Adaptive Ensemble Models of Extreme Learning Machines for Time Series Prediction*

Mark van Heeswijk[1,3,**], Yoan Miche[1,2], Tiina Lindh-Knuutila[1,4], Peter A.J. Hilbers[3], Timo Honkela[1], Erkki Oja[1], and Amaury Lendasse[1]

[1] Adaptive Informatics Research Centre, Helsinki University of Technology
P.O. Box 5400, 02015 TKK - Finland
heeswijk@cis.hut.fi
[2] INPG Grenoble - Gipsa-Lab, UMR 5216
961 rue de la Houille Blanche, Domaine Universitaire, 38402 Grenoble - France
[3] Eindhoven University of Technology
Den Dolech 2, P.O. Box 513, 5600 MB Eindhoven - The Netherlands
[4] International Computer Science Institute of University of California,
947 Center Street, Suite 600, Berkeley, CA 94704 - USA

Abstract. In this paper, we investigate the application of adaptive ensemble models of Extreme Learning Machines (ELMs) to the problem of one-step ahead prediction in (non)stationary time series. We verify that the method works on stationary time series and test the adaptivity of the ensemble model on a nonstationary time series. In the experiments, we show that the adaptive ensemble model achieves a test error comparable to the best methods, while keeping adaptivity. Moreover, it has low computational cost.

Keywords: time series prediction, sliding window, extreme learning machine, ensemble models, nonstationarity, adaptivity.

1 Introduction

Time series prediction is a challenge in many fields. In finance, experts predict stock exchange courses or stock market indices; data processing specialists predict the flow of information on their networks; producers of electricity predict the load of the following day [1,2]. The common question in these problems is how one can analyze and use the past to predict the future.

A common assumption in the field of time series prediction is that the underlying process generating the data is stationary and that the data points are independent and identically distributed (IID). Under this assumption, the training data is generally a good indication for what data to expect in the test phase.

However, a large number of application areas of prediction involve nonstationary phenomena. In these systems, the IID assumption does not hold since the

* This work was supported by the Academy of Finland Centre of Excellence, Adaptive Informatics Research Centre.

** Corresponding author.

C. Alippi et al. (Eds.): ICANN 2009, Part II, LNCS 5769, pp. 305–314, 2009.

system generating the time series changes over time. Therefore, contrary to the stationary case, one cannot assume that one can use what has been learned from past data and one has to keep learning and adapting the model as new samples arrive. Possible ways of doing this include: 1) retraining the model repeatedly on a finite window of past values and 2) using a combination of different models, each of which is specialized on part of the state space.

Besides the need to deal with nonstationarity, another motivation for such an approach is that one can drop stationarity requirements on the time series. This is very useful, since often we cannot assume anything about whether or not a time series is stationary.

To construct the ensemble model presented in this paper, a number of Extreme Learning Machines (ELMs) [3] of varying complexity are generated, each of which is individually trained on the data. After training, these individual models are combined in an ensemble model. The output of the ensemble model is a weighted linear combination of the outputs of the individual models. During the test phase, the ensemble model adapts this linear combination over time with the goal of minimizing the prediction error: whenever a particular model has bad prediction performance (relative to the other models) its weight in the ensemble is decreased, and vice versa. A detailed description can be found in Section 2.3.

In the first experiment, we test the performance of this adaptive ensemble model in repeated one-step ahead prediction on a time series that is known to be stationary (the Santa Fe A Laser series [4]). The main goal of this experiment is to test the robustness of the model and to investigate the different parameters influencing the performance of the model. In the second experiment, the model is applied to another time series (Quebec Births [5]) which is nonstationary and more noisy than the Santa Fe time series.

Ensemble methods have been applied in various forms (and under various names) to time series prediction, regression and classification. A non-exhaustive list of literature that discusses the combination of different models into a single model includes *bagging* [6], *boosting* [7], *committees* [8], *mixture of experts* [9], *multi-agent systems for prediction* [10], *classifier ensembles* [11], among others. Out of these examples, our work is most closely related to [10], which describes a multi-agent system prediction of financial time series and recasts prediction as a classification problem. Other related work includes [11], which deals with classification under concept drift (nonstationarity of classes). The difference is that both papers deal with classification under nonstationarity, while we deal with regression under nonstationarity.

In Section 2, the theory of ensemble models and the ELM are presented, as well as how we combine both of them in the adaptive ensemble method. Section 3 describes the experiments, the datasets used and discusses the results.

