Applying Support Vector Machines and Boosting to a Non-Intrusive Monitoring System for Household Electric Appliances with Inverters

Takashi Onoda†, Gunnar Rätsch‡ and Klaus-Robert Müller‡*
† CIRL CRIEPI, 2-11-1 Iwadokita, Komae-shi, Tokyo, 201-8511 Japan
‡ GMD FIRST, Kekuléstr. 7, 12489 Berlin, Germany
* University of Potsdam, Computer Science Dept., Am Neuen Palais 10, 14469 Potsdam, Germany
onoda@criepi.denken.or.jp, {raetsch,klaus}@first.gmd.de

ABSTRACT

A non-intrusive load monitoring system has been developed for estimating the behavior of *individual* electrical appliances from the measurement of the *total* household load demand curve. The system is useful for monitoring both inverter and non-inverter type appliances that change their mode of operation over time. The total load demand is measured at the entrance of the feeder line into the house and the operating status of household electric appliances can be identified with the help of Support Vector Machines (SVM), Boosting, RBF and neural network techniques by analyzing the characteristic frequency content from the load curve of the household. Load curve measurements of air-conditioners, refrigerators (inverter type and non-inverter type), incandescent light, fluorescence light and television systems are used as examples for training and test data. So far only a small data set was measured for this feasibility study and our experiments show a great potential for machine learning techniques. In particular the Boosting algorithm exhibits accurate classification of the operating status both for inverter and non-inverter type electric appliances.

1. Introduction

Annual electricity consumption in residential areas in 1996 was 240TWh and is equivalent to 27% of the total consumption in Japan. The amount of electricity consumption in residential sectors is 3.1 times higher than in 1973 (first oil crisis). Compared with the electricity consumption in residential sectors in 1986, the electricity consumption in 1996 increased by 60% and this consumption per household increased by 39%. The increase has been caused mainly by a growing popularization of air conditioners, dryers and electric carpets which consume high ammounts of electricity, as well as an increase in the electricity consumption of refrigerators and television systems which have already been popular appliciances in households. Fig. 1 shows the respective shares of the annual electricity consumption for various types of electric household appliances [10].

The Third Conference of the Parties (COP3) of the UN Framework Convention on Climate Change has agreed to reduce greenhouse gases by 6%. In the revised energy-saving law which the National Diet adopted in May 1998, harder energy saving standards will be applied to designated electric appliances or equipments such as air conditioners, refrigerators, fluorescent lamps, television systems, video tape recorder systems, computers, magnetic disk drives, copiers, by the end of May of 1999. Under these social circumstances, Japanese household electric appliance manufacturers have begun to enter the market with products of much higher energy-efficiency, following up on the first product of this kind, which was an air conditioner with inverter system[1] in 1982. Now all air conditioners for households on the market have inverter systems to

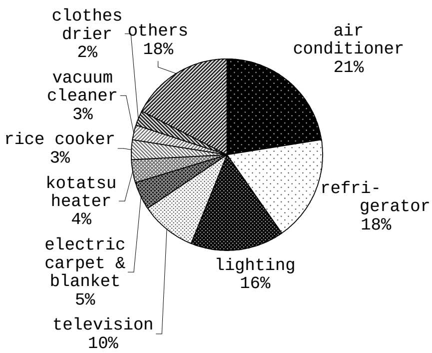

Fig. 1: Annual Electricity Consumption by Household Electric Appliances in 1996

[1] An inverter system controls e.g. the rotation speed of a motor (as in air-conditioner) by changing the frequency of the electric current.

reach a higher energy efficiency. Recently, also electric equipments with inverter systems, such as refrigerators, vacuum cleaners and microwave ovens, have emerged on the market, aiming to save energy. However, overall demand increases too greatly, to be able to compensate this additional demand with inverter systems.

The most difficult problem for power companies is to handle short-term peak loads for which more and more power plants need to be built that give security against a peak load instigated power failure. Alternatively one could also think of the following future scenario, where the power companies would be able to (remotely) control new generation household appliances. At the time of peak demand, the power companies would simply remotely down-regulate a household appliance, e.g. an air conditioner. So every individual air conditioner would loose 0.5% efficiency, without being noticeable by the home owners. This proposed technique would be ideally suited for the power companies to cut rsp. smooth the peak demands.

