

Modelling of Web Domain Visits by Radial Basis Function Neural Networks and Support Vector Machine Regression

Vladimír Olej, Jana Filipová

Institute of System Engineering and Informatics
Faculty of Economics and Administration, University of Pardubice, Studentská 84
532 10 Pardubice, Czech Republic,
{vladimir.olej@upce.cz, jana.filipova@upce.cz}

Abstract. The paper presents basic notions of web mining, radial basis function (RBF) neural networks and ϵ -insensitive support vector machine regression (ϵ -SVR) for the prediction of a time series for the website of the University of Pardubice. The model includes pre-processing time series, design RBF neural networks and ϵ -SVR structures, comparison of the results and time series prediction. The predictions concerning short, intermediate and long time series for various ratios of training and testing data. Prediction of web data can be benefit for a web server traffic as a complicated complex system.

Keywords: Web mining, radial basis function neural networks, ϵ -insensitive support vector machine regression, time series, prediction.

1 Introduction

Data modelling (prediction, classification, optimization) obtained by web mining [1] from the log files and data on a virtual server, as a complicated complex system that affects these data, there are a few problems worked out. The web server represents a complicated complex system; virtual system usually works with several virtual machines that operate over multiple databases. The quality of the activities of the system also affects the data obtained using web mining. Given virtual system is characterized by its operational parameters, which are changing over time, so it is a dynamic system. The data show a nonlinear characteristics, are heterogeneous, inconsistent, missing and uncertain. Currently there are a number of methods for modelling of the data obtained by web mining. These methods can generally be divided into methods of modelling with unsupervised learning [2,3] and supervised learning [4]. The present work builds on the modelling of web domains visit with uncertainty [5,6].

In the paper is presented a problem formulation with the aim of describing the time series web upce.cz (web presentation visits), including possibilities of pre-processing which is realized by means of simple mathematic-statistic methods. Next, there are introduced basic notions of RBFs [7] neural networks and ϵ -SVRs [4,8] for time

series prediction. Further, the paper includes a comparison of the prediction results of the model designed. The model represents a time series prediction for web upce.cz with the pre-processing done by simple mathematical statistical methods. The prediction is carried out through RBF neural networks and ε -SVRs for different durations of time series and various ratios $O_{\text{train}}:O_{\text{test}}$ training O_{train} and testing O_{test} dates ($O = O_{\text{train}} \cup O_{\text{test}}$). There is expected, that for shorter time series prediction it would be better to use RBF neural networks and for longer ε -SVRs.

2 Problem Formulation

The data for prediction of the time series web upce.cz over a given time period was obtained from Google Analytics. This web mining tool, which makes use of JavaScript code implemented for web presentation, offers a wide spectrum of operation characteristics (web metrics). Metrics provided by Google Analytics can be divided into following sections: visits, sources of access, contents and conversion. In the section 'visits' it can be monitored for example the number of visitors, the number of visits and number of pages viewed as well as the ratio of new and returning visitors. Indicator geolocation, i.e. from which country are visitors most often, is needed to be known because of language mutations, for example. In order to predict the visit rate to the University of Pardubice, Czech Republic website (web upce.cz) it is important monitoring the indicator of the number of visits within a given time period. One 'visit' here is defined as an unrepeated combination of IP address and cookies. A sub-metric is absolutely unique visit defined by an unrepeatability of IP address and cookies within given time period. The basic information obtained from Google Analytics about web upce.cz during May 2009 considered of the following: The total visit rate during given monthly cycles. A clear trend is obvious there, with Monday having the highest visit rate, which in turn decreases as the week progresses; Saturday has the lowest visit rate; The average number of pages visited is more than three; A visitor stays on certain page five and half a minutes on average; The bounce rate is approximately 60%; Visitors generally come directly to the website, which is positive; The favourite pages is the main page, followed by the pages of the Faculty of Economic and Administration and the Faculty of Philosophy.

