
Human Activity Recognition on Smartphones for Mobile Context Awareness

Davide Anguita, Alessandro Ghio, Luca Oneto,

DITEN Department, University of Genova
Via Opera Pia 11A, I-16145 Genova, Italy

{Davide.Anguita,Alessandro.Ghio,Luca.Oneto} @unige.it

Xavier Parra

CETpD - Universitat Politècnica de Catalunya
Vilanova i la Geltrú 08800, Spain
xavier.parra@upc.edu

Jorge L. Reyes-Ortiz*

CETpD - Universitat Politècnica de Catalunya
Vilanova i la Geltrú 08800, Spain
and

DITEN Department, University of Genova
Via Opera Pia 11A, I-16145 Genova, Italy

jorge.luis.reyes@estudiant.upc.edu

Abstract

Activity-Based Computing [1] aims to capture the state of the user and its environment by exploiting heterogeneous sensors in order to provide adaptation to exogenous computing resources. When these sensors are attached to the subject's body, they permit continuous monitoring of numerous physiological signals. This has appealing use in healthcare applications, e.g. the exploitation of Ambient Intelligence (AmI) in daily activity monitoring for elderly people. In this paper, we present a system for human physical Activity Recognition (AR) using smartphone inertial sensors. As these mobile phones are limited in terms of energy and computing power, we propose a novel hardware-friendly approach for multiclass classification. This method adapts the standard Support Vector Machine (SVM) and exploits fixed-point arithmetic. In addition to the clear computational advantages of fixed-point arithmetic, it is easy to show the regularization effect of the number of bits and then the connections with the Statistical Learning Theory. A comparison with the traditional SVM shows a significant improvement in terms of computational costs while maintaining similar accuracy, which can contribute to develop more sustainable systems for AmI.

1 Introduction

Since the appearance of the first commercial hand-held mobile phones in 1979, it has been observed an accelerated growth in the mobile phone market which has reached by 2011 near 80% of the world population [2]. This shows that in a very short time, mobile devices will become easily accessible to

*This work was supported in part by the Erasmus Mundus Joint Doctorate in Interactive and Cognitive Environments, which is funded by the EACEA Agency of the European Commission under EMJD ICE FPA n 2010-0012.

virtually everybody. Smartphones, which are a new generation of mobile phones, are now offering many other features such as multitasking and the deployment of a variety of sensors, in addition to the basic telephony. Current efforts attempt to incorporate all these features while maintaining similar battery lifespans and device dimensions.

The integration of these mobile devices in our daily life is rapidly growing. It is envisioned that such devices will seamlessly keep track of our activities, learn from them, and subsequently help us to make better decisions regarding our future actions [3]. This is one key concepts in which AmI relies on. In this paper, we employ smartphones for human Activity Recognition with potential applications in assisted living technologies. We take into account current hardware limitations and propose a new alternative for AR that requires less computational resources to operate since it is based on just fixed-point arithmetic. Less computational resources means not only speed but also less power consumption and this is a key issue in smartphones applications [4]. Adopting a fixed point arithmetic has also a strong connection with the problem of regularization and Statistical Learning Theory (SLT) [5] as described in [6] and [7].

AR aims to identify the actions carried out by a person given a set of observations of itself and the surrounding environment. Recognition can be accomplished, for example, by exploiting the information retrieved from inertial sensors such as accelerometers [8]. In some smartphones these sensors are embedded by default and we benefit from this to classify a set of physical activities (*standing, walking, laying, walking, walking upstairs and walking downstairs*) by processing inertial body signals through a supervised Machine Learning (ML) algorithm for hardware with limited resources.

This paper is structured in the following way: The state of the art regarding previous work is depicted in Section 2. The description of the adopted methodology is presented in Section 3. There, the experimental set up for capturing the data, the mathematical description for the proposed Multiclass Hardware Friendly Support Vector Machine (MC-HF-SVM) and the connection with the SLT are explained. Experimental results and conclusions of this research are described in Sections 4 and 5.