2 Methodology

2.1 Ensemble Models

In ensemble methods, several individual models are combined to form a single new model. Commonly, this is done by taking the average or a weighted average

of the individual models, but other combination schemes are also possible [11]. For example, one could take the best n models and take a linear combination of those. For an overview of ensemble methods, see [8].

Ensemble methods rely on having multiple good models with sufficiently uncorrelated error. The individual models are typically combined into a single ensemble model as follows:

$$\hat{y}_{ens}(t) = \frac{1}{m} \sum_{i=1}^{m} \hat{y}_i(t)), \tag{1}$$

where $\hat{y}_{ens}(t)$ is the output of the ensemble model, $\hat{y}_i(t)$ are the outputs of the individual models and m is the number of models.

Following [8], it can be shown that the variance of the ensemble model is lower than the average variance of all the individual models:

Let $y(t)$ denote the true output that we are trying to predict and $\hat{y}_i(t)$ the estimation for this value of model i. Then, we can write the output $\hat{y}_i(t)$ of model i as the true value $y(t)$ plus some error term $\epsilon_i(t)$:

$$\hat{y}_i(t) = y(t) + \epsilon_i(t). \tag{2}$$

Then the expected square error of a model becomes

$$\mathbb{E}[\{\hat{y}_i(t) - y(t)\}^2] = \mathbb{E}[\epsilon_i(t)^2]. \tag{3}$$

The average error made by a number of models is given by

$$E_{avg} = \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}[\epsilon_i(t)^2]. \tag{4}$$

Similarly, the expected error of the ensemble as defined in Equation 1 is given by

$$E_{ens} = \mathbb{E}\Big[\Big\{\frac{1}{m} \sum_{i=1}^{m} \hat{y}_i(t) - y(t)\Big\}^2\Big] = \mathbb{E}\Big[\Big\{\frac{1}{m} \sum_{i=1}^{m} \epsilon_i(t)\Big\}^2\Big]. \tag{5}$$

Assuming the errors $\epsilon_i(t)$ are uncorrelated (i.e. $E[\epsilon_i(t)\epsilon_j(t)] = 0$) and have zero mean ($E[\epsilon_i(t)] = 0$), we get

$$E_{ens} = \frac{1}{m} E_{avg}. \tag{6}$$

Note that these equations assume completely uncorrelated errors between the models, while in practice errors tend to be highly correlated. Therefore, errors are often not reduced as much as suggested by these equations, but can be improved by using ensemble models. It can be shown that $E_{ens} < E_{avg}$ always holds. Note that this only tells us that the test error of the ensemble is smaller than the average test error of the models, and that it is not necessarily better than the best model in the ensemble. Therefore, the models used in the ensemble should be sufficiently accurate.

2.2 ELM

The ELM algorithm is proposed by Guang-Bin Huang *et al.* in [3] and makes use of Single-Layer Feedforward Neural Networks (SLFN). The main concept behind ELM lies in the random initialization of the SLFN weights and biases. Under the condition that the transfer functions in the hidden layer are infinitely differentiable, the optimal output weights for a given training set can be determined analytically. The obtained output weights minimize the square training error. The trained network is thus obtained in very few steps and is very fast to train, which is why we use them in the adaptive ensemble model.

Below, we review the main concepts of ELM as presented in [3]. Consider a set of M distinct samples $(\mathbf{x}_i, y_i)$ with $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \mathbb{R}$; then, a SLFN with N hidden neurons is modeled as the following sum

$$\sum_{i=1}^{N} \beta_i f(\mathbf{w}_i \mathbf{x}_j + b_i), j \in [1, M], \tag{7}$$

with f being the activation function, $\mathbf{w}_i$ the input weights to the i^{th} neuron in the hidden layer, b_i the biases and β_i the output weights.

In the case where the SLFN would perfectly approximate the data (meaning the error between the output $\hat{y_i}$ and the actual value y_i is zero), the relation is

$$\sum_{i=1}^{N} \beta_i f(\mathbf{w}_i \mathbf{x}_j + b_i) = y_j, j \in [1, M], \tag{8}$$

which can be written compactly as

$$\mathbf{H}\beta = \mathbf{Y}, \tag{9}$$

where $\mathbf{H}$ is the hidden layer output matrix defined as

$$\mathbf{H} = \begin{pmatrix} f(\mathbf{w}_1 \mathbf{x}_1 + b_1) & \cdots & f(\mathbf{w}_N \mathbf{x}_1 + b_N) \\ \vdots & \ddots & \vdots \\ f(\mathbf{w}_1 \mathbf{x}_M + b_1) & \cdots & f(\mathbf{w}_N \mathbf{x}_M + b_N) \end{pmatrix}, \tag{10}$$

and $\beta = (\beta_1 \dots \beta_N)^T$ and $\mathbf{Y} = (y_1 \dots y_N)^T$.