The measurement of the visit rate of the University of Pardubice web page, (web upce.cz) took place time periods of regular, controlled intervals. The result represents a time series. The pre-processing of data was realized by means of simple mathematic-statistic methods: a simple moving average (SMA), a central moving average (CMA), a moving median (MM) along with simple exponential smoothing (SES) and double exponential smoothing (DES) at time t . Pre-processing was used with aim of smoothing the outliers, while maintaining the physical interpretation of data.

The general formulation of the model of prediction of the visit rate for upce.cz can be stated in thusly: $y' = f(x^t_1, x^t_2, \dots, x^t_m)$, $m=5$, where y' is the number of daily web visits in time $t+1$, y is the number of daily web visits in time t , x^t_1 is SMA, x^t_2 is CMA, x^t_3 is MM, x^t_4 is SES, and x^t_5 is DES at time t . An example of the pre-processing of the time series of web upce.cz is represented in Fig. 1.

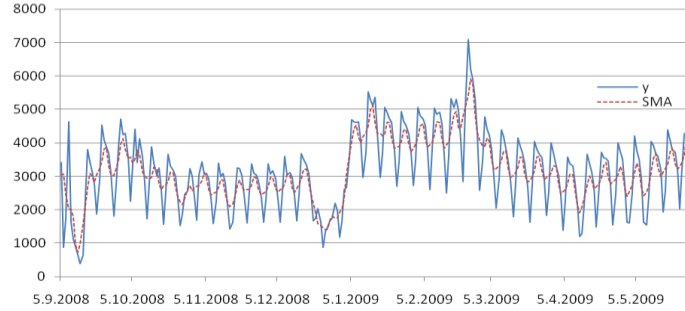


Fig. 1. The pre-processing of web upce.cz visits by SMA

On the basis of the different durational time series of web upce.cz, concrete prediction models for the visit rate to upce.cz web can be defined thusly

$$\begin{aligned}
 y' &= f(\text{TS}_{\text{SMA}}^{\text{S}}, \text{TS}_{\text{CMA}}^{\text{S}}, \text{TS}_{\text{MM}}^{\text{S}}, \text{TS}_{\text{SES}}^{\text{S}}, \text{TS}_{\text{DES}}^{\text{S}}), \\
 y' &= f(\text{TS}_{\text{SMA}}^{\text{I}}, \text{TS}_{\text{CMA}}^{\text{I}}, \text{TS}_{\text{MM}}^{\text{I}}, \text{TS}_{\text{SES}}^{\text{I}}, \text{TS}_{\text{DES}}^{\text{I}}), \\
 y' &= f(\text{TS}_{\text{SMA}}^{\text{L}}, \text{TS}_{\text{CMA}}^{\text{L}}, \text{TS}_{\text{MM}}^{\text{L}}, \text{TS}_{\text{SES}}^{\text{L}}, \text{TS}_{\text{DES}}^{\text{L}}),
 \end{aligned} \tag{1}$$

where TS - time series, S - short TS (264 days), I - intermediate TS (480 days), L - long TS (752 days) at time t .

3 Basic Notions of RBF Neural Networks and ε -SVR

The term RBF neural network [7] refers to any kind of feed-forward neural networks that uses RBF as an activation function. RBF neural networks are based on supervised learning. The output $f(\mathbf{x}, H, \mathbf{w})$ RBF of a neural network can be defined this way

$$f(\mathbf{x}, H, \mathbf{w}) = \sum_{i=1}^q w_i \times h_i(\mathbf{x}), \tag{2}$$

where $H = \{h_1(x), h_2(x), \dots, h_i(x), \dots, h_q(x)\}$ is a set of activation functions RBF of neurons (of RBF functions) in the hidden layer and w_i are synapse weights. Each of the m components of vector $\mathbf{x} = (x_1, x_2, \dots, x_k, \dots, x_m)$ is an input value for the q activation functions $h_i(x)$ of RBF neurons. The output $f(\mathbf{x}, H, \mathbf{w})$ of RBF neural network represents a linear combination of outputs from q RBF neurons and corresponding synapse weights \mathbf{w} .

