Municipal Revenue Prediction by Ensembles of Neural Networks and Support Vector Machines

PETR HÁJEK, VLADIMÍR OLEJ
Institute of System Engineering and Informatics
Faculty of Economics and Administration
University of Pardubice, Studentská 84, Pardubice
CZECH REPUBLIC
petr.hajek@upce.cz, vladimir.olej@upce.cz

Abstract: - Municipalities have to to pay increasing attention to the importance of revenue prediction due to fiscal stress. Currently, judgmental, extrapolative, and deterministic models are used for municipal revenue prediction. In this paper we present the designs of neural network and support vector machine ensembles for a real-world regression problem, i.e. prediction of municipal revenue. Base learners, as well as linear regression models are used as benchmark methods. We prove that there is no single ensemble method suitable for this regression problem. However, the ensembles of support vector machines and neural networks outperformed the base learners and linear regression models significantly.

Key-Words: - Municipal revenue, prediction, regression, support vector machine ensembles, modelling, neural network ensembles.

1 Introduction

Accurate revenue prediction assists municipalities in determining both long-term and short-term revenue. The knowledge of future revenue is also important for accurate plans of future expenditures [1]. Thus, municipalities can predict the balance of the budget.

Currently, the three general types of methods are used for the prediction of municipal revenue, namely judgmental (expert) methods, extrapolative (trend) methods, and deterministic (econometric) methods [2]. The judgmental methods are generally the most simplistic and less expensive than the other methods. The accuracy of expert prediction is dependent on the experience and knowledge of the expert [3]. The extrapolative methods are quantitative approaches, based on time series analysis to utilize mathematical models that use historical values to predict future values. Deterministic models allow the researcher to consider the simultaneous effects of several input variables that determine the level of revenue [3].

There are several parameters that affect municipal revenue, e.g. population, income per capita, property value, employment, business activity and interest rates [4]. The evaluation of the effects of these parameters on future revenue can also assist in budgetary decision-making process. A problem defined this way represents a regression problem with many input variables (affecting municipal revenue) and one output variable (municipal revenue).

Linear regression models (LRMs) have been applied for municipal revenue prediction so far [2]. However, the economy from which revenue come is increasingly complex. The input variables are inter-dependent and, moreover, the relations between the input variables and the predicted revenue are not necessarily linear. Therefore, the methods making it possible to model such complex relationships are suitable for municipal revenue prediction. The examples of such methods are e.g. neural networks (NNs), support vector machines (SVMs), and many others.

Neural networks are appropriate for municipal revenue prediction due to their ability to learn, generalize and model non-linear relations. Another important quality of NNs, except their ability to learn based on finding dependencies in training data and representing those in synapse weights, is the ability to generalize gained knowledge. Neural networks have been successfully applied in both classification [5], [6], [7] and regression tasks [8], [9].

Support vector machines represent an essential kernel-based method. Many variants of SVMs have been developed since SVMs were proposed [10], e.g. least squares SVMs, robust SVMs, etc. They represent approximate implementation of structural risk minimization method [11], [12]. This principle is based on the fact that the testing error is limited with the sum of training error and the expression depending on Vapnik-Chervonenkis dimension [11].

The article has the structure as follows. First, input variables (parameters) for the prediction model will be designed. Further, basic notions of NN and SVM ensembles will be introduced. A model design for the prediction of the municipal revenue will be realized in the next part of the paper. The results for feed-forward NN (FFNN) and SVM ensembles will be compared to both the base learners and the LRM ensembles across several ensemble methods. The impact of input parameters (variables) on the revenue of municipalities will be further analyzed on the data for Czech municipalities.

2 Parameters Design for Municipal Revenue Prediction

The design of parameters for municipal revenue modelling is realized with the view of involving all relevant factors of municipal revenue in the Czech Republic. As a result, the factors affecting municipal budgetary policy will be obtained.

Municipal revenue has been affected by economic transformation in the Czech Republic. For example, new competences allowed municipalities to increase their revenue by credits or loans practically without any restrictions. Thus, the need arises for the analysis of municipal revenue including the possibility of municipal revenue prediction. Both the economic parameters (population size, national GDP, number of enterprises, etc.) and the financial parameters (tax revenue, assets, own revenue, grants, etc.) represent the inputs of the model, see Table 1. The descriptive statistics on the used data (452 municipalities in the years 2003-2006) are presented in the Appendix.

The economic parameters determine the range of public goods and services provided and, thus, the revenue from the state budget and the revenue for municipal services. The financial parameters represent the main sources of municipal revenues (taxes, fees, assets, and grants).

Based on the presented facts, the following data matrix **P** can be designed

$$\mathbf{P} = \begin{array}{c|ccccc:c} & x_1 & \dots & x_k & \dots & x_m & y_i \\ \hline o_1 & x_{1,1} & \dots & x_{1,k} & \dots & x_{1,m} & y_1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ o_i & x_{i,1} & \dots & x_{i,k} & \dots & x_{i,m} & y_i \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ o_n & x_{n,1} & \dots & x_{n,k} & \dots & x_{n,m} & y_n \end{array},$$

where: – $o_i \in O$, $O=\{o_1,o_2, \dots ,o_i, \dots ,o_n\}$ are objects (municipalities),

– x_k is the k-th parameter,

– $x_{i,k}$ is the value of the parameter x_k for the i-th object $o_i \in O$,

– y_i is the output represented by the municipal revenue size.