Given the randomly initialized first layer of the ELM and the training inputs $\mathbf{x}_i \in \mathbb{R}^d$, the hidden layer output matrix $\mathbf{H}$ can be computed. Now, given $\mathbf{H}$ and the target outputs $y_i \in \mathbb{R}$ (i.e. $\mathbf{Y}$), the output weights β can be solved from the linear system defined by Equation 9. This solution is given by $\beta = \mathbf{H}^{\dagger}\mathbf{Y}$, where $\mathbf{H}^{\dagger}$ is the Moore-Penrose generalized inverse of the matrix $\mathbf{H}$ [12]. This solution for β is the unique least-squares solution to the equation $\mathbf{H}\beta = \mathbf{Y}$. Overall, the ELM algorithm then is:

Algorithm 1. ELM

Given a training set $(\mathbf{x}_i, y_i), \mathbf{x}_i \in \mathbb{R}^d, y_i \in \mathbb{R}$, an activation function $f : \mathbb{R} \mapsto \mathbb{R}$ and N the number of hidden nodes,

1: - Randomly assign input weights $\mathbf{w}_i$ and biases b_i, $i \in [1, N]$;
2: - Calculate the hidden layer output matrix $\mathbf{H}$;
3: - Calculate output weights matrix $\beta = \mathbf{H}^{\dagger}\mathbf{Y}$.

Theoretical proofs and a more thorough presentation of the ELM algorithm are detailed in the original paper [3].

2.3 Adaptive Ensemble Model of ELMs

When creating a model to solve a certain regression or classification problem, it is unknown in advance what the optimal model complexity and architecture is. Also, we cannot always assume stationarity of the process generating the data (i.e. in cases where the IID assumption does not hold). Therefore, since the information that has been gathered from past samples can become inaccurate, it is needed to keep learning and keep adapting the model once new samples become available. Possible ways of doing this include: 1) retraining the model repeatedly on a finite window into the past and 2) use a combination of different models, each of which is specialized on part of the state space. In this paper, we employ both strategies in repeated one-step ahead prediction on (non)stationary time series. On the one hand, we use diverse models and adapt the weights with which these models contribute to the ensemble. On the other hand, we retrain the individual models on a limited number of past values (sliding window) or on all known values (growing window).

Adaptation of the Ensemble. The ensemble model consists of a number of randomly initialized ELMs, which each have their own parameters (details are discussed in the next subsection). The model ELM_i has an associated weight w_i which determines its contribution to the prediction of the ensemble. Each ELM is individually trained on the training data and the outputs of the ELMs contribute to the output $\hat{y}_{ens}$ of the ensemble as follows: $\hat{y}_{ens}(t+1) = \sum_i w_i \hat{y}_i(t+1)$.

Once initial training of the models on the training set is done, repeated one-step ahead prediction on the 'test' set starts. After each time step, the previous predictions $\hat{y}_i(t-1)$ are compared with the real value $y(t-1)$. If the square error $\epsilon_i(t-1)^2$ of ELM_i is larger than the average square error of all models at time step $t-1$, then the associated ensemble weight w_i is decreased, and vice versa. The rate of change can be scaled with a parameter α, called the learning rate. Furthermore, the rate of change is normalized by the number of models and the variance of the time series, such that we can expect similar behaviour on time series with different variance and ensembles with a different number of models. The full algorithm can be found in Algorithm 2.

Adaptation of the Models. As described above, ELMs are used in the ensemble model. Each ELM has a random number of input neurons, random number of hidden neurons, and random variables of the regressor as input.

Besides changing the ensemble weights w_i as a function of the errors of the individual models at every time step, the models themselves are also retrained. Before making a prediction at time step t, each model is either retrained on a past window of n values $(\mathbf{x}_i, y_i)_{t-n}^{t-1}$ (sliding window), or on all values known so far $(\mathbf{x}_i, y_i)_{1}^{t-1}$ (growing window). Details on how this retraining fits in with the rest of the ensemble can be found in Algorithm 2.

