by 53 features. Features 1-8 are derived from an analysis of the heart rate while features 9-15 are obtained from Lorentz plots. The third and fourth feature categories are computed from ECG-Derived Respiration (EDR) time series, either from the linear coefficients of their auto-regressive (AR) model (features 16-24) or from their power spectral analysis (PSD) (features 25-53).

Similarly to [8], to reduce this set we analyse their correlations and iteratively remove the ones that are highly correlated to others. To this end, we first compute pairwise Pearson coefficients ρ between features:

$$\rho(\mathbf{F}_{j1}, \mathbf{F}_{j2}) = \frac{cov(\mathbf{F}_{j1}, \mathbf{F}_{j2})}{\sigma(\mathbf{F}_{j1}), \sigma(\mathbf{F}_{j2})} \quad (4)$$

where cov is the covariance between the features $j1$ and $j2$, and σ their standard deviation. Higher ρ values indicate a stronger correlation. Figure 3 shows the correlation matrix

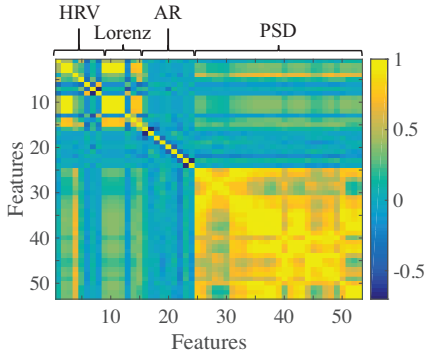


Fig. 3: Correlation coefficient matrix for the baseline feature set, comprising 53 features.

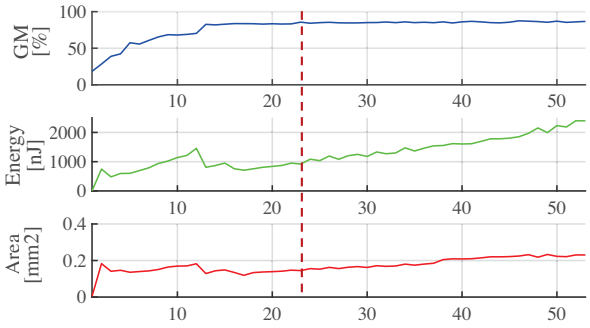


Fig. 4: Classification performance and resource requirements, when varying the number of features.

computed for all feature tuples, indicating that most PSD, some HRV and some Lorenz features have high mutual Pearson coefficients, and therefore encode information redundantly.

In a second step, we sum column-wise the coefficients in the correlation matrix, identifying the feature having the highest aggregated Pearson coefficient. By iterating on those two phases, increasingly reduced feature sets are identified, for which we trained different SVM implementations and we synthesised the corresponding pipelines.

The experimental results are shown in Figure 4, both in terms of geometric mean of sensitivity and specificity (GM, top), as well as energy for the classification of a test vector (middle) and area of the inference hardware (bottom). We considered a 64-bit implementation, which has the same accuracy as an equivalent floating point version. Counter-intuitively, between 15 and 8 features we measured an increase in resource requirements as we decreased the number of features, because more SVs were selected during training. More importantly, we observed that GM values slowly worsen for set sizes greater than 15 features, and drop significantly for smaller ones. Then, by selecting only 23 features (a design choice highlighted with the dashed line in Figure 4), energy and area costs are reduced by 65% and 42%, respectively, with a marginal classification performance decrease of 1.2% in terms of GM. This feature set comprises six HRV features, nine from the auto-regressive model, four extracted from Lorentz plots and four from the PSD analysis.

Reducing the number of support vectors. The number of support vectors tends to grow linearly with the size of the

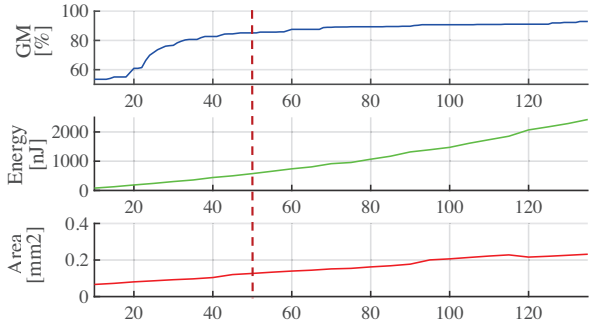


Fig. 5: Classification performance and resource requirements, when varying the SV set size.

