available at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/compag

Application of distributed SVM architectures in classifying forest data cover types

Mira Trebar[a,*], *Nigel Steele*[b]

[a] *University of Ljubljana, Faculty of Computer and Information Science, Trzaska 25, 1000 Ljubljana, Slovenia*
[b] *Department of Mathematical Sciences, Coventry University, Priory Street, Coventry CV1 5FB, UK*

ARTICLE INFO

Article history:
Received 1 June 2007
Received in revised form
27 November 2007
Accepted 4 February 2008

Keywords:
Support vector machine
Classification
Distributed architecture
Imbalanced data
Training subsets

ABSTRACT

In many 'real-world' applications, a classification of large data sets, which are often also imbalanced, is difficult due to the small, but usually more interesting classes. In this study, a large data set, forest cover type classes, which is actually multi-class classification defined with seven imbalanced classes and used as a resource inventory information was analyzed and evaluated. The data set was transformed into seven new data sets and a support vector machine (SVM) was employed to solve a binary classification problem of balanced and imbalanced data sets with various sizes. In the two approaches considered, the use of distributed SVM architectures, which basically reduces the complexity of the quadratic optimization problem of very large data sets, and the use of two sampling approaches for classification of imbalanced data sets were combined and results presented. The experimental results of distributed SVM architectures show the improvement of the accuracy for larger data sets in comparison to a single SVM classifier and their ability to improve the correct classification of the minority class.

1. Introduction

Accurate classification information is vital to any research institute or management agency, dealing with large real data sets. As an example of large data sets, forest cover types (UCIKDD archive, 2005) which came from US Forest Service Inventory information and a geographic information system, as an imbalanced multi-class classification problem, was studied and analyzed (Blackard and Dean, 1999). The main goal was to obtain the inventory information which is vital to any management agency for adjoining lands that are not directly under its control and is often impossible to collect this inventory data. The classification accuracy results of multi-class classification with neural networks having seven outputs defining a predicted class outperformed the results of discriminant analysis. The same data set was also used in some other applications where it was transformed into a binary classification problem with class 2 separated from the other six classes and it actually became a classification problem of balanced data set. In first case (Collobert et al., 2002), a parallel mixture of SVMs outperformed both a single SVM and also a neural network model (multi-layer perceptron) in terms of training, validation and test accuracy. It was also much faster in the training phase than a single SVM since the mixture can easily be parallelized and performed on cluster of computers. In the second case of binary classification of cover data type, the impact was also on the training of very large data sets where a cascade model of SVMs was defined (Graf et al., 2005). The samples in forest cover type training data set were split into subsets and optimized in parallel with multiple SVMs. The generalization of the model and the accuracy results were good and also improved after more iterations of optimization.

[*] *Corresponding author*. Tel.: +386 1 4768 254; fax: +386 1 4264 647.
E-mail address: mira.trebar@fri.uni-lj.si (M. Trebar).

0168-1699/$ – see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.compag.2008.02.001

Besides neural networks, the support vector machines (SVMs) are presented as an effective machine learning method (Vapnik, 2000) used in two-class (binary) classification problems where the samples belong to one of two classes, either positive class (+1) or negative class (−1). SVMs have the property of encapsulating all the information from the data set using a reduced number of samples, called support vectors (Kecman, 2001), which is usually much smaller than the number of training samples. The results on a set of text classification experiments (Joachims, 2002) using SVMs yielded lower accuracy error than many other classification methods, for example, Naive Bayes or neural networks. SVMs can also be used in a multi-class classification where the problem is transformed into several two-class classifiers, each trained to separate one class from the rest. After the training, they are combined to carry out the multi-class classification according to specified rules such as a majority voting mechanism, estimation of a class probability (Tax and Duin, 2002), least-squares estimation-based weighting and double layer hierarchical combining (Kim et al., 2003) or the decision directed acyclic graph (Platt and Cristianini, 2000).

The two-class classification systems, including SVMs usually tend to optimize the overall accuracy which is very appropriate and yields acceptable results for balanced data sets where the number of samples in both classes is nearly equal. In case of imbalanced data sets, it is often misleading due to small number of samples in the minority class which is usually more interesting and the classifier should tend to give the smallest possible number of misclassifications. The class imbalance problem is one of the problems that emerged when machine learning became more important in the world of business, industry and scientific research (Chawla et al., 2003). Approaches for solving the problem are usually divided into two directions: (i) sampling approaches, for example, oversampling the minority class or under-sampling the majority or; (ii) algorithm based approaches. In the over-sampling technique, the minority class is used to replicate the minority samples and in under-sampling, the majority samples are removed from the majority class. The sampling approaches were used in classification problems of imbalanced data sets which had from 100 to 20,000 samples (Batista et al., 2004), trained with neural networks, and also in a multi-class classification on less than 30,000 samples where the combination of two-class classifiers with an ensemble of SVMs classifiers and combining both over-sampling and under-sampling (Liu et al., 2006) were presented. In another algorithm based approach, a new method of pruning to search iteratively for a subset of support vectors from a large class used as samples to build a two-class classifier (Chen et al., 2005) is used where the classification rate of the small class is improved without significantly reducing the classification rate of the other class. The second algorithm approach (Kotsiantis and Pintelas, 2003) presents a mixture of expert agents method that combines distributed artificial intelligence and machine learning.

In the last few years, several attempts to classify large data sets with various training algorithms or models of SVMs have been published. Training time of an SVM is usually acceptable for small data sets, which is not the case in large data sets where the complexity of SVMs training increases very fast. To improve the training time, a sequential minimal optimization algorithm (Platt, 1999) breaks a large quadratic programming problem into a series of smaller quadratic programming problems. In text classification, the SVM^{light} fast implementation algorithm (Joachims, 2004) makes large scale SVM training more practical by using the idea of successive 'shrinking' of the optimization problem. Recently, the parallelization approaches which efficiently scale to large data sets, have been implemented on support vector machines. One of the approaches defines a parallel mixture of SVMs (Collobert et al., 2002), where each SVM is trained on a small subset of samples and can be implemented in parallel. The parallel SVMs which are solved independently, are organized in a cascade and can be spread over multiple processors (Graf et al., 2005). Constructing a support vector machine ensemble is also proposed as providing good generalization performance (Kim et al., 2003).

The main objective of this paper is to examine the classification performance of forest cover type as a large data set, which is basically multi-class classification problem transformed into seven binary classification problems. In the data set, there are actually imbalanced classes which were evaluated with three distributed SVM architectures. The performance obtained and the classification results were compared with the results obtained using a single SVM on the same binary classification data sets.

2. Forest cover type data set

For this study, a large data set, the forest cover type data set from UCI KDD archive (UCIKDD archive, 2005) was used. The following seven forest cover type classes used in a classification problem were:

- C1: Spruce/fir (*Picea engelmannii* and *Abies laciocarpa*),
- C2: Lodgepole pine (*Pinus contorta*),
- C3: Ponderosa pine (*Pinus ponderosa*),
- C4: Cottonwood/willow (*Populus angustifolia*, *Populus deltoides*, *Salix bebbiana*, *Salix amygdaloides*),
- C5: Aspen (*Populus tremuloides*),
- C6: Douglas-fir (*Pseudotsuga menziesii*),
- C7: Krummholz (Engelmann spruce (*Picea engelmannii*), subalpine fir (*Abies lasiocarpa*) and Rocky Mountain bristlecone pine (*Pinus aristata*)).

