Journal of Data Science 12(2014), 727-754

Support Vector Machines Classification on Class Imbalanced Data: A Case Study with Real Medical Data

Krystallenia Drosou [1], Stelios Georgiou [2,3], Christos Koukouvinos[1*] and Stella Stylianou[2]

[1] *Department of Mathematics, National Technical University of Athens*
[2] *Department of Mathematics, University of the Aegean*
[3] *School of Mathematical and Geospatial Sciences, RMIT University*

Abstract: support vector machines (SVMs) constitute one of the most popular and powerful classification methods. However, SVMs can be limited in their performance on highly imbalanced datasets. A classifier which has been trained on an imbalanced dataset can produce a biased model towards the majority class and result in high misclassification rate for minority class. For many applications, especially for medical diagnosis, it is of high importance to accurately distinguish false negative from false positive results. The purpose of this study is to successfully evaluate the performance of a classifier, keeping the correct balance between sensitivity and specificity, in order to enable the success of trauma outcome prediction. We compare the standard (or classic) SVM (C SVM) with resampling methods and a cost sensitive method, called Two Cost SVM (TC SVM), which constitute widely accepted strategies for imbalanced datasets and the derived results were discussed in terms of the sensitivity analysis and receiver operating characteristic (ROC) curves.

Key words: class imbalance, support vector machines, cost sensitive learning, TC SVM, resampling methods, SMOTE SVM, medical data

1. Introduction and motivation

Support vector machines (SVMs), a powerful machine learning technique, were introduced by Vapnik (Vapnik (1995) and Cortes and Vapnik (1995), Burges (1998), Cristianinio and Shawe-Taylor (2000), Scholkopf and Smola (2001)) and successfully applied in various real-world problems, ranging from image retrieval (Tong and Chang (2001)) and handwriting recognition (Cortes (1995)) to face detection (Osuna et al. (1997)) and speaker identification (Schmidt, M.(1996)). SVMs have found popularity among machine learning researchers and statisticians due to its theoretical and practical advantages which justify its improved performance in binary classification scenario.

* Corresponding author.

However, standard SVMs, instead of their effectiveness in balanced datasets, could be proved inappropriate when they are faced with imbalanced data. The issue concerning imbalanced data is recognized as a crucial problem in machine learning community (Chawla, et al. (2004)). In these cases, classifiers tend to be overpowered by the majority class and ignore the minority examples assuming an equal misclassification error. Therefore, the produced models are, often, biased toward the majority class while having a low performance on the minority class. Furthermore, classifiers are typically designed to maximize the overall accuracy which is not an appropriate evaluation measure for imbalanced data. As a consequence, in order to handle imbalanced data we should both, consider improved algorithms and choose other metrics, such as Geometric mean and AUC to measure the performance, instead of accuracy. In parallel with, for many applications, especially for medical diagnosis where normal cases are the majority, it is more important the correct balance between sensitivity and specificity means since we have to accurately distinguish false negative results from false positives. Numerous recent works, including preprocessing and algorithmic methods have been proposed and dealt with the crucial problem of imbalanced data. These techniques can be sorted into two different categories: preprocessing the data by oversampling the minority instances or undersampling the majority instances and algorithmic methods including cost-sensitive learning (Batuvita and Palade (2013)). In our comparative study we use a cost sensitive learning technique proposed by Veropoulos et al. (1999) called "TC SVM" due to the fact that it uses two costs for the two different classes. In addition we applied two different forms of re-sampling methods, namely, random over-sampling and random under-sampling as well. Last but not least we present a combination of a widely used method called Synthetic Minority Oversampling Technique (SMOTE) proposed by Chawla et al.(2002) with random undersampling and the results were developed in the last section.

Parpoula et al. (2013) have already dealt with the analysis of a large dimensional Trauma dataset; however, their study lies on the comparison of several high-powered data mining techniques. The motivation of conducting the present study, applied in the medical dataset in question, is not only to enable the success trauma outcome prediction, improving the quality of the prediction model, but also to successfully evaluate the performance of a classifier faced with imbalanced data and keeping the correct balance between sensitivity and specificity. In this way, we compare the performance of the standard SVM with the TC SVM, random over-/under-sampling and a combination of SMOTE method with undersampling, and the derived results were discussed in terms of the sensitivity analysis. The merits of our comparative study through a real medical data set show the effectiveness of the considered approaches.

The rest of this paper is organized as follows. In Section 2, we present a theoretical background of the considered SVM classifiers. In Section 3, we present the SVM analysis and we carry out a comparative study for the considered methods in terms of accuracy, Geometric mean and the Area Under the Roc Curve (AUC). We also describe the performance criteria used for the evaluation of the employed methods. In conclusion, in Section 4, we summarize the results of our study and we highlight some concluding remarks. Note here that, we use classic and standard SVM with the same meaning as soft margin SVM. Moreover, we also use Gaussian or Radial or RBF kernel, consider exactly the same.

2. Theoretical background

In this section we briefly summarize the basic concept of the considered methods by providing a short but required theoretical background. Firstly, we discuss the main problem of soft margin SVM and then the modifications resulting in TC SVM method. Subsequently, we discuss the main concept of the pre-processing methods that we have applied in our analysis. Last subsection contains the metrics examined in our work.

2.1 Introduction to Support Vector Machines

SVM algorithm aims to find the optimal separating hyperplane which effectively separates the data points into the labeled classes. Let us consider that we have a binary classification problem. The data points are mapped into a high-dimensional feature space (Hilbert space) by a kernel function K (dot products between data points). For input points $\boldsymbol{x_i} \in R^p$ and label of the class of data y_i $(i = 1 \dots n)$, the decision function in the feature space can be considered as follows

$$\boldsymbol{f(x) = sign\left(\sum_{i=1}^{n} a_i y_i K\left(x, x_j\right) + b\right)} \tag{1}$$

where b is the model bias. Note that only those points which lie closest to the hyperplane have $a_i > 0$ and consist the support vectors. Let us assume the primal optimization problem in order to obtain the necessary parameters. The soft margin optimization problem (Cortes and Vapnik (1995)) can be formulated as:

$$\begin{aligned} &\text{min}imize \quad \tfrac{1}{2}\|w\|^2 + C\ \textstyle\sum_{i=1}^{n} \xi_i \\ &\boldsymbol{subject\ to} \qquad \boldsymbol{y_i\ (x_i \cdot w + b)\ - 1 + \xi_i \geq 0} \qquad \boldsymbol{\forall i\ with\ \xi_i \geq 0} \end{aligned} \tag{2}$$

where w is the weight vector normal to the hyperplane, ξ_i are the slack variables that hold for misclassification examples and, consequently, the term $\sum_{i=1}^{n} \xi_i$ can be considered as a measure of the amount of total misclassifications of the model (esp. the training errors). The trade-off between maximization of the margin and minimization of error is controlled by cost parameter C. The Lagrangian optimization problem of (2), used for finding the parameter b and coefficients a_i, has the following formulation:

$$\boldsymbol{maximize}\ \left[\sum_{i=1}^{n} a_i\ - \frac{1}{2}\sum_{i,j} a_i a_j\ y_i y_j K(x_i, x_j)\right] \tag{3}$$

$$subject\ to \quad 0\ \leq\ a_i\ \leq C\ \ \forall i, \qquad \sum_{i=1}^{n} a_i\ y_i\ =\ 0$$

which satisfy KKT conditions.

Note here that SVM address with the problem of moderately imbalanced data in more effective way, compared to other classifiers, due to the fact that SVM only takes into account those instances that are close to the boundary, means the support vectors, for building its model (for more details see Akbani et al. (2004)). More specifically, Akbani et al. (2004) have argued that due to the constraint $\sum_{i=1}^{n} a_i\, y_i = 0$, the coefficients a_i of each positive support vector are fewer than the negative support vectors, and as a result must be larger in magnitude than the a_i values correspond to the negative support vectors. The a_i in question, act as weights in the final classifier and consequently receive a higher weight than negative, something that counter-balance, in some extent, the effect of support vector imbalance.