2 Related Work

The development of AR applications using smartphones has several advantages such as easy device portability without the need for additional fixed equipment, and comfort to the user due to the unobtrusive sensing. This contrasts with other established AR approaches which use specific purpose hardware devices such as in [9] or sensor body networks [10]. Although the use of numerous sensors could improve the performance of a recognition algorithm, it is unrealistic to expect that the general public will use them in their daily activities because of the difficulty and the time required to wear them. One drawback of the smartphone-based approach is that energy and services on the mobile phone are shared with other applications and this become critical in devices with limited resources.

ML methods that have been previously employed for recognition include Naive Bayes, SVMs, Threshold-based and Markov chains [10]. In particular, we make use of SVMs for classification as it was also used in [11] and [12]. Although it is not fully clear which method performs better for AR, SVMs have confirmed successful application in several areas including heterogeneous types of recognition such as handwritten characters [13] and speech [14].

In ML, fixed-point arithmetic models have been previously studied [15, 16] initially because devices with floating-point units were unavailable or expensive. The possibility of retaking these approaches for AmI systems that require either low cost devices or to allow load reduction in multitasking mobile devices has nowadays become particularly appealing. Anguita et al. in [17] introduced the concept of a Hardware-Friendly SVM (HF-SVM). This method exploits fixed-point arithmetic in the feed-forward phase of the SVM classifier, so as to allow the use of this algorithm in hardware-limited devices. In this paper, we extend this model for multiclass classification and we show, both in theory and with the experimental results, how the number of bits have a strong regularization effect.

The SVM algorithm was originally proposed only for binary classification problems but it has been adapted using different schemes for multiclass problems such as in [13]. In particular, we have chosen the One-Vs-All (OVA) method as its accuracy is comparable to other classification methods as demonstrated by Rifkin and Klautau in [18], and because its learned model uses less memory

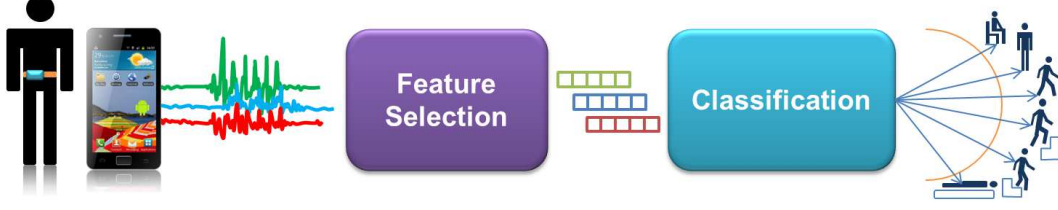


Figure 1: Activity Recognition process pipeline.

when compared for instance to the One-Vs-One (OVO) method. This is advantageous when used in limited resources hardware devices.

3 Methodology

3.1 Experimental Setup

The experiments have been carried out with a group of 30 volunteers within an age bracket of 19-48 years. Each person performed the six activities previously mentioned wearing the smartphone on the waist. The experiments have been video-recorded to facilitate the data labeling. The obtained database has been randomly partitioned into two sets, where 70% of the patterns has been used for training purposes and 30% as test data: the training set is then used to train a multiclass SVM classifier which is described in the following section. A Samsung Galaxy S2 smartphone has been exploited for the experiments, as it contains an accelerometer and a gyroscope for measuring 3-axial linear acceleration and angular velocity respectively at a constant rate of 50Hz, which is sufficient for capturing human body motion¹.

For AR purposes, we have developed a smartphone application based on the Google Android Operating System. The recognition process starts with the acquisition of the sensor signals, which are subsequently pre-processed by applying noise filters and then sampled in fixed-width sliding windows of 2.56 sec and 50% overlap. From each window, a vector of 17 features is obtained by calculating variables from the accelerometer signals in the time and frequency domain (e.g. mean, standard deviation, signal magnitude area, entropy, signal-pair correlation, etc.). Fast Fourier Transform is used for finding the signal frequency components. Finally, these patterns are used as input of the trained SVM Classifier for the recognition of the activities. The entire AR process pipeline is as shown in Figure 1.