The activation function $h_i(x)$ of an RBF neural network in the hidden layer belongs to a special class of mathematical functions whose main characteristic is a monotonous rising or falling at an increasing distance from center c_i of the activation function $h_i(x)$ of an RBF. Neurons in the hidden layer can use one of several activation functions $h_i(x)$ of an RBF neural network, for example a Gaussian activation function (a one-dimensional activation function of RBF), a rotary Gaussian activation function (a two-dimensional RBF activation function), multisquare and inverse multisquare activation functions or Cauchy's functions. Results may be presented in this manner

$$h(\mathbf{x}, C, R) = \sum_{i=1}^q \exp\left(-\frac{\|\mathbf{x} - c_i\|^2}{r_i}\right), \quad (3)$$

where $\mathbf{x}=(x_1, x_2, \dots, x_k, \dots, x_m)$ represents the input vector, $C=\{c_1, c_2, \dots, c_i, \dots, c_q\}$ are centres of activation functions $h_i(x)$ an RBF neural network and $R=\{r_1, r_2, \dots, r_i, \dots, r_q\}$ are the radiuses of activation functions $h_i(x)$.

The neurons in the output layer represent only a weighted sum of all inputs coming from the hidden layer. An activation function of neurons in the output layer can be linear, with the unit of the output eventually being convert by jump instruction to binary form. The RBF neural network learning process requires a number of centres c_i of activation function $h_i(x)$ of the RBF neural networks to be set as well as for the most suitable positions for RBF centres c_i to be found. Other parameters are radiuses of centres c_i , rate of activation functions $h_i(x)$ of RBFs and synapse weights $W(q, n)$. These are setup between the hidden and output layers. The design of an appropriate number of RBF neurons in the hidden layer is presented in [4]. Possibilities of centres recognition c_i are mentioned in [7] as a random choice. The position of the neurons is chosen randomly from a set of training data. This approach presumes that randomly picked centres c_i will sufficiently represent data entering the RBF neural network. This method is suitable only for small sets of input data. Use on larger sets, often results in a quick and needless increase in the number of RBF neurons in the hidden layer, and therefore an unjustified complexity of the neural network. The second approach to locating centres c_i of activation functions $h_i(x)$ of RBF neurons can be realized by a K-means algorithm.

In nonlinear regression ε -SVR [4,8,9,10,11] minimizes the loss function $L(d, y)$ with insensitive ε [4,11]. Loss function $L(d, y) = |d - y|$, where d is the desired response and y is the output estimate. The construction of the ε -SVR for approximating the desired response d can be used for the extension of loss function $L(d, y)$ as follows

$$L_\varepsilon(d, y) = \begin{cases} |d - y| - \varepsilon & \text{for } |d - y| \geq \varepsilon \\ 0 & \text{else} \end{cases}, \quad (4)$$

where ε is a parameter. Loss function $L_\varepsilon(d, y)$ is called a loss function with insensitive ε .

Let the nonlinear regression model in which the dependence of the scalar d vector \mathbf{x} expressed by $d=f(\mathbf{x}) + n$. Additive noise n is statistically independent of the input vector \mathbf{x} . The function $f(\cdot)$, and noise statistics are unknown. Next, let the sample training data (\mathbf{x}_i, d_i) , $i=1, 2, \dots, N$, where \mathbf{x}_i and d_i is the corresponding value of the output model d . The problem is to obtain an estimate of d , depending on \mathbf{x} . For further progress is expected to estimate d , called y , which is widespread in the set of nonlinear basis functions $\varphi_j(\mathbf{x})$, $j=0, 1, \dots, m_1$ this way

$$y = \sum_{j=0}^{m_1} w_j \varphi_j(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}), \quad (5)$$

where $\boldsymbol{\varphi}(\mathbf{x})=[\varphi_0(\mathbf{x}), \varphi_1(\mathbf{x}), \dots, \varphi_{m_1}(\mathbf{x})]^T$ and $\mathbf{w}=[w_0, w_1, \dots, w_{m_1}]$. It is assumed that $\varphi_0(\mathbf{x})=1$ in order to the weight w_0 represents bias b . The solution to the problem is to minimize the empirical risk