2.2 Approaches for imbalanced data learning

2.2.1 Cost sensitivity SVM (TC SVM) for imbalanced data

As we can conclude from equation (2) the cost C given to positive and negative class is exactly the same. However, in case of imbalanced data, as we have already mentioned, the same cost could be result to a biased model toward the majority class and as a consequence could provide suboptimal results. Veropoulos et al. (1999) proposed a cost sensitive method (Two-cost method) to deal with the above problem revealed in SVMs. They generalize the soft margin approach so that the formulation of the Lagrangian contains two misclassification costs, one for each class examples. More specifically the reformulation of the optimization problem having two errors given as follows:

$$minimize\ \{\frac{1}{2}\|\boldsymbol{w}\|^2 + C^+ \sum_{\{i|\, y_i=+1\}}^{n} \xi_i + C^- \sum_{\{i|\, y_i=-1\}}^{n} \xi_i$$

$$subject\ to \qquad [\, y_i\, (\boldsymbol{x_i} \cdot \boldsymbol{w} + b) - 1 + \xi_i\,] \geq 0\, ,\ \xi_i \geq 0, i = 1, \dots, n$$

And the Lagrangian takes the following form

$$L_P \equiv \frac{1}{2}\|\boldsymbol{w}\|^2 + C^+ \sum_{\{i|\, y_i=+1\}}^{n} \xi_i + C^- \sum_{\{i|\, y_i=-1\}}^{n} \xi_i$$

$$- \sum_{i=1}^{n} a_i\, [\, y_i\, (\boldsymbol{x_i} \cdot \boldsymbol{w} + b) - 1 + \xi_i\,] - \sum_{i=1}^{n} \mu_i\, \xi_i\, ,$$

where $\mu_i \geq 0$ and $\alpha_i \geq 0$. The dual formulation gives the Lagrangian

$$L_D \equiv \sum_{i=1}^{L} a_i \; - \frac{1}{2} \sum_{i\,,j} a_i a_j \; y_i y_j \; K(x_i, x_j)$$

subject to $0 \; \leq \; a_i \; \leq C^+$, if $\; y_i = +1$ and $0 \leq \; a_i \; \leq C^-, \;\; if \; y_i = -1$.

This dual optimization problem can be solved in just the same way as solving the standard SVM optimization problem. Good results can be obtained, as indicated in Akbani et al. (2004), by setting the ratio C^+/C^- equal to the minority to majority class ratio.

2.2.2 Sampling methods

Data preprocessing methods can be used to balance the datasets before training SVM models. In data level, there are methods for balancing the classes consist of resampling the original data set either by over-sampling the minority or by under-sampling the majority class, until when there is a balance ratio between the two classes. Apart from random over-/under-sampling there are synthetic generation methods like SMOTE (Chawla et al. (2002)) or like ROSE (Menardi and Torelli (2013)). Resampling methods have been addressed to train SVM models with imbalanced data in many different fields (see for example Akbani et al. (2004), Yuan et al. (2006), Batuwita and Palade (2009), Batuwita and Palade (2009)). However, such methods have revealed significant disadvantages. On the one hand, under-sampling may throw out useful information acquired from data, while over-sampling increase the computational burden since it increases the size of the data.

Random Sampling SVM

Random over-sampling constitutes the simplest method that increases the minority class examples. It randomly replicates existing instances in the minority class so that it balances the class distribution. Random over-sampling doesn't put additional information but it increases the weight of minority examples by replication. However, there is a problem that has been generally occurred, that is the over-fitting problem. As a consequence, even though we have high accuracy in training set, the classification performance of test set will likely be worse. Chawla et al. (2002) proposed Synthetic Minority Over-sampling Technique (SMOTE) in order to avoid over-fitting problem in random over-sampling. SMOTE method generates synthetic data based on the feature space similarities between minority instances. These examples will be generated by using the information from the k-nearest neighbours of each instance of the minority class. More precisely, this method finds the k-nearest neighbours of each minority example , randomly selects one of them, and multiplies the corresponding feature vector difference with a randomly taken number between 0 and 1 so as to produce a new minority example in the neighborhood. It should be mentioned that SMOTE not only avoids over-fitting,

but it also causes the decision boundaries for the minority class to move towards the majority class.

Random under-sampling, contrary to oversampling, removes randomly majority instances keeping all examples of minority class. The training process becomes faster since many majority examples are ignored. However, the main disadvantage of random under-sampling is that potentially useful data are lost. There are some heuristic under-sampling methods which try to remove superfluous instances which will not affect the classification accuracy of the training set (Hart (1968)).

Undersampling and SMOTE Combination

SMOTE (Chawla et. al. 2002) as we have already mentioned is a well-known algorithm to fight the unbalanced problem to many learning algorithms. The general idea of this method is to artificially generate new examples of the minority class using the nearest neighbors of these cases. In the present modification, we simultaneously under-sample the majority class examples, leading to a more balanced dataset and avoiding over-fitting.

In conclusion it should be noted that when focusing on approaches at the data level (means rebalancing the data distribution), there are two important problems associated with a SVM classifier. The first one is that over-sampling methods significantly increase the dataset size leading to bigger computational time and overfitting of data. Secondly an optimal ratio of class distribution is empirically determined by grid search procedures.

2.3 Metrics for evaluating model performance

Traditionally, the performance of a binary classifier is accomplished by using metrics derived from the confusion matrix (Table 1). More precisely, given a classifier and a record, there are four possible scenarios:

Table 1: Confusion Matrix

		Predicted	
		Positives	Negatives
Real	Positives	TP	FN (Type II error)
	Negatives	FP (Type I error)	TN

True Positives (TP) where positive records are correctly predicted as positive, False Negatives (FN) where positive records are incorrectly identified as negative, False Positives (FP) where negative records are classified as positive ones, and finally True Negatives (TN) where negative records are correctly identified as negative. Using a two-by-two confusion matrix we can easily represent these possible outcomes and compute the measures are followed.

Accuracy is the most common measure used for quantify the performance of a classifier. Despite the efficacy of accuracy measure on balanced data sets using standard SVM, overall accuracy in case of imbalanced data, constitutes an inappropriate metric. For instance, a classifier that predicts all samples as negative has high accuracy (4) but it cannot detect rare positive samples.

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP+TN}}{\boldsymbol{TP}+\mathbf{FP}+\mathbf{FN}+\mathbf{TN}} \tag{4}$$

Consequently, the performance of such systems, in order to get optimal balance classification ability, described effectively in terms of sensitivity (or true positive rate or positive class accuracy) and specificity (or true negative rate or negative class accuracy)

$$\boldsymbol{Sensitivity} = \frac{\boldsymbol{TP}}{\boldsymbol{TP+FN}} = \mathbf{1} - \mathbf{Type\ II\ error} \tag{5}$$

$$\boldsymbol{Specificity} = \frac{\boldsymbol{TN}}{\boldsymbol{TN+FP}} = \mathbf{1} - \mathbf{Type\ I\ error} \tag{6}$$

More precisely, sensitivity measures the proportion of actual positives that are correctly identified as such, meaning that it measures the percentage of people who are having the disease and they are correctly identified as having the disease. The specificity measures the proportion of actual negatives which identified correctly meaning that it measures the percentage of people who are not having the disease and they are correctly identified as healthy. As far as the Type error I as concerned, it occurs when the null hypothesis is true, but it is rejected. In medical diagnosis an example of type I error includes a test that indicates a patient to have a disease when in fact the patient does not have the disease. A typical example of medical experiments regarding Type II error would be a failure to detect the disease in a patient who really has the disease. It should be noted that a test with high sensitivity has low type II error and a test with high specificity has low type I error.