So, power companies need means to control the electric energy demand in order to avoid (1) a big power cut in the near future and (2) the construction of in principle unnecessary additional power plants. A prerequisite for controlling the electric energy demand, however, is the ability to correctly and non-intrusively detect and classify the operating status of electric appliances of individual households. Our goal is therefore to develop exactly such a non-intrusive measuring system for assessing the status of electric appliances with and without inverters. Particularly for inverter type electric appliances this is a hard problem, whereas non-intrusive measuring systems have already been developed for conventional on-off (non-inverter) operating electric equipments [8, 4]. Fig. 2 illustrates electric load curves of an air conditioner with inverter system and a refrigerator without inverter system. Clearly, the load curve of the air conditioner is more complex than that of the refrigerator. The operating behavior of the refrigerator is characterized by repeated on-off operation, which is a comparatively simple pattern, whereas the operating behavior of the air conditioner depends on the modulation of the inverter system and has a more complicated electric load curve.

This paper presents the first evaluation of Support Vector Machines (SVM), Boosting, RBF and neural network techniques to classify the operating status of electric appliances (with and without inverter) for the purpose of constructing a non-intrusive measuring system.

Section 2 shows a data measurement system which measures harmonic waves of some electric appliances. Note that we could only access very few data points since the data was gathered manually (automatic data assessment is currently being developed). The SVM and Boosting algorithms are shown in Section 3 and Section 4 gives a first experimental study evaluating SVMs and Boosting algorithms in this task. Finally, results and related work are discussed in Section 5.

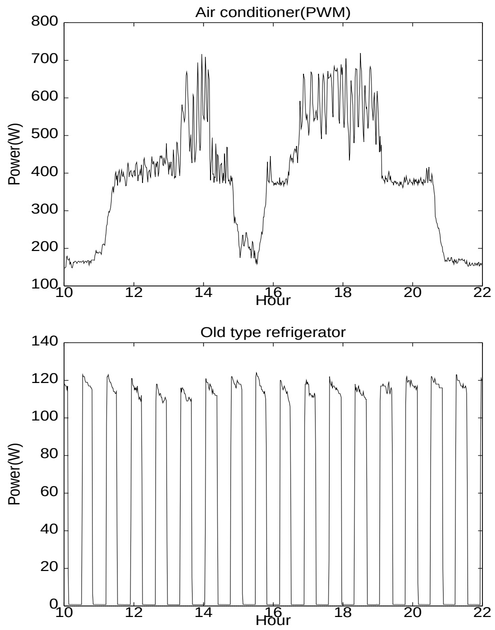

Fig. 2. Load Curves of Household Electric Appliances

2. Data Measurement System

The circuit configuration for measuring the data has harmonic wave analyzers as shown in Fig. 3. In order to access the load state from the feeder,

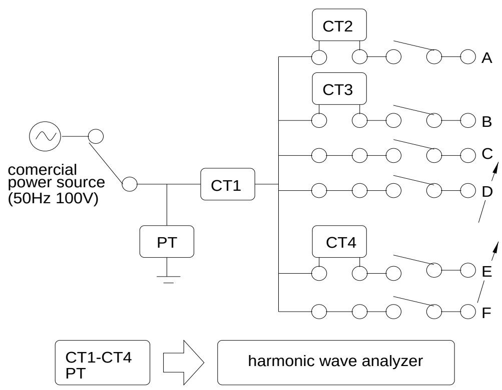

Fig. 3: Structure of the measurement system of harmonic waves for training and test data

the electric current sensor CT1 is placed in the line, whereas CT2, CT3 and CT4 are measuring the load state of each single appliance. The informations at CT2-4 are used to obtain stable points of the respective appliances at certain operating power levels (e.g. 0%, 60%, 100%), so that stable point information becomes available at CT1. By appropriately changing the combination of switches 1-6, the combination of electric appliances can be changed

Table 1. List of considered electric appliances

Symbol	Electric Appliance
A	air conditioner with inverter (1000W)
B	refrigerator with inverter (220W)
C	old type refrigerator(150W)
D	incandescent (100W $\times$ 10)
E	fluorescence (20W $\times$ 10)
F	television system (80W)

and various load states can be measured. The current at CT1 (at stable points) is analyzed for frequency content. In particular the features – used later as input data for the learning algorithms – are obtained from the current at the power system. They consist of odd-order harmonic currents up to the 13th order and respective the phase angles (expressed by sin) that can be measured by a special purpose hardware device at a stable load point. The corresponding outputs are ± 1 values, with -1 indicating a non-operating, and 1 indicating an operating electric appliance.