3.2 The Multiclass HF-SVM model

Consider a dataset consisting of l patterns where each one is a pair of the type $(\mathbf{x}_i, y_i) \forall i \in \{1, \dots, l\}$, $\mathbf{x}_i \in \mathbb{R}^m$, and $y_i = \pm 1$. A standard binary SVM can be learned by solving a Convex Constrained Quadratic Programming (CCQP) minimization problem which is given by the following formulation [17]:

$$\min_{\alpha} \quad \frac{1}{2} \alpha^T Q \alpha - \mathbf{r}^T \alpha \quad (1)$$

$$0 \leq \alpha_i \leq C \quad \forall i \in \{1, \dots, l\}, \quad (2)$$

$$\mathbf{y}^T \alpha = 0, \quad (3)$$

where C is the regularization parameter, $r_i = 1 \quad \forall i \in \{1, \dots, l\}$ and Q is the symmetric positive semidefinite $l \times l$ kernel matrix where $q_{ij} = y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$.

¹The dataset obtained in this session of experiments will be soon available on the site <http://www.smartlab.ws/> with a detailed description of the data that have been collected.

After solving this CCQP problem, the $\alpha_i \forall i \in \{1, \dots, l\}$ values can be found and used to predict the class of any new pattern using the Feed-Forward Phase (FFP) formulation of the SVM:

$$f(\mathbf{x}) = \sum_{i=1}^l y_i \alpha_i K(\mathbf{x}_i, \mathbf{x}) + b, \quad (4)$$

where b is the bias term and is obtained by using the method proposed in [19].

Clearly, this output is not valid for use in a fixed-point arithmetic approach as these α_i values are intrinsically real numbers ranging between zero and C . Hence a normalization procedure is proposed that will not affect the sign of the classifier output but only its magnitude, maintaining the performance of the SVM as it is known that the class is only determined by the feed-forward function sign. The HF-SVM described in [17] proposes a new vector β and it is defined as:

$$\beta_i = \alpha_i \frac{2^k - 1}{C} \quad \forall i \in \{1, \dots, l\}, \quad (5)$$

where k is the number of bits and $\beta_i \in \mathbb{N}^0$. Also the bias term b of the FFP formulation is removed since we use an RBF kernel such as the Laplacian that has infinite VC dimension and the bias b becomes unnecessary [5]. The modified formulation is:

$$\min_{\beta} \quad \frac{1}{2} \beta^T Q \beta - s^T \beta \quad (6)$$

$$0 \leq \beta_i \leq \frac{2^k - 1}{C} \quad \forall i \in \{1, \dots, l\}, \quad (7)$$

where $s_i = (2^k - 1) / C \quad \forall i \in \{1, \dots, l\}$. Note that the cost function keeps unchanged but Eq. (3) disappears.

Lastly, to hold true the assumption of having a FFP with only integer values, the kernel $K(\cdot, \cdot)$ and the input vector \mathbf{x} are also represented with u and v bits respectively [17]:

$$0 \leq K(\mathbf{x}_i, \mathbf{x}) \leq 1 - 2^{-u} \quad \forall i \in \{1, \dots, l\}, \quad (8)$$

$$0 \leq x_{i,j} \leq 1 - 2^{-v} \quad \forall i \in \{1, \dots, l\} \quad \forall j \in \{1, \dots, m\}. \quad (9)$$

The modified FFP formulation with the β vector is:

$$f(\mathbf{x}) = \sum_{i=1}^l y_i \beta_i K(\mathbf{x}_i, \mathbf{x}). \quad (10)$$

We opted for a Laplacian kernel, instead of the more conventional Gaussian kernel, as it is more convenient for hardware limited devices because it can be easily computed using shifters:

$$K(\mathbf{x}_i, \mathbf{x}_j) = 2^{-\gamma \|\mathbf{x}_i - \mathbf{x}_j\|_1}, \quad (11)$$

where $\gamma > 0$ is the kernel hyperparameter and the norm is expressed as $\|\mathbf{x}\|_1 = \sum_{i=1}^m |x_i|$.