Table 1 Municipal revenue parameters design

x_k	Description of parameter
x_1	Municipal population
x_2	Municipal size coefficient
x_3	National GDP
x_4	Enterprises in municipality
x_5	Physical person income tax
x_6	Business tax
x_7	Tax on capital yield
x_8	Corporate income tax
x_9	Corporate income tax (municipal.)
x_{10}	Value added tax
x_{11}	Fees from selected services
x_{12}	Real estate tax
x_{13}	Long-term intangible assets
x_{14}	Long-term tangible assets
x_{15}	Long-term financial assets
x_{16}	Own revenue
x_{17}	Lease revenue
x_{18}	Revenue from interests
x_{19}	Revenue from intangible assets sale
x_{20}	Revenue from fiscal assets sale
x_{21}	Population under 14 years
x_{22}	Current grants
x_{23}	Capital grants
y	Total revenue

Higher value of the parameter x_1 entails higher municipal tax revenues. Larger municipalities have a higher share in tax yield, because the more populated municipalities have higher expenditures for infrastructure and other public goods. Therefore, higher population guarantees future municipal revenues for the creditors. On the other hand, more populated municipalities are likely to have higher demands for public expenditures, leading to higher levels of public debt. The size category x_2 represents the competences of municipality. Larger competences force municipalities to provide more public goods, leading to the growth of expenditures and debt.

National GDP influence the tax base of the state. Consequently, municipalities obtain more revenue with a higher GDP due to a share on the taxes collected by the state. Higher number of enterprises brings higher business taxes. The revenue from selected taxes makes a considerable proportion of total revenue. The income from long-term assets (lease, sale) are represented by the parameters x_{13}-x_{15},x_{17}, and x_{19}-x_{20}. Other own revenue is included in the parameters x_{16} and x_{18}. Higher proportion of own revenue in total revenues x_{16} entails higher fiscal autonomy of the municipality. Parameters x_{21}-x_{23} are linked to revenue from grants (the highest share of grants goes to education).

Due to high correlation among input variables, data pre-processing is carried out by means of principal components analysis (PCA). The dimensionality of the original set of input parameters $x_1,x_2, \dots ,x_{23}$ is reduced in four principal components $c_1,c_2, \dots ,c_4$ in this way, see Table 2. The results of the PCA are represented by the fraction of the explained variation R^2X, by the eigenvalues, and by the fraction of predicted variation Q^2. The fraction of the explained variation R^2X and the fraction of predicted variation Q^2 are defined as follows

$$R^2X = 1 - \frac{\mathrm{RSS}}{\mathrm{SS}} , \quad (1)$$

$$Q^2 = 1 - \frac{\mathrm{PRSS}}{\mathrm{RSS_c}} , \quad (2)$$

where RSS is residual sum of squares, SS is sum of squares, PRSS is predictive residual sum of squares, and RSS_c is residual sum of squares for the previous component.

When selecting the number of principal components, one has to bear in mind the trade off between model complexity and the goodness of fit. We selected only statistically significant variables with the Q^2>-0.1. As a result, R^2X=89.43% of the sum of squares has been explained by all the extracted components.

Table 2 Principal components analysis

	R^2X	Eigenvalues	Q^2
c_1	0.755768	17.38266	0.728513
c_2	0.063700	1.46510	0.120088
c_3	0.044054	1.01323	-0.090726
c_4	0.030764	0.70757	-0.100000

In Table 3 the loadings of municipal revenue parameters in the constructed components are shown. The component c_1 represents most of the original parameters (except for the parameter x_3). Similarly, the component c_2 represents the parameters x_2, x_{16}, x_{17}, and x_{23} especially. The component c_3 stands for the national GDP, and the component c_4 is addressed to the revenue from intangible assets sale.

Table 3 Loadings of municipal revenue parameters $x_1,x_2, \dots ,x_{23}$ in principal components $c_1,c_2, \dots ,c_4$

	c_1	c_2	c_3	c_4
x_1	0.992	-0.090	-0.003	-0.011
x_2	0.490	0.354	-0.007	0.001
x_3	0.008	0.041	0.989	0.150
x_4	0.987	-0.134	0.002	-0.017
x_5	0.983	-0.140	0.020	-0.068
x_6	0.959	-0.071	-0.027	0.057
x_7	0.972	-0.147	0.013	-0.044
x_8	0.980	-0.141	0.027	-0.083
x_9	0.904	0.164	0.004	-0.037
x_{10}	0.980	-0.145	0.026	-0.068
x_{11}	0.945	-0.078	0.023	-0.066
x_{12}	0.980	-0.117	0.002	-0.030
x_{13}	0.969	-0.047	0.033	-0.085
x_{14}	0.985	-0.010	0.014	-0.055
x_{15}	0.943	-0.150	0.027	-0.071
x_{16}	0.680	0.557	-0.046	0.034
x_{17}	0.647	0.653	-0.022	0.040
x_{18}	0.859	0.038	0.001	-0.006
x_{19}	0.562	-0.125	-0.165	0.796
x_{20}	0.900	-0.079	0.010	0.065
x_{21}	0.991	-0.045	-0.010	0.000
x_{22}	0.986	-0.055	-0.020	0.010
x_{23}	0.514	0.623	-0.004	-0.001

3 Feed-Forward Neural Networks and Support Vector Machines

3.1 Feed-Forward Neural Networks

A FFNN [13] uses neurons connected among themselves in layers. The neurons of adjoining layers are connected so that the output of one neuron is distributed into the inputs of the neurons in the following layer. As a result, the input values only move from input to hidden layers and, at the same time, from hidden to output layers.

A vector **x** of input values is presented to the input layer. This pattern is then expanded (transformed) through the FFNN using synapse weights w_{ij} and activation functions f up to the outputs of the FFNN. The values of potentials ξ are computed consecutively from input to output layer as follows:

$$\xi_j = \sum_i w_{ij} * y_i , \quad (3)$$

where w_{ij} are synapse weights between the i-th and the j-th neuron, i and j stand for indices passing through neurons of two adjoining layers, and y_i is the output of the i-th neuron.