At this state of the project all measurements and the control of the individual appliances were done manually, which makes the process of gathering data very expensive and tedious and which gives an explanation why we have so few data points. Currently, however, an automatic measuring system is being built. Table 1 lists the electric appliances which are used for the measurements. The harmonic wave data is obtained under combined conditions between the fixed conditions of a-f and variable conditions 1-4. In this sense, measurement data under 24 (6×4) combined conditions are obtained as training data (see Table 2). Another 12 data points (not included in the table) are observed as test data. This data set was particularly designed to estimate the performance on inverter systems [19].

Table 2. Types of measurement for training data

Fixed Condition

Symbol	condition
a	power of incandescent 60%, power of fluorescence s 100%
b	power of incandescent s 60%, old type refrigerator ON
c	power of incandescent s 60%, television system ON
d	power of fluorescence 100%, old type refrigerator ON
e	power of fluorescence 100%, television system ON
f	old type refrigerator ON, television system ON

Variable Condition

No.	condition
1	power of air conditioner 0%
2	power of air conditioner 60%
3	power of air conditioner 100%
4	power of refrigerator 100%

3. Constructing the Metering System

Fig. 4 shows a sketch of the large margin classifier metering systems. For the metering system we investigate two types of large margin classifiers: SVMs and Boosting. These methods have already been recognized as very useful classifiers for pattern recognition, e.g. in OCR systems. Furthermore we compare the results with RBF, neural network and classical K Nearest Neighbor (KNN) techniques.

3.1. Support Vector Machines

Formally, the Support Vector Machine (SVM) [18] like any other classification method aims to estimate a classification function $f : \mathcal{X} \to \{\pm 1\}$ using labeled training data from $\mathcal{X} \times \{\pm 1\}$. Moreover this function f should even classify unseen examples correctly.

In order to construct good classifiers by learning, two conditions have to be respected. First, the training data must be an unbiased sample from the same source (pdf) as the unseen test data. This concerns the experimental setup. Second, the size of the class of functions from which we choose our estimate f, the so-called capacity of the learning machine, has to be properly restricted according to statistical learning theory [18]. If the capacity is too small, complex discriminant functions cannot be approximated sufficiently well by any selectable function f in the chosen class of functions – the learning machine is too simple to learn well. On the other hand, if the capacity is too large, the learning machine bears the risk of overfitting. In neural network training, overfitting is avoided by early stopping, regularization or asymptotic model selection [1, 11, 12, 13].

For SV learning machines that implement linear discriminant functions in feature spaces, the capacity limitation corresponds to finding a large margin separation between the two classes. The margin ϱ is the minimal distance of training points to the separation surface, i.e.

$$\varrho = \min_{i=1,\dots,\ell} \rho(\mathbf{z}_i, f) \qquad (1)$$

where

$$\rho(\mathbf{z}_i, f) = y_i f(\mathbf{x}_i), \qquad (2)$$

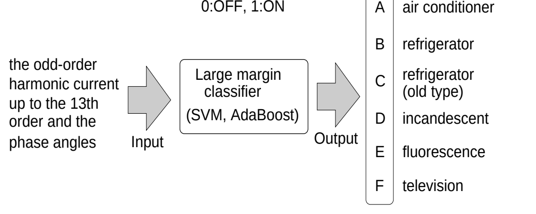

Fig. 4. Sketch of large margin classifiers metering systems

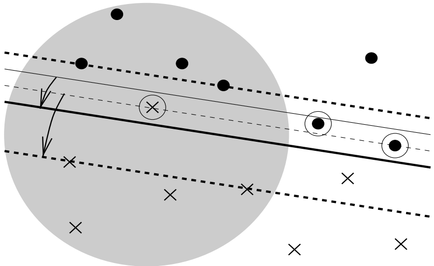

Fig. 5: A binary classification toy problem: This problem is to separate black circles from white circles. The shaded region consists of training examples, the other regions of test data. The data can be separated with a margin indicated by the slim dashed lines, implicating the slim solid line as discriminate function. Misclassifying one training example(a circled white circle) leads to a considerable extension(arrows) of the margin(fat dashed and solid lines) and this fat solid line can classify two test examples(circled black circles) correctly.