The complete learning process for each SVM consists of performing grid search model selection of the C and γ hyperparameters that converge with the minimum validation error. A k -fold cross validation with $k = 10$ is employed for each hyperparameter pair.

The output of the FFP varies depending on each learned SVM model as these are not normalized. Our extension of this binary problem for the multiclass case employs the OVA method in which each class c is compared against the other classes. This evidently requires a method to allow comparability between the output of each SVM. For this reason, we have opted to compute probability estimates for each SVM $p_c(\mathbf{x})$ and choose the one with the highest probability as the actual class c^* of each test pattern.

We have developed the following approach using the J.Platt's method for estimating probability estimates [20]. The training dataset and the learned SVM model are employed to fit the output of the FFP $f(\mathbf{x})$ with a sigmoid function of the form:

$$p(\mathbf{x}) = \frac{1}{1 + e^{(Af(\mathbf{x}) + B)}}, \quad (12)$$

in which $p(\mathbf{x})$ is the probability estimate, and A and B are parameters which are properly fitted on the available learning samples.

Taking into account that we have the fixed-point arithmetic restriction, the sigmoid function cannot be directly applied on $f(\mathbf{x})$. To solve this, we have designed a method based on Look-Up-Tables (LUTs). By defining a fixed number of bits t , it is possible to map the probability estimates $p(\mathbf{x})$ given $f(\mathbf{x})$ without requiring floating-point arithmetic. It has been observed that $t = 8$ is suitable for this application and it only requires a LUT with 256 elements.

3.3 HF-SVM and Statistical Learning Theory

In this section we try to investigate how the adoption of a fixed-point arithmetic affects the generalization ability of a classifier in the form of Eq. (10). In order to do this we describe each parameter β_i as an integer value of k bits:

$$\beta_i = \sum_{j=1}^{k-1} b_i^j 2^j, \quad (13)$$

where b_i^j is a binary valued variable $b_i^j \in \{0, 1\}$ and therefore β_i can be expressed as an integer variable such that $0 \leq \beta_i \leq 2^k - 1$. Since each b_i^j belongs to a finite set, for a fixed training set of cardinality l and a fixed kernel (with its hyperparameter), the number of classifiers that we can represent is finite. According to the notation of [5] we call N_f^l the number of classifiers that we can built with b_i^j , $i \in \{1, \dots, l\}$ and $j \in \{0, \dots, k-1\}$. Consequently we can exploit the well known Vapnik's generalization bounds for finite hypothesis sets [5] which uses N_f^l as measure of complexity. Let then d^β the number of nonzero parameters, $\beta_i \neq 0$, then:

$$N_f^l(k, d^\beta) \leq \sum_{i=1}^{d^\beta} \binom{l}{d^\beta} \left[(2^k - 1)^{d^\beta} - (2^{k-1} - 1)^{d^\beta} \right], \quad (14)$$

where we take into account the fact that if all the parameters are even numbers, they can be divided by two without changing the class estimate. If, instead, d^b is the number of nonzero parameters, $b_i^j \neq 0$, then

$$N_f^l(k, d^b) \leq \sum_{i=1}^{d^b} \binom{l}{d^b} \binom{k}{d^b}. \quad (15)$$

In the Statistical Learning Theory framework and in particular in the Structural Risk Minimization framework [5] we have to define a nested structure of the hypothesis sets ($\mathcal{H}_1 \subseteq \mathcal{H}_2 \subseteq \dots$) with increasing complexity before seeing the data. Then the generalization capability of a model can be controlled by choosing the appropriate set by finding the best compromise between complexity and learning error. In this way a good generalization capability on previously unseen data can be guaranteed [5, 21].