The structure of a FFNN is presented in Fig. 1, where the hidden neurons are connected to one output neuron through the synapse weights w_{jk}.

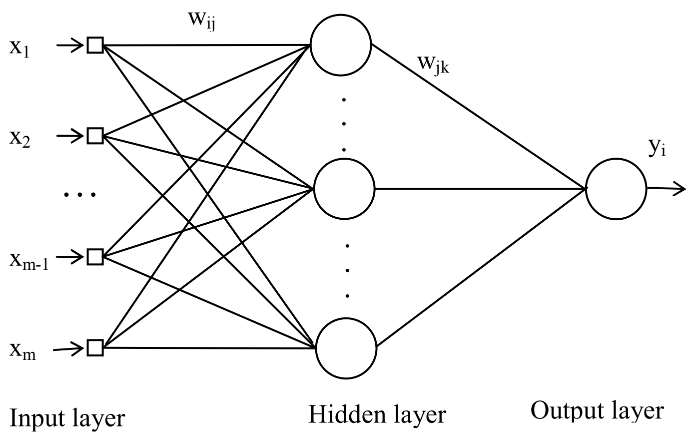

Fig. 1 General structure of a feed-forward neural network

The resulting weighted values are added together producing a weighted sum given to a transfer function. The outputs of the transfer function are distributed to the output layer. They are multiplied by a weight w_{jk}, and based on the resulting weighted values, the weighted sum is put to a transfer function, which outputs a value y_i representing a predicted value. The FFNNs are based on supervised learning. The objective of the learning lies in obtaining such a setting of synapse weights w_{ij} that the deviation (error) E between actual and target outputs of the FFNN is minimum for the given training patterns. The partial derivative of the error E, with respect to the synapse weights, represents the minimization of the error E by gradient method. Backpropagation algorithm is a standard learning algorithm of the FFNNs [13]. Since the FFNN uses the gradient method in the learning process, it is possible that the learning algorithm gets stuck in local minimum within the error function. This can be solved, for example, by adding noise to the equation for synapse weights adaptation, adding neurons, setting the learning rate, or adding momentum.

3.2 Support Vector Machines

The design of SVMs [10], [11], [12] depends on the non-linear projection of the input space Ξ into multidimensional space Λ, and on the construction of an optimal hyperplane. This operation is dependent on the estimation of inner product kernel referred to as kernel function.

Let **x** be a vector from input space Ξ of dimension m. Next let $g_j(\mathbf{x})$, j=1,2, … ,m be a set of non-linear transformations from input space Ξ into multidimensional space Λ of dimension p. Then the hyperplane can be defined as a decision surface as follows [11]

$$\sum_{j=1}^{m} \mathbf{w}_j g_j(\mathbf{x}) + b = 0 , \tag{4}$$

where $\mathbf{w}_j$, j=1,2, … ,m is the vector of weights connecting the p-dimensional space Λ with the n-dimensional output space Π, b is bias.

In a linearly separable case, the algorithm of SVMs tries to find the separating hyperplane with the widest margin [10]. If patterns are linearly separable, it is always possible to find weights **w** and bias b so that the inequalities bounding this optimization problem are satisfied. If the algorithm for separable data is used for inseparable data, it does not find an acceptable solution. Then this fact is realized by means of bounding this optimization problem in the following way: $\mathbf{x}.\mathbf{w} + b \geq +1 - \xi_i$ for $y_i = +1$, $\mathbf{x}.\mathbf{w} + b \geq -1 - \xi_i$ for $y_i = -1$, where ξ_i, i=1,2, … ,u, and (.) represents the dot product of two vectors **x** and **w**. Lots of $g_j(\mathbf{x})$ represent input supported by weight $\mathbf{w}_j$ through the p-dimensional space Λ. Further, let vector $\mathbf{g}(\mathbf{x})=[g_0(\mathbf{x}),g_1(\mathbf{x}), \dots ,g_m(\mathbf{x})]^T$ be defined. Then the vector $\mathbf{g}(\mathbf{x})$ represents the image derived in the p-dimensional space Λ related to the input vector **x**. Hence, decision hyperplane $w^T\mathbf{g}(\mathbf{x})=0$ can be defined based on this image. Then based on [10] and [11], after the application of optimization condition into the Lagrange equation, the following form of weights is obtained

$$\mathbf{w} = \sum_{i=1}^{N} \alpha_i d_i \mathbf{g}(\mathbf{v}_i), \tag{5}$$

where the vector $\mathbf{g}(\mathbf{v}_i)$ from the p-dimensional space Λ corresponds to the i-th support vector $\mathbf{v}_i$, α_i are Lagrange multipliers determined in the optimization process, and d_i represents the shortest distance separating the hyperplane from the nearest positive or negative patterns.