As mentioned in Section 2.2, ELMs are very fast to train. In order to further speed up the retraining of the ELMs, we make use of PRESS statistics, which allow you to add and remove samples from the training set of a linear model and give you the linear model that you would have obtained, had you trained it on the modified training set. Since an ELM is essentially a linear model of the responses of the hidden layer, PRESS statistics can be applied to (re)train the ELM in an incremental way. A detailed discussion of incremental training of ELMs with PRESS statistics falls outside the scope of this paper, but details on PRESS statistics can be found in [13].

Algorithm 2. Adaptive Ensemble of ELMs

Given a set $(\mathbf{x}(t), y(t)), \mathbf{x}(t) \in \mathbb{R}^d, y(t) \in \mathbb{R}$, and m models,

Create m random ELMs: $(ELM_1 \ldots ELM_m)$
Train each of the ELMs individually on the training data
Initialize each w_i to $\frac{1}{m}$
while $t < t_{end}$ **do**
  generate predictions $\hat{y}_i(t+1)$
  $\hat{y}_{ens}(t+1) = \sum_i w_i \hat{y}_i(t+1)$
  t = t + 1
  compute all errors $\rightarrow \epsilon_i(t-1) = \hat{y}_i(t-1) - y(t-1)$
  for i = 1 to #models **do**
    $\Delta w_i = -\epsilon_i(t-1)^2 + mean(\epsilon(t-1)^2)$
    $\Delta w_i = \Delta w_i \cdot \alpha / (\#models \cdot \mathbf{var}(y))$
    $w_i = \max(0, w_i + \Delta w_i)$
    Retrain ELM_i
  end for
  renormalize weights $\rightarrow \mathbf{w} = \mathbf{w} / ||\mathbf{w}||$
end while

3 Experiments

3.1 Experiment 1: Stationary Time Series

The Santa Fe Laser Data time series [4] has been obtained from a far-infrared-laser in a chaotic state. This time series has become a well-known benchmark

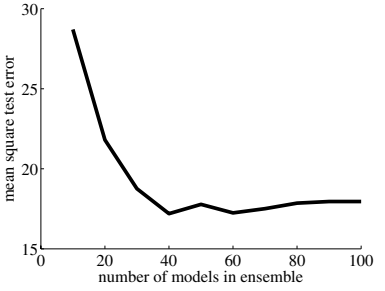

Fig. 1. MSE_{test} of ensemble on laser time series for varying number of models (no window retraining, learning rate 0.1)

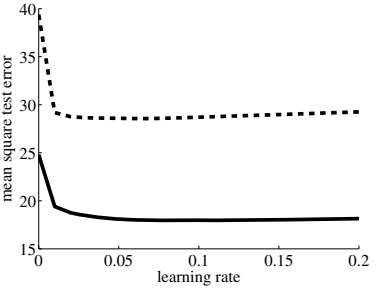

Fig. 2. MSE_{test} of ensemble on laser time series as a function of learning rate (no window retraining), for 10 models (*dotted line*) and 100 models (*solid line*)

in time series prediction since the Santa Fe competition in 1991. It consists of approximately 10000 points and the time series is known to be stationary.

The adaptive ensemble model is trained on the first 1000 values of the time series, after which sequential one-step ahead prediction is performed on the following 9000 values. This experiment is repeated for various combinations of learning rate α and number of models in the ensemble. Each ELM has a regressor size of 8 (of which 5 to 8 variables are randomly selected) and between 150 and 200 hidden neurons with a sigmoid transfer function.

Figure 1 shows the effect of the number of models on the prediction accuracy. It can be seen that the number of models strongly influences the prediction accuracy and that at least 60 models are needed to get good prediction accuracy. Figure 2 shows the effect of the learning rate on the prediction accuracy. The influence of the various (re)training strategies can be found in Table 1. This table also shows that the method is able to achieve a prediction accuracy comparable to the best methods [14].

Table 1. MSE_{test} of ensemble for laser (training window size 1000)

		retraining		
learning rate	#models	none	sliding	growing
0.0	10	39.39	58.56	34.16
0.1	10	28.70	37.93	18.42
0.0	100	24.80	33.85	20.99
0.1	100	17.96	27.30	**14.64**

Figures 3 and 4 show the adaptation of some of the ensemble weights over the length of the entire prediction task.