training set. This effect (known as "curse of kernelization" [10]) may lead to an over-design of the local memory storing the vectors, or to an inflexible implementations, unable to exploit updates of the training data. To counter it, we adopt a strategy firstly introduced in [10], which imposes a bound on the number of SVs. It implements a budgeting approach through the iterative removal of the least significant SV from the training set according to the following norm:

$$\|SV_i\| = \|\alpha_i\|^2 \times k(\mathbf{x}_i, \mathbf{x}_i) \quad (5)$$

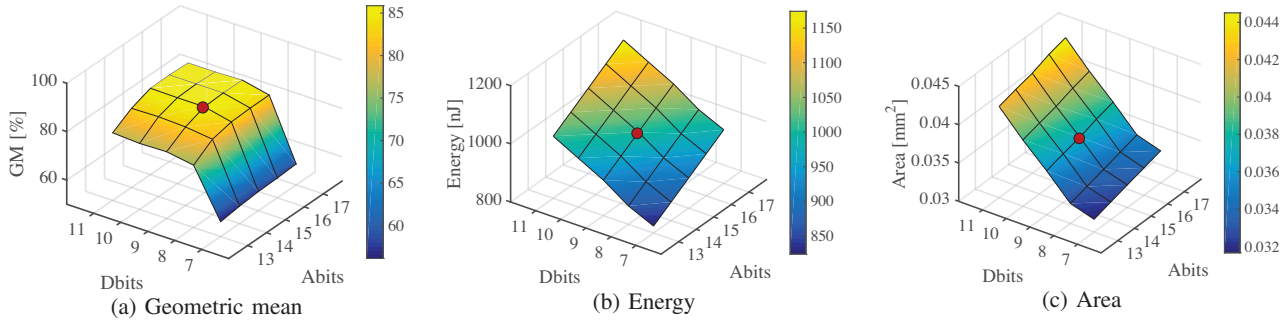
The reduced test set is then employed to re-train the SVM parameters. The result of the ensuing exploration, under different SV budget thresholds, is shown in Figure 5. The area benefits of small SV budgets derive from requiring a smaller local memory, which also reduce its static (leakage) and dynamic (energy-per-access) energy consumption. A further factor impacting energy efficiency is the reduced workload required by a smaller SV cardinality. Classification performance is instead only marginally affected by the removal of low-norm support vectors up until only around 50 elements are present, and sharply worsens after that. At this design point, the GM is 1.5% less with respect to the un-budgeted case, with an energy reduction of 76% and an area reduction of 45%.

Reducing bitwidths. A further avenue to decrease resource requirements is to limit the range and precision to represent features, parameters, and intermediate values. In this way, the size of the SVs local memory, as well as the width of arithmetic operators, can be tailored, saving both area and energy. To this end, we first discard the least significant bits at the output of the kernel computation and the square operator. Then, we quantise the $\alpha_i y_i$ values, which are bounded by construction between 1 and -1. Finally, we limit the maximum magnitude of the features, and express them with limited precision.

In this last regard, we only consider feature values ranges in the form $[-2^{R_j}, 2^{R_j}], j = 1..N_{feat}$, for which up- and down-scaling can be implemented efficiently with shift operations instead of dividers. The R_j parameter related to feature j is the smallest values that verifies the inequalities

$$avg(\mathbf{F_j}) - \sigma(\mathbf{F_j}) > -2^{R_j}; \quad avg(\mathbf{F_j}) + \sigma(\mathbf{F_j}) < 2^{R_j} - 1 \quad (6)$$

where $avg(\mathbf{F_j})$ is the average of the values assigned to feature j in the SV set and $\sigma(\mathbf{F_j})$ its standard deviation. If a feature value (both in the SVs and in the test vector) exceeds its



(a) Geometric mean (b) Energy (c) Area

Fig. 6: Performance/requirements exploration varying the width of the data ($Dbits$) and the parameters ($Abits$) representations.

range, it is saturated to the admissible maximum / minimum. Precision reduction is then performed by only considering the bits in the interval $[R_j - 1; R_j - Dbits]$.

While this strategy mandates a scale-back operation during the kernel computation and a dedicated memory to store the scale factors, the resulting area and energy overheads are dwarfed by the gains ensuing from adopting small bitwidths, as showcased in Figure 6. Across all experiments illustrated therein, the least significant ten bits are discarded both after the dot product and after the square operations, with no impact on classification performance. The point marked with a red circle corresponds to employing 9 bits to represent features and 15 bits for the coefficients, which exhibits negligible GM loss of 1% compared to a floating point implementation. When instead the same bitwidth is considered throughout the pipeline, and the same scaling factor is employed among features and among $\alpha_i y_i$, 64 bits are required to reach the same GM, resulting in a design having 2.4X more energy and 6.2X more area.