Support vectors $\mathbf{v}_i$ consist of small subset of training data extracted by the algorithm. Then as equation (5) is substituted into $\mathbf{w}^T\mathbf{g}(\mathbf{x})=0$, the decision hyperplane is computed in the p-dimensional space Λ. Inner product of two vectors $\mathbf{g}^T(\mathbf{v}_i)\mathbf{g}(\mathbf{x})$ is derived in the p-dimensional space Λ in relation to input vector **x** and support vector $\mathbf{v}_i$. Now, the inner product kernel $k(\mathbf{x},\mathbf{x}_i)$ can be defined this way

$$k(\mathbf{x}, \mathbf{v}_i) = \mathbf{g}^T(\mathbf{v}_i)\mathbf{g}(\mathbf{x}) = \sum_{j=0}^{m} \mathbf{g}_j(\mathbf{v}_i)\mathbf{g}_j(\mathbf{x}), \quad (6)$$

for i=1,2, … ,N. As obvious from this equation, the kernel function $k(\mathbf{x},\mathbf{v}_i)$ is a symmetrical function with respect to its arguments. The most important fact is that the kernel function $k(\mathbf{x},\mathbf{v}_i)$ can be used in order to construct an optimal hyperplane in the p-dimensional space Λ without considering separate p-dimensional space Λ in explicit form. Based on given facts, it is possible to find linear separators in the p-dimensional space Λ so that $(\mathbf{x},\mathbf{v}_i)$ is replaced by kernel function $k(\mathbf{x},\mathbf{v}_i)$. Accordingly, the process of learning can be realized so that only kernel functions $k(\mathbf{x},\mathbf{v}_i)$ can be computed instead of full list of attributes for each data point. Evidently, the found linear separators can be transformed back into the original space Ξ. In this way any non-linear boundaries between patterns can be obtained. Various kernel functions $k(\mathbf{x},\mathbf{v}_i)$ representing different spaces can be used for modelling, e.g. linear, polynomial, RBF, and sigmoid kernel function [10], [11]. Then the output $f(\mathbf{x}_t)$ of SVMs is defined this way

$$f(\mathbf{x}_t)=\sum_{i=1}^{N} \alpha_i y_i k(\mathbf{v}_i, \mathbf{x}_t) + b, \quad (7)$$

where $\mathbf{x}_t$ is the evaluated pattern, N is the number of support vectors, $\mathbf{v}_i$ are support vectors, α_i are Lagrange multipliers determined in the optimization process, and $k(\mathbf{v}_i,\mathbf{x}_t)$ is actual kernel function $k(\mathbf{x},\mathbf{v}_i)$. A general structure of SVMs is shown in Fig. 2.

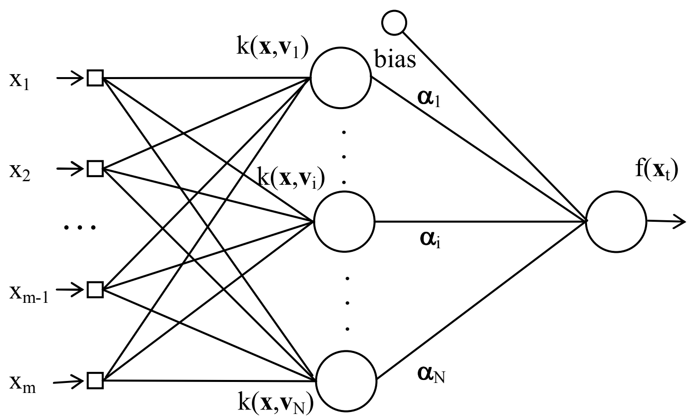

Fig. 2 General structure of Support Vector Machines

Support vectors $\mathbf{v}_i$ represent the component of predictors' structure, and their number N is cut during the optimization process [10], [11], [12]. In regression problems, the quality of estimation is measured by the loss function

$$\begin{aligned} L_\varepsilon(y, f(\mathbf{x},\mathbf{w})) &= 0 && \text{if } |y - f(\mathbf{x},\mathbf{w})| \le \varepsilon \\ L_\varepsilon(y, f(\mathbf{x},\mathbf{w})) &= |y - f(\mathbf{x},\mathbf{w})| - \varepsilon && \text{otherwise} \end{aligned}, \quad (8)$$

where ε is a threshold. In regression case the loss function $L_\varepsilon(y,f(\mathbf{x},\mathbf{w}))$ is used to penalize errors that are greater than the threshold ε.

4 Neural Network and Support Vector Machine Ensembles

The idea of the NN and SVM ensemble has been proposed in [14] and [15]. The boosting method was used to train each individual NN (SVM) and another NN (SVM) was applied for combining several NNs (SVMs).

The basic idea behind NN (SVM) ensembles is to predict a given input pattern by obtaining a prediction from each copy of the NN (SVM) and then using a consensus scheme to decide the collective prediction [14], [15], [16], [17], [18]. We refer to the set of NNs (SVMs) used as an ensemble. Ensembles are desirable due to the basic fact that selection of the weights **w** is an optimization problem with many local minima. As each NN (SVM) makes generalization errors on different subsets of the input space, we shall argue that the collective decision produced by the ensemble is represented by a lower error than the decision made by any of the individual NN (SVMs) [14], [15]. The conclusion is that the ensemble can be more accurate than any one NN or SVM for both classification and regression problems. As for combining the predictions of NNs and SVMs, the most prevailing approaches are simple averaging or weighted averaging for regression problems [16].

For the prediction of municipal revenue we will use four ensemble methods, i.e. bagging, dagging, stochastic gradient boosting, and rotation forest.

Bagging was proposed by Breiman [19] based on bootstrapping defined by Efron and Tibshirani [20]. It generates several training sets from the original training set and then trains a component neural network from each of those training sets. When bootstrap samples are used the method is called bagging, for disjoint samples we call it dagging [21].

Stochastic gradient boosting (SGB) is proposed by Friedman [22]. It incorporates randomness as an integral part of the gradient boosting procedure. At each iteration a subsample of the training data is drawn at random. This subsample is then used to fit the base learner and compute the model update for the current iteration.

Rotation forest (RF), proposed by Rodriguez et al. [23], is a method for generating ensembles based on feature extraction. To create the training data for a base predictor, the feature set is randomly split into K subsets and PCA is applied to each subset. All principal components are retained in order to preserve the variability information in the data. Thus, K axis rotations take place to form the new features for a base classifier. The idea of the rotation approach is to encourage simultaneously individual accuracy and diversity within the ensemble.