The forest cover type for the 30 m × 30 m cells obtained from US Forest Service (USFS) Region 2 Resource Information System (RIS) data represents the primary dominant tree species currently found in four wilderness areas (Neota with 3904 ha, Rawah with 29628 ha, Comanche Peak with 27389 ha, Cache la Poudre with 3817 ha) located in the Roosevelt National Forest of northern Colorado. In these areas, the existing forest cover types are primarily a result of natural ecological processes which experienced relatively little human disturbances in the past. Some interesting background information for these four wilderness areas are the mean elevational value and the primary tree species found there. Neota area has the highest mean elevational value and a primary major tree species spruce/fir (C1). Rawah and Comanche Peak areas would have a lower mean elevational value and lodgepole pine (C2) as their primary tree species, followed by spruce/fir (C1) and aspen (C5). Cache la Poudre area would have the lowest mean elevational value and Ponderosa pine (C3), Douglas-fir (C6) and Cottonwood/willow (C4).

From US Geological Survey (USGS) and USFS original data, a total number of 581,012 observations with 54 attributes was defined in the cover type data set (UCIKDD archive, 2005). The distribution of seven data classes is presented as class C1 with 211,840 observations, class C2 with 283,301 observations, class C3 with 35,754 observations, class C4 with 2747 observations, class C5 with 9493 observations, class C6 with 17,367 observations and class C7 with 20,510 observations. The forest cover type data set was treated as a multi-class classification problem with imbalanced number of observations in the last five classes. The total number of 54 attributes of cover type data availably include the following 12 measures defined with 10 independent quantitative variables, four binary wilderness areas and forty binary soil type variables. The quantitative variables are:

- Elevation in range 1859–3858 (m).
- Aspect (azimuth from true north) in range 0–360 (azimuth).
- Slope in range 0–66 (°).
- Horizontal _ Distance _ To _ Hydrology in range 0–1397 (m).
- Vertical _ Distance _ To _ Hydrology in range −173–601 (m).
- Horizontal _ Distance _ To _ Roadways in range 0–7117 (m).
- Hillshade _9 a.m. in range 0–255 (index).
- Hillshade _ Noon in range 0–255 (index).
- Hillshade _3 p.m. in range 0–255 (index).
- Horizontal _ Distance _ To _ Fire _ Points in range 0–7173 (m).

In observations, the wilderness area designation and soil type information are defined with binary variables treated as multiple binary values where a value ‘0’ would represent an absence and a value ‘1’ would represent a presence of a specific wilderness area or soil type.

3. Support vector machines

The support vector machine is a linear classifier (Vapnik, 2000) in the data space where a maximum hyperplane that separates a class of positive samples from a class of negative samples is constructed. This solution is generalized to a non-linear classifier, so called non-linear SVM, where the input samples are mapped from the original data space into a high-dimensional feature space, where they become linearly separable and there construct an optimal hyperplane.

Assume that there is a set of training samples given as $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \Re^n$, $y_i \in \{-1, +1\}$, $i = 1, \ldots, p$ for which there exists a hyperplane which linearly or nonlinearly separates both classes. The decision function of a nonlinear SVM for classification problem is defined as

$$y = \sum_{i=1}^{p} \alpha_i y_i K(\mathbf{x}, \mathbf{x}_i) + b \tag{1}$$

where $\mathbf{x}$ is a test sample, y_i the i-th class label and $\mathbf{x}_i$ the i-th training sample, p the number of training samples and $\boldsymbol{\alpha} = \{\alpha_1, \ldots, \alpha_p\}$ and b are the parameters of the model. The use of kernel function $K(\mathbf{x}, \mathbf{x}_i)$ assures that the mapping operation and all calculations associated with it are actually carried out in the data space using scalar products. One of the most common kernel functions used in SVMs is a Gaussian Radial Basis function defined as

$$K(\mathbf{x}, \mathbf{x}_i) = \exp\left\{-\frac{\|\mathbf{x} - \mathbf{x}_i\|^2}{\sigma^2}\right\} \tag{2}$$

with common variance σ which is experimentally defined. To find the coefficients $\alpha_i, i = 1, \ldots, p$, it is sufficient to solve the optimization problem in the dual space by finding the minimum of the objective function

$$Q(\boldsymbol{\alpha}) = \sum_{i=1}^{p} \alpha_i - \frac{1}{2} \sum_{i=1}^{p} \sum_{j=1}^{p} \alpha_i y_i \alpha_j y_j K(\mathbf{x}_i, \mathbf{x}_j) \tag{3}$$

as subject to

$$\sum_{i=1}^{p} \alpha_i y_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le U, i = 1, \ldots, p \tag{4}$$

where the upper bound U on the parameters α_i determines how severely classification errors are to be penalized and is also denoted as the soft margin optimal hyperplane parameter.

4. Experiments

To evaluate the performance of the proposed distributed SVM architectures and to compare the classification results with the single SVM, a series of experiments by using the SVM$^{\text{light}}$ implementation of support vector machines, version 6.01 (Joachims, 2004) was used. The optimization algorithm is fast and can handle problems with large number of samples efficiently. The software provides the performance measures used in the experiments to evaluate the performance of distributed SVMs.

The most common performance measure used to evaluate proposed classification methods is accuracy measured on training and test sets which treats both classes of observations equally and it is not evident which class performs better. This is usually very satisfactory but incomplete and not applicable to highly imbalanced data sets where there always exists one class with a small number of samples and another class with a large number of samples. In this study of both balanced and imbalanced data sets, performance measures of learning algorithms besides classification accuracy a true negative rate, a true positive rate, precision and an F-measure were also used. They are based on the confusion matrix shown in Table 1.

These metrics are widely used for comparison of imbalanced data sets. All the performance measures use the confusion matrix and are defined as:

- True negative rate (Acc^-) = TN/(TN + FP) is the percentage of negative cases correctly classified as belonging to the negative class.

Table 1 – Two-class confusion matrix

	Predicted positive class	Predicted negative class
Actual positive class	TP (true positive)	FN (false negative)
Actual negative class	FP (false positive)	TN (true negative)

- True positive rate (Acc^+) = TP/(TP + FN) is the percentage of positive cases correctly classified as belonging to the positive class.
- Accuracy = (TP + TN)/(TP + FP + TN + FN) is the parameter of the test defined as a proportion of false positives and true negatives.
- Precision = TP/(TP + FP) defines the degree to which further measurements will show the same or similar results.
- F-measure = (2 × Precision × Acc^+)/(Precision + Acc^+) measures the overall performance of the minority class.

The desirable performance of the classifier should be defined with high accuracy measure for balanced data sets and high true positive rate while maintaining reasonable true negative rate for imbalanced data sets.