Kubat and Matwin (1997) based on these two measures proposed Geometric mean, a geometric mean of sensitivity and specificity

$$\boldsymbol{Geometric\ mean} = \sqrt{\boldsymbol{Sensitivity} * \boldsymbol{Specificity}} \tag{7}$$

Moreover, Receiver Operating Characteristic (ROC) curves are another way besides confusion matrices to examine the performance of a classifier in a much more intuitive and robust way. A ROC curve (Pepe (2000)) is used to evaluate the performance of a system with dichotomous outcome. The trade-off between sensitivity and specificity can be represented graphically as a ROC curve. The Area Under the Curve (AUC) can indicate balance classification ability between sensitivity and specificity as a function of varying a classification threshold. For more details we refer to Swets and Pickett (1982). Consequently, in order to handle imbalanced data we should consider other measures, such as Geometric mean and AUC.

3. Application – Comparative results

In this section we compare the performance of the two different methods, SVM and Two-cost SVM random sampling (random oversampling and random undersampling), a combination of SMOTE and random undersampling as well as a new proposed method called ROSE on a large dimensional Trauma data set consisting of $N = 8862$ patients and 41 factors that include demographic, transport and intrahospital data. The main aim is to provide an unbiased estimation of each model's discrimination. In this way the values of performance criteria are calculated from a data set which is not used in the model building process, constitute a portion of the original data set and called test set. A classifier should present high values of accuracy, sensitivity, specificity, AUROC and geometric mean and the model's generalization performance is often estimated by the holdout validation. In our study we deal with a large data set that is split randomly into a training set, containing 75% of cases (6647) and the test set, containing 25% of cases (2215) in order to evaluate the performance of classifiers on new data. Our medical dataset is highly imbalanced since it consists of 446 positive instances (majority class) and 8416 instances of negative instances (minority class). This makes imperative both the use of pre-processing methods to balance the dataset and cost sensitive learning methods that give another weights into the two different classes. In addition the use of more robust measures than accuracy, like Geometric mean and AUC will provide more reliable conclusions. Our motivation for conducting this study comes from medical decision support something indicates that the choice of a medical data set was imperative. For each patient the target attribute, variable y is binary and denotes the probability of death. Specifically variable y, expressed in the form of two categories -1 and 1, where -1 represent the survival, while the value of 1 the death. According to medical advices, all the prognostic factors should be treated equally during the statistical analysis and there is no factor that should be always maintained in the model. The names of these factors are included in the Appendix Section. The analysis, which contains all steps of data pre-processing and model development, was carried out using R codes and the algorithms were implemented using simultaneously the packages 'e1071' and 'DMwR'.

3.1 Standard SVM

For a standard SVM classifier we should determine not only the kernel function but also the regularization parameter C the value of gamma in case of a Gaussian (RBF) kernel and the degrees of freedom in case of polynomial kernel. The issue of model selection in support vector machine is vital and influence the overall performance of the classifier, making SVM quite sensitive to the selection of these parameters.

Applying a 10-fold cross validation we obtain the cost value for the best performance in terms of error rate, equal to 2. Figure 1 illustrates the changes in classification error for different values of cost parameter in case of a standard linear SVM. Besides the cost parameter, the intrinsic parameters of SVM classifier greatly affect its performance. For a Gaussian (RBF)

basis kernel apart from the regularization parameter C, the value of gamma should be selected from several candidates.

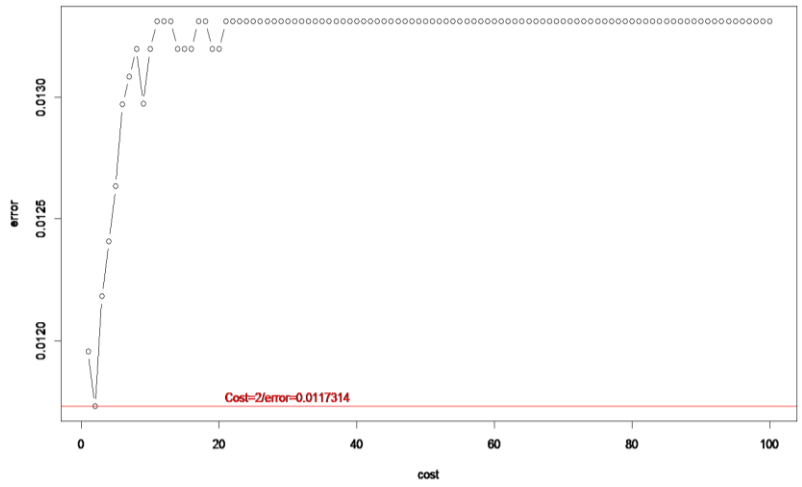

Figure 1: Performance of SVM with a linear kernel for different values of cost parameter. Red line shows the cost with the best performance in terms of error rate.

The gamma value should normally be between 1/k (=0.0244) and 6/k (=0.14634), where k represents the data dimension (41 in our study). Performing a grid search we chose the one that result in the best performance. Figure 2 displays the difference in error, changing the gamma parameter. The optimal value of gamma (=0.03125) showed in the following figure (red line), gives the smallest error rate. We performed a selection of gamma parameter in SVM and in Table 2 are illustrated some selected values.

Table 2: Model selection (some selected values) for gamma parameter in SVM with a Gaussian (RBF) kernel

gamma	Error	dispersion
0.03125	0.01275349	0.003503342
0.06250	0.01918199	0.005812675
0.12500	0.03666481	0.008780307
0.25000	0.04874225	0.007506852
0.50000	0.05032355	0.007295674
1.00000	0.05032355	0.007295674
2.00000	0.05032355	0.007295674

The estimated measures in Table 3 are obtained using $C = 2$ for the linear kernel, $C = 1$ for the sigmoid, polynomial and Gaussian kernel and $gamma = 0.03125$ for the Gaussian kernel. If the kernel type is set to polynomial or sigmoid the parameter bias sets the offset parameter in the kernel function and the default value 0 is suitable in most cases. Only if kernel type is set to polynomial the parameter degree is enabled and is set to be equal to 3.

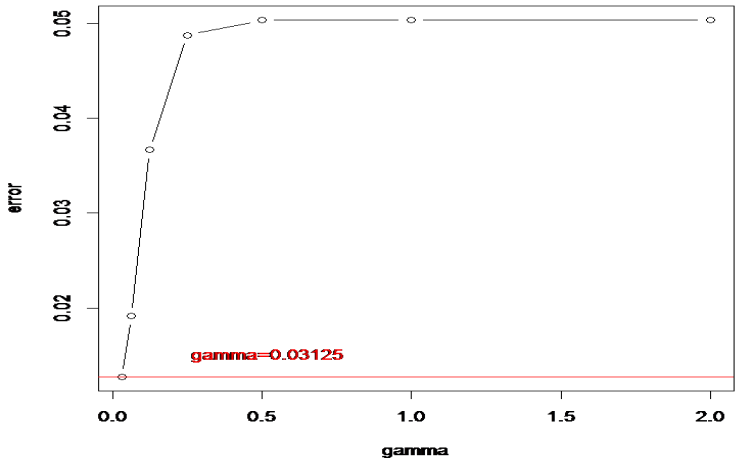

Figure 2: Performance of SVM with a Gaussian (RBF) kernel for different values of gamma parameter

Table 3 shows the performance of SVM using different kernels. Both SVM with a linear and SVM with a Gaussian kernel have the highest classification accuracy, sensitivity, specificity AUC and Geometric mean. Gaussian kernel reaches the percentage of 0.9848, 0.77922, 0.99821, 0.8866 and 0.8796 for accuracy, sensitivity, specificity AUC and Geometric mean respectively. Almost similar results were given for the linear kernel. The second best results were taken using a Sigmoid kernel regarding accuracy measure. However Sigmoid has the worst performance assuming the results for the most robust metrics as AUC and Geometric mean.

