

# Prediction of Power Load Demand Using Modified Dynamic Weighted Majority Method

Radoslav Nemeč<sup>(✉)</sup>, Viera Rozinajová, and Marek Lóderer

Faculty of Informatics and Information Technologies, Slovak University  
of Technology, Ilkovičová 2, 841 04 Bratislava, Slovakia  
{xnemečr, viera.rozinajova, marek\_loderer}@stuba.sk

**Abstract.** The paper deals with the prediction of electricity demand, using data from smart meters obtained in defined time steps. We propose the modification of ensemble learning method called Dynamic Weighted Majority (DWM). The data are represented by data streams. According to our experiments, the proposed solution offers favorable alternative to current solutions. We also focus on the comparison of proposed ensemble method with single predictions used in the model.

**Keywords:** Prediction · Data streams · Ensemble learning · Dynamic weighted majority · Smart meter

## 1 Introduction

Predicting future values is a topic that fascinates mankind for decades. Man is a curious creature and his interest in knowing the future directly or indirectly affects his life. The known and popular applications of prediction are the weather forecasting prediction of growth or depreciation of finance. However, an increasingly popular area of prediction is power load demand, which is the main goal of this paper. Due to non-stationarity and special characteristic of dynamicity, the prediction of electricity loads is a difficult problem. Moreover, predicting electricity loads is very important for suppliers as well as for its customers. If it is precise, it could help companies to create major decisions about its purchase. Using the correct prediction could also give customers an opportunity to schedule their energy consumption, which may affect the saving of finance.

In this paper we will focus on different approaches of machine learning and data mining algorithms. The main part will describe the proposed approach, which is based on ensemble learning method called Dynamic Weighted Majority [12]. This method was modified to fit and also solve our prediction problem. Ensemble learning approaches are nowadays becoming very popular and currently they are subject to many research works in various cases of study. Their relevance and achieved results are one of the main motivating factors for their usage in our work.

This paper is organized as follows: Sect. 2 contains a summary of the related work. In Sect. 3 we describe our proposed approach (the Modified Dynamic Weighted Majority method and Dynamic Weighted Majority method with decomposition and

parameter optimization). Description of used datasets, experimental evaluation and results are presented in Sect. 4 and the conclusion is in Sect. 5.

## 2 Related Work

The prediction is based on the values measured in the past, which are used to predict the future values. To achieve the precise prediction, it is very important to adjust the prediction to the problem, which is to be solved. The load prediction, also known as load forecasting, could be in general divided into three categories: short-term, medium-term and long-term forecasts [4, 20]. The type of prediction is one of the facts, which had to be considered before implementing the solution. The second important fact, which should be considered, is related to the type of learning we want to use. There are two known types of frequently used learning algorithms for the prediction. The first one is called offline learning. In this case, the learning methods are basically focused on datasets which are static. The arrival of any other data after training is not expected [14]. The electricity load data are formed as data streams and they arrive in specified time intervals meaning that the size of this dataset grows continuously. This fact led us to analyze and use the second type of learning method that is called online learning. Methods for this type of learning could provide i.e. lifelong learning. This means that the models can obtain new information that can consequently be used to adapt to new changes in data stream [14].

Over the last decades various online and offline models for improving electricity load prediction accuracy have been developed. For example regression models [25], methods designed for time series, ARMA and ARIMA models [16] or Holt-Winters exponential smoothing [21]. In recent decade, intelligent forecasting models like Support Vector Regression, Artificial Neural Networks or Expert Systems have been developed and used [8, 20].

Till now, there is no single model which is capable to provide best forecasting results for every kind of data. Big effort has been spend to overcome this issue.

One of the most popular methodologies used to solve this issue is called Ensemble Learning [13]. The main idea of Ensemble Learning is that proper combination of predictions of different base models can create more accurate result in comparison to the result provided by the best individual model.

Recently, an interesting example of Ensemble Learning was proposed. It used a cuckoo search algorithm to find optimal weights to combine four forecasting models based on different types of neural networks [24]. Each neural network forecasts electric load demand based on historical load data. The forecasting results of the combined models were significantly improved compared to the results of the individual models.

In work [18] a Pattern Forecasting Ensemble Model (PFEM) for electricity demand time series is proposed. It is based on the previous PSF algorithm, but uses a combination of five separate clustering models. Published results indicate that proposed approach gives more accurate results compared to all the other five individual models.