$$R_{emp} = \frac{1}{N} \sum_{i=1}^N L_{\varepsilon}(d_i, y_i), \quad (6)$$

under conditions of inequality $\|\mathbf{w}\|^2 \leq c_0$, where c_0 is a constant. The restricted optimization problem can be rephrased using two complementary sets of non-negative variables. Additional variables ξ and ξ' describe loss function $L_{\varepsilon}(d,y)$ with insensitivity ε . The restricted optimization problem can be written as an equivalent to minimizing the cost functional

$$\phi(w, \xi, \xi') = C \left(\sum_{i=1}^N (\xi_i + \xi'_i) \right) + \frac{1}{2} w^T w, \quad (7)$$

under the constraints of two complementary sets of non-negative variables ξ and ξ' . The constant C is a user-specified parameter. Optimization problem (7) can be easily solved in the dual form. The basic idea behind the formulation of the dual-shaped structure is the Lagrangian function [9], the objective function and restrictions. Can then be defined Lagrange multipliers with their functions and parameters which ensure optimality of these multipliers. Optimization the Lagrangian function only describes the original regression problem. To formulate the corresponding dual problem a convex function can be obtained (for shorthand)

$$Q(\alpha_i, \alpha'_i) = \sum_{i=1}^N d_i (\alpha_i - \alpha'_i) - \varepsilon \sum_{i=1}^N (\alpha_i + \alpha'_i) - \frac{1}{2} \sum_{i=1}^N \sum_{j=1}^N (\alpha_i - \alpha'_i) (\alpha_j - \alpha'_j) K(x_i, x_j), \quad (8)$$

where $K(x_i, x_j)$ is kernel function defined in accordance with Mercer's theorem [4,8]. Solving optimization problem is obtained by maximizing $Q(\alpha, \alpha')$ with respect to Lagrange multipliers α and α' and provided a new set of constraints, which hereby incorporated constant C contained in the function definition $\phi(w, \xi, \xi')$. Data points covered by the $\alpha \neq \alpha'$, define support vectors [4,8,9].

4 Modelling and Analysis of the Results

The designed model in Fig. 2 demonstrates prediction modelling of the time series web upce.cz. Data pre-processing is carried out by means of data standardization. Thereby, the dependency on units is eliminated. Then the data are pre-processed through simple mathematical statistical methods (SMA, CMA, MM, SES, and DES). Data pre-processing makes the suitable physical interpretation of results possible. The pre-processing is run for time series of different duration namely (TS^S_{SMA}, TS^S_{CMA}, TS^S_{MM}, TS^S_{SES}, TS^S_{DES}), (TS^I_{SMA}, TS^I_{CMA}, TS^I_{MM}, TS^I_{SES}, TS^I_{DES}), (TS^L_{SMA}, TS^L_{CMA}, TS^L_{MM}, TS^L_{SES}, TS^L_{DES}). The prediction is made for the aforementioned pre-processed time series S, I, and L with the help of RBF neural networks, ε -SVR with a polynomial kernel function and ε -SVR with a RBF kernel function for various sets of training O_{train} and testing O_{test} data.