4.1. *Experimental data sets and subsets*

For this study, where the support vector machine classifiers were used, two binary classes of observations were required and analyzed. A forest cover type is a multi-class classification problem with seven classes and for this purpose, from the original cover type data set, a set of seven new data sets denoted as DS*i*-581, $i = 1, \ldots, 7$ were defined. They contained observations falling into one of two binary classes where the observations of one cover type class (positive class) were separated from the other six cover type classes (negative class). In each data set, the total number of observations was $N = N1 + N2$, where $N1$ was the number of observations in the negative class (−1) and $N2$ was the number of observations in the positive class (+1). The seven data sets, produced for binary classification were: DS1-581 (C1 with $N2 = 211,840$ observations separated from the other six classes); DS2-581 (C2 with $N2 = 283,301$ observations separated from the other six classes); DS3-581 (C3 with $N2 = 35,754$ observations separated from the other six classes); DS4-581 (C4 with $N2 = 2747$ observations separated from the other six classes); DS5-581 (C5 with $N2 = 9493$ observations separated from the other six classes); DS6-581 (C6 with $N2 = 17,367$ observations separated from the other six classes); DS7-581 (C7 with $N2 = 20,510$ observations separated from the other six classes). In the first two binary classification problems, there exists nearly balanced number of observations in negative ($N1$) and positive class ($N2$) while in the other five classification problems there exists an highly imbalanced number of observations as shown in Fig. 1.

After that, from each data set DS*i*-581, $i = 1, \ldots, 7$, additional data sets with smaller number of observations were generated by randomly sampling the data observations from a total number of 581,012 observations. These data sets were: DS*i*-30 with 30,000 observations, DS*i*-60 with 60,000 observations and DS*i*-300 with 300,000 observations, where $i = 1, \ldots, 7$. The ratio of cover type classes for all data sets has been retained very similar as it is shown in Fig. 2. The mean values of observations in these data sets were: C1(36.43%), C2(48.76%), C3(6.20%), C4(0.46%), C5(1.60%), C6(3.01%) and C7(3.55%).

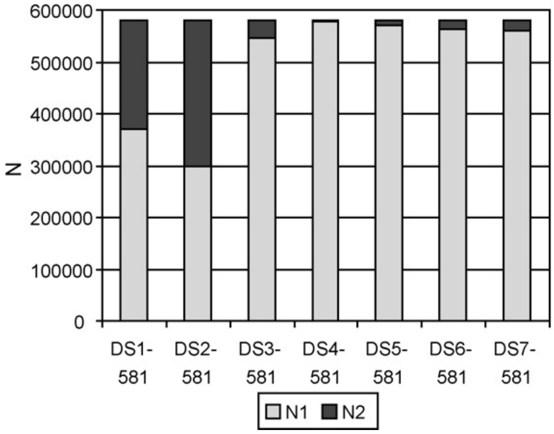

Fig. 1 – Forest cover type data sets defined as binary classification problem.

Sometimes raw data are not in the most convenient form for used classification method and it can be advantageous to modify them prior to analysis which was also the case in our study. Very often, when neural networks are used, the data observations are scaled to lie in the range between zero and one which can also be used in SVM classifiers. For the forest cover types, this data transformation was also tested where the accuracy results show the improvement but it was still not satisfactory. Finally, a ratio scale was defined by dividing each observation by a constant 1000 which leaves the ratio of values unaffected but it brings the Gaussian kernels of SVMs into an appropriate range.

To evaluate the classification performance of SVMs, all data sets were randomly distributed into training and test data sets with following distributions:

- The training set with 90% of observations and the test set with 10% of observations for DS*i*-30, DS*i*-60, DS*i*-300, DS*i*=581, $i = 1, \ldots, 7$ data sets.
- The training set with 70% of observations and the test set with 30% of observations for DS*i*-60, DS*i*-300, DS*i* =581, $i = 1, \ldots, 7$ data sets.

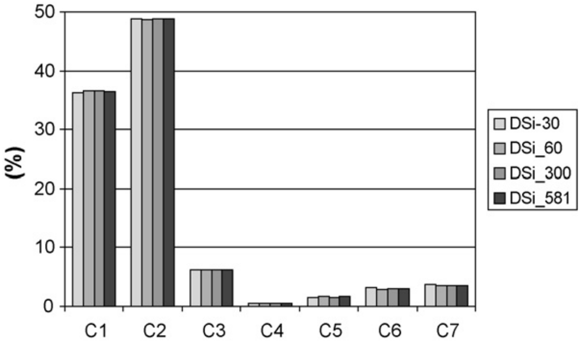

Fig. 2 – Ratio of forest cover type classes.

Furthermore, the partition of all training sets into M training subsets was performed. This number of training subsets was experimentally obtained in Trebar and Steele (2007), where the experiments with $M = 2, 4, 8, 16, 32$ were analysed. For the cover type data set, a binary classification problem (class C2 was separated from the other six classes) was evaluated with two-layered SVMs and more evident improvement of the training time by keeping or even improving the accuracy performance of a single SVM classifier was obtained with $M = 4$ subsets. This number of subsets indicated the optimal solution of training time and performance results.

Two different methods of sampling for generating training subsets were used in the training sets partition:

- Random subsets were defined by sampling without replacement therefore the observations were selected at random from both classes (positive and negative) in the training set and placed into $M = 4$ training subsets. This partition was used in all training sets for DS*i*-30, DS*i*-60, DS*i*-300, DS*i*-581, $i = 1, \ldots, 7$ data sets. The number of observations in each training subset was defined as $N1/4 + N2/4$.
- Semi-random subsets were defined by sampling with replacement where the observations of positive class was repeated. From training sets of DS*i*-60, DS*i*-300, DS*i*-581, $i = 3, \ldots, 7$ data sets with imbalanced classes ($N1 \gg N2$) the observations from a larger class (negative class) were selected at random and placed into $M = 4$ training subsets and the total number of observations from the other class with a smaller number (positive class) were placed into all M subsets. The number of observations in each training subset was defined as $N1/4 + N2$. This determination of training subsets is a non-heuristic method of over-sampling that aims to balance the class distribution through the replication of minority class observations in each subset.

4.2. *Distributed SVM architectures*

The proposed distributed SVM architectures with the independent SVMs organized in the layered architectures were used in the experiments: (i) a two-layered SVM architecture ($L = 2$), (ii) a three-layered binary cascade SVM architecture ($L = 3$) and (iii) a three-layered combinatorics SVM architecture ($L = 3$). The number of layers defined the number of training steps in distributed SVM architecture.

The two-layered SVM architecture shown in Fig. 3 consists of $M = 4$ independent SVMs on the first layer and a single SVM on the output layer. The training algorithm is performed in two steps, beginning with parallel training of first layer of SVMs with the input data samples forming subsets $\mathbf{x}_s(k)$, and desired class labels $y_s(k)$, $k = 1, \ldots, 4$. The results of first layer SVMs are subsets of support vectors $\mathbf{x}^1_{sv}(k)$ with class labels $y^1_{sv}(k)$ and they form a new set of samples used in second step of training. Finally, a new set of support vectors with learning parameters is obtained and used to evaluate performance of the described two-layered SVM architecture on the training and testing sets.