Another approach is proposed in work [7], where an ensemble of online regression and option trees is introduced. Mendes-Moreira et al. provide an exhausting research on topic of ensemble approaches for Ensemble [13].

Other successful implementation of Ensemble Learning approach is Dynamic Weighted Majority (DWM). It was introduced by Kolter and Maloof in 2003 [12]. The method was presented as a new ensemble method for tracking concept drifts in data streams. It was mainly designed to solve the classification problems. In our work, we adjust the DWM method to solve regression problems e.g. prediction of electricity loads. During the analysis of available solutions we also found a modification of DWM that is called Additive Expert Ensembles (AddExp) [11]. This modified method became also one of the methods which are included in our proposed solution.

### 3 Proposed Method

We propose the Modified Dynamic Weighted Majority method for time series prediction. This approach was chosen for its ability to adapt to changes in the distribution of a target variable in time series data. It has a potential to obtain more accurate prediction results than a single base prediction method.

#### 3.1 Modified Dynamic Weighted Majority Method

Authors of the original DWM describe the main functionality of the method by four mechanisms: (1) models of the ensemble are trained based on their performance, (2) each model of ensemble is weighted, (3) models are removed from ensemble based on their performance, and (4) new models are added to the ensemble based on global performance i.e. the performance of the whole ensemble. As mentioned in previous chapter, DWM method was until now mainly used for solving classification problems and we could not find any described modification of this method for the prediction of electricity loads. So we decided to modify it to fit to our problem [12]. Following pseudo-code represents the simplified modified version of DWM method, which we have used in our implementation.

```

1. Choose_Experts() //choose, set initial experts
2.  $e_m \leftarrow \text{Train\_Expert}(\text{train}_1)$ 
3.  $\text{pred}_m \leftarrow \text{Predict}(e_m)$ 
4. for  $i \leftarrow 2, \dots, n$  //loop over examples
5.   for  $j \leftarrow 1, \dots, m$  //loop over experts
6.      $\text{error} \leftarrow \text{Mape}(\text{pred}_j, \text{real}_{i-1})$  //evaluate prediction
7.     if ( $\text{error} \geq \gamma$ ) //modify expert weight
8.        $w_j \leftarrow \text{Decrease\_Expert\_Weight}(w_j, \beta)$ 
9.     else
10.       $w_j \leftarrow \text{Increase\_Expert\_Weight}(w_j, \beta)$ 
11.    end;
12.  end;
13.   $\Lambda \leftarrow \text{Integratation}(\text{pred}, w)$  //compute the global prediction
14.  if( $i \bmod p = 0$ )
15.     $r \leftarrow \text{Remove\_Experts}(e, \theta)$  //remove "weak" experts
16.    for  $k \leftarrow 1, \dots, r$  //replace removed experts
17.       $e_k \leftarrow \text{Create\_New\_Expert}$ 
18.    end;
19.  end;
20.  for  $j \leftarrow 1, \dots, m$  //train experts on next example
21.     $e_j \leftarrow \text{Train\_Expert}(\text{train}_i)$ 
22.     $\text{pred}_j \leftarrow \text{Predict}(e_j)$ 
23.  end;
24.  output  $\Lambda$ 
25. end;

```

The pseudo-code of DWM method use the following parameters:

- $\beta$  – factor for increasing/decreasing model weight
- $p$  – number of iterations between the models removal and creation
- $\Lambda, \lambda$  – global prediction of ensemble and local prediction of model
- $w_j$  – actual weight of model  $j$
- $\text{win}$  – predefined length of sliding window (e.g. number of days for training)
- $\text{train}_i$  – training data chunk of size  $\text{win}$  for iteration  $i$
- $\text{real}_{i-1}$  – data chunk of real values from previous iteration  $i$
- $\text{pred}_m$  – predicted values from model  $m$
- $\theta$  – threshold for removing experts according to their actual weights
- $\gamma$  – parameter, which represents a threshold for an acceptance of expert prediction.