In our case the complexity of the class can be defined through two quantities, k and d^β (or d^b). Starting from the set \mathcal{H}_1 with complexity $N_f^l(1, 1)$ we can increase the complexity by increasing the number of bits $k \rightarrow k + 1$ or by decreasing the sparsity of the representation $d^\beta, d^b \rightarrow d^\beta, d^b + 1$. In other words we have to search the best class which is more sparse as possible (smaller d^β or d^b) and represented with the minimum number of bits k .

Obviously a classifier that belongs to a space with smaller complexity is also more energy efficient respect to the one that belongs to a space with higher complexity.

Increasing the complexity of the space has also direct consequence on the generalization ability of the classifier since according to the bound of Vapnik [5], which holds with probability $(1 - \delta)$:

$$\pi \leq \nu + \sqrt{\frac{\ln [N_f^l(k, d)] - \ln(\delta)}{2l}} \quad (16)$$

where π is the generalization error and ν is the error obtained by the learning machine on the dataset.

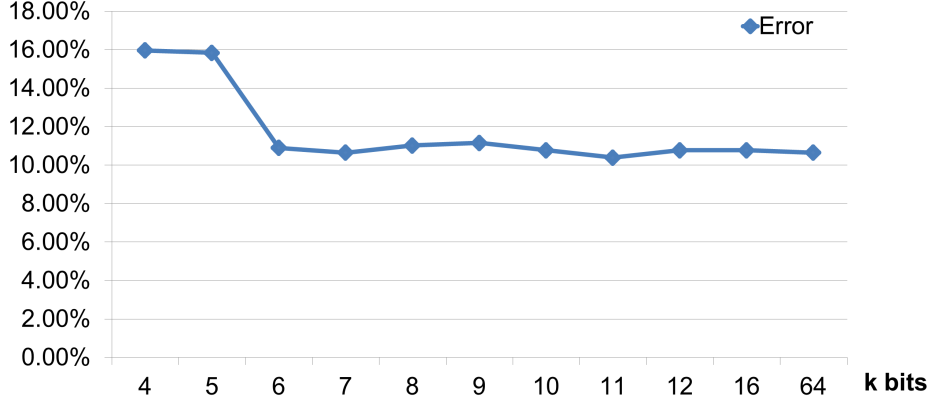


Figure 2: Comparison between the MC-SVM and the MC-HF-SVM: Classification test error for the MC-HF-SVM with different values of k against the MC-SVM which is represented with $k = 64$ bits.

The result is similar to what is presented in [7] and [6]. The important results of this section is that the number of bits in the HF-SVM have a strong regularization affect with an impact on the generalization ability on a classifier. Between two classifier with approximately the same performance we have to choose the one that can be represented whit less number of bits since it is more energy efficient and it has more capacity of performing well on previously unseen data.

Finally we want to point out that the bound of Eq. (16) is very loose since it is data independent. Data dependent bounds have been developed in the last ten years [22, 23] that give more tight bounds on the generalization ability of the classifier and have shown to perform well on real world problems [21, 24]. For these reasons a very interesting topic of research will be to understand how the fixed-point arithmetic can influence the estimation of these bounds.

4 Experimental results

For evaluating the performance of the MC-HF-SVM, a set of experiments were carried out using the AR dataset described in this paper. They consisted of learning SVM models with different number of bits k for β estimation and then comparing their performance in terms of test data error against the standard floating-point Multiclass SVM (MC-SVM). The results of this comparison are depicted in Figure 2.

The experiment shows that for this dataset $k = 6$ bits are sufficient for achieving a performance comparable with the MC-SVM approach that uses 64-bit floating-point arithmetic. The test error remains stable (around 1% variation) for k values from 64 to 6 bits, but it increases noticeably to 15% when it reaches 5 bits. Moreover, it is also seen from the graph that some values of k produced smaller errors than the one obtained with the MC-SVM. This finding coincides with the discussion of Section 3.3: smaller number of bits can increase the generalization ability of the classifier. Then the truncation of the model parameters produces a regularization effect.