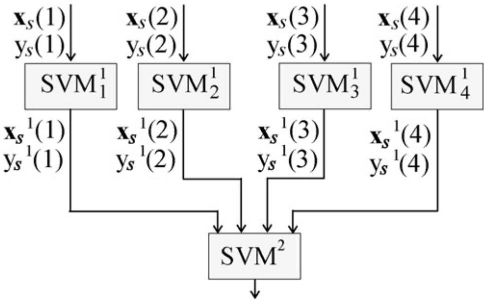
$\mathbf{x}_s(1)$ $y_s(1)$
$\mathbf{x}_s(2)$ $y_s(2)$
$\mathbf{x}_s(3)$ $y_s(3)$
$\mathbf{x}_s(4)$ $y_s(4)$
SVM^1_1
SVM^1_2
SVM^1_3
SVM^1_4
$\mathbf{x}_s{}^1(1)$ $y_s{}^1(1)$
$\mathbf{x}_s{}^1(2)$ $y_s{}^1(2)$
$\mathbf{x}_s{}^1(3)$ $y_s{}^1(3)$
$\mathbf{x}_s{}^1(4)$ $y_s{}^1(4)$
SVM^2

Fig. 3 – Two-layered SVM architecture.

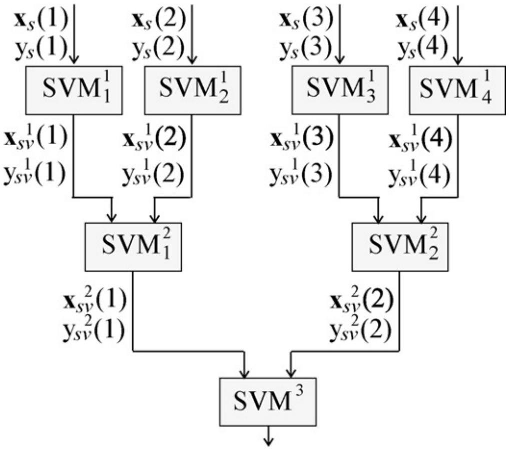
$\mathbf{x}_s(1)$ $y_s(1)$
$\mathbf{x}_s(2)$ $y_s(2)$
$\mathbf{x}_s(3)$ $y_s(3)$
$\mathbf{x}_s(4)$ $y_s(4)$
SVM^1_1
SVM^1_2
SVM^1_3
SVM^1_4
$\mathbf{x}^1_{sv}(1)$ $y^1_{sv}(1)$
$\mathbf{x}^1_{sv}(2)$ $y^1_{sv}(2)$
$\mathbf{x}^1_{sv}(3)$ $y^1_{sv}(3)$
$\mathbf{x}^1_{sv}(4)$ $y^1_{sv}(4)$
SVM^2_1
SVM^2_2
$\mathbf{x}^2_{sv}(1)$ $y^2_{sv}(1)$
$\mathbf{x}^2_{sv}(2)$ $y^2_{sv}(2)$
SVM^3

Fig. 4 – Three-layered binary cascade SVM architecture.

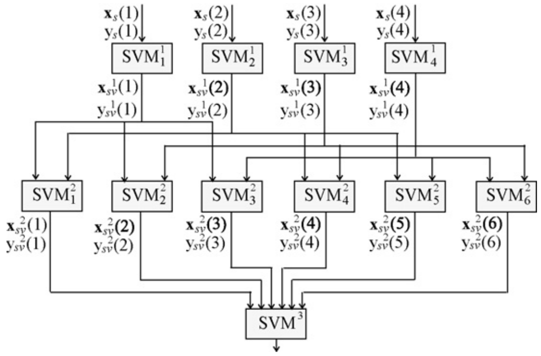
$\mathbf{x}_s(1)$ $y_s(1)$
$\mathbf{x}_s(2)$ $y_s(2)$
$\mathbf{x}_s(3)$ $y_s(3)$
$\mathbf{x}_s(4)$ $y_s(4)$
SVM^1_1
SVM^1_2
SVM^1_3
SVM^1_4
$\mathbf{x}^1_{sv}(1)$ $y^1_{sv}(1)$
$\mathbf{x}^1_{sv}(2)$ $y^1_{sv}(2)$
$\mathbf{x}^1_{sv}(3)$ $y^1_{sv}(3)$
$\mathbf{x}^1_{sv}(4)$ $y^1_{sv}(4)$
SVM^2_1
SVM^2_2
SVM^2_3
SVM^2_4
SVM^2_5
SVM^2_6
$\mathbf{x}^2_{sv}(1)$ $y^2_{sv}(1)$
$\mathbf{x}^2_{sv}(2)$ $y^2_{sv}(2)$
$\mathbf{x}^2_{sv}(3)$ $y^2_{sv}(3)$
$\mathbf{x}^2_{sv}(4)$ $y^2_{sv}(4)$
$\mathbf{x}^2_{sv}(5)$ $y^2_{sv}(5)$
$\mathbf{x}^2_{sv}(6)$ $y^2_{sv}(6)$
SVM^3

Fig. 5 – Three-layered combinatorics SVM architecture.

The three-layered binary cascade SVM architecture in Fig. 4 consists of $M = 4$ independent SVMs on the first layer, two independent SVMs on the second layer $M/2$ and a single SVM on the output layer. The first step of training process includes the parallel training of input data samples forming subsets $\mathbf{x}_s(k)$, and desired class labels $y_s(k)$, $k = 1, \ldots, 4$. The second step performs the training of two sets of support vectors from the first layer where the first set of samples includes the $\mathbf{x}^1_{sv}(1)$, $\mathbf{x}^1_{sv}(2)$ support vectors with class labels $y^1_{sv}(1)$, $y^1_{sv}(2)$ and the second set of samples includes $\mathbf{x}^1_{sv}(3)$, $\mathbf{x}^1_{sv}(4)$ support vectors with class labels $y^1_{sv}(3)$, $y^1_{sv}(4)$. The last step, the training of the output layer $L = 3$ with $\mathbf{x}^2_{sv}(1)$, $\mathbf{x}^2_{sv}(2)$ support vectors with class labels $y^2_{sv}(1)$, $y^2_{sv}(2)$ were used as a new set of samples to obtain the final set of support vectors.

The three-layered combinatorics SVM architecture in Fig. 5 consists of a first parallel layer of SVMs ($M = 4$), of a second parallel layer of SVMs where their number is defined by the

Table 2 – 10-fold cross-validation accuracy results (random training subsets)

Data	Training set (%)				Test set (%)			
	SVM	R4	R4-2	R4-6	SVM	R4	R4-2	R4-6
DS1-30	96.50	96.36	96.28	97.75	90.73	90.51	90.39	90.43
DS2-30	95.69	95.51	95.46	97.23	88.76	88.64	88.38	88.66
DS3-30	99.30	99.25	99.24	99.52	98.08	98.00	97.96	98.01
DS4-30	99.95	99.95	99.94	99.95	99.72	99.72	99.72	99.70
DS5-30	99.58	99.56	99.56	99.61	99.04	99.00	98.98	98.79
DS6-30	99.40	99.37	99.37	99.58	98.52	98.42	98.39	98.28
DS7-30	99.79	99.76	99.75	99.84	99.13	99.08	99.06	99.05

binomial coefficient $M!/(2!(M-2)!)$, and a single SVM on the output layer. The first step of training process includes the parallel training of input data samples forming subsets $\mathbf{x}_s(k)$, and desired class labels $y_s(k)$, $k = 1, \ldots, 4$. The second step performs the training of six sets of support vectors from the first layer where each new set of samples is formed from the support vectors of two SVMs. The last step, the training of output layer $L = 3$ with $\mathbf{x}^2_{sv}(k)$, support vectors with class labels $y^2_{sv}(k)$, $k = 1, \ldots, 6$ were used to obtain the final set of support vectors.