Combining approximation techniques. Even higher efficiency gains are attained when all three strategies are performed in sequence. The performance of the resulting pipelines, at subsequent optimisation stages, are illustrated in the left part of Figure 7, considering a) a reduction in the number of features from 53 to 30, b) restricting the size of the support vector set to 68 vectors, c) the use of 9 bits for representing features and of 15 bits to encode $\alpha_i y_i$ values.

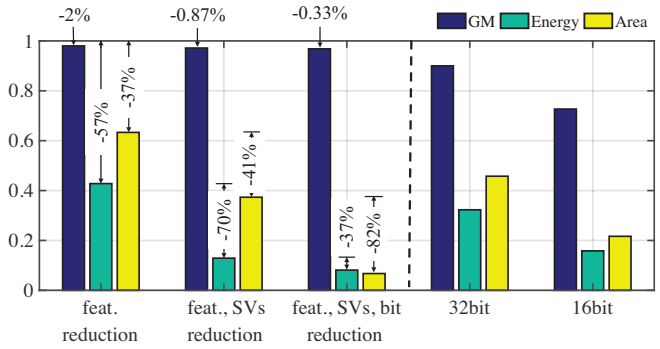


Fig. 7: Left: classification performance and required resources of inference pipelines, when each optimisation is applied in sequence. Above each bar, percentages indicate differences with respect to the previous optimisation step. Right: performance of 32-bits and 16-bits pipelines. All data is normalised with respect to a 64-bits implementation.

Overall, efficiency gains of 12.5X and 16X are attained, in terms of energy and area footprints, for a GM loss of less then 3.2%. Figure 7 also reports, for comparison, the performance achievable by a more limited strategy where, as the only optimisation, two parameters are adopted to homogeneously scale features and parameters, respectively. The resulting implementations are clearly sub-optimal, with the 32-bit pipeline demanding 7X more area and 4X more energy, while also having a 7% lower GM with respect to our fully optimised design.

IV. Conclusion

Health monitoring applications can provide life-saving detections of acute episodes. Nonetheless, their implementation in resource-constrained WBSNs mandates a careful tailoring of the ensuing workloads. Against this backdrop, and targeting the identification of epileptic seizure from ECG features, we have shown that the efficiency of SVM inference can be increased by on order of magnitude through resource-aware optimisations.

References

[1] Y. Hao et al., "Wireless body sensor networks for health-monitoring applications," *Physiological measurement*, vol. 29, no. 11, 2008.

[2] R. Braojos et al., "Ultra-low power design of wearable cardiac monitoring systems," *ACM DAC*, 2014.

[3] S. S. Basu et al., "An inexact ultra-low power bio-signal processing architecture with lightweight error recovery," *IEEE CODES+ISSS*, 2017.

[4] K. Vandecasteele et al., "Automated epileptic seizure detection based on wearable ECG and PPG in a hospital environment," *MDPI Sensors*, vol. 17, no. 10, 2017.

[5] J. Pavei et al, "Early seizure detection based on cardiac autonomic regulation dynamics," *Frontiers in physiology*, vol. 8, 2017.

[6] F. Forooghifar et al., "Self-aware wearable systems in epileptic seizure detection," *Euromicro DSD*, 2018.

[7] P. J. Fleming et al., "How not to lie with statistics: the correct way to summarize benchmark results," *ACM Communications*, 1986.

[8] L. Yu et al., "Efficient feature selection via analysis of relevance and redundancy," *JMLR*, vol. 5, 2004.

[9] G. Roffo et al., "Infinite latent feature selection: A probabilistic latent graph-based ranking approach," *IEEE CVPR*, 2017.

[10] Z. Wang et al., "Breaking the curse of kernelization: Budgeted stochastic gradient descent for large-scale SVM training," *JMLR*, vol. 13, 2012.

[11] T. Le et al., "Budgeted semi-supervised support vector machine," *UAI*, 2016.

[12] B. Lesser et al., "Effects of reduced precision on floating-point SVM classification accuracy," *Elsevier PCS*, vol. 4, 2011.

[13] D. Anguita et al., "Energy efficient smartphone-based activity recognition using fixed-point arithmetic," *J.UCS*, vol. 19, no. 9, May 2013.

[14] S. J. E. Wilton et al., "CACTI: An enhanced cache access and cycle time model," *IEEE JSSC*, vol. 31, no. 5, 1996.