and f is the linear discriminant function in some feature space

$$f(\mathbf{x}) = (\mathbf{w} \cdot \Phi(\mathbf{x})) + b = \sum_{i=1}^{\ell} \alpha_i y_i (\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x})) + b, \tag{3}$$

with $\mathbf{w}$ expressed as $\mathbf{w} = \sum_{i=1}^{\ell} \alpha_i y_i \Phi(\mathbf{x}_i)$. The quantity Φ denotes the mapping from input space $\mathcal{X}$ by explicitly transforming the data into a feature space $\mathcal{F}$ using $\Phi : \mathcal{X} \to \mathcal{F}$.In order to train and classify, all that SVMs use are dot products of pairs of data points $\Phi(x), \Phi(y) \in \mathcal{F}$ in feature space (cf. Eq. 3). A kernel function k allows to implicitly define the feature space (Mercer's Theorem, e.g. [2]) via

$$k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})). \tag{4}$$

Note that there is no need to use or know the form of Φ, because the mapping is never performed explicitly! The introduction of Φ in the explanation above was for purely didactical and not algorithmical purposes. Therefore, we can computationally afford to work in implicitly very large (e.g. 10^{10}-dimensional) feature spaces. SVM can avoid overfitting by controlling the capacity and maximizing the margin. Simultaneously, SVMs learn which of the features implied by the kernel k are distinctive for the two classes, i.e. instead of finding well-suited features by ourselves (which can often be difficult), we can use the SVM to *select* them from an extremely rich feature space.

With respect to good generalization, it is often profitable to misclassify some outlying training data points in order to achieve a larger margin between the other training points (see Fig. 5 for an example). This soft-margin strategy can also learn non-separable data. The trade-off between margin size and number of misclassified training points is then controlled by the regularization parameter C (softness of the margin). The following quadratic program (QP) (see e.g. [18, 17]):

$$\begin{array}{lll} \min & \|\mathbf{w}\|^2 + C\sum_{i=1}^{\ell}\xi_i & \\ \text{s.t.} & \rho(\mathbf{z}_i, \boldsymbol{\alpha}) \geq 1 - \xi_i & \text{for all } 1 \leq i \leq \ell \\ & \xi_i, \alpha_i \geq 0 & \text{for all } 1 \leq i \leq \ell \end{array} \tag{5}$$

leads to the SV soft-margin solution allowing for some errors.

3.2. Boosting

AdaBoost also aims at estimating the classification function $f : \chi \to \{\pm 1\}$ using labeled training data from $\chi \times \{\pm 1\}$ – such that it classifies unseen examples well.

Let $\{h_t(\mathbf{x}) : t = 1, \ldots, T\}$ be an ensemble of T predictors defined on input vector $\mathbf{x}$ and $\boldsymbol{\alpha} = [\alpha_1 \ldots \alpha_T]$ their weights (with which they are combined) satisfying $\alpha_t \geq 0$. The combined hypothesis generates the label that is the (signum of the) weighted majority of the votes:

$$f_{\boldsymbol{\alpha}}(\mathbf{x}) := \|\boldsymbol{\alpha}\|_1^{-1} \sum_t \alpha_t h_t(\mathbf{x}). \tag{6}$$

In order to train this ensemble of T predictors $\{h_t(\mathbf{x})\}$ and $\boldsymbol{\alpha}$, several algorithms have been proposed: Bagging, where the weighting is simply $\alpha_t = 1/T$ [3] and AdaBoost/Arcing, where the weighting scheme is more complicated [16, 3].