5 Model Design and Analysis of Results

The model design of municipal revenue aims to realize the regression problem, where the dependent variable is represented by the size of the municipal revenue.

Data are pre-processed for the purposes of modelling. First, data are standardized in order to eliminate the dependence on the units. Then, the original set of parameters is reduced by the PCA as presented in the previous chapter.

We use polynomial kernel function with the polynomial degree of three and the shifting parameter c=10.0 for the SVMs. FFNNs and LRMs are used for the comparison with the SVMs as presented in Fig. 3. Especially FFNNs [13] are suitable for regression problem realization. The backpropagation algorithm with momentum is applied for the learning of the FFNN. The number of neurons in the input layer n_1=4, the number of neurons in the hidden layer n_2=3, learning rate η=0.01, and momentum mom=0.2 represent the input parameters of the FFNN.

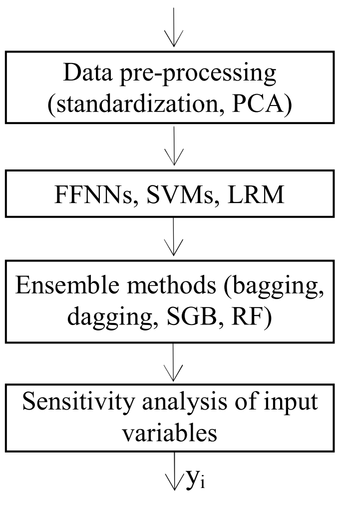

Fig. 3 Regression model design for municipal revenue prediction

The ensembles of the underlying predictors are created by means of bagging, dagging, SGB, and RF which were introduced previously. The learning parameters of the ensemble methods are as follows. For the bagging, the number of iterations=10 and the size of each bag (as a percentage of the training set)=100%. For the daging, the number of folds to use for splitting the training set into smaller chunks for the base regression model=10. For the SGB, the number of iterations=10 and shrinking rate=1.0. For the RF, the number of iterations=10, the percentage of instances to be removed=50%, and the projection method is represented by the PCA.

The sensitivity analysis of input parameters consists in the testing of input factors' c_1,c_2, … ,c_4 influence on the municipal revenue y_i.

The model design is verified on the sample of the municipalities in the Pardubice Region, Czech Republic. Input data set contains the values of parameters c_1,c_2, … ,c_4 on 452 municipalities of the Pardubice Region for the years 2003-2006 (one election cycle). We tested the SVM ensembles on 4 data sets, with four different prediction time horizons (years t=1,2,3,4).

The quality of the models is measured using the relative absolute error (RAE) and relative root squared error (RRSE). As a result, we refer the errors RAE_1,RAE_2, ... ,RAE_4, and $RRSE_1$,$RRSE_2$, … ,$RRSE_4$ for t=1,2,3,4.

For the RAE, the comparison of the FFNN ensembles is presented in Fig. 4, while the results for the SVM and LRM ensembles are shown in Fig. 5 and Fig. 6, respectively.

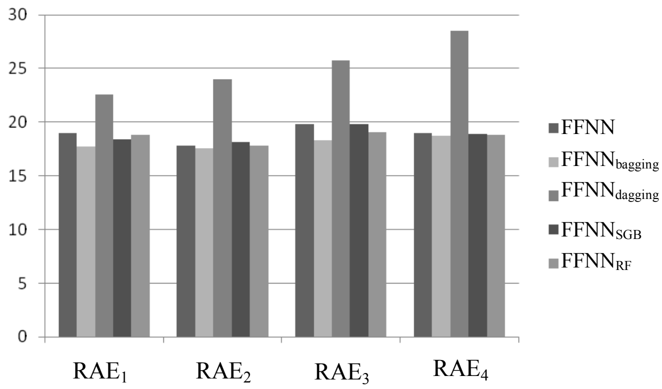

Fig. 4 Relative absolute errors for FFNN ensembles

Stochastic gradient boosting show best results for all the prediction time horizons when using SVMs. We tested the results with a paired t-test, and the RAE for the SGB is significantly lower than the RAE for the other methods at 5%. For the LRM ensembles, dagging provides the lowest RAE. The RAE for the dagging is significantly lower than the RAE for the other methods at 5%. On the other hand, bagging outperformed the other ensemble methods for the FFNNs with a significantly lower RAE at 5%. When comparing the best results of all the presented experiments, the SVM ensembles created by the SGB show significantly better results for the following prediction time horizons: t=1,3,4. For the prediction time horizon t=2, the FFNN ensemble created by the bagging technique is superior to the other methods.