Table 3: Comparison of standard SVM performance for different kernels on Trauma dataset

Kernel	Accuracy		Sensitivity		Specificity		AUC		Geometric mean	
	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Linear	0.9929	0.9875	0.86986	0.84416	0.99929	0.99499	0.9204	0.9037	0.9171	0.8991
Gaussian	0.9924	0.9848	0.84932	0.77922	0.99982	0.99821	0.9474	0.8866	0.9459	0.8796
Polynomial	0.9748	0.9702	0.52397	0.50000	0.99822	0.99606	0.9703	0.9043	0.9698	0.8996
Sigmoid	0.98	0.977	0.75685	0.78571	0.99163	0.98282	0.805	0.832	0.7848	0.8183

It should be mentioned that there are an overfitting of data, especially in case of non-linear kernels considering the above measures. In parallel with, conducting SVM classification without selective sampling, we observed that the g-mean values are consistently low.

3.2 Approaches for imbalanced data learning

Two-cost SVM

Applying Two-cost SVM one should determine two cost, as concluded from the aforementioned theory in the previous section. For achieving expected classification results, the misclassification costs play a crucial role in the construction of a cost sensitive learning model. We discover the optimal parameters based on different evaluation functions such as Geometric mean and AUC. For our Trauma dataset the minority class consists of positive instances and the majority class consists of negative instances. The two cost parameters are the minority cost (C^+) referred to positive instances and the majority cost (C^-) referred to the negative instances. We can reduce the effects of class imbalance by assigning a higher classification cost for the minority class examples than the majority class examples. Veropoulos et al. (1999) and Akbani et al. (2004) suggested the inverse ratio between the two class sizes as a good choice that improves the performance of the TC SVM method. After performing a search among different values for the two costs we confirm the mentioned result.

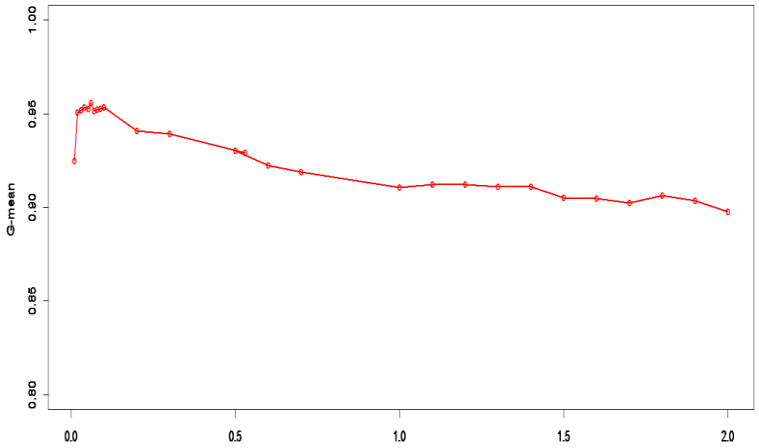

Figure 3: Geometric mean (y-axis) measure changing the cost of majority class (x-axis)

The ratio between the minority and majority class for trauma dataset is equal to 0.05299. More specifically, by setting the cost of the minority class equal to 1 and changing the cost of majority class we performed a search among many values. We execute the analysis for values varied from 0.01 to 2.0. The most accurate results in terms of Geometric mean measure were given for values 0.04, 0.0529(=ratio) and 0.06 of the majority cost as concluded from Figure 4. The best performance gives the value 0.06. However, the two other values gave almost similarly results. We finally chose the inverse ratio between the two classes, setting the ratio equal to the minority to majority class ratio ($C^- = C^+ * 0.05299$).

Figure 4 illustrates in separate graphs the performance for accuracy, sensitivity and specificity, changing the cost of majority class. Dashed grey line shows the value of each measure in case of standard SVM.

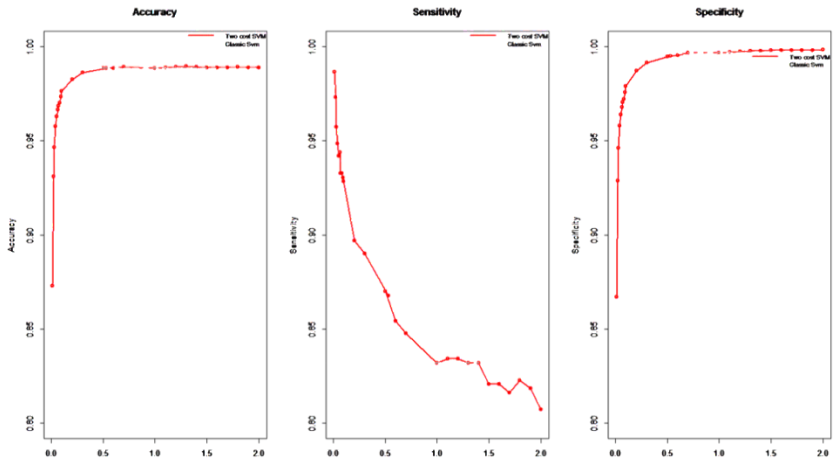

Figure 4: Performance in terms of the three measures (Accuracy, Sensitivity, Specificity) changing the cost of majority class (x-axis) (solid red line: Two-cost SVM; dashed line: Classic SVM)

In Figure 5 we consider the comparisons mentioned below but in the same graph. The vertical grey line indicates the cost of majority class when it was set to be equal to the ratio of the two classes.

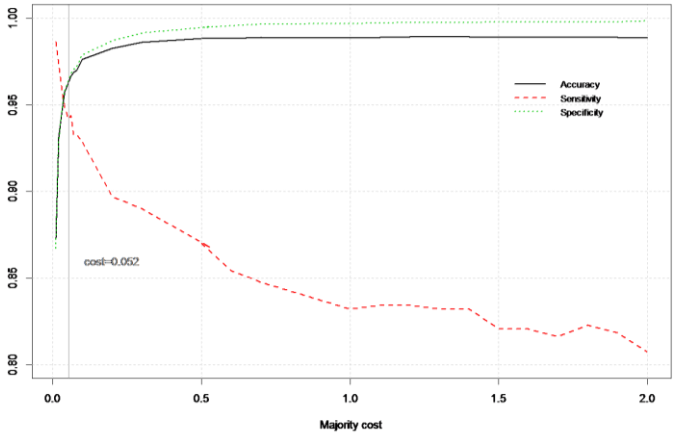

Figure 5: Comparison performance of accuracy, sensitivity and specificity majority class (x-axis) (solid black line: Accuracy; dashed red line: Sensitivity; dashed green line: Specificity). Vertical line indicates the cost of majority class when it was set to be equal to the ratio of the two classes.

As we can conclude from Figure 6, sensitivity was continually increasing as support vectors were increasing. In contrast, accuracy and specificity gathered higher values for fewer support vectors. Note here that increasing the majority cost we have fewer support vectors, as well.

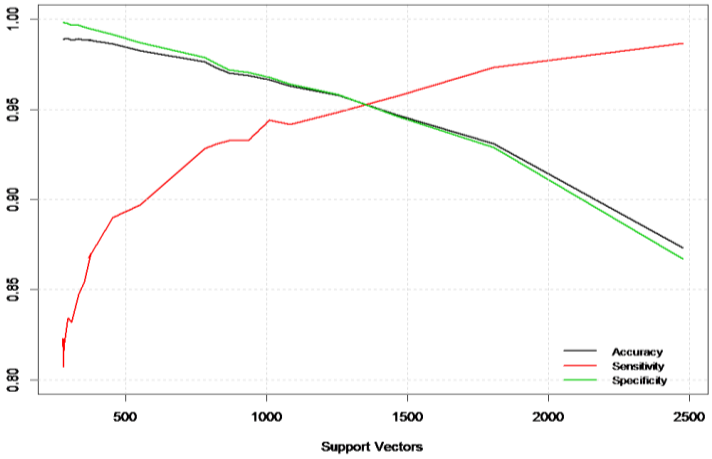

Figure 6: Three different measures (y-axis) versus support vectors (x-axis) in two-cost SVM.

Some Comparisons among C and TC SVM

Some comparisons between SVM and TC SVM are contained in order to obtain the importance of the applied methodology. First of all, we present the performance for the linear case and the other kernels are followed after we had chosen the best parameters. Table 4 shows the acquired results where SVM gathers higher accuracy in both train and test set.