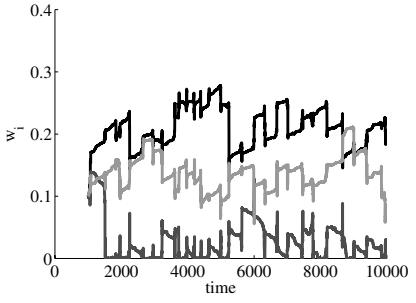

Fig. 3. plot showing part of the ensemble weights w_i adapting over time during sequential prediction on laser time series (#models=10, learning rate=0.1, no window retraining)

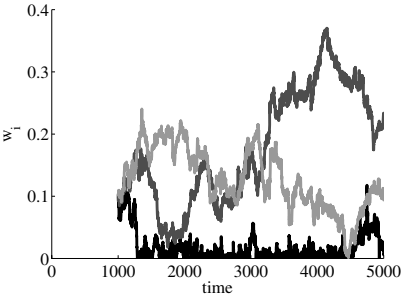

Fig. 4. plot showing part of the ensemble weights w_i adapting over time during sequential prediction on qbirths time series (#models=10, learning rate=0.1, no window retraining)

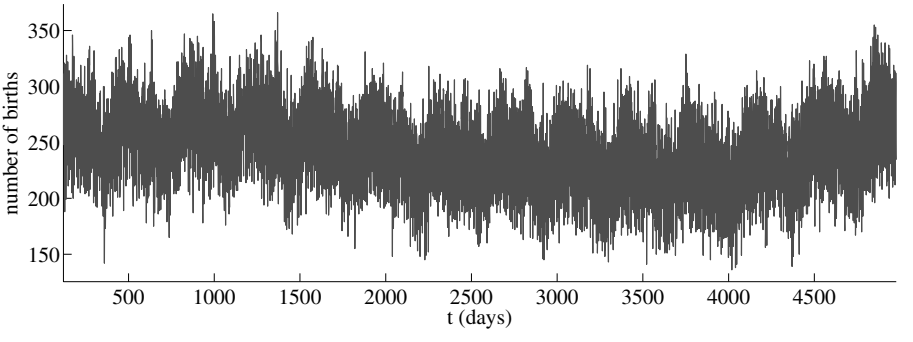

Fig. 5. The Quebec Births time series

3.2 Experiment 2: Nonstationary Time Series

The Quebec Births time series [5] consists of the number of daily births in Quebec over the period of January 1, 1977 to December 31, 1990. It consists of approximately 5000 points, is nonstationary and more noisy than the Santa Fe Laser Data.

The adaptive ensemble model is trained on the first 1000 values of the time series, after which sequential one-step ahead prediction is performed on the following 5000 values. This experiment is repeated for varying learning rates α and numbers of models. Each ELM has a regressor size of 14 (of which 12 to 14 variables are randomly selected) and between 150 and 200 hidden neurons.

Figure 6 shows the effect of the number of models on the prediction accuracy. It can be seen that the number of models strongly influences the prediction accuracy, as was the case with the Santa Fe time series. However, we need more

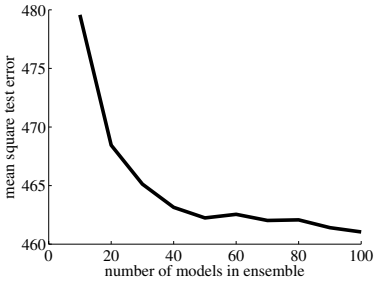

Fig. 6. MSE_{test} of ensemble on Quebec Births time series for varying number of models (sliding window retraining, learning rate 0.1)

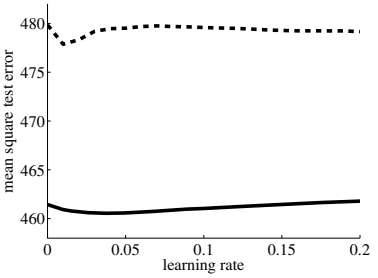

Fig. 7. MSE_{test} of ensemble on Quebec Births time series as a function of learning rate (retraining with sliding window of size 1000), for 10 models (*dotted line*) and 100 models (*solid line*)

Table 2. MSE_{test} of ensemble for Quebec Births (training windows size 1000)

		retraining		
learning rate	#models	none	sliding	growing
0.0	10	594.04	479.84	480.97
0.1	10	582.09	479.58	476.87
0.0	100	585.53	**461.44**	469.79
0.1	100	567.62	**461.04**	468.51

models in order to get a good prediction accuracy. Figure 7 shows the effect of the learning rate on the prediction accuracy. The influence of the various (re)training strategies can be found in Table 2.