The classification results of the MC-SVM and the MC-HF-SVM with $k = 8$ bits² for the test data are depicted by means of a confusion matrix in Table 1 and 2, where estimates of the overall accuracy, recall and precision are also given. 789 test samples were evaluated with approximately equal number of samples per class. Both confusion matrices show similar outputs varying slightly in the classification accuracy of the activities *walking downstairs* and *walking upstairs*. They also expose some false predictions mostly in the dynamic activities. Static activities instead perform better, particularly the *laying* activity which obtained an accuracy of 100%.

²We decide to show the result for $k = 8$ since the 8 bits fixed-point arithmetic is already implemented on almost all new smartphones.

Method	MC-SVM						
Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	Recall
Walking	109	0	5	0	0	0	95.6
Upstairs	1	95	40	0	0	0	69.8
Downstairs	15	9	119	0	0	0	83.2
Standing	0	5	0	132	5	0	93.0
Sitting	0	0	0	4	108	0	96.4
Laying	0	0	0	0	0	142	100
Precision %	87.2	87.2	72.6	97.1	95.6	100	89.3

Table 1: Confusion Matrix of the classification results on the test data using the traditional floating-point MC-SVM. Rows represent the actual class and columns the predicted class. The diagonal entries (in bold) show the number of test samples correctly classified.

Method	MC-HF-SVM $k = 8$ bits						
Activity	Walking	Upstairs	Downstairs	Standing	Sitting	Laying	Recall %
Walking	109	2	3	0	0	0	95.6
Upstairs	1	98	37	0	0	0	72.1
Downstairs	15	14	114	0	0	0	79.7
Standing	0	5	0	131	6	0	92.2
Sitting	0	1	0	3	108	0	96.4
Laying	0	0	0	0	0	142	100
Precision %	87.2	81.7	74.0	97.8	94.7	100	89.0

Table 2: Confusion Matrix of the classification results on the test data using the MC-HF-SVM with $k = 8$ bits. Rows represent the actual class and columns the predicted class. The diagonal entries (in bold) show the number of test samples correctly classified.

5 Conclusions

In this paper, we proposed a new method for building a multiclass SVM using integer parameters. The MC-HF-SVM is an appealing approach for use in Aml systems for healthcare applications such as activity monitoring on smartphones. This alternative that employs fixed-point calculations, can be used for AR because it requires less memory, processor time and power consumption. It provides accuracy levels comparable to traditional approaches (or greater) such as the MC-SVM that uses floating-point arithmetic. In addition the fixed-point arithmetic produces a regularization effect that affects the generalization ability of the predictive model that we are building and increases the accuracy of the model itself on previously unseen data.

The experimental results confirm that even with a reduction of bits equal to 6 for representing the learned MC-HF-SVM model parameter β , it is possible to substitute the standard MC-SVM. This outcome brings positive implications for smartphones because it could help to release system resources and reduce energy consumption. Future work will present a publicly available AR dataset to allow other researchers to test and compare different learning models.