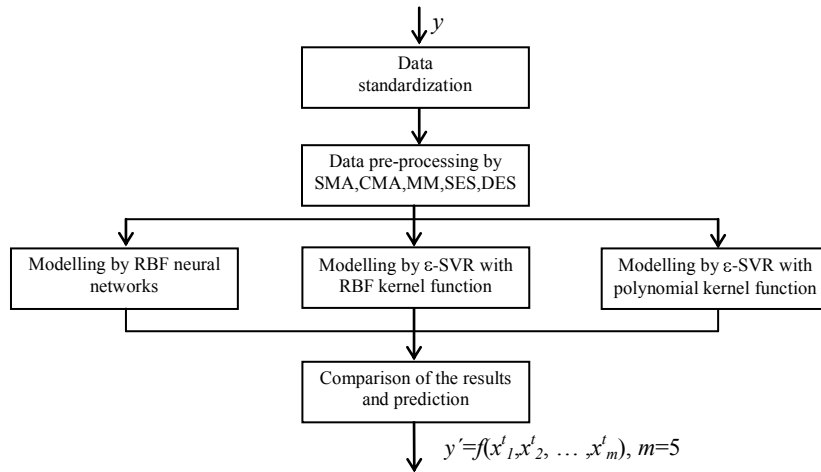


Fig. 2. Prediction modelling of time series for web.upce.cz

In Fig. 3 are shown the dependencies of Root Mean Squared Error (RMSE) on the number of neurons q in the hidden layer. The dependencies RMSE on the parameter μ are represented in Fig. 4. In Fig. 5 the dependencies RMSE are on the parameter ν for training O_{train} and as well as the testing O_{test} set, where: \bullet - O_{train_S} , \blacklozenge - O_{train_L} , \blacksquare - O_{train_L} , \times - O_{test_S} , \blacktriangle - O_{test_L} , $*$ - O_{test_L} . The parameter μ allows for an overrun of the local extreme in the learning process and the following progress of learning. The parameter ν represent the selection of centers RBFs as well as guarantees the correct allocation of neurons in the hidden layer for the given data entering the RBF neural network.

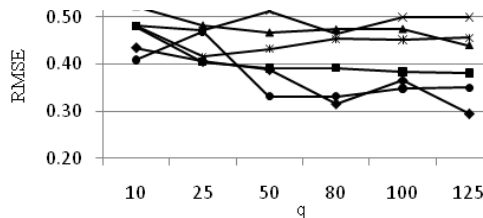


Fig. 3. RMSE dependencies on the parameter q

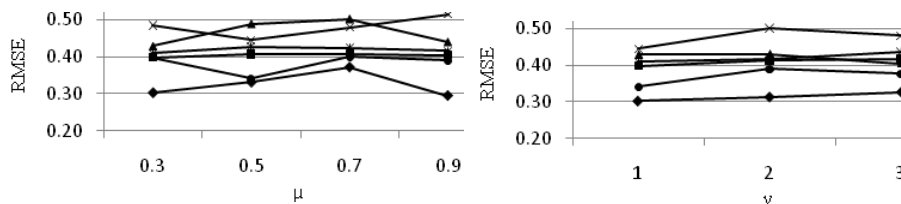


Fig. 4. RMSE dependencies on the parameter μ **Fig. 5.** RMSE dependencies on the parameter ν

Conclusions presented in [4,7,10] are verified by the analysis of results (with 10-fold cross validation). $RMSE_{\text{train}}$ is lowered until it reaches to the value 0.3, when q neurons in the hidden layer increases in size 125 and with the greater ratio $O_{\text{train}}:O_{\text{test}}$.

RMSE_{test} decreases only when the number q RBF neurons increases at a significantly slower ratio than the ratio $O_{\text{train}}:O_{\text{test}}$. Minimum RMSE_{test} moves right with the increasing ratio $O_{\text{train}}:O_{\text{test}}$. Next, determination of the optimal number of q neurons in the hidden layer is necessary. Respectively, a lower number q neurons represents the increasing RMSE_{test}. In Table 1, Table 2, and Table 3 are shown optimized results of the analysis of the experiments for different parameters of the RBF neural networks (with RBF activation function) with different durations of time series, various ratios of $O_{\text{train}}:O_{\text{test}}$ training O_{train} and testing O_{test} of data sets and same amount of learning at $p=600$ cycles. Table 2 builds on the best set of q in Table 1 and Table 3 builds on the best set of q (Table 1) and μ (Table 2).

Table 1. Optimized results of RMSE analysis using the amount of neurons q in the hidden layer (μ and ν are constant for given TS)

TS	$O_{\text{train}}:O_{\text{test}}$	Number q of neurons	μ	ν	RMSE _{train}	RMSE _{test}
S	50:50	80	0.9	1	0.383	0.408
S	66:34	80	0.9	1	0.331	0.463
S	80:20	80	0.9	1	0.343	0.365
I	50:50	125	0.9	1	0.372	0.316
I	66:34	125	0.9	1	0.295	0.440
I	80:20	125	0.9	1	0.352	0.326
L	50:50	25	0.9	1	0.409	0.402
L	66:34	25	0.9	1	0.404	0.415
L	80:20	25	0.9	1	0.409	0.390