Table 4: Performance comparison for standard SVM and TC svm with linear kernel

Linear	Accuracy		Sensitivity		Specificity		AUC		Geometric mean	
SVM	Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
C	0.9929	0.9875	0.86986	0.84416	0.99929	0.99536	0.9346	0.9198	0.9323	0.9166
TC	0.9717	0.9648	0.94863	0.93506	0.97293	0.96643	0.9608	0.9507	0.9607	0.9506

Comparing standard and TC SVM, the first one has higher specificity which means that the classifier recognizes more actual negatives; in other words this means that using TC SVM we obtain lower Type I error rate. This measure alone does not tell us how well the classifier recognizes positive cases and so it is necessary to take into consideration both sensitivity and

specificity of the used classifiers. When the two algorithms are evaluated against the sensitivity, TC SVM has clear advantage having highest percentage, which means that the Type II error rates are lower than the one of C SVM (classic or standard SVM).

Figure 8 displays the ROC curves derived from the two considered methods. The further the curve lays above the reference line, the more accurate the test. The AUROC achieved the value of 0.9198 for linear C SVM and higher value for TC SVM equals to 0.9507. Not only in terms of AUC but also of Geometric mean the cost sensitive method outweighs the standard SVM.

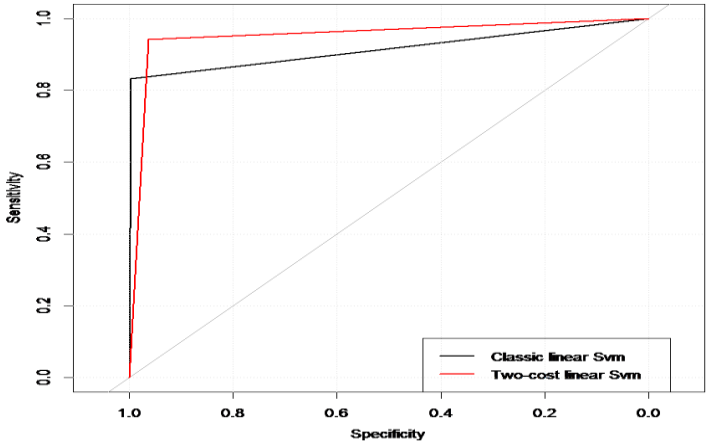

Figure 7: Roc curves comparison for linear case

Table 5 describes the performance for the standard SVM and TC SVM for the nonlinear case. The best measure in Geometric mean was gathered for TC SVM using Gaussian with a radial basis kernel. Comparative results are taken using a sigmoid kernel for all the considered metrics, achieving the ratio of 95.20% for Geometric mean for the TC method.

Table 5: Performance comparison for the two different SVM techniques with different kernels (non-linear case)

Kernel	SVM	Accuracy		Sensitivity		Specificity		AUC		Geometric mean	
		Train	Test	Train	Test	Train	Test	Train	Test	Train	Test
Gaussian	C	0.9924	0.9848	0.84932	0.77922	0.99982	0.99607	0.9246	0.8876	0.9215	0.8809
	TC	0.9655	0.9563	0.93493	0.94805	0.96706	0.95679	0.951	0.9524	0.9509	0.9524
Polynomial	C	0.9748	0.9702	0.52397	0.50000	0.99822	0.99607	0.7611	0.748	0.7232	0.7057
	TC	0.9749	0.9692	0.70890	0.71429	0.98878	0.98321	0.8488	0.8488	0.8372	0.8380
Sigmoid	C	0.98	0.977	0.75685	0.78571	0.99163	0.98750	0.8742	0.8866	0.8663	0.8808
	TC	0.969	0.9631	0.91096	0.94156	0.97204	0.96429	0.9415	0.9520	0.9410	0.9520

In the above Table, C is an abbreviation for classic or standard SVM and TC for TC SVM

It should be noted that using the cost sensitive learning method it reduces the problem of the overfitting. Almost similar results were given considering the AUC metric instead of Geometric mean measure. Gaussian kernel has clearly the highest Geometric mean and AUC compared to all non-linear kernels considering TC SVM whereas polynomial kernel has the lowest. The difference between the two kernels, Gaussian and sigmoid, is so small that both achieve good results for all measures. Furthermore, cost- sensitive SVM performs well for the linear case. In accordance with the AUC measure, polynomial kernel has the worst results. Figure 7 displays a comparison in respect to the Geometric mean confirming the above conclusions. Comparing TC with C SVM for a Gaussian kernel, it can be inferred that the first method outperforms the second one in terms of geometric mean and AUC. Unlike, as far polynomial kernel as concerned, the difference between the two compared methods is considerably higher than the previous presented kernels.

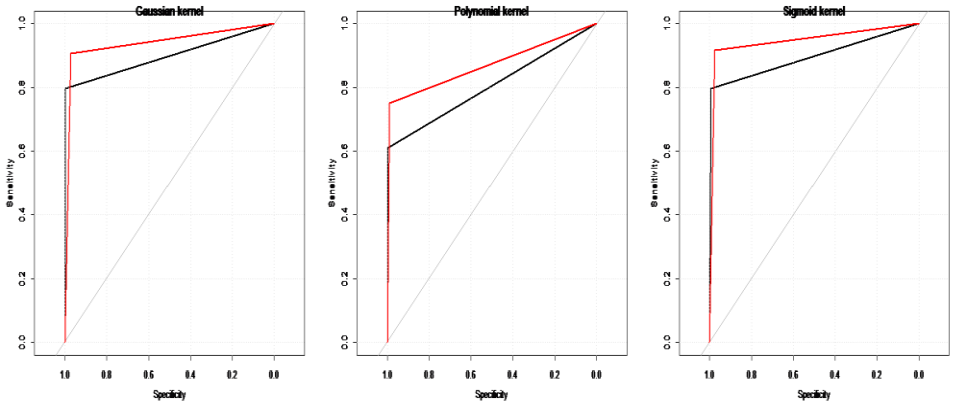

Figure 8: Roc curves comparison for non-linear case in Test set. Red curves represent TC SVM and black curves Classic SVM.

Figure 8 displays the ROC curves derived from all SVMs with the three non-linear kernels. For the ROC curves in Figure 8, regarding the Gaussian kernel, the TC method performs better on the average compared with the other kernels though the difference between sigmoid kernel is not statistically significant. As we can infer from Figure 8, Polynomial kernel shows the worst performance.

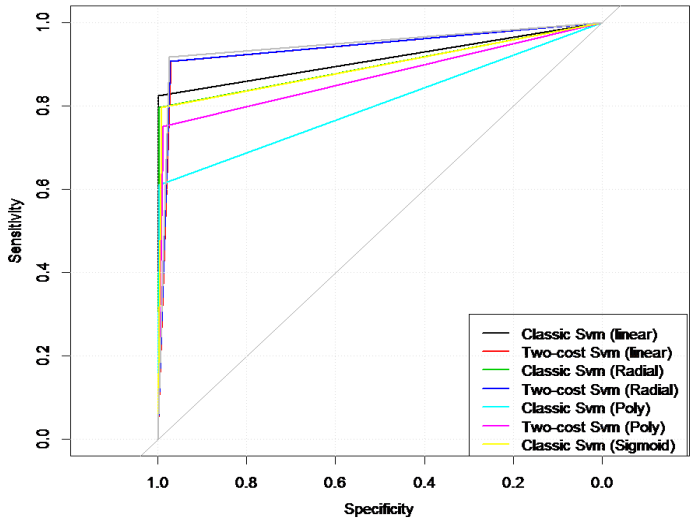

Figure 9: ROC curves derived from all kernels using both methods

Figure 9 illustrates the performance of these two methods on Trauma dataset for all the examined kernels. In addition, it ranks the best candidate models according to the AUC criterion and helps the experimenter to choose the best approach for a given analysis. The highest AUC was obtained for the TC method with a Gaussian kernel (AUC=0.9524) and the second highest was marked for both TC method using a linear kernel (AUC=0.9507) and Sigmoid kernel (AUC=0.9520) with the second slightly outperforms the first one. Almost similar results were showed for the standard linear SVM (AUC=0.9198) and standard SVM with a Gaussian kernel (AUC=0.8876). The AUROC for the Polynomial kernel revealed the lowest value equal to 0.748, 0.8488 for standard and TC SVM respectively. In Figure 9 we mean Linear kernel with linear, Gaussian kernel with Radial, Polynomial kernel with abbreviation Poly and sigmoid kernel with Sigmoid.