The DWM method begins with initialization of prediction models (lines 1–3). During the initialization, the method sets the initial weights, trains chosen models on the first data chunk and obtains the prediction. After the initialization phase, the method continues in two loop cycles. The outer loop goes through the data chunks of incoming

data stream and the inner loop passes through all models of the ensemble. For the evaluation of local prediction (line 6), we used the mean absolute percentage error (MAPE), which represents a frequently used prediction accuracy measure. MAPE is defined by Eq. (1).

$$MAPE = \frac{1}{n} \sum_{t=1}^n \frac{|\hat{y}_t - y_t|}{y_t} \times 100 \quad (1)$$

where  $y_t$  is a real consumption,  $\hat{y}_t$  is a predicted load and  $n$  is a length of the time series.

Subsequently, the method evaluates the performance of current model by increasing or decreasing its actual weight (lines 7–11). The process of increasement of model weight is defined by Eq. (2).

$$w_j^{new} = w_j * (1 + \beta) \quad (2)$$

The process of decreasement of model weight is defined by Eq. (3).

$$w_j^{new} = w_j * \beta \quad (3)$$

Then the method normalizes the models weights to an interval  $<0, 1>$  by subprocedure ‘Normalize\_Experts\_Weights’. Subsequently the global prediction of the whole ensemble is computed by Eq. (4).

$$\hat{f}_{global} = \frac{\sum_{i=1}^n (w_i * \hat{f}_i)}{\sum_{i=1}^n (w_i)} \quad (4)$$

where  $n$  is the number of prediction models in ensemble,  $w_i$  is the assigned weight of  $i$ th model and  $\hat{f}_i$  is the prediction of  $i$ th model.

If the predefined number of iterations was reached or the prediction error of the ensemble is higher than the specified threshold (parameter  $\theta$ ), the method proceeds into the removal phase, where the “weak” models are removed and replaced by new created models (line 18). Subprocedure ‘Create\_New\_Expert’ is used to choose new prediction model from available set of base learners that is not presented in ensemble yet. Subsequently, all models are trained on the following data chunk by subprocedure ‘Train\_Expert’ (line 22).

As the base learners of our DWM method, we chose six prediction models that are frequently used for prediction of electricity consumption:

1. Resilient Backpropagation Neural Network (NN) is an artificial neural network with learning heuristic for supervised learning which performs a local adaptation of the weight-updates according to the behavior of the error function [1]. Attributes to neural network are load and lag of 3 days and training window of size 3 days. NN used tanh activation function and sum of squares error as an error function. NN had 3 hidden layers. The number of neurons in each hidden layer and learning rate was determined by Particle Swarm Optimization algorithm (PSO).

2. Recursive Partitioning and Regression Trees (RT) is a method using 2 stages [2]. The first stage the model is created by splitting the data into subgroups using best split variables. The second stage consists of cross-validation used to trim the created tree. The attributes for RT are load and time vector. See Eqs. (5), (6) and (7).

$$\{Day_i\}_{i=1}^d \tag{5}$$

$$Day_i = 1, 2, \dots, period \tag{6}$$

$$|Day_i| = period \tag{7}$$

where *period* is the number of daily electric load measurements per end-user and *d* is the number of days in training window. The maximum depth of final tree was set to value 20.

3. Support Vector Regression (SVR) tries to find regression function that can best approximate the actual output vector with an error tolerance [25]. SVR maps the input data in to a higher dimensional feature space, in which the training data may exhibit linearity. For this purpose various kernels are used. Attributes for SVR were modeled as binary (dummy) variables representing the sequence numbers in regression model. Variable equals 1 in the case when they point to the *i*th value of the period, where  $i = (1, 2, \dots, period)$ .
4. Seasonal decomposition of time series by loess (STL): is a method [6] that decomposes a seasonal time series into three components: trend, seasonal and irregular. The resulting three components are forecasted separately by ARIMA (STL + ARIMA). ARIMA is one of the most frequently used forecasting methods [8]. It consists of three parts: autoregressive (AR), moving average (MA) and the differencing processes (I).
5. Exponential smoothing (STL + EXP) forecasts the future values of a time series as a weighted average of past values [15]. The weights decay exponentially with time as the observations get older.
6. Holt-Winters prediction method (HW) is used when the data shows not only the trend, but the seasonality as well [23]. The additional formula and smoothing parameter are introduced to handle the seasonality. The additive version of HW is used.

All three methods STL + ARIMA, STL + ETS and HW receive only the load time series.