Table 2. Optimized results of RMSE analysis using parameter μ (q and ν are constant for given TS)

TS	$O_{\text{train}}:O_{\text{test}}$	Number q of neurons	μ	ν	RMSE _{train}	RMSE _{test}
S	50:50	80	0.5	1	0.473	0.385
S	66:34	80	0.5	1	0.341	0.445
S	80:20	80	0.5	1	0.311	0.408
I	50:50	125	0.3	1	0.313	0.344
I	66:34	125	0.3	1	0.302	0.428
I	80:20	125	0.3	1	0.343	0.365
L	50:50	25	0.3	1	0.400	0.387
L	66:34	25	0.3	1	0.398	0.409
L	80:20	25	0.3	1	0.406	0.391

Table 3. Optimized results of RMSE analysis using parameter ν (q and μ are constant for given TS)

TS	$O_{\text{train}}:O_{\text{test}}$	Number q of neurons	μ	ν	RMSE _{train}	RMSE _{test}
S	50:50	80	0.5	1	0.385	0.473
S	66:34	80	0.5	1	0.341	0.445
S	80:20	80	0.5	1	0.311	0.408
I	50:50	125	0.3	3	0.357	0.375
I	66:34	125	0.3	3	0.327	0.405
I	80:20	125	0.3	3	0.376	0.441
L	50:50	25	0.3	1	0.400	0.387
L	66:34	25	0.3	1	0.398	0.409
L	80:20	25	0.3	1	0.406	0.391

The dependencies RMSE on the parameter C are represented in Fig. 6, the dependencies RMSE on the parameter β in Fig. 7, the dependencies RMSE on the parameter γ in Fig. 8 for training O_{train} as well as testing O_{test} set, where: \bullet - O_{train_S} , \blacklozenge - O_{train_I} , \blacksquare - O_{train_L} , \times - O_{test_S} , \blacktriangle - O_{test_I} , $*$ - O_{test_L} . The parameters C , ε are functions of kernel functions $k(\mathbf{x}, \mathbf{x}_i)$ [4,11] variations. In the learning process ε -SVR are set using 10-fold cross validation. Parameter C controls the trade off between errors of the ε -SVR of training data and margin maximization; ε [4] selects support vectors in the regression structures, and β represents the rate of polynomial kernel function $k(\mathbf{x}, \mathbf{x}_i)$. The coefficient γ characterizes polynomial and RBF kernel function.

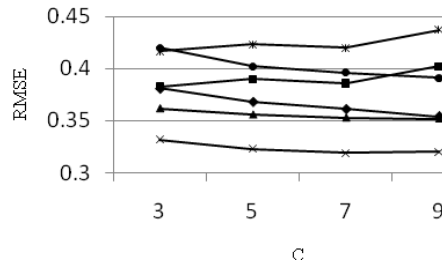


Fig. 6. RMSE dependencies on the parameter C

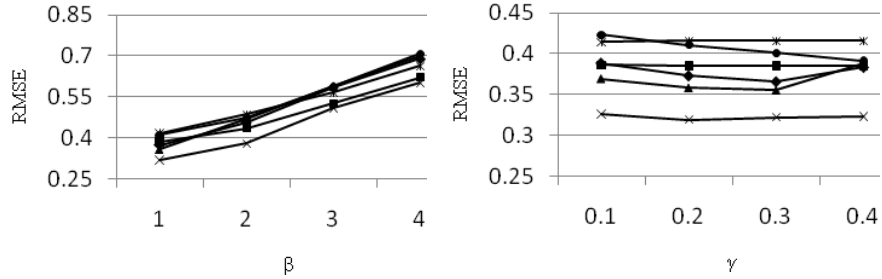


Fig. 7. RMSE dependencies on the parameter β **Fig. 8.** RMSE dependencies on the parameter γ

The confirmation of conclusions presented in [4,11] is verified by the analysis of results. $RMSE_{\text{test}}$ for the ε -SVR with RBF kernel function lowers toward zero value with decreasing C (in the case the user experimentation) and higher ratio of $O_{\text{train}}:O_{\text{test}}$. In the use of 10-fold cross validation, $RMSE_{\text{test}}$ moves towards zero with an increase of parameter γ for O_{train_S} . In the ε -SVR with polynomial kernel function $RMSE_{\text{test}}$ significantly decreases when the parameter β decreases (Fig. 7). Minimum $RMSE_{\text{test}}$ moves right with an increase of the parameter γ (the minimum is between 0.2 to 0.3), whereas an indirect correlation between the ratio $O_{\text{train}}:O_{\text{test}}$ and $RMSE_{\text{test}}$ obtains.