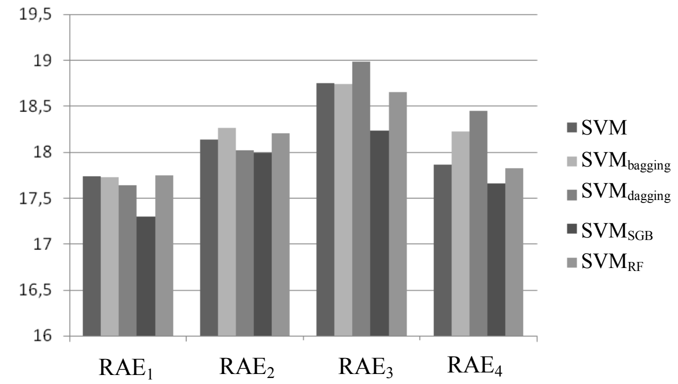

Fig. 5 Relative absolute errors for SVM ensembles

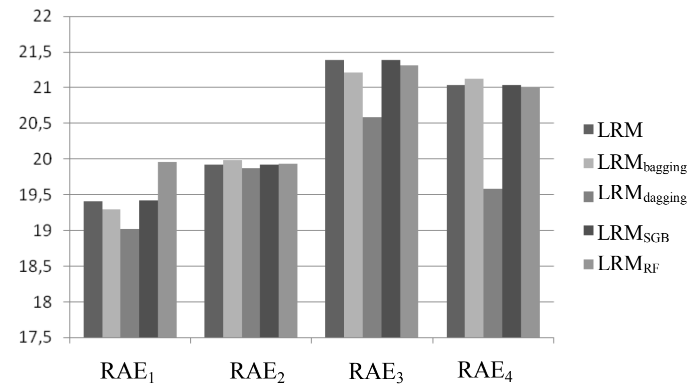

Fig. 6 Relative absolute errors for LRM ensembles

For the RRSE, the results of the ensemble methods are ambiguous. As illustrated by the results given in Table 4, the bagging is superior to other methods for t=1 in the case of the SVM and the LRM, while for t=2 in the case of the FFNN. For t=2,3,4, the RF is better or equal to the rest of ensemble methods for both the SVM and the LRM. For the FFNN ensembles, the SGB provides the lowest $RRSE_1$ and $RRSE_3$, while the RF is significantly more accurate for t=3.

When comparing the best results of all the presented experiments, the FFNN base learner, the FFNN ensemble created by the SGB, and the SVM ensemble created by the RF and bagging show significantly better results for the prediction time horizon t=1 at 5%. For the prediction time horizon t=2, the FFNN ensemble produced by the bagging technique, the $SVM_{bagging}$ and SVM_{RF} outperform the other methods. For t=3, the $FFNN_{RF}$ and SVM_{RF} provides a significantly lower $RRSE_3$ than the compared regression models. In the case of t=4, the FFNN ensemble constructed by the SGB significantly outperforms the other models. In Table 4, the results which are significantly better at 5% are in bold.

Table 4 Relative root squared errors for FFNN, SVM and LRM ensembles

	$RRSE_1$	$RRSE_2$	$RRSE_3$	$RRSE_4$
FFNN	**15.95**	17.41	18.96	16.13
$FFNN_{bagging}$	16.75	**16.09**	19.17	21.53
$FFNN_{dagging}$	45.56	50.22	52.82	63.46
$FFNN_{SGB}$	**15.91**	17.42	18.68	**15.19**
$FFNN_{RF}$	15.98	17.21	**18.43**	15.44
SVM	15.99	17.04	18.16	15.81
$SVM_{bagging}$	**15.90**	**16.25**	18.62	19.67
$SVM_{dagging}$	17.62	18.14	22.66	22.52
SVM_{SGB}	16.34	16.77	18.56	16.72
SVM_{RF}	**15.94**	**16.17**	**18.13**	15.79
LRM	16.09	17.81	18.73	17.05
$LRM_{bagging}$	16.01	17.28	19.06	22.29
$LRM_{dagging}$	18.78	21.51	23.66	26.45
LRM_{SGB}	16.09	17.00	18.73	17.05
LRM_{RF}	18.09	17.00	18.66	16.99

In this study the calculation of variables' importance is performed using sensitivity analysis. The values of each input variable are randomized and the effect on the quality of the model (RAE) is measured. Finally the contributions of input variables are standardized so that the contribution of the most important input variable is 100%, and the contributions of other input variables are related to this variable. The resulting relative contributions of input variables of SVM ensembles created by the SGB are presented in Table 5.

Table 5 Relative contributions [%] of input variables for different prediction time horizons

	t=1	t=2	t=3	t=4
c_1	100.000	100.000	100.000	100.000
c_2	0.356	0.334	0.401	0.432
c_3	0.077	0.155	0.350	0.405
c_4	0.015	0.003	0.043	0.105

Considering the results obtained by the SVMs, parameter c_1 proved to be the most important factor of municipal revenue y_i. There is strong evidence of a relationship of municipal revue to both the economic (population, enterprises) and the financial (previous revenue, assets, etc.) parameters.

6 Related Literature

In previous studies, ensemble methods have been mostly applied in classification problems. This holds for both the artificial datasets and real-world datasets. The examples of the latter ones are face recognition [24],

[25], medical diagnosis [26], [27], [28], [29], abnormal internet protocols detection [30], gene expression data classification [31], etc.

There are also studies addressing the regression problem. However, they have been realized on artificial data only, e.g. [18].

Zhou, Wu, and Tang [16] point out the relationship between the ensemble and its component NNs. The results showed that it may be better to ensemble only a subset of the available NNs, i.e. not all of them. This conclusion is based on experiments conducted on both classification and regression artificial datasets.

A cooperative co-evolution approach was proposed by [32] for designing NN ensembles. In this work, the cooperation of the NNs is encouraged so that each NN is evaluated throughout the evolutionary process using a multi-objective method. For each NN, different objectives are defined.

The NN ensembles are effective only in such cases where the individual NNs are as accurate and diverse as possible [33]. However, these two conditions are conflicting. This issue has been discussed by [34] in order to find an optimal compromise between these objectives.

Less attention has been paid to SVM ensembles in the literature [15], [35], [36], [37]. In [15], the SVM ensemble based on the bagging and boosting methods was proposed for classification problems. Simulation results for the hand-written digit recognition and the fraud detection showed that the SVM ensemble with bagging or boosting outperforms a single SVM in terms of classification accuracy.

7 Conclusion

The article presents the design of a model for municipal revenue prediction. Support vector machines, FFNNs, and LRMs are used for the modelling. In order to improve the prediction performance of these methods we applied several methods for the creation of predictors' ensembles. The proposed models have been applied to the data sample of municipalities in the Pardubice Region. The impact of economic and financial parameters on the revenue of municipalities was examined during one political cycle (4 years). The impact of macroeconomic variables increases with a longer prediction time horizon.