The main differences between the original and modified version of DWM are:

- Integration function – the original version uses the result of the best model as global prediction. In modified version, the global prediction is calculated as a weighted average of predictions of all models in ensemble that allows solving regression problems.
- New threshold parameter  $\gamma$  for an evaluation of expert prediction based on prediction error.

- The weights of models are updated at each iteration, unlike the original version, where the weights were updated periodically after  $p$  iterations.
- Modified version of DWM starts with  $m$  prediction models. The original version starts with one model.

### 3.2 Dynamic Weighted Majority with Decomposition

In addition to the modified DWM described in previous section, we have proposed further modification of this method. The main difference lies in decomposition of the time series. Each part of incoming data stream is represented as a time series that can be further categorized to three types of time series patterns: trend, seasonal and cyclic [3]. Our method is based on the prediction of each of these patterns using different prediction model. Obtained results can be then combined to one final prediction. Therefore, from six models, which were used as prediction models in modified DWM, variations of size three without repetitions were created. As the result we have created 120 different variations where each one represents one predictive model. We suppose that the prediction of each part with different model could lead to higher probability to create a better model, whose prediction error will be lower than error of previous six models. Results of this proposed method (named as DWM+) can be found in evaluation chapter.

In this case, we have used the method of additive expert ensembles with its core functionality based on classical DWM. This method was proposed in 2005 by the same authors as the original DWM method [14]. The main difference lies in modified training process of models that are currently included in ensemble. However, in original DWM method, each predictive model learns i.e. updates its learned information every time when the new data chunk of data stream arrives. Here comes the main question: Why are we updating the learned information, if the last prediction error was low?

In this modified version, we re-train only those models, whose prediction error from the last iteration is higher than defined threshold i.e. an acceptable error rate was achieved. We have used other models, which had prediction error under the threshold, in iteration with previously learned information.

As the name of method AddExp already suggests, another feature of this method is based on adding new models to the ensemble. In our proposed method, we add a new model to ensemble every time when the prediction of the whole ensemble i.e. global prediction exceeds the defined threshold. New added model will be trained for the first time on new data chunk and this prediction can help the whole ensemble to decrease the global prediction error.

### 3.3 Technical Issues

After the implementation and initial testing of this method, we encountered the problem of the rapid increase in the number of models over the longer term prediction (e.g. prediction for the one-year period). This problem significantly affects the time and

space complexity of this method. To solve this problem, we have suggested a pruning method which is based on following two parameters: maximal number of models in ensemble and pruning threshold. The pruning method is applied every time when the size of ensemble reaches the defined maximal value. After that, all models, whose prediction error from the last iteration was higher than pruning threshold, were removed from the ensemble. Removed models were then replaced by new models that were chosen by proposed heuristic. This heuristic relies on choosing the best predictive models from the last iteration of method i.e. iteration where the last models have been removed. The main idea of this heuristic is to add models with better results in comparison with the results of removed models. So, if the number of better models is lower than number of removed models, it is not necessary to preserve the original number of models. Pruning will be applied again when the size of ensemble reaches the maximum number of models.

During implementation and initial testing of proposed method and predictive models, we used recommended configuration parameters, described in the studied papers [10, 11, 15, 19]. The electricity load forecasting represents a special type of prediction problem therefore we decided to optimize parameters of DWM method and its modified version AddExp, to fit our electricity loads dataset exactly.

To perform the optimization task, one of the Biologically Inspired Algorithms (BIA), called Particle Swarm Optimization (PSO), was used. The main reason of choosing the BIAs is their excellent ability to optimize various problems. This approach can lead to solving the optimization problem in a different way in comparison to classical optimization methods [5].

### 3.4 Optimization of Parameters by Particle Swarm Optimization

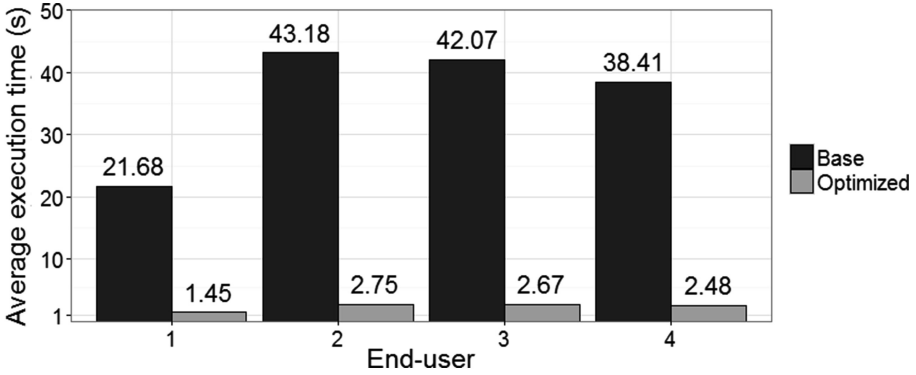
PSO algorithm represents a popular biological inspired method, which is frequently used for optimization problems. More information about this algorithm and used implementation library could be found in [9].