In Table 4 and Table 5 are shown the optimized results of the analysis of the experiments for different parameters of the ε -SVR (with RBF and a polynomial kernel function) with different durations of time series, various ratios $O_{\text{train}}:O_{\text{test}}$ training O_{train} and testing O_{test} of data sets and same amount of learning at $p=600$ cycles. The tables are not represented by the partial results for the various parameters, but only the resulting set of parameters for the TS and the ratio $O_{\text{train}}:O_{\text{test}}$.

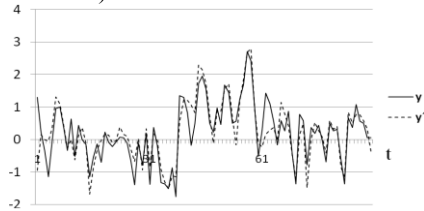
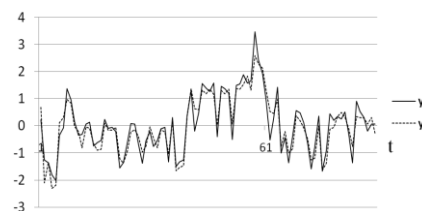
Table 4. Optimized results of RMSE analysis using parameter C

TS	$O_{\text{train}}:O_{\text{test}}$	C	ε	γ	$\text{RMSE}_{\text{train}}$	$\text{RMSE}_{\text{test}}$
S	50:50	8	0.2	0.4	0.343	0.414
S	66:34	10	0.1	0.4	0.365	0.314
S	80:20	10	0.1	0.4	0.355	0.306
I	50:50	10	0.1	0.4	0.331	0.369
I	66:34	10	0.1	0.4	0.342	0.361
I	80:20	9	0.1	0.4	0.350	0.336
L	50:50	6	0.1	0.2	0.382	0.409
L	66:34	4	0.1	0.2	0.383	0.416
L	80:20	4	0.1	0.2	0.384	0.445

Table 5. Optimized results of RMSE analysis using parameters γ and β

TS	$O_{\text{train}}:O_{\text{test}}$	C	ε	γ	β	$\text{RMSE}_{\text{train}}$	$\text{RMSE}_{\text{test}}$
S	50:50	10	0.1	0.2	1	0.411	0.319
S	66:34	10	0.2	0.2	2	0.474	0.379
S	80:20	10	0.1	0.2	3	0.587	0.509
I	50:50	6	0.1	0.3	1	0.385	0.416
I	66:34	10	0.1	0.3	2	0.436	0.486
I	80:20	6	0.1	0.3	3	0.526	0.567
L	50:50	10	0.1	0.2	2	0.455	0.468
L	66:34	6	0.1	0.2	1	0.385	0.416
L	80:20	8	0.1	0.2	1	0.385	0.443

The original value of TS^S (O_{test}) y compared with predicted values of y' (in which time is $t+\Delta t$, $\Delta t=1$ day) using RBF neural network is displayed in Fig. 9. In Fig. 10 TS^S (O_{test}) y is then compared to predicted values y' using the ε -SVR (with RBF kernel function).

**Fig. 9.** TS^S with values predicted by RBF**Fig. 10.** TS^S with values predicted by SVR

In Table 6, there is a comparison of the $\text{RMSE}_{\text{train}}$ and $\text{RMSE}_{\text{test}}$ on the training and testing set to other designed and analyzed structures. For example, it was used a fuzzy inference system (FIS) Takagi-Sugeno [5,6], intuitionistic fuzzy inference system (IFIS) Takagi-Sugeno [5,6], feed-forward neural networks (FFNNs), RBF neural networks and ε -SVR¹ (ε -SVR²) with RBF (polynomial kernel function) with pre-processing input data by simple mathematical statistical methods.