Resampling Methods

Random Sampling SVM

Random Over-sampling (SVM-RO)

Learning with over-sampled training sets was repeated 20 times for each size of the increased training sets. Then we chose the increased training set that produced the maximum gmean value for the original training set.

Random Under-sampling (SVM-RU)

We also conducted random under-sampling of the majority instances. In the same way as oversampling, learning with under-sampled training sets was repeated 20 times for each size of the reduced training set. Then we chose the reduced training set that produced the maximum gmean value for the original training set.

SMOTE-SVM and undersampling combination

As far as SMOTE algorithm as concerned, for the calculation of K-nearest neighbors, K was set to 5. Learning was performed using 20 independent synthetically enhanced datasets and then in order to identify the best synthetic sample size we calculate the maximum Geometric mean. In example, an increment of 300% is selected if the maximum average gmean of the original training set appears when 300% of new synthetic instances are added into the training dataset. While increasing minority instances gradually and simultaneously reduced the majority class examples, we observed for each combination Geometric mean values of the original training sets for each experimental dataset. Using SVM-SMOTE, the number of synthetic instances to achieve the desired class balance is unknown and empirical studies must be performed.

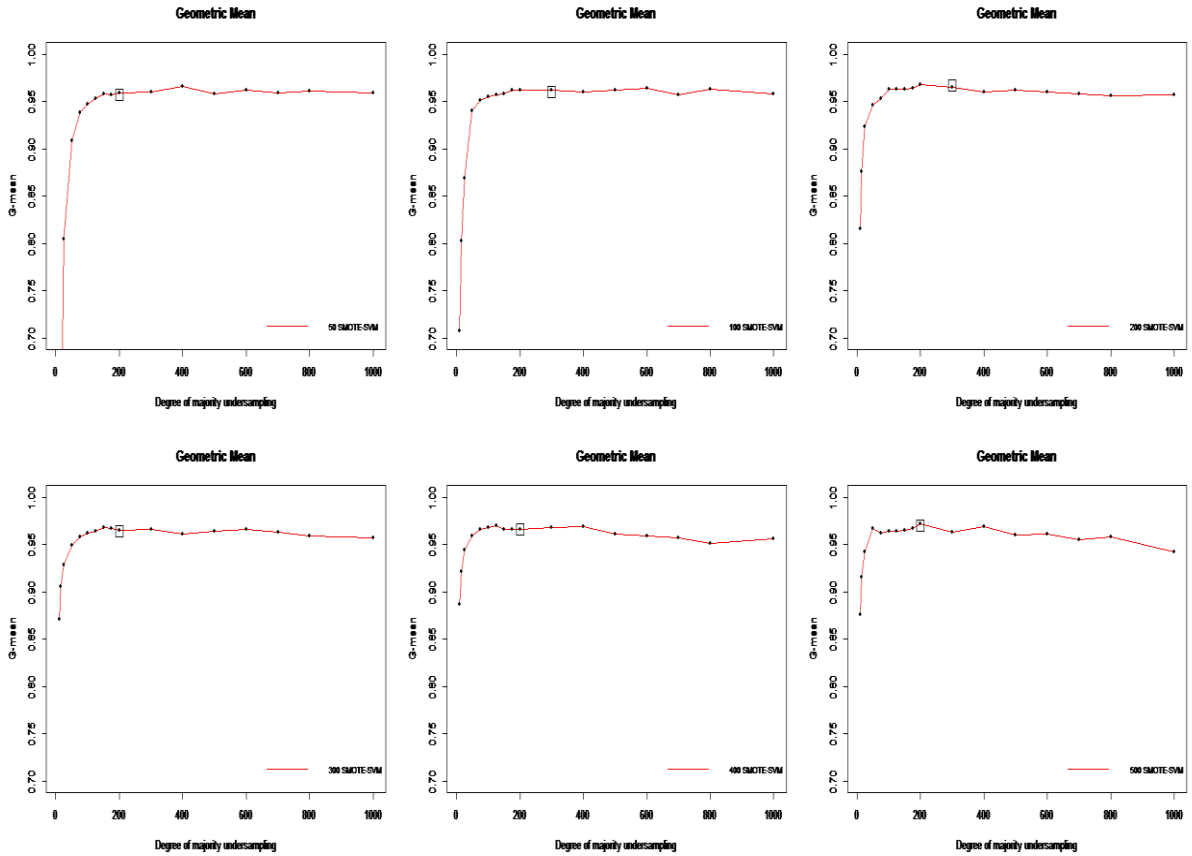

Figure 10: Geometric mean values of the training dataset in terms of increase of synthetic minority instances by SMOTE

Minority class was over-sampled at 50%, 100%, 200%, 300%, 400%, 500% and the majority class was under-sampled 10%, 15%, 25%, 50%, 75%, 100%, 125%, 150%, 175%, 200%, 300%, 400%, 500%, 600%, 700%, 800%, 1000%, 2000% as presented in Table 6. We chose the aforementioned rates according to Chawla et al. (2002). Due to its performance, a Gaussian kernel function with a radial basis $k(x, x') = \exp(-\|x - x'\|^2/\sigma^2)$ is used for SVM classification. We remind that it presents the best performance among all the kernels (linear and nonlinear) in our data set.

Table 6: Grid search for different combinations of SMOTE SVM and random undersampling

Gaussian Kernel	Geometric mean					
Under-sampling %	50 SMOTE	100 SMOTE	200 SMOTE	300 SMOTE	400 SMOTE	500 SMOTE
10%	0.2300692	0.7077196	0.8152652	0.8712314	0.8864990	0.8759041
15%	0.5211030	0.8021911	0.8761406	0.9051832	0.9212721	0.9150150
25%	0.8045620	0.8689910	0.9230279	0.9287044	0.9446142	0.9417371
50%	0.9083428	0.9401976	0.9461497	0.9487433	0.9593880	0.9665661
75%	0.9384779	0.9508191	0.9534347	0.9584441	0.9654686	0.9614618
100%	0.9469343	0.9553015	0.9626488	0.9622824	0.9677662	0.9641782
125%	0.9531918	0.9567008	0.9631719	0.9644199	0.9695098	0.9637715
150%	0.9580112	0.9583828	0.9627070	**0.9677358**	0.9661614	0.9649674
175%	0.9566272	0.9617051	0.9641814	0.9665047	0.9655365	0.9669310
200%	0.9588921	0.9623313	**0.9675190**	0.9645545	0.9657113	**0.9718652**
300%	0.9603146	0.9618026	0.9645217	0.9662298	0.9678828	0.9633213
400%	**0.9661793**	0.9602417	0.9596261	0.9608790	**0.9689449**	0.9684944
500%	0.9578417	0.9619651	0.9621467	0.9635404	0.9613943	0.9604165
600%	0.9619149	**0.9634723**	0.9597816	0.9662038	0.9592851	0.9605706
700%	0.9589905	0.9568111	0.9575871	0.9631608	0.9572629	0.9554419
800%	0.9613749	0.9632656	0.9561161	0.9594724	0.9510912	0.9575539
1000%	0.9593712	0.9578405	0.9574465	0.9569860	0.9560181	0.9421841
2000%	0.9457649	0.9434341	0.9355585	0.9301694	0.9403617	0.9284354

From SVM-SMOTE, the maximum Geometric mean was found for increments of 500% with the combination of 200% undersampling ratio, achieving the value of 97.18652% for the Geometric mean.