Before the optimization of DWM parameters took places, we decided to optimize parameters of neural network predictive model [17]. The main reason for the optimization of this model was its high time complexity that was needed in order to obtain the prediction. Optimized parameters include: number of hidden neurons and hidden layers, interval for learning rate of network and threshold for stopping criteria.

Figure 1 shows a comparison of an optimized version of model to a neural net model with default parameters. The comparison is based on the average execution time.

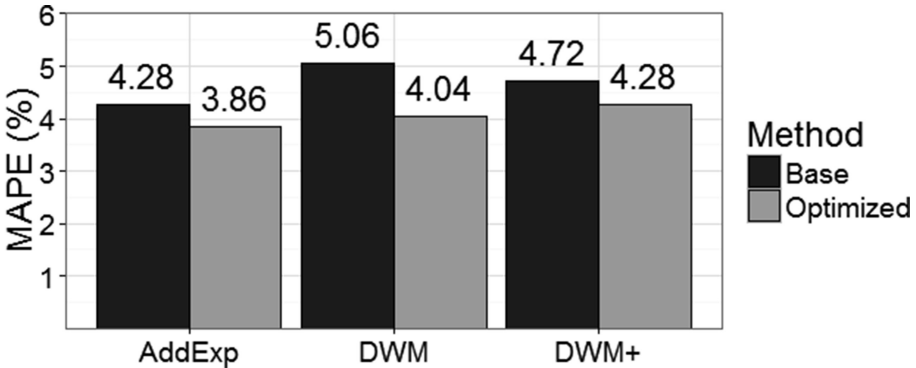
The time interval, which was chosen for the optimization task of this model, represents one month period of predictions. During this period, PSO algorithm was optimizing the obtained results in terms of execution time and prediction error i.e. MAPE. The x axis in Fig. 1 represents selected end-users from our dataset. The y axis shows an average time of execution for described time period. As we can see, the optimization by PSO algorithm helped the neural network to decrease the execution time for some end-users more than fifteen times. It is important to mention that this time reduction also helped to improve the whole ensemble of predictive models. Then we continued with optimization of DWM method and its modified version.





**Fig. 1.** Average execution time of prediction for 4 end-users using neural network optimization

The Fig. 2 shows the obtained results from optimization of proposed DWM methods for the same period as previously optimized neural network model. We have optimized following parameters: decrease/increase factor for model weights, prediction acceptance threshold and parameter representing initial ensemble size. In Fig. 2 the x axis represents the tested methods and the y axis shows the mean absolute percentage error of prediction. Except the time reduction, the optimization brought also a moderate decrease of prediction error. The results indicate that the prediction error of the optimized version was reduced in some cases by more than one percent in comparison to original method.



**Fig. 2.** MAPE of AddExp and DWM methods optimized by PSO

## 4 Evaluation

In this chapter we focus on comparison of errors between the predictions of three modifications of original DWM method and six base predictive models. Before we describe the designed experiments and results, we focus on electricity loads dataset used in our study.

### 4.1 Data Description

Dataset used in this paper is represented by electricity load records of Slovak companies from 1.7.2013 to 16.02.2015. These records were obtained by smart meters that send information about the actual electricity consumption every 15 min. Our dataset consists of ca. 490mil records from 21 502 different end-users.

Records in a modified version of the dataset contain the following attributes: date, time and electricity load.

The whole dataset was transformed to a stream of chunks where each part represented one shift of sliding window for 96 records i.e. one day load records.

Figures 3 and 4 represent the comparisons of all predictive models that were used in this study. For the purpose of testing we have chosen two different time periods from the data set. Figure 3 shows results from one month prediction and Fig. 4 represents the results obtained from prediction of one year period. For better evaluation of the results, we also provide a numeric comparison, which can be seen in Table 1.