Table 6. Comparison of the $\text{RMSE}_{\text{train}}$ and $\text{RMSE}_{\text{test}}$ on the training and testing data to other designed and analyzed structures of fuzzy inference systems and neural networks

	FIS	IFIS	FFNN	RBF	ε -SVR ¹	ε -SVR ²
$\text{RMSE}_{\text{train}}$	0.221	0.224	0.593	0.311	0.331	0.385
$\text{RMSE}_{\text{test}}$	0.237	0.239	0.687	0.408	0.369	0.416

5 Conclusion

The proposed model consists of data pre-processing and actual prediction using RBF neural networks as well as ε -SVR with polynomial and RBF kernel functions. Furthermore, the modelling was done for time series of different lengths and different parameters of neural networks. The analysis results for various ratios of $O_{\text{train}}:O_{\text{test}}$ show trends for $\text{RMSE}_{\text{train}}$ and $\text{RMSE}_{\text{test}}$. From the analysis of all obtained results of modelling time series by RBF neural network (ε -SVR) shows that $\text{RMSE}_{\text{test}}$ takes minimum values [4,10,11] for TS^1 (TS^S). Further direction of research in the area of modelling web domain visits (excluding the use of uncertainty [5,6] and modelling using RBF neural networks and machine learning using ε -SVR) is focused on different structures of neural networks. The crux of modelling are different lengths of time series, various ratios of $O_{\text{train}}:O_{\text{test}}$ and different techniques of their partitioning.

Prediction using web mining gives better characterization of webspace. On the basis of that system engineers can better characterize the load of complex virtual system and its dynamics. The RBF neural networks (ε -SVR) design was carried out in SPSS Clementine (STATISTICA).

Acknowledgments.

This work was supported by the scientific research project of Ministry of Environment, the Czech Republic under Grant No: SP/4i2/60/07.

References

- [1] Cooley, R., Mobasher, B., Srivistava, J.: Web Mining: Information and Pattern Discovery on the World Wide Web. In: 9th IEEE International Conference on Tools with Artificial Intelligence, ICTAI '97, Newport Beach, CA (1997)
- [2] Krishnapuram, R., Joshi, A., Yi, L.: A Fuzzy Relative of the K-medoids Algorithm with Application to Document and Snippet Clustering. In: IEEE International Conference on Fuzzy Systems, pp.1281-1286. Korea (1999)
- [3] Pedrycz, W.: Conditional Fuzzy C-means. Pattern Recognition Letters, 17, 625-632 (1996)
- [4] Haykin, S.: Neural Networks: A Comprehensive Foundation. Prentice-Hall Inc., New Jersey (1999)
- [5] Olej, V., Hájek, P., Filipová, J.: Modelling of Web Domain Visits by IF-Inference System. WSEAS Transactions on Computers, Issue 10, 9, 1170-1180 (2010)
- [6] Olej, V., Filipová, J., Hájek, P.: Time Series Prediction of Web Domain Visits by IF-Inference System. In: Proc. of the 14th WSEAS International Conference on Systems, Latest Trends on Computers, N. Mastorakis at al. (eds.), Vol.1, Greece, pp.156-161 (2010)
- [7] Broomhead, D.S., Lowe, D.: Multivariate Functional Interpolation and Adaptive Networks. Complex Systems, 2, 321-355 (1988)
- [8] Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and other Kernel-based Learning Methods. Cambridge University Press, Cambridge (2000)
- [9] Smola, A., Scholkopf, J.: A Tutorial on Support Vector Regression. Statistics and Computing, 14, 199-222 (2004)
- [10] Niyogi, P., Girosi, F.: On the Relationship between Generalization Error, Hypothesis Complexity, and Sample Complexity for Radial Basis Functions. Massachusetts Institute of Technology Artificial Intelligence Laboratory, Massachusetts (1994)
- [11] Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer-Verlag New York (1995)