The training of distributed SVM architectures was performed layer by layer from the first layer with four independent SVMs to the output layer with only one SVM. For the comparison on training and test sets, presented in the paper, the output layer SVM$^{\text{light}}$ optimization parameters were used.

5. Results

In the experiments, the following distributed SVM architectures were used: (i) for random training subsets: a two-layered distributed SVM architecture (R4), a three-layered binary cascaded SVM architecture (R4-2) and a three-layered combinatorial SVM architecture (R4-6), and (ii) for semi-random training subsets: a two-layered distributed SVM architecture (SR4), a three-layered binary cascaded SVM architecture (SR4-2) and a three-layered combinatorial SVM architecture (SR4-6).

At the beginning, to determine the optimization parameters of SVMs, the DSi-30, $i = 1, \ldots, 7$ data sets were used. The training sets were used for a single SVM and four random training subsets from each training data set were used for distributed SVM architectures. In the SVM$^{\text{light}}$ implementation, with a radial basis function, the Gaussian kernels with a variance $(\sigma = 1, 4, 16, 25)$ and an upper bound $(U = 1, 5, 10)$ were chosen in a 10-fold cross-validation. For determining SVM parameter values, the accuracy results obtained on the test sets were used. Table 2 lists the accuracy results of the selected SVM model with the variance $\sigma = 16$ and upper bound $U = 5$.

For training data sets, at least one or more distributed SVM architectures produced better accuracy results than single SVM while for test data sets a single SVM is the most accurate approach which indicates that the use of distributed SVM architectures is not justified for smaller data sets. The classification accuracy results were actually very similar which was a satisfactory reason for continuing the experiments with defined SVM parameters ($\sigma = 16$, $U = 5$) on larger data sets.

5.1. *Accuracy comparison*

To further show the performance of other, larger data sets, the comparison of the accuracy results of the single SVM and the distributed SVM architectures using random and semi-random training subsets is presented. For three groups of data sets with different number of samples (DSi-60, DSi-300 and DSi-581, $i = 1, \ldots, 7$) the experiments were performed. For cover type data sets where the classes C1 and C2 are separated from the other classes, the number of samples in positive and negative class is very similar and the experiments were performed only with random training subsets. For all other data sets with classes C3, C4, C5, C6, C7 separated from the other classes, the random and semi-random training subsets were used.

The accuracy results presented in Table 3 obtained by the distributed SVM architectures for cover type classes C1 and C2 separated from the other classes outperform the results of the single SVM on test data sets with 10% of observations. For the test sets with 30% of observations this was the case for larger data sets DSi-300 and DSi-581, $i = 1, 2$ while for small data sets

Table 3 – Comparison of classification accuracy (random subsets: C1, C2)

Data	Test set (10%)				Test set (30%)			
	SVM	R4	R4-2	R4-6	SVM	R4	R4-2	R4-6
DS1-60	92.70	92.97	92.78	93.18	91.98	91.82	91.79	91.96
DS1-300	95.77	95.84	95.86	96.27	95.30	95.46	95.45	95.98
DS1-581	96.69	96.86	96.89	97.19	96.36	96.49	96.50	96.92
DS2-60	90.75	90.90	90.80	91.30	90.87	90.74	90.64	90.83
DS2-300	94.90	95.08	95.08	95.61	94.48	94.58	94.61	95.23
DS2-581	95.89	96.03	96.06	96.54	95.65	95.77	95.79	96.35

Table 4 – Comparison of classification accuracy (random subsets: C3, C4, C5, C6, C7)

Data	Test set (10%)				Test set (30%)			
	SVM	R4	R4-2	R4-6	SVM	R4	R4-2	R4-6
DS3-60	98.60	98.60	98.63	98.72	98.44	98.42	98.38	98.39
DS3-300	99.08	99.07	99.08	99.19	99.10	99.11	99.11	99.19
DS3-581	99.34	99.36	99.35	99.44	99.29	99.32	99.32	99.39
DS4-60	99.73	99.77	99.73	99.75	99.78	99.74	99.71	99.71
DS4-300	99.88	99.89	99.89	99.88	99.86	99.87	99.87	99.86
DS4-581	99.89	99.91	99.90	99.91	99.89	99.89	99.89	99.89
DS5-60	99.22	99.22	99.27	99.02	99.09	99.05	98.99	99.13
DS5-300	99.53	99.58	99.56	99.59	99.47	99.49	99.50	99.51
DS5-581	99.57	99.58	99.58	99.66	99.58	99.59	99.59	99.63
DS6-60	98.75	98.68	98.67	98.88	98.73	98.68	98.64	98.63
DS6-300	99.32	99.36	99.35	99.38	99.26	99.24	99.25	99.30
DS6-581	99.43	99.48	99.49	99.51	99.43	99.45	99.45	99.46
DS7-60	99.40	99.37	99.42	99.32	99.32	99.27	99.25	99.27
DS7-300	99.74	99.71	99.73	99.73	99.67	99.67	99.66	99.69
DS7-581	99.78	99.77	99.77	99.78	99.76	99.76	99.77	99.76

DS1-60 and DS2-60 single SVM gave slightly better results. The following Table 4 lists the accuracy results for classes C3, C4, C5, C6, C7 separated from the other classes. The distributed SVMs produced slightly better accuracy results for test sets with 10% of observations for all other data sets, except for DS7-300 and DS7-581 where they are almost the same. For test sets with 30% of observations, a single SVM produced better results on small data sets DS*i*-60, $i = 3, 4, 6, 7$ while for all other data sets the distributed SVM architectures gave better results.

For the second group of experiments with semi-random data sets, the accuracy results obtained by single SVM and the distributed SVM architectures shown in Table 5 for test data sets with 10% of observation are very similar or equal. A single SVM produced slightly better results for following data sets: DS3-581, DS4-60, DS5-60, DS5-300, DS7-300 and DS7-581 and slightly worse accuracy results for all other data sets. In majority of test sets with 30% of observations, single SVM was slightly better than the distributed SVM architectures in most cases except for DS3-60, DS3-300, DS3-581, DS5-60, DS6-581 and DS7-300 data sets.

When comparing the accuracy results of binary classification problems, for many data sets distributed SVM architectures using random and semi-random subsets perform well or even better in comparison to single SVM. For all cover type classes at least one or even more SVM architectures are comparable or even better than the single SVM in terms of accuracy measure.