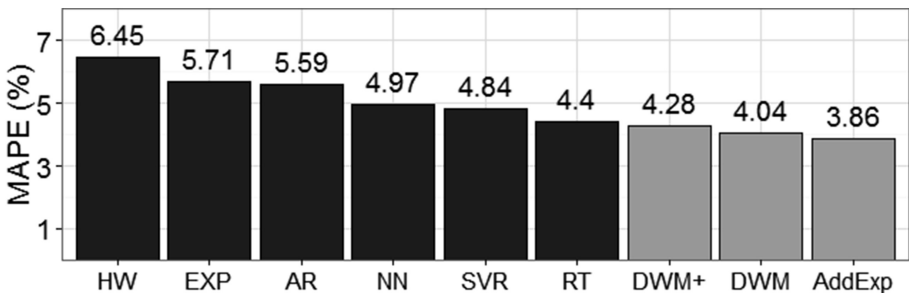


Fig. 3. Comparison of all tested predictive models (month prediction)

As we see, prediction errors obtained from one year period are higher than prediction errors obtained from shorter period i.e. one month. This fact was caused mainly due to the presence of concept drifts that occurred in this one year period. However, in both tested time intervals, all three modified versions of original DWM reached the lowest prediction errors in comparison to other predictive models. This result represents a main achievement of this evaluation, which was based on the assumption that ensemble methods could obtain more precise prediction than single models. The modified method AddExp, which represents an extended version with a decomposition of time series, reached the lowest prediction error.

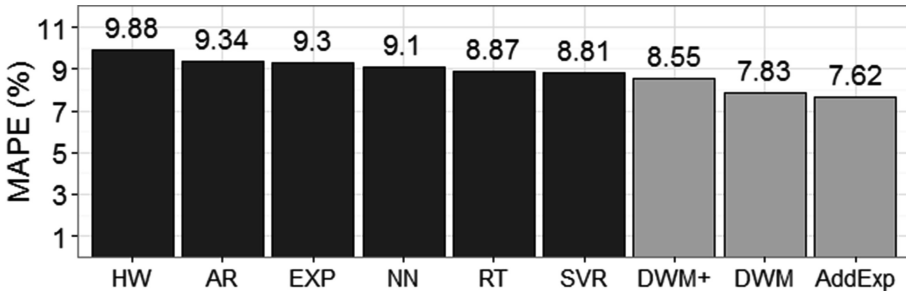


Fig. 4. Comparison of all tested predictive models (year prediction)

Table 1. Obtained results of Mean Absolute Percentage Error (MAPE)

Name of predictive model	MAPE (%) month	MAPE (%) year
Additive Expert Ensembles (AddExp)	3,86	7,62
ARIMA (AR)	5,59	9,34
Dynamic Weighted Majority (DWM)	4,04	7,83
Dynamic Weighted Majority with decomp. (DWM +)	4,28	8,55
Exponential Smoothing (EXP)	5,71	9,30
Holt-Winters (HW)	6,45	9,88
Neural Network (NN)	4,97	9,10
Regression Tree (RT)	4,40	8,87
Support Vector Regression (SVR)	4,84	8,81

Based on further investigation of obtained predictions, we also provide a study on prediction error progress on previous one month interval.

In Fig. 5 we can see a more detailed view of prediction error progress of six models. Graphs on the left side represent the implemented methods based on DWM. Graph on the right side represent the error of three models, whose error development compared to previous methods looks more irregular. This fact can also be seen in Table 2 that represents a histogram of models used during the one year prediction. The methods with lowest prediction errors are preferred mostly and therefore they are frequently used.