Table 7: Comparison of % Minority correct for different undersampling ratio changing the Over-sampling rate

Gaussian Kernel	% Minority correct			
Smote	200% under-sampling	300% under-sampling	400% under-sampling	500% under-sampling
50%	**0.9566563**	0.9487952	0.9554896	0.9444444
100%	0.9534161	0.9386503	0.9375000	0.9266055
200%	0.9509202	0.9494048	0.9335260	0.9216301
300%	0.9495549	0.9427711	0.9369369	0.9230769
400%	0.9549550	0.9501466	0.9341317	0.9120235
500%	0.9489489	0.9316770	0.9221557	0.9221557

Table 7 shows a search among different ratios of under-sampling for 50%, 100%, 200%, 300%, 400%, 500% SMOTE-SVM respectively. Figure 11 shows the percentage of minority correct values of the original training sets as instances added by SMOTE with 4 different under-sampling ratios. The highest value revealed with the combination of 50-SMOTE SVM and 200% under-sampling.

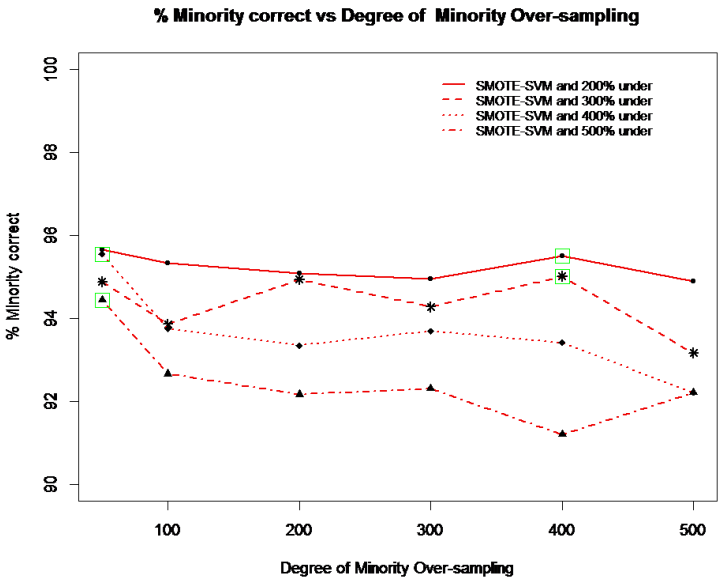

Figure 11: A graphical presentation of % Minority correct for different undersampling ratio changing the Over-sampling rate

3.3 Experimental Results and Discussion

The Geometric mean values for the original training set using the 5 different methods are shown in the below table. Comparison of differences for all pairs of methods illustrated that SMOTE-SVM and oversampling have the best performance in test set.

Table 8: Geometric mean of Training sets obtained from 4 different methods

Gmean of training set in original data		
Linear kernel	Train	Test in original train data
C SVM	0.9192	0.9248457
TC SVM	0.98138	0.9699514
SVM-RU	0.9824	0.9579060
SVM-RO	0.9792	0.9789352
SMOTE	0.9681217	0.9687785

However using random-oversampling there is a problem of overfitting of data something that it is more likely to happen using nonlinear kernels. For this reason, SMOTE SVM seems to have the best performance. SMOTE SVM slightly outperformed SVM-RU and the biased

Penalty method (TS SVM). Comparing the three non-linear kernels for the considered methods we observe that the Gaussian kernel reveals the highest performance for the Geometric mean metric. As far as the methods as concerned SMOTE SVM has the highest performance compared to C SVM, TC SVM and SVM-RU. In case of SVM-RO did not show a significant mean difference for the gaussian and sigmoid kernel, whereas SVM-RO outperforms SMOTE SVM in case of polynomial kernel.

Table 9: Geometric mean of Training sets obtained from 4 different methods (nonlinear case)

	Gmean of training set in original data		
	Gaussian kernel	Polynomial	Sigmoid
C SVM	0.8904698	0.7866470	0.8596290
TC SVM	0.9567920	0.8732783	0.9561292
SVM-RU	0.9562475	0.8782818	0.9479898
SVM-RO	0.9840769	0.9529313	0.9568208
SMOTE SVM	0.9718652	0.880000	0.9606813

Despite the high values of Geometric mean that creates, oversampling leads to a large increase on the training set something that not only increases the computational burden of the learning algorithm but it also leads to overfitting problems. SMOTE SVM in combination with undersampling doesn't reveal overfitting and simultaneously keeps smaller data set compared to oversampling method. Consequently, as we anticipated, the randomness of under-sampling did not produce a consistent result as compared with SMOTE. As we can conclude, oversampling increases to a large extent the training size giving a computational burden to the SVM algorithm which applied in the second step.

4. Concluding Remarks

The main aim of this study was to develop a model that will enable successful prediction of the trauma outcome. Many strategies have been proposed dealing with imbalanced data, some of which have been applied in the present analysis. At data level, sampling is the most common approach, with oversampling outperforms the random undersampling. At the algorithmic level, solutions using adjusting costs have been widely proposed. An alternative cost sensitive SVM (TC SVM) strategy was used, since classic SVMs are proved inappropriate to deal with imbalanced datasets. We investigated the effect of incorporating the TC SVM on a learned SVM model using a medical dataset. In general, the TC SVM seem to outperform the classic SVM for all kernels employed in this comparative trauma study in terms of the criteria AUC and Geometric mean, something that confirms the significance of the TC method for imbalanced data. The experimental results presented in this study have demonstrated that the

TC method provides a very competitive solution to other existing standard methods, in optimization of Geometric mean and AUC for combating imbalanced classification problems. These results confirm the advantages of the considered approach, showing the promising perspective and new understanding of cost sensitive learning. On the other hand, sampling methods seem to outperform the C SVM. Especially a combination of SMOTE SVM with undersampling has revealed the best performance considering not only the Geometric mean and computational time, but also the overfitting problems that have been created using the other methods.

This paper presents a comparative analysis of different SVM strategies on real medical data. Evaluating the reliability of classifier algorithms is essential to ensure data quality. We used the Geometric mean measure and the Area Under the Roc Curve, both obtained by sensitivity and specificity, for the comparison of algorithms in order to provide useful results. Note that these two metrics gave almost similar results. In this way we make some comparisons only in terms of Geometric mean. It is obvious that the effort of health care to prevent patients' death is a huge problem that arises, forcing researchers to be more careful in their research. Sensitivity and specificity measure the prognostic model's ability to recognize the patients of a certain group (survivors or non-survivors). The value of this comparative study is the ability to calculate Type I and Type II error rates, giving lower Type II error with the cost sensitive and data preprocessing methods and as a consequence higher sensitivity compared to C SVM. This issue is of high importance for medical diagnosis due to the fact that the presented methodology gives us the ability to recognize the patients which are going to die and they are provided by an appropriate treatment. In this way, many deaths would be avoided. This method may assist as guidelines for improving the quality of treatment and therefore survivability of a patient through optimal trauma management. Although, Parpoula et al. (2013) have already dealt with the analysis of the Trauma dataset; their study focuses on the comparison of several data mining techniques including standard SVM. Our motivation for conducting this study is different because what we want to achieve is the balance between sensitivity and specificity enable the success trauma outcome prediction. The effectiveness of the considered approach is obvious.

We hope this work will convince experimenters to use not only standard SVM techniques but also reformulations of SVMs for the extraction of useful patterns when they deal with imbalanced medical datasets. Support Vector Machines are a powerful predictive tool and the use of the SVMs classifiers as an alternative method for supporting medical knowledge discovery is one of the most promising topics for further research.

Acknowledgements

The authors would like to thank the Editor and the Referee for their valuable comments and suggestions that resulted in improving the quality of presentation of this manuscript.

The research of the first author (K.D.) was financially supported by a scholarship awarded by Captain Fanourakis Foundation and State Scholarships Foundation.