5.2. *Comparison of true negative and true positive rate*

The overall classification accuracy was presented as the percentage of correctly classified number of samples in the training set or in the test set. This was acceptable and reasonable in balanced data sets where the number of observations in both classes (positive class and negative class) is comparable. For the imbalanced data sets where the number of observations in the positive class is very small, the results of cover type classes (C*i*) separated from the other classes are examined as the comparison between the true negative rate (C*i*-N) and true

Table 5 – Comparison of classification accuracy (semi-random subsets: C3, C4, C5, C6, C7)

Data	Test set (10%)				Test set (30%)			
	SVM	SR4	SR4-2	SR4-6	SVM	SR4	SR4-2	SR4-6
DS3-60	98.60	98.42	98.43	98.63	98.44	98.32	98.34	98.48
DS3-300	99.08	99.08	99.08	99.20	99.10	98.97	98.98	99.15
DS3-581	99.34	99.13	99.13	99.29	99.29	99.13	99.14	99.30
DS4-60	99.73	99.70	99.70	99.68	99.78	99.74	99.74	99.74
DS4-300	99.88	99.86	99.86	99.86	99.86	99.86	99.86	99.81
DS4-581	99.89	99.89	99.89	99.90	99.89	99.87	99.87	99.89
DS5-60	99.22	98.97	98.97	98.97	99.09	99.13	99.10	99.11
DS5-300	99.53	99.50	99.50	99.53	99.47	99.42	99.41	99.47
DS5-581	99.57	99.51	99.51	99.60	99.58	99.51	99.51	99.57
DS6-60	98.75	98.73	98.72	98.82	98.73	98.63	98.62	98.66
DS6-300	99.32	99.20	99.18	99.36	99.26	99.13	99.13	99.26
DS6-581	99.43	99.36	99.35	99.50	99.43	99.32	99.32	99.46
DS7-60	99.40	99.25	99.25	99.42	99.32	99.15	99.15	99.27
DS7-300	99.74	99.63	99.63	99.71	99.67	99.63	99.63	99.70
DS7-581	99.78	99.72	99.72	99.74	99.76	99.70	99.70	99.74

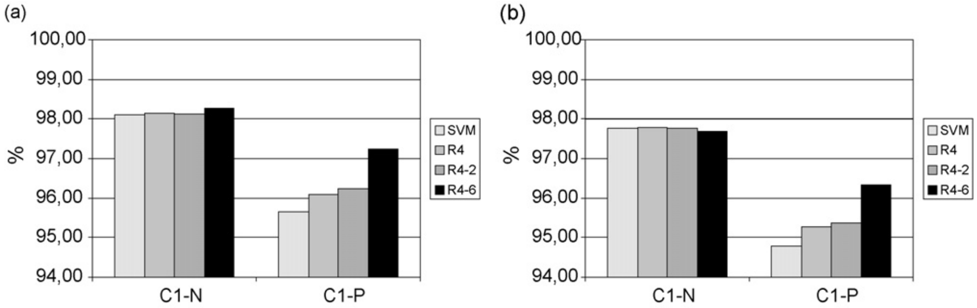

Fig. 6 – Comparison of true negative rate (C1-N), true positive rate (C1-P): (a) training set and (b) test set.

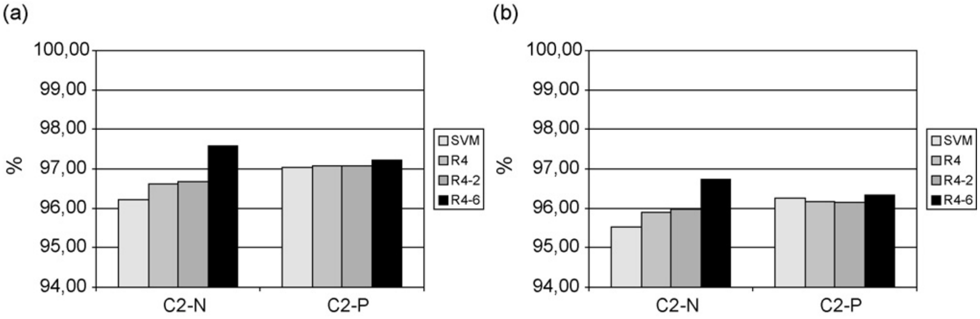

Fig. 7 – Comparison of true negative rate (C2-N), true positive rate (C2-P): (a) training set and (b) test set.

positive rate (Ci-P) in the training and test sets. For the largest data sets DSi-581, the comparisons of the single and the distributed SVM architectures on balanced and imbalanced data sets are presented.

For first two data sets (DS1-581, DS2-581) for training data sets with 90% of observation and test data sets with 10% of observations, the negative and in positive classes ($N1 \cong N2$) are comparable. For class C1 as a positive class with less than 40% of observations is shown the improvement of true positive rate on training and test data set in Fig. 6 for distributed SVMs and slightly improved true negative rate of training set and deteriorated true negative rate of test set. By using distributed SVM architectures, the true positive rate is improved with the improved overall accuracy presented in Table 3. In Fig. 7 with a similar number of observations in positive class and negative class the single SVM and distributed SVM architectures gave comparable true positive rates and true negative rates which are still slightly improved with distributed SVMs. In all other data sets (DS3-581, DS4-581, DS5-581, DS6-581 and DS7-581), the number of observations in the positive class is much smaller than in negative class. For the random and semi-random training subsets used in the experiments, the true negative rate is very similar but there is a great difference in the true positive rate which is improved. The true negative rate slightly deteriorates in case of semi-random subsets used in distributed SVMs.

The true positive rate was already improved in the case of distributed SVM architectures with random subsets and was significantly improved with semi-random subsets used in distributed SVM architectures. The improvement shown in Figs. 8–12 varies with the data sets. The use of semi-random subsets gave much the same rates for all data sets on training samples and slightly worse true positive rates on test sets.

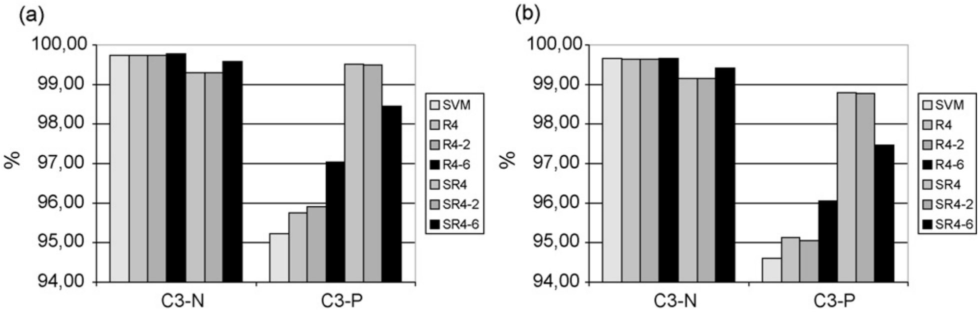

Fig. 8 – Comparison of true negative rate (C3-N), true positive rate (C3-P): (a) training set and (b) test set.

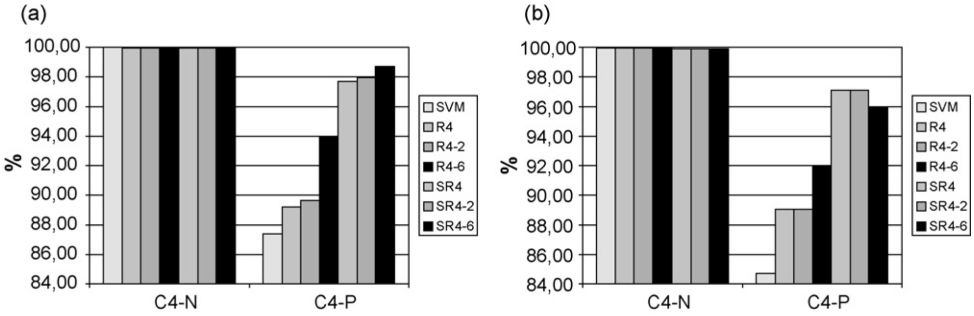

Fig. 9 – Comparison of true negative rate (C4-N), true positive rate (C4-P): (a) training set and (b) test set.