In the following we give a brief description of AdaBoost and Arcing. AdaBoost aims to minimize a function of the margins $\rho(\mathbf{z}_i, \boldsymbol{\alpha}) := \rho(\mathbf{z}_i, f_{\boldsymbol{\alpha}})$ (cf. Eq. 2) on the training set

$$\mathcal{G}(\boldsymbol{\alpha}) = \sum_{i=1}^{\ell} \exp\left\{-\|\boldsymbol{\alpha}\|_1 (\rho(\mathbf{z}_i, \boldsymbol{\alpha}) - \phi)\right\}. \tag{7}$$

In order to find the predictor h_t, the learning examples $\mathbf{z}_i$ are weighted with $w^t(\mathbf{z}_i)$. Using a bootstrap on this weighted sample distribution we train h_t. Alternatively a weighted error function (e.g. weighted mean squared error) can be employed. The weights for the next iteration, $w^{t+1}(\mathbf{z}_i)$, are then updated according to

$$w^{t+1}(\mathbf{z}_i) := \frac{\exp\left\{-\|\boldsymbol{\alpha}^t\|_1 \rho(\mathbf{z}_i, \boldsymbol{\alpha}^t)\right\}}{\sum_j \exp\left\{-\|\boldsymbol{\alpha}^t\|_1 \rho(\mathbf{z}_j, \boldsymbol{\alpha}^t)\right\}}. \tag{8}$$

We can minimize $\mathcal{G}(\boldsymbol{\alpha})$ with respect to α_t by setting

$$\alpha_t = \underset{\alpha_t \geq 0}{\operatorname{argmin}}\, \mathcal{G}(\boldsymbol{\alpha}). \tag{9}$$

Note: For $\phi \equiv 0$ and $h_t(\mathbf{x}_i) \in \{\pm 1\}$ we can solve (9) analytically [3] and get $\alpha_t = \frac{1}{2} \log \frac{1-\epsilon_t}{\epsilon_t}$ [5].

Several researchers [3, 6, 9, 14] have noticed, that this procedure is equivalent to a constrained gradient descent procedure in the space of combined hypotheses that minimize (7). Rewriting (8) this connection becomes more apparent:

$$w^{t+1}(\mathbf{z}_i) := \frac{\partial\mathcal{G}(\boldsymbol{\alpha}^t)/\partial\rho(\mathbf{z}_i,\boldsymbol{\alpha}^t)}{\sum_j \partial\mathcal{G}(\boldsymbol{\alpha}^t)/\partial\rho(\mathbf{z}_j,\boldsymbol{\alpha}^t)}\,, \qquad (10)$$

i.e. first a gradient (in function space [9]) is computed and then the optimal step size is used. For an algorithmic description see pseudocode in Fig. 6.

Algorithm AdaBoost($\mathcal{G}$)
Input:
ℓ examples $(\mathbf{x}_1, y_1), \ldots, (\mathbf{x}_\ell, y_\ell)$
Initialize:
$w^1(\mathbf{z}_i) = 1/\ell$ for all $i = 1 \ldots \ell$
Do for $t = 1, \ldots, T$,
1. Train base classifier using the example distribution $\mathbf{w}^t$ and obtain $h_t : \mathbf{x} \mapsto [-1, 1]$
2. compute optimal α_t according to (9)
3. update example distribution $\mathbf{w}^{t+1}$ with (10)
Output:
final hypothesis $f(\mathbf{x}) = \frac{1}{\|\boldsymbol{\alpha}\|}\sum_{t=1}^{T}\alpha_t h_t(\mathbf{x})$

Fig. 6. The *generalized* AdaBoost algorithm using $\mathcal{G}$

Moreover, connections between Boosting and Large Margins Classifiers have been shown [3, 15]. Setting $\phi = \min_i \rho(\mathbf{z}_i, \boldsymbol{\alpha}^{t-1})$ makes it possible to prove that the so called Arc-GV algorithm [3] converges to the maximum margin solution (for $\phi \equiv 0$ this is also believed, but not proven).

From these facts it is clear that AdaBoost certainly has problems with overfitting, if the data is noisy (the same applies for SVMs with hard margin) [7, 14]. To avoid overfitting several suggestions [9, 14, 15] have been made to change AdaBoost's error function (7). Here, we adopt one idea – RoBoost/ν-Arc [15]. It combines the maximum margin property mentioned above, the slack-variable approach of SVMs and the ν-trick [17]. The basic idea is to use AdaBoost/Arc-GV to approximate the solution of the following linear program (LP) [14] that uses SVM's slack-variables:

$$\begin{array}{lll} \max & \tilde{\varrho} - C\sum_{i=1}^{\ell}\xi_i & \\ \text{s.t.} & \rho(\mathbf{z}_i,\boldsymbol{\alpha}) \geq \tilde{\varrho} - \xi_i & \text{for all } 1 \leq i \leq \ell \\ & \xi_i, \alpha_t, \tilde{\varrho} \geq 0 & \text{for all } 1 \leq t \leq T \\ & \|\boldsymbol{\alpha}\|_1 = 1 & \text{and } 1 \leq i \leq \ell \end{array} \qquad (11)$$

where C again controls the trade-off between the margin size and the amount of allowed margin errors. Interestingly, for $C = \frac{1}{\nu\ell}$ it can be shown [17, 15] that ν directly controls the fraction of patterns within the margin area, i.e. $\nu\ell$ is the number of patterns that are allowed to have $\xi_i > 0$.

To exploit AdaBoost's and Arc-GV's maximum margin property for approximating (11), one transforms the LP to an equivalent min-max program. It turns out [15] that only the definition of the margin has to be changed:

$$\widetilde{\rho}(\mathbf{z}_i,\boldsymbol{\alpha}) := \rho(\mathbf{z}_i,\boldsymbol{\alpha}) + \xi_i - \frac{1}{\nu\ell}\sum_i \xi_i, \qquad (12)$$

where $\xi_i := \max(\rho(\mathbf{z}_i,\boldsymbol{\alpha}) - \widetilde{\varrho}_\nu(\boldsymbol{\alpha}), 0)$ and $\widetilde{\varrho}_\nu(\boldsymbol{\alpha}) := \operatorname{argmax}_{\widetilde{\varrho}>0}\, \widetilde{\varrho} - \nu^{-1}\ell^{-1}\sum_i \xi_i$. Plugging definition (12) into (7) we obtain the **RoBoost** algorithm for $\phi \equiv 0$. Using $\phi \equiv \widetilde{\varrho} - \nu^{-1}\ell^{-1}\sum_{i=1}^{\ell}\xi_i$ we get the **ν-Arc** algorithm. They are essentially the same [15], but use a slightly different strategy for approximating the solution of (11).

Interestingly, for $\nu = 0$ we recover the original AdaBoost/Arc-GV algorithm and for $\nu = 1$ we (almost) obtain Bagging.

4. Experimental Results

In our comparison on the data described in Section 2 we use RBF networks [1, 14], KNN, SVMs (cf. Section 3.1) and ν-Arc[2] (cf. Section 3.2). Furthermore, we state the results achieved using Multi-Layer-Perceptrons (MLPs) given in [19].

4.1. Model Selection

The data set is very small, so one has to be careful with finding a good model, that is at the same time representative enough, but not overly complex. To get reliable results, also the experimental setup has to be quite sophisticated. We start with summarizing, which model parameters have been chosen.

The SVM has the regularization parameter C that determines the trade-off between the size of the margin and the number of margin errors. Furthermore, we have to specify the kernel, which implies the non-linear mapping $\Phi(\cdot)$. In our experiments we use a RBF kernel [18] (cf. Eq. 4)

$$k(\mathbf{x},\mathbf{y}) = \exp\left(\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\sigma^2}\right)$$

that has an additional parameter σ that determines the width. For ν-Arc we use RBF networks as base hypotheses. As model parameters we have to find the regularization parameter ν, which controls the number of margin errors, and the optimal number T of Boosting iterations. The RBF network optimizes a regularized (weight decay) mean-squared error function. Model parameters are the number of RBF centers m (i.e. number of hidden units) and the regularization parameter λ and the number of iterations I for optimization.

That means for each method we have to find two or three model parameters. Because the data is extremely small, it becomes very difficult to get estimates for good model parameters. Therefore, we use **Leave-One-Out** (LOO) cross-validation [1], which is known to be rather reliable, but computationally very expensive. We scan the following ranges in the model parameter space:

[2] We experienced that the results of ν-Arc and RoBoost are very similar and therefore we use only ν-Arc.

SVM	$C=10^{-1}\dots10^5$	$\sigma=10^{-3}\dots10^3$
ν-Arc	$\nu=0\dots1$	$T=1\dots30$
RBF	$m=3\dots10$	$\lambda=10^{-8}\dots1$
	$I=0\dots15$	

4.2. Experiment Setup and Results

In a first experiment we use the original data (24 train, 12 test points). This particular realization [19] of the data was designed to evaluate the performance of predicting the operation status of air conditioners – but it also gives some insight to the performance on other tasks. In Table 3 one can find the generalization errors in % for the first experiment (see [19] for their experimental setup for MLPs). One finds, that the first two (air conditioner, inverter fridge) and the last two (fluorescence light, TV) classification tasks are easy to solve for all methods (on this realization). The most difficult task is the third (old fridge). The best performance is achieved by ν-Arc that classifies all test patterns correctly.