Support vector machines and FFNNs have proven to be an appropriate method for the prediction of municipal revenue, as they are able to learn, generalize and model non-linear relationships while maintaining the speed and robustness of computation. Moreover, the chosen ensemble methods improved the results of the SVMs and FFNNs significantly.

This is in line with previous studies for it was shown that the NN and SVM ensembles with bagging, boosting, and other ensemble methods outperform a single NN (SVM) in terms of classification or prediction accuracy greatly [14], [15], [38]. We proved that the same results hold also for this real-world regression problem.

Compared to other methods (FFNNs and LRM), SVM ensembles were superior for t=1,3,4, while FFNN ensembles outperformed the rest of the methods for t=2 in terms of RAE. Out of the selected methods for ensembles' creation, dagging produced best results for LRM ensembles, while bagging was superior for FFNN ensembles. For RRSE, FFNN and SVM ensembles constructed by bagging, SGB, and RF provided best results for different prediction time horizons.

Based on presented facts we can state that the designed model can serve as a tool for decision-making support in the municipal policy.

The experiments were carried out in Weka in MS Windows XP operation system.

Acknowledgement

This work was supported by the scientific research of the Czech Science Foundation, under Grant No: 402/09/P090 with title Modelling of Municipal Finance by Computational Intelligence Methods, and Grant No: 402/08/0849 with title Model of Sustainable Regional Development Management.

References:

[1] J. Beckett-Camarata, Revenue Forecasting Accuracy in Ohio Local Governments, *Journal of Public Budgeting, Accounting & Financial Management*, Vol.18, No.1, 2006, pp.77-99.

[2] Ch.G. Reddick, Assessing Local Government Revenue Forecasting Techniques, *International Journal of Public Administration*, Vol.27, No.8-9, 2004, pp.597-613.

[3] L. Schroeder, Forecasting Local Revenues and Expenditures, *Management Policies in Local Government Finance*, 4th Ed., R.J. Aronson, E. Schwartz, (Eds.), ICMA, Washington, D.C., 1996.

[4] K. Nollenberger, *Evaluating Financial Condition: A Handbook for Local Government*, International City/County Management Association, Washington, D.C., 2003.

[5] P. Hájek, V. Olej, Air Quality Modelling by Kohonen's Self-organizing Feature Maps and LVQ Neural Networks, *WSEAS Transactions on Environment and Development*, Vol.4, No.1, 2008, pp.45-55.

[6] H. Xiao, X. Zhang, Comparison Studies on Classification for Remote Sensing Image Based on Data Mining Method. *WSEAS Transactions on Computers*, Vol.7, No.5, 2008, pp.552-558.