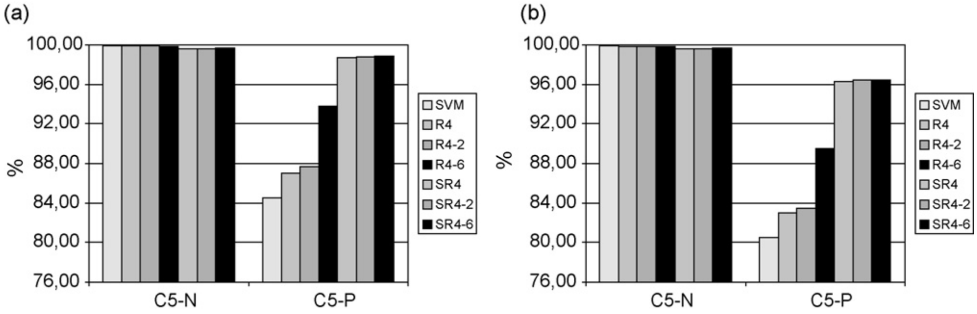

Fig. 10 – Comparison of true negative rate (C5-N), true positive rate (C5-P): (a) training set and (b) test set.

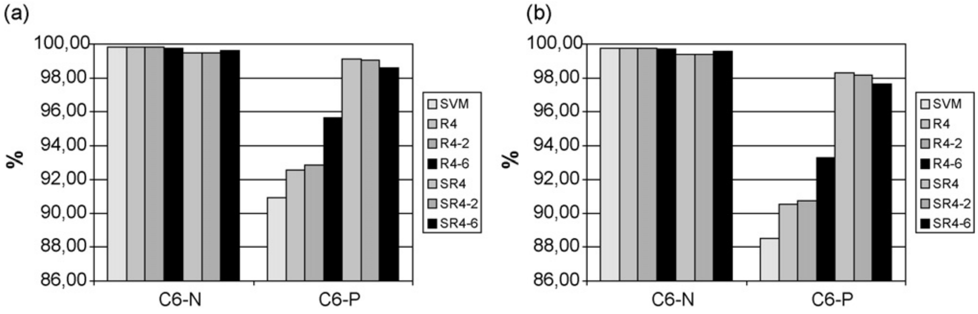

Fig. 11 – Comparison of true negative rate (C6-N), true positive rate (C6-P): (a) training set and (b) test set.

5.3. *Comparison of precision and F-measure*

Finally, the comparison of the precision and *F*-measure for the forest cover type data sets DS*i*-581 with 90% of training and 10% of test observations used in the experiments are presented. As presented in Table 6, the precision for random subsets of first three data sets (DS1-581, DS2-581, DS3-581) was slightly improved for distributed SVMs, for next data sets (DS4-581, DS5-581, DS6-581, DS7-581), where as the number of positive class observations were very small, the preci-

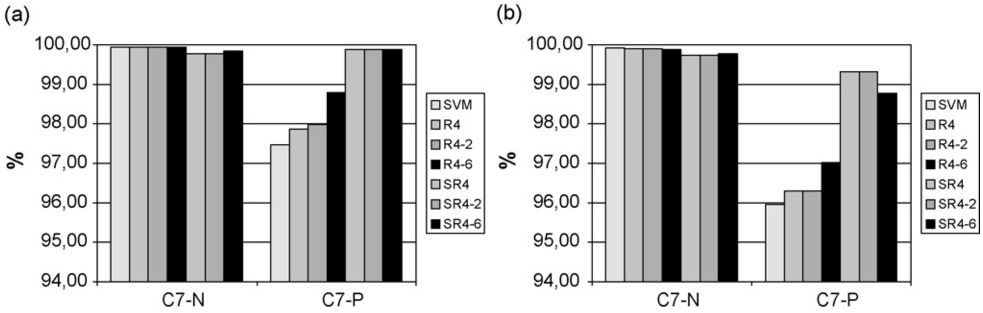

Fig. 12 – Comparison of true negative rate (C7-N), true positive rate (C7-P): (a) training set and (b) test set.

Table 6 – DSi-581: test sets precision

Data	SVM	Random subsets			Semi-random subsets		
		R4	R4-2	R4-6	SR4	SR4-2	SR4-6
DS1-581	96.07	96.10	96.07	95.98	–	–	–
DS2-581	95.35	95.70	95.78	96.57	–	–	–
DS3-581	94.65	94.52	94.47	94.84	88.39	88.41	91.56
DS4-581	91.70	91.04	89.71	89.05	82.35	82.61	84.57
DS5-581	92.38	90.68	90.10	89.85	78.73	78.54	82.21
DS6-581	92.15	91.77	92.11	90.55	83.35	83.12	87.24
DS7-581	97.67	97.19	97.05	96.84	93.18	93.14	94.10

sion slightly deteriorates. The results for semi-random data sets show that the precision obtained by a single SVM is significantly better than by distributed SVMs which is very significant for DS5-581 data set.

The comparison of *F*-measure for the larger data sets (DSi-581) used in the experiments is presented in Table 7. It can be seen that for the three-layered combinatorics SVM architecture (R4-6) with random training subsets the results were better than for single SVM for all data sets while the other two architectures (R4, R4-2) had slightly worse results for DS7-581 data set. In case of semi-random subsets, the values of *F*-measure were improved only for two imbalanced data sets (DS4-581 and DS5-581), where as the number of observations in positive class were very small, and were slightly worse for other three imbalanced data sets (DS3-581 and DS6-581, DS7-581).

5.4. *Discussion*

This study demonstrated the capability of support vector machines used as a single SVM and layered distributed SVM architectures to handle balanced and imbalanced data sets in the binary classification problem of forest cover type classes. A series of experiments with different number of observations, from small number to large number were conducted to demonstrate the link-up between the number of observations, the distribution of classes and the use of different SVM classifiers. By using different values of Gaussian kernel parameters and the number of training subsets the best SVM model was used in all experiments.

In order to illustrate the justification and advantages of distributed SVMs, the following findings could be highlighted:

- The distributed SVMs architectures with random defined training subsets produced better accuracy results for six data sets with larger number of observations (DS*i*-300, DS*i*-581, $i = 1, \ldots, 6$) and practically the same accuracy results for DS7-300, DS7-581. This was not always the case when using semi-random subsets where the results are slightly better or very similar for half of data sets when using single SVM.
- Most valuable information was the result which indicates very notable improvement of correct classifications of a small class in imbalanced data sets. The distributed SVMs improved the true positive rate for all data sets and slightly deteriorated the true negative rate.
- The most imbalanced data sets DS4-30, DS4-60, DS4-300, DS4-581, with the smallest number of observations (less than 1%) in positive class produced very similar or the same accuracy results for all SVM architectures and single SVM. There was also no difference of accuracy results for random subsets and semi-random subsets. But there was a significant improvement of true positive rate for all distributed SVM architectures with random training subsets and particularly for semi-random subsets.
- The last two measures, precision and *F*-measure were used to present the impact of the imbalance in the distribution of positive and negative class in the classification results. With the improvement of correct classification of positive class with less than 5% of observations the precision changed to the worse due to the ratio of wrong classifications of classes which increased in negative class more than decreased in positive class. The distributed SVMs architectures improved the *F*-measure for random and semi-random subsets.
- The number of training subsets used in this study was $M = 4$. Some additional tests, not presented in this paper,