Table 3: Generalization Errors in % on the original realization of the data for MLPs [19], RBF networks [14], ν-Arc and SVMs. The model parameters are found by Cross-Validation (see text).

Method	MLP	RBF	ν-Arc	SVM
Air Cond.	0.8	0.0	0.0	0.0
Fridge	0.0	0.0	0.0	0.0
old Fridge	27.5	14.0	0.0	8.3
incand. Light	16.7	0.0	0.0	0.0
fluor. Light	0.0	0.0	0.0	0.0
TV	2.5	0.0	0.0	0.0
Av.Perf.	7.9	2.3	0.0	1.4
Av.Inv.S.	0.4	0.0	0.0	0.0

The problem with this first experiment is that the test set is too small to get good generalization error estimates (and error bars). Therefore, we setup the second experiment, that uses random splits of the whole data set: First, we generate 20 realizations of training and test data. We split the 36 patterns into 30 training and 6 test patterns. Then we do the model selection procedure on each single data set and compute the test error for the selected model. Finally, we have 20 (almost independent) estimates of the generalization error (each on 6 test patterns). These estimates are averaged to get a more reliable estimate and a confidence interval was computed (see Table 4). We observe that ν-Arc yields a better performance than SVMs and RBF networks.

4.3. Discussion

In the first experiment based on the original data [19], where the non-intrusive metering system has to predict only the operating status of air conditioner and refrigerator, then neural networks (MLP, RBF), SVMs and ν-Arc can almost completely ($\approx$ 100%) detect the system state on untrained data. However, it is difficult for MLP and RBF to detect the operating status of electric appliances that do not have an inverter system. One reason for this might be that these electric appliances scarcely transmit harmonic waves and the data simply does not contain the information needed to make the classification.

The SVM made only one misclassification, which is an operating status of an old type refrigerator, that does not have an inverter system. The ν-Arc algorithms gives a perfect classification.

In the second experiment using random splits, ν-Arc followed by RBF achieve the best generalization performance in comparison to the other methods methods (see Table 4). The results of the second experiment show the validity of the first experiment from a statistical point of view. We observe from the tables that the best average performance is achieved by ν-Arc. Clearly, the classical KNN yields the worst classification rate.

We believe that the worse performance of SVM compared to ν-Arc is due to the fixed kernel (width). RBF and neural networks can detect multi-scaling information in the data, but optimize a suboptimal error function (mean squared error). ν-Arc can combine both virtues – maximizing the margin and "looking" at different scales of the data.

5. Conclusion

In order to estimate the states of electric appliances in a general household, we proposed a construction of a non-intrusive load monitoring system based on measuring harmonic waves, and evaluated the applicability of neural networks (MLP & RBF), KNN, SVM and AdaBoost. The electric appliances consisted of inverter and non-inverter type electric appliances.

At this point our claim is (1) that our careful leave-one-out strategy already reveals the potential of the classifiers on the task (in a controlled statistical setting) and (2) from the application point of view that the classifier precision is already reliable enough to assess the system state of the electric ap-

Table 4: Generalization Errors (incl. confidence intervals) in % averaged over 20 random splits into train and test data (84% train:16% test) for RBF networks, ν-Arc, SVM with RBF-kernel and KNN. The model parameters are found by Cross-Validation on each realization of the training data separately.

	RBF	ν-Arc	SVM	KNN
Air Cond.	3.7±2.0	**2.5±1.8**	5.8±2.2	6.7±2.2
Fridge	0.0±0.0	0.0±0.0	0.0±0.0	0.0±0.0
old Fridge	26.9±2.5	**25.0±3.9**	35.0±3.8	43.3±4.7
incand. L.	24.4±4.2	**21.7±4.2**	23.3±4.7	23.3±3.9
fluor. L.	16.4±3.6	**15.0±4.5**	15.8±4.8	39.2±5.0
TV	2.8±1.4	1.7±1.7	**0.8±0.8**	5.0±2.1
Av. Perf.	11.9±0.9	**11.0±1.2**	13.5±1.2	19.6±1.1
Av.Inv.S.	2.1±1.2	**1.3±2.1**	2.9±1.1	3.3±1.1

pliances. The latter point is an interesting novelty as previous techniques [8] have only been able to assess the status of non-inverter type appliances, whereas we can now do so for *both* inverter and non-inverter type appliances.