References

- [1] Nigel Davies, Daniel P. Siewiorek, and Rahul Sukthankar. Activity-based computing. *IEEE Pervasive Computing*, 7(2):20–21, 2008.
- [2] Jessica Ekholm and Sylvain Fabre. Forecast: Mobile data traffic and revenue, worldwide, 2010-2015. In *Gartner Mobile Communications Worldwide*, 2011.
- [3] Diane J. Cook and Sajal K. Das. Pervasive computing at scale: Transforming the state of the art. *Pervasive and Mobile Computing*, 8:22–35, 2012.
- [4] N. Györfi, Á. Fábián, and G. Hományi. An activity recognition system for mobile phones. *Mobile Networks and Applications*, 14(1):82–91, 2009.
- [5] Vladimir N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, 1995.
- [6] Hartmut Neven, Vasil S Denchev, Geordie Rose, and William G Macready. Training a binary classifier with the quantum adiabatic algorithm. *Arxiv preprint arXiv08110416*, page 11, 2008.
- [7] Davide Anguita and Dario Sterpi. Nature inspiration for support vector machines. In *Proceedings of the 10th international conference on Knowledge-Based Intelligent Information and Engineering Systems - Volume Part II*, pages 442–449, 2006.
- [8] Felicity R Allen, Eliathamby Ambikairajah, Nigel H Lovell, and Branko G Celler. Classification of a known sequence of motions and postures from accelerometry data using adapted gaussian mixture models. *Physiological Measurement*, 27(10):935, 2006.
- [9] A. Rodríguez-Molinero, D. Pérez-Martínez, A. Samá, P. Sanz, M. Calopa, C. Gálvez, C. Pérez-López, J. Romagosa, and A. Catalá. Detection of gait parameters, bradykinesia and falls in patients with parkinson’s disease by using a unique triaxial accelerometer. *World Parkinson Congress, Glasgow*, 2007.
- [10] Andrea Mannini and Angelo Maria Sabatini. Machine learning methods for classifying human physical activity from on-body accelerometers. *Sensors*, 10(2):1154–1175, 2010.
- [11] Nishkam Ravi, Nikhil D, Preetham Mysore, and Michael L. Littman. Activity recognition from accelerometer data. In *In Proceedings of the Seventeenth Conference on Innovative Applications of Artificial Intelligence (IAAI)*, pages 1541–1546, 2005.
- [12] Jennifer R. Kwapisz, Gary M. Weiss, and Samuel A. Moore. Activity recognition using cell phone accelerometers. *SIGKDD Explor. Newsl.*, 12(2):74–82, 2011.
- [13] Y. LeCun, L. Jackel, L. Bottou, A. Brunot, C. Cortes, J. Denker, H. Drucker, I. Guyon, U. Mller, E. Sckinger, P. Simard, and V. Vapnik. Comparison of learning algorithms for handwritten digit recognition. In *International Conference on Artificial Neural Networks*, pages 53–60, 1995.
- [14] A. Ganapathiraju, J.E. Hamaker, and J. Picone. Applications of support vector machines to speech recognition. *Signal Processing, IEEE Transactions on*, 52(8):2348 – 2355, 2004.
- [15] John Wawrzynek, Krste Asanovic, Nelson Morgan, and Senior Member. The design of a neuro-microprocessor. *VLSI for Neural Networks and Artificial Intelligence*, 4:103–7, 1993.
- [16] Davide Anguita and Benedict A. Gomes. Mixing floating- and fixed-point formats for neural network learning on neuroprocessors. *Microprocess. Microprogram.*, 41(10):757–769, 1996.
- [17] D. Anguita, A. Ghio, S. Pischitta, and S. Ridella. A hardware-friendly support vector machine for embedded automotive applications. In *Neural Networks, 2007. IJCNN 2007. International Joint Conference on*, pages 1360 –1364, 2007.
- [18] Ryan Rifkin and Aldebaro Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
- [19] S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to platt’s smo algorithm for svm classifier design. *Neural Comput.*, 13(3):637–649, 2001.
- [20] John C. Platt. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. In *Advances in Large Margin Classifiers*, pages 61–74, 1999.
- [21] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. In-sample and out-of-sample model selection and error estimation for support vector machines. *Neural Networks and Learning Systems, IEEE Transactions on*, 23(9):1390–1406, 2012.

- [22] P.L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: Risk bounds and structural results. *The Journal of Machine Learning Research*, 3:463–482, 2003.
- [23] P.L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *The Annals of Statistics*, 33(4):1497–1537, 2005.
- [24] D. Anguita, A. Ghio, L. Oneto, and S. Ridella. The impact of unlabeled patterns in rademacher complexity theory for kernel classifiers. In *Proceeding of the Neural Information Processing Systems*, pages 1009–1016, 2011.