In Table 2, we can see three models that were characterized by their irregular error development in Fig. 5. They were also less used in original DWM, where all parts of time series are predicted by same model. On the other hand, we can notice that these same three models were the most common in AddExp method. This fact proves the following statement: Predictive models ARIMA, Holt-Winter and model of exponential smoothing achieve better prediction results, if they predict only a part of time series than a whole data stream chunk. This statement could lead to reverse fact that the predictive models like Regression tree and Support Vector Regression model can be more frequently used in AddExp model in case they are a part of ensemble, where they

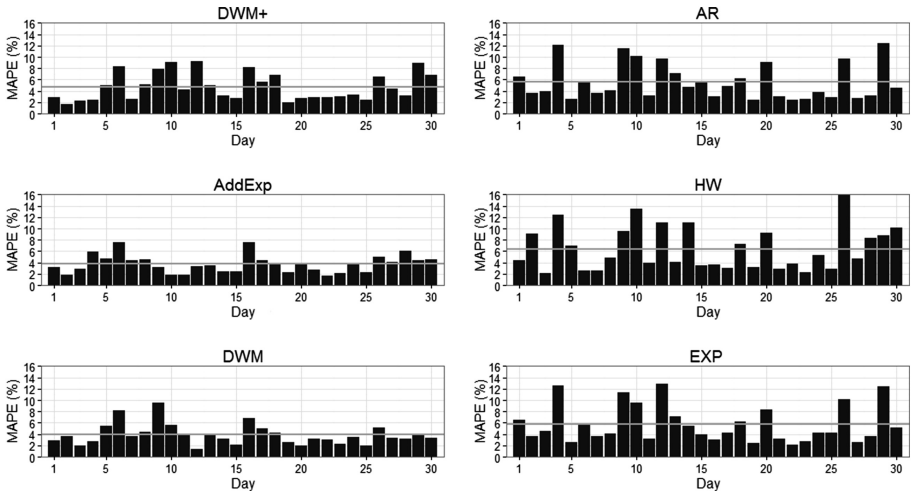


Fig. 5. Prediction error progress (month prediction)

Table 2. Histogram of models usage

Model name	AddExp decomposition			AddExp	DWM
	Trend	Season	Cyclic	Sum	Sum
AR	655	481	386	<b>1522</b>	<b>72</b>
EXP	568	460	458	<b>1486</b>	<b>93</b>
HW	450	636	616	<b>1702</b>	<b>60</b>
NN	55	82	25	<b>162</b>	<b>102</b>
RT	470	447	508	<b>1425</b>	<b>104</b>
SVR	359	451	564	<b>1374</b>	<b>106</b>

could predict all three parts of time series. Consequently, this fact can lead to a reduction of prediction error of AddExp method.

## 5 Conclusion

In this paper, we proposed three modifications of original Dynamic Weighted Majority method that were applied as a solution to prediction problem of electricity load records. We also focused on decomposition of time series, optimization of parameters by PSO algorithm, or modifications based on extension of original DWM method with the aim of further prediction accuracy improvement.

The results of proposed solutions were compared to six predictive models. Tested models were compared by their prediction error that was represented by Mean Absolute Percentage Error metric. The prediction was performed for two different time intervals

from dataset. In all tested cases, which were applied, the proposed methods achieved a lower prediction error than other prediction models. The best results i.e. the lowest prediction error or most regular error development, were reached by method AddExp. We believe that further improvements of this method could be reached by applying described modification of models variations in time series decomposition. Additional modifications could be a subject of future work of our study. The accuracy of proposed ensemble could be improved by developing a mechanism to keep the threshold values and models input settings continuously up-to-date.

Our modified ensemble learning methods i.e. DWM, DWM + and AddExp method proved their suitability for solving the electricity power load demand predictions. The proposed method is applicable generally for the time series prediction problems in various domains.

**Acknowledgements.** This work was partially supported by the Research and Development Operational Programme for the project “International Centre of Excellence for Research of Intelligent and Secure Information-Communication Technologies and Systems”, ITMS 26240120039, co-funded by the ERDF and the Scientific Grant Agency of The Slovak Republic, grant No. VG 1/0752/14 and VG 1/0646/15. The authors would also like to thank for financial contribution from the STU Grant scheme for Support of Young Researchers.