So far the data was collected manually which took a lot of time, man power, i.e. money. A larger data set will have been collected by the time of presentation at NC2000, so a much more extensive study can be presented then. Nevertheless our results make us think that the future vision of a control system to balance load peaks – a large progress from the the environmental point of view – might not at all belong in the realm of science fiction anymore.

Acknowledgments: We greatfully acknowledge valuable discussions with A. Smola, B. Schölkopf, S. Mika.

References

[1] C. Bishop, *Neural Networks for Pattern Recognition.* Oxford: Clarendon Press, 1995.

[2] B. Boser, I. Guyon, and V. Vapnik, "A training algorithm for optimal margin classifiers," in *5th Annual ACM Workshop on COLT*, (D. Haussler, ed.), (Pittsburgh, PA), pp. 144–152, ACM Press, 1992.

[3] L. Breiman, "Prediction games and arcing algorithms," Technical Report 504, Statistics Department, University of California, December 1997.

[4] J. Carmichael, "Non-intrusive appliance load monitoring system," EPRI Journal, Electric Power Research Institute, 1990.

[5] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *EuroCOLT: European Conference on Computational Learning Theory*, LNCS, 1994.

[6] J. Friedman, "Greedy function approximation," Technical Report, Department of Statistics, Stanford University, Feb. 1999.

[7] A. Grove and D. Schuurmans, "Boosting in the limit: Maximizing the margin of learned ensembles," in *Fifteenth National Conference on Artifical Intelligence*, 1998.

[8] W. Hart, "Non-intrusive appliance load monitoring," *Proceedings of the IEEE*, vol. 80, no. 12, 1992.

[9] L. Mason, J. Baxter, P. Bartlett, and M. Frean, "Functional gradient techniques for combining hypotheses," in *Advances in Large Margin Classifiers*, (A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds.), pp. 221–247, Cambridge, MA: MIT Press, 1999.

[10] "Outline of supply and demand of electric power 1997, miti," 1997. (in Japanese).

[11] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion - determining the number of hidden units for an artificial neural network model," *IEEE Transactions on Neural Networks*, vol. 5, no. 6, pp. 865–872, 1994.

[12] T. Onoda, "Neural network information criterion for the optimal number of hidden units," in *Proc. ICNN'95*, pp. 275–280, 1995.

[13] J. Orr and K.-R. Müller, eds., *Neural Networks: Tricks of the Trade.* LNCS 1524, Springer Verlag, 1998.

[14] G. Rätsch, T. Onoda, and K.-R. Müller, "Soft margins for AdaBoost," Technical Report NC-TR-1998-021, Department of Computer Science, Royal Holloway, University of London, Egham, UK, 1998. Machine Learning to appear.

[15] G. Rätsch, B. Schökopf, A. Smola, S. Mika, T. Onoda, and K.-R. Müller, "Robust ensemble learning," in *Advances in Large Margin Classifiers*, (A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, eds.), pp. 207–219, Cambridge, MA: MIT Press, 1999.

[16] R. Schapire, Y. Freund, P. Bartlett, and W. Lee, "Boosting the margin: a new explanation for the effectiveness of voting methods," in *Proc. 14th International Conference on Machine Learning*, pp. 322–330, Morgan Kaufmann, 1997.

[17] B. Schölkopf, A. Smola, R. Williamson, and P. L. Bartlett, "New support vector algorithms," NeuroCOLT Technical Report NC-TR-98-031, Royal Holloway College, University of London, UK, 1998. To appear in *Neural Computation.*

[18] V. Vapnik, *The Nature of Statistical Learning Theory.* Springer, 1995.

[19] K. Yoshimoto and Y. Nakano, "Non-intrusive load monitoring system part i: Identification of inverter-driven appliances by a neual network," Technical Report, Central Institute of Electric Power Industry, 1999. (in Japanese).