Table 7 – DSi-581: *F*-measure of test sets

Data	SVM	Random subsets			Semi-random subsets		
		R4	R4-2	R4-6	SR4	SR4-2	SR4-6
DS1-581	95.42	95.68	95.72	96.15	–	–	–
DS2-581	95.80	95.94	95.97	96.45	–	–	–
DS3-581	94.63	94.83	94.76	95.44	93.30	93.00	94.42
DS4-581	88.05	90.04	89.38	90.48	89.11	89.26	89.91
DS5-581	86.04	86.69	86.65	89.71	86.64	86.57	88.75
DS6-581	90.31	91.16	91.41	91.88	90.22	90.02	92.14
DS7-581	96.80	96.74	96.67	96.93	96.15	96.13	96.38

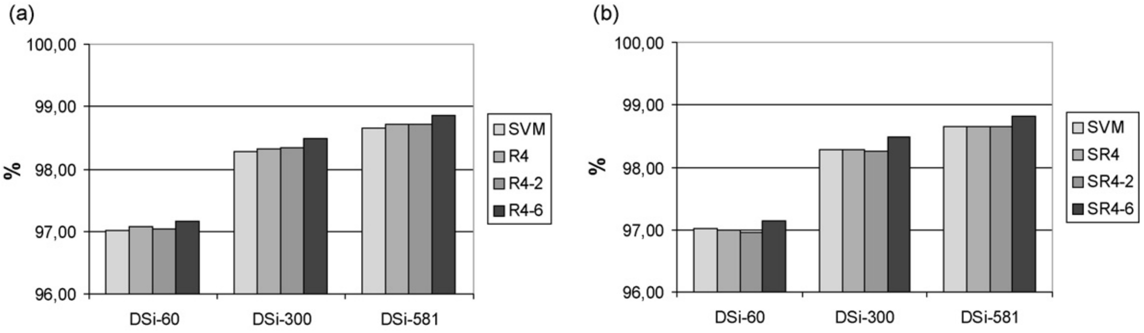

Fig. 13 – DS6-581: Average of accuracy for all forest cover types: (a) random subsets and (b) random and semi-random subsets.

showed that there was no improvement of the performance on other configurations of layered SVMs in comparison to presented and analyzed results. The larger number of training subsets implied the larger number of layers but there was also no improvement of performance.

Finally, an average of the individual accuracy measure for all cover types in three data sets DSi-60, DSi-300 and DSi-581, $i = 1, \ldots, 7$ is presented. The average accuracies of distributed SVM architectures with random training subsets shown in Fig. 13 a outperform the average accuracy of a single SVM. In Fig. 13 b is shown an average accuracy of two random training subsets DSi-60, DSi-300 and DSi-581, $i = 1, 2$ and semi-random subsets for the other data sets DSi-60, DSi-300 and DSi-581, $i = 3, \ldots, 7$. In this case, a single SVM performed slightly better in comparison to two-layered and three-layered binary cascade SVM architectures and slightly worse in comparison to the three-layered combinatorics SVM architecture.

6. Conclusion

This study of distributed support vector machines evaluated the classification accuracy of balanced and imbalanced data sets on forest cover type classes. For two balanced data sets, only the random training subsets were generated while for imbalanced data sets, both the random and the semi-random training subsets were generated. The experiments were performed on small and large data sets to show the performance of the classification of three distributed SVM architectures in comparison to the single SVM. In particular, the distributed SVM architectures were trained with $M = 4$ random subsets and they produced good classification results, with similar or even greater accuracies in case of large data sets.

The comparison, based on data partition into semi-random subsets, shows a great improvement in the classification true positive rate of the minority class. For a data set with smaller number of samples (DSi-60), used in the experiments, the semi-random training subsets produced very similar results in all experiments while with the larger number of samples in the data sets (DSi-300, DSi-581), the improvement of distributed SVM architectures is more evident for minority class observations while there is very small deterioration of true negative rate for the majority class observations.

Support vector machines based on distributed SVM architectures have been shown to be a valuable tool for the classification of large data sets, with balanced and specially imbalanced number of observations when the correct classification of minority class is required. With the parallel implementation of SVMs, the training time can be significantly decreased and is viable alternative to a single SVM training with comparable or even better classification performance of binary classifications. And finally, distributed SVMs architectures also improved an average of accuracy for all forest cover types classes.

REFERENCES

Batista, G., Prati, R.C., Monard, M.C., 2004. A study of the behaviour of several methods for balancing machine learning training data. Sigkdd Explorations 6 (1), 20–29.

Blackard, J.A., Dean, D.J., 1999. Comparative accuracies of artificial neural networks and discriminant analysis in predicting forest cover types from cartographic variables. Comput. Electron. Agric. 24, 131–151.

Chawla, N.V., Japkowicz, N., Kolcz, A., 2003. Editorial: special issue on learning from imbalanced data sets. Sigkdd Explorations 6 (1), 1–6.

Chen, X.-W., Gerlach, B., Casasent, D., 2005. Pruning support vectors for imbalanced data classification. In: IEEE Proceedings of International Joint Conference on Neural Networks, 1883–1888, Montreal, Canada.

Collobert, R., Bengio, S., Bengio, Y., 2002. A parallel mixture of SVMs for very large scale problem. Neural Comput. 14 (5), 2757–2767.

Graf, T.P., Cossato, E., Bottou, L., Durdanovic, I., Vapnik, V., 2005. Parallel support vector machines: the cascade SVM. In: Nineteenth Annual Conference on Neural Information Processing Systems, Vancouver.

Joachims, T., 2002. Learning to Classify Text using Support Vector machines Methods, Theory and Algorithms. Kluwer Academic Publishers, Boston, USA.

Joachims, T., 2004. SVMlight support vector machine. http://www.cs.cornell.edu/People/tj/svm-light.

Kecman, V., 2001. Learning and Soft Computing. MIT Press, London.

Kim, C.H., Pang, S., Je, H.-M., Kim, D., Bang, S.Y., 2003. Constructing support vector machine ensemble. Pattern Recognit. 36, 2757–2767.

Kotsiantis, S.B., Pintelas, P.E., 2003. Mixture of expert agents for handling imbalanced data sets. Ann. Mathemat. Comput. Teleinform. 1 (1), 46–55.

Liu, Y., An, A., Huang, X., 2006. Boosting prediction accuracy on imbalanced datasets with SVM ensembles. In: 10th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining, Lecture Notes in Computer Science. Springer, Singapore, pp. 107–118.

Platt, C.J., 1999. Fast training of support vector machines using sequential minimal optimization. http://research.micro-soft.com/ jplatt/abstracts/SMO.html.

Platt, C.J., Cristianini, N., 2000. Large Margin DAGs for Multiclass Classification. MIT Press, pp. 547–553.

Tax, D.M.J., Duin, R.P.W., 2002. Using two-class classifiers for multiclass classification, http://ict.ewi.tudelft.nl/ davidt/papers.

Trebar, M., Steele, N., 2007. An implementation of a two-layered SVM classifier in condor. Electrotech. Rev. 74 (3), 107–112.

UCI KDD archive, 2005. http://kdd.ics.uci.edu/databases/covertype/covertype.html.

Vapnik, V.N., 2000. The Nature of Statistical Learning Theory. Springer-Verlag, New York.