[7] S. M. H. Hitam, M.Y. Muslan, M.M. Deris, M.Y. Saman, Image Texture Classification using Gray Level Co-Occurrence Matrix and Neural Network. *WSEAS Transactions on Computers*, Vol.3, No.4, 2004, pp.1151-1156.
[8] Y.M. Wang, S. Traore, T. Kerh, Neural Network Approach for Estimating Reference Evapotranspiration from Limited Climatic Data in Burkina Faso, *WSEAS Transactions on Computers*, Vol.6, No.7, 2008, pp.704-713.
[9] Y.M. Wang, S. Traore, T. Kerh, Monitoring Event-Based Suspended Sediment Concentration by Artificial Neural Network Models, *WSEAS Transactions on Computers*, Vol.5, No.7, 2008, pp.359-368.
[10] S. Abe, *Support Vector Machines for Pattern Classification*, Springer-Verlag, London, 2005.
[11] V. Vapnik, *The Nature of Statistical Learning Theory*, Springer-Verlag, New York, 1999.
[12] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*, Cambridge University Press, Cambridge, 2000.
[13] S.S. Haykin, *Neural Networks: A Comprehensive Foundation*, Prentice-Hall, Upper Saddle River, 1999.
[14] L.K. Hansen, P. Salamon, Neural Network Ensembles, *IEEE Transactionson Pattern Analysis and Machine Intelligence*, Vol.12, No.10, 1990, pp.993-1001.
[15] H.Ch. Kim, S. Pang, H.M. Je, D. Kim, S.Y. Bang, Constructing Support Vector Machine Ensemble, *Pattern Recognition*, Vol.36, 2003, pp.2757-2767.
[16] Z.H. Zhou, J. Wu, W. Tang, Ensembling Neural Networks: Many Could Be Better Than All, *Artificial Intelligence*, Vol.137, No.1-2, 2002, pp.239-263.
[17] B. Bakker, T. Heskes, Clustering Ensembles of Neural Network Models, *Neural Networks*, Vol.16, No.2, 2003, pp.261-269.
[18] M.F. Amasyali, O.K. Ersoy, *A Study of Meta Learning for Regression*, ECE Technical Reports, Purdue University, 2009.
[19] L. Breiman, Bagging Predictors, *Machine Learning,* Vol.24, No.2, 1996, pp.123-140.
[20] B. Efron, R. Tibshirani, *An Introduction to the Bootstrap*, Chapman and Hall, London, 1993.
[21] K.M. Ting, I.H. Witten, Stacking Bagged and Dagged Models, *International Conference on Machine Learning,* Tennessee, 1997, pp.367-375.
[22] J.H. Friedman, Stochastic Gradient Boosting, *Computational Statistical Data Analysis*, Vol.38, No.4, 2002, pp.367-378.
[23] J.J. Rodriguez, L.I. Kuncheva, C.J. Alonso, Rotation Forest: A New Classifier Ensemble Method, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.28, No.10, 2006, pp.1619-1630.
[24] S. Gutta, H. Wechsler, Face Recognition using Hybrid Classifier Systems, *Proceedings of ICNN96*, Washington, IEEE Computer Society Press, Los Alamitos, CA, 1996, pp.1017-1022.
[25] F.J. Huang, Z.H. Zhou, H.J. Zhang, T.H. Chen, Pose Invariant Face Recognition, *Proceedings of 4th IEEE International Conference on Automatic Face and Gesture Recognition*, Grenoble, France, IEEE Computer Society Press, Los Alamitos, CA, 2000, pp.245-250.
[26] Z.H. Zhou, Y. Jiang, Y.B. Yang, S.F. Chen, Lung Cancer Cell Identification based on Artificial Neural Network Ensembles, *Artificial Intelligence in Medicine*, Vol.24, No.1, 2002, pp.25-36.
[27] Z.H. Zhou, Y. Jiang, Medical Diagnosis with C4.5 Rule Preceded by Artificial Neural Network Ensemble, *IEEE Transactions on Information Technology in Biomedicine*, Vol.7, No.1, 2003, pp.37-42.
[28] P. Cunningham, J. Carney, S. Jacob, Stability Problems with Artificial Neural Networks and the Ensemble Solution, *Artificial Intelligence in Medicine*, Vol.20, No.3, 2000, pp.217-225.
[29] S.B. Cho, H.H. Won, Cancer Classification using Ensemble of Neural Networks with Multiple Significant Gene Subsets, *Applied Intelligence*, Vol.26, 2007, pp.243-250.
[30] Q. Wu, Q. Feng, Abnormal BGP Routing Dynamics Detection by Active Learning Using Bagging on Neural Networks, *Studies in Computational Intelligence*, Vol.208, 2009, pp.61-72.
[31] W. Chen, H. Lu, M. Wang, Ch. Fang, Gene Expression Data Classification using Artificial Neural Network Ensembles Based on Samples Filtering, *International Conference on Artificial Intelligence and Computational Intelligence*, Los Alamitos, 2009, pp.626-628.
[32] N. García-Pedrajas, C. Hervás-Martínez, D. Ortiz-Bo, Cooperative Coevolution of Artificial Neural Network Ensembles for Pattern Classification. *IEEE Transactions on Evolutionary Computation*, Vol.9, No.3, 2005, pp.271-302.
[33] A. Krogh, J. Vedelsby, Neural Network Ensembles, Cross Validation, and Active Learning, *Advances in Neural Information Processing*, Vol.7, 1995, pp.231-238.
[34] P.M. Granitto, P.F. Verdes, H.A. Ceccatto, Neural Network Ensembles: Evaluation of Aggregation Algorithms, *Artificial Intelligence*, Vol.163, No.2, 2005, pp.139-162.

[35] Q. Hu, Z. He, Z. Zhang, Y. Zi, Fault Diagnosis of Rotating Machinery Based on Improved Wavelet Package Transform and SVMs Ensemble, *Mechanical Systems and Signal Processing*, Vol.21, No.2, 2007, pp.688-705.
[36] P. Kang, S. Cho, EUS SVMs: Ensemble of Under-Sampled SVMs for Data Imbalance Problems, *ICONIP 2006, Lecture Notes in Computer Science 4233*, Part 1, Springer-Verlag, Heidelberg, 2006, pp.837-846.
[37] X. Li, L. Wang, E. Sung, A Study of AdaBoost with SVM Based Weak Learners, *Proceedings of International Joint Conference on Neural Networks*, Montreal, Canada, 2005, pp.196-201.
[38] P. Hajek, V. Olej, Municipal Revenue Prediction by Support Vector Machine Ensembles, Proceedings of the 14th WSEAS International Conference on Computers, Corfu, Greece, 2010, pp.325-330.

Appendix

Descriptive statistics of the data

x_k	Mean	Median	Std dev	Range
x_1	1124.167	340	4686.797	23–89245
x_2	0.566733	0.5881	0.048781	0.4213–0.8487
x_3	5.46	6.1	1.2079	3.6%–6.8%
x_4	230.6333	65	1128.506	5–22481
x_5	1813481	469598	9530747	$17801–213*10^6$
x_6	712947	128444.4	3731031	$-34876–83*10^6$
x_7	106266.3	28386	562725.6	$0–12.84*10^6$
x_8	1947483	513428	10156085	$4226–233*10^6$
x_9	422947.1	0	2430280	$0–56.9*10^6$
x_{10}	2824631	757773	14631054	$0–334.1*10^6$
x_{11}	1127604	139425	5776355	$0–113.4*10^6$
x_{12}	590741.5	252608	2140671	$0–43.38*10^6$
x_{13}	691051.4	143839	3561062	$0–79.03*10^6$
x_{14}	97479806	22139145	4.51E+08	$8.4*10^4–10*10^9$
x_{15}	11177475	626000	72913011	$0 – 1.45*10^9$
x_{16}	624876.3	113282	2084369	$0–26.52*10^6$
x_{17}	727486	66652	2838835	$0–38.48*10^6$
x_{18}	97028.52	19330.57	415742.3	$0–7.46*10^6$
x_{19}	11119.79	0	82139.4	$0–2.77*10^6$
x_{20}	1192925	26331	7013374	$0–176.91*10^6$
x_{21}	171.1774	55	637.1748	1–11814
x_{22}	26019190	318965	1.8E+08	$549–3.6*10^9$
x_{23}	2047989	50000	6968154	$0–102.6*10^6$
y	20940601	4234476	93139725	$2.8*10^5–2*10^9$