3.3 Discussion

The experiments clearly show that it is important to have a sufficient number of models (more is generally better). Furthermore, the shape of the learning rate graph is independent of the number of models, which means that these parameters can probably be optimized independently from each other. We are currently performing a more thorough statistical analysis for determining the best strategy for optimizing the parameters. However, the results suggest that choosing the number of models high and choosing a sufficiently large learning rate (i.e. $\alpha = 0.1$) is a good and robust strategy.

The results also show that the proposed adaptive ensemble method is able to achieve a prediction accuracy comparable to the best methods [14], and is able to do so for both stationary and nonstationary series. An added advantage of the method is that it is adaptive and has low computational cost.

4 Conclusions

We have presented an adaptive ensemble model of Extreme Learning Machines (ELMs) for use in repeated one-step ahead prediction. The model has been tested on both stationary and nonstationary time series, and these experiments show that in both cases the adaptive ensemble method is able to achieve a prediction accuracy comparable to the best methods. An added advantage of the method is that it is adaptive and has low computational cost. Furthermore, the results suggest that we can make good guesses for the parameters of the method (i.e. choose number of models sufficiently large and choose learning parameter $\alpha = 0.1$). We are currently performing more thorough statistical analysis of the model, in order to determine the best strategy for optimizing the parameters. In addition, we would like to extend the model to include other models like OP-ELM [15] and investigate how we can guide adding new models to the ensemble in an online fashion, in order to introduce an extra degrees of adaptivity.

References

1. Sorjamaa, A., Hao, J., Reyhani, N., Ji, Y., Lendasse, A.: Methodology for long-term prediction of time series. Neurocomputing 70(16-18), 2861–2869 (2007)
2. Simon, G., Lendasse, A., Cottrell, M., Fort, J.-C., Verleysen, M.: Time series forecasting: Obtaining long term trends with self-organizing maps. Pattern Recognition Letters 26(12), 1795–1808 (2005)
3. Huang, G.-B., Zhu, Q.-Y., Siew, C.-K.: Extreme learning machine: Theory and applications. Neurocomputing 70(1-3), 489–501 (2006)
4. Weigend, A., Gershenfeld, N.: Time Series Prediction: Forecasting the Future and Understanding the Past. Addison-Wesley, Reading (1993)
5. Quebec Births Data, `http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/misc/qbirths.dat`
6. Breiman, L.: Bagging predictors. In: Machine Learning, pp. 123–140 (1996)
7. Schapire, R.E., Freund, Y., Bartlett, P., Lee, W.S.: Boosting the margin: a new explanation for the effectiveness of voting methods. The Annals of Statistics 26, 322–330 (1998)
8. Bishop, C.M.: Pattern Recognition and Machine Learning (Information Science and Statistics). Springer, Secaucus (2006)
9. Jacobs, R.A., Jordan, M.I., Nowlan, S.J., Hinton, G.E.: Adaptive mixtures of local experts. Neural Computation 3, 79–87 (1991)
10. Raudys, S., Zliobaite, I.: The multi-agent system for prediction of financial time series. In: Rutkowski, L., Tadeusiewicz, R., Zadeh, L.A., Żurada, J.M. (eds.) ICAISC 2006. LNCS (LNAI), vol. 4029, pp. 653–662. Springer, Heidelberg (2006)
11. Kuncheva, L.I.: Classifier ensembles for changing environments. MCS, 1–15 (2004)
12. Rao, C.R., Mitra, S.K.: Generalized Inverse of Matrices and Its Applications. John Wiley & Sons Inc., Chichester (1972)
13. Myers, R.H.: Classical and Modern Regression with Applications, 2nd edn. Duxbury, Pacific Grove (1990)
14. Suykens, J.A.K., Gestel, T.V., Brabanter, J.D., Moor, B.D., Vandewalle, J.: Least Squares Support Vector Machines. World Scientific, Singapore (2002)
15. Miche, Y., Sorjamaa, A., Lendasse, A.: OP-ELM: Theory, experiments and a toolbox. In: Kůrková, V., Neruda, R., Koutník, J. (eds.) ICANN 2008, Part I. LNCS, vol. 5163, pp. 145–154. Springer, Heidelberg (2008)