Appendix

Trauma Data
Y: 0 (survival), 1 (death)

–Continuous covariates:
x1: weight, kg
x2: age, years
x3: Glasgow Coma Score, score
x4: pulse, N/min
x6: systolic arterial blood pressure, mmHg
x7: diastolic arterial blood pressure, mmHg
x8: Hematocrit (Ht), %
x9: haemoglobin (Hb), g/dl
x11: white cell count, /ml
x15: glucose, mg %
x16: creatinine, mg %
x18: amylase, score
x20: Injury Severity Score, score
x21: Revised Trauma Score, score

–Categorical covariates:
x19: evaluation of disability (0 = expected permanent big, 1 = expected permanent small, 2 = expected impermanent big, 3 = expected impermanent small, 4 = recovery)
x23: cause of injury (0 = fall, 1 = trochee accident, 2 = athletic, 3 = industrial, 4 = crime, 5 = other)
x24: means of transportation (0 = airplane, 1 = ambulance, 2 = car, 4 = on foot)
x25: Ambulance (0 = no, 1 = yes)
x26: hospital of records
x27: substructure of hospital (0 = orthopaedic, 1 = CT, 2 = vascular surgeon, 3 = neurosurgeon, 4 = Intensive Care Unit)
x28: comorbidities (0 = no, 1 = yes)
x31: sex (0 = female, 1 = male)
x35: doctor's speciality (0 = angiochirurgeon, 1 = non specialist, 2 = general doctor 3 = general surgeon, 4 =jawbonesurgeon, 5 = gynaecologist, 6 = thoraxsurgeon, 7 = neurosurgeon, 8 = orthopaedic, 9 = urologist, 10 = paediatrician, 11 =children surgeon, 12 = plastic surgeon)
x36: major doctor (0 = no, 1 = yes)
x41: dysphoria (0 = no, 1 = yes)
x52: collar (0 = no, 1 = yes)
x55: immobility of limbs (0 = no, 1 = yes)

x56: fluids (0 = no, 1 = yes)
x64: Radiograph E.R. (0 = no, 1 = yes)
x66: US (0 = no, 1 = yes)
x67: urea test (0 = no, 1 = yes)
x71: destination after the emergency room (0 = other hospital, 1 = clinic, 2 = unit of high care, 3 = intensive care unit I.C.U, 4 = mortuary, 5 = operating room)
x72: surgical intervention (0 = no, 1 = yes)
x86: arrival at emergency room (0 = 00:00-04:00, 1 = 04:01-08:00, 2 = 08:01-12:00, 3 = 12:01-16:00, 4 =16:01-18:00, 5 = 18:01-20:00, 6 = 20:01-24:00)
x87: exit from emergency room (0 = 00:00-04:00, 1 = 04:01-08:00, 2 = 08:01-12:00, 3 = 12:01-16:00, 4 =16:01-18:00, 5 = 18:01-20:00, 6 = 20:01-24:00)
x101: head injury (0 = none, 1 = AIS ≤ 2, 2= AIS > 2)
x102: face injury (0 = none, 1 = AIS ≤ 2, 2= AIS > 2)
x104: breast injury (0 = none, 1 = AIS ≤ 2, 2= AIS > 2)
x106: spinal column injury (0 = none, 1 = AIS ≤ 2, 2= AIS > 2)
x107: upper limbs injury (0 = none, 1 = AIS ≤ 2, 2= AIS > 2)
x108: lower limbs injury (0 = none, 1 = AIS ≤ 2, 2= AIS > 2)

References

[1] Akbani, R., Kwek, S. and Japkowicz, N. (2004). Applying support vector machines to imbalanced datasets, in *Proceedings of the 15th European Conference on Machine Learning*, 39-50.

[2] Batuwita, R. and Palade, V. (2010). Efficient resampling methods for training support vector machines with imbalanced data sets, in *Proceedings of the International Joint Conference on Neural Networks*. 1 - 8.

[3] Batuwita, R. and Palade, V. (2013). Class Imbalance Learning Methods for Support Vector Machines, in *Imbalanced Learning: Foundations, Algorithms, and Applications* (eds H. He and Y. Ma), John Wiley & Sons, Inc., Hoboken, NJ, Ch5.

[4] Burges, C. J. C. (1998). A tutorial on support vector machines for pattern recognition. *Data Mining and Knowledge Discovery* **2**, no. 2, 121–167.

[5] Chang, C.-C. and Lin, C.-J. (2011). \LIBSVM: A library for support vector machines. *ACM Transactions on Intelligent Systems and Technology* **2**, 27:1 - 27:27.

[6] Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. Journal of Artificial Intelligence Research, 16:321-357.

[7] Chawla, N.V., Japkowicz, N., Kolcz, A. (2004). Editorial: special issue on learning from imbalanced data sets. SIGKDD Explorations Special Issue on *Learning from Imbalanced Datasets* **6** (1):1-6.

[8] Choi, J. M. (2010). A Selective Sampling Method for Imbalanced Data Learning on Support Vector Machines. *Graduate Theses and Dissertations*. Paper 11529.

[9] Cortes, C. (1995). *Prediction of Generalisation Ability in Learning Machines.* PhD thesis, Department of Computer Science, University of Rochester.

[10] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine Learning* **20**, no. 3, 273-297.

[11] Cristianinio N. and Shawe-Taylor J. (2000). *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press.

[12] Hart, P. (1968). The condensed nearest neighbor rule. *IEEE Transactions on Information Theory*, **IT-14**, 515-516.

[13] Kubat M., and Matwin, S. (1997). Addressing the curse of imbalanced training sets: one-sided selection. In *Proceedings of the 14th International Conference on Machine Learning* (ICML1997), 179–186.

[14] Menardi, G. and Torelli, N. (2013). Training and assessing classification rules with imbalanced data. Data Mining and Knowledge Discovery, DOI: 10.1007/s10618-012-0295-5, to appear.

[15] Osuna, E., Freund, R., Girosi, F. (1997). Training support vector machines: An application to face detection. In *Proceedings Computer Vision and Pattern Recognition '97*, 130-136.

[16] Parpoula, C., Drosou K., Koukouvinos, C. (2013). Large-Scale Statistical Modelling via Machine Learning Classifiers, *Journal of Statistics Applications and Probability* **2**, No. 3, 203-222.

[17] Pepe, M. S., (2000). Receiver operating characteristic methodology, *Journal of the American Statistical Association* **95**, 308-11.

[18] Scholkopf, B. and Smola, A. (2001). *Learning with Kernels: Support Vector Machines, Regularization, Optimization*, Beyond. Cambridge, MA, USA: MIT Press.

[19] Schmidt, M. (1996). Identifying speakers with support vector machines. In *Proceedings of Interface '96*, Sydney.

[20] Swets J.A. and Pickett R.M. (1982). *Evaluation of Diagnostic Systems: Methods form Signal Detection Theory*. Academic Press, New York.

[21] Tong, S. and Chang, E. (2001). Support Vector Machine Active Learning for Image Retrieval. *Proceedings of ACM International Conference on Multimedia*, 107-118.

[22] Vapnik, V., (1995). *The Nature of Statistical Learning Theory*. Springer-Verlag New York, Inc.

[23] Veropoulos, K., Campbell, C., Cristianini, N. (1999). Controlling the sensitivity of support vector machines. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 55-60.

[24] Yuan, J. Li, J. and Zhang, B. (2006). Learning concepts from large scale imbalanced data sets using support cluster machines in *Proceedings of the 14th annual ACM international conference on Multimedia*, 441-450, ACM.

Received December 12, 2013; accepted July 26, 2014.

Krystallenia Drosou
Department of Mathematics
National Technical University of Athens
Zografou 15773, Athens, Greece
drosou.kr@gmail.com

Stelios Georgiou
Department of Mathematics
University of the Aegean
Karlovassi 83200, Samos, Greece
stgeogiou@aegean.gr

Christos Koukouvinos
Department of Mathematics
National Technical University of Athens
Zografou 15773, Athens, Greece
ckoukouv@math.ntua.gr

Stella Stylianou
Department of Mathematics
University of the Aegean
Karlovassi 83200, Samos, Greece
sstylian@aegean.gr