## References

1. Anastasiadis, A.D., Magoulas, G.D., Vrahatis, M.N.: New globally convergent training scheme based on the resilient propagation algorithm. *Neurocomputing* **64**, 253–270 (2005)
2. Archer, K.J.: rpartOrdinal: an R package for deriving a classification tree for predicting an ordinal response. *J. Stat. Softw.* **34**(7), 1–17 (2010)
3. Athanasopoulos, G., Hyndman, R.J.: *Forecasting: principles and practice*. OTexts, Melbourne (2013)
4. Aung, Z., Toukhy, M., Williams, J.: Towards Accurate Electricity Load Forecasting in Smart Grids, vol. 8, pp. 51–57 (2012)
5. Binitha, S., Siva Sathya, S.: A survey of bio inspired optimization algorithms. *Int. J. Soft Comput. Eng.* **2**(2), 137–151 (2012)
6. Cleveland, R.B., Cleveland, W.S., McRae, J.E., Terpenning, I.: STL: a seasonal-trend decomposition procedure based on loess. *J. Official Stat.* **6**(1), 3–73 (1990)
7. Ikonomovska, E., Gama, J., Džeroski, S.: Learning model trees from evolving data streams. *Data Min. Knowl. Disc.* **23**(1), 128–168 (2011)
8. Hong, W.-C.: Modeling for Energy Demand Forecasting. In: Hong, W.-C. (ed.) *Intelligent Energy Demand Forecasting*. Lecture Notes in Energy, vol. 10, pp. 21–40. Springer, Heidelberg (2013)
9. Kennedy, J., Eberhart, R.: Particle swarm optimization. In: *Proceedings of IEEE International Conference on Neural Networks*, Washington, DC (1995)
10. Maqsood, I., Khan, M.R.: Dynamic weighted majority: an ensemble method for drifting concepts. *JMLR* **8**, 2755–2790 (2007)
11. Maqsood, I., Khan, M.R.: Using additive expert ensembles to cope with concept drift. In: *Proceedings of the Twenty Second ACM International Conference on Machine Learning (ICML 2005)*, Bonn, Germany (2005)

12. Maqsood, I., Khan, M.R.: Dynamic weighted majority: a new ensemble method for tracking concept drift. In: Proceedings of the Third International IEEE Conference on Data Mining (ICDM 2003), Los Alamitos (2003)
13. Mendes-Moreira, J., Soares, C., Jorge, A.M., Freire de Sousa, J.: Ensemble approaches for regression: a survey. *ACM Comput. Surv.* **45**(1), 1–40 (2012)
14. Narasimhamurthy, A., Kuncheva, L.I.: A framework for generating data to simulate changing environments. In: Proceedings of the 25th IASTED International Multi-Conference on Artificial Intelligence and Applications (AIA 2007), Innsbruck, Austria (2007)
15. Neupane, B.: Ensemble Learning-based Electricity Price, Masdar Institute of Science and Technology (2013)
16. Pappas, S.Sp., Ekonomou, L., Karampelas, P., Karamousantas, D.C., Katsikas, S.K., Chatzarakis, G.E., Skafidas, P.D.: Electricity demand load forecasting of the Hellenic power system using an ARMA model. In: *Electric Power Systems Research*, vol. 80, pp. 256–264 (2010)
17. Riedmiller, M., Braun, H.: Rprop - a fast adaptive learning algorithm. In: Proceedings of the International Symposium on Computer and Information Science VII, Antalya, Turkey (1992)
18. Shen, W., Babushkin, V., Aung, Z., Woon, W.L.: An ensemble model for day-ahead electricity demand time series forecasting. In: Proceedings of the 4th International Conference on Future Energy Systems - e-Energy 2013, pp. 51–62. ACM, New York (2013)
19. Sidhu, P., Bhatia, M.P.S.: Empirical support for concept drifting approaches: results based on new performance metrics. *IJISA* **7**(6), 1–20 (2015)
20. Singh, A., Khatoon, S.: An overview of electricity demand forecasting techniques. *Netw. Complex Syst.* **3**(3), 38–48 (2013)
21. Taylor, J.W., McSharry, P.E.: Short-term load forecasting methods: an evaluation based on european data. *IEEE Trans. Power Syst.* **22**(4), 2213–2219 (2007)
22. Therneau, T., et al.: rpart: recursive partitioning and regression trees. R package version 4.1–10, (2014). <http://cran.r-project.org/package=rpart>
23. Winters, P.R.: Forecasting sales by exponentially weighted moving averages. *Manage. Sci.* **6**(3), 324–342 (1960)
24. Xiao, L., Wang, J., Hou, R., Wu, J.: A combined model based on data pre-analysis and weight coefficients optimization for electrical load forecasting. *Energy* **82**, 1–26 (2015)
25. Zhang, F., et al.: A conjunction method of wavelet transform-particle swarm optimization-support vector machine for streamflow forecasting. *J. Appl. Math.* **2014**, 1–10 (2014)