GMDH: An R Package for Short Term Forecasting via GMDH - Type Neural Network Algorithms

by Osman Dag and Ceylan Yozgatligil

Abstract Group Method of Data Handling (GMDH) - type neural network algorithms are the heuristic self organization method for modelling the complex systems. GMDH algorithms are utilized for a variety of purposes, which are identification of physical laws, extrapolation of physical fields, pattern recognition, clustering, approximation of multidimensional processes, forecasting without models and so on. In this study, an R package, **GMDH**, is presented to make short term forecasting through GMDH - type neural network algorithms. It has options to use different transfer functions, sigmoid, radial basis, polynomial, and tangent functions, simultaneously or separately. Data on cancer death rate of Pennsylvania from 1930 to 2000 are used to illustrate the implementation of **GMDH** package. We include ARIMA models and exponential smoothing methods for the comparison purpose.

Introduction

Time series data are ordered successive observations which are measured in equally or unequally spaced time. Time series data include dependency among successive observations. Hence, the order of the data is important. They are commonly appearing in various areas, such as medical studies, economics, energy industry, agriculture, meteorology, and so on. Modelling time series data is the method which utilizes history of the data and makes forecasting by the help of this history. Sometimes, statistical models are not sufficient to solve the problems, such as pattern recognition, forecasting, identification, and so on. Extracting the information from the measurements has advantages while modelling complex systems since there is no enough prior information and/or no theory is defined to model the complex systems. Selecting model automatically is a powerful way for the researchers who are interested in the result and do not have sufficient statistical knowledge and sufficient time (Mueller et al., 1998).

The objective of this study is to develop an R package to make forecasting. Some of recent softwares developed for time series are Dunsmuir and Scott (2015); Shang (2013); Holmes et al. (2012); Fraley et al. (2011); Hyndman and Khandakar (2008). In this study, we focused on the development of R package for short term forecasting via Group Method of Data Handling (GMDH) algorithms. The background of GMDH - type neural network is based on the end of the 1960s and the beginning of 1970s. First, Ivakhnenko (1966) introduced a polynomial, which is the basic algorithm of GMDH, to construct higher order polynomial. Also, Ivakhnenko (1970) introduced heuristic self-organization method which constructed the main working system of GMDH algorithm. Heuristic self-organization method defines the way that the algorithm follows by the rules such as external criteria. GMDH method, convenient for the complex and unstructured system, has superiority on the high order regression (Farlow, 1981).

Kondo (1998) proposed GMDH - type neural network in which the algorithm works according to the heuristic self-organization method. Kondo and Ueno (2006a,b) proposed GMDH algorithm which has a feedback loop. According to the algorithm, the output obtained from the last layer is set as a new input variable, if threshold is not satisfied in the last layer. The system of algorithm is organized by heuristic self-organization method where sigmoid transfer function is integrated. Kondo and Ueno (2007) proposed logistic GMDH-type neural network. The difference from conventional GMDH algorithm was that they take linear function of all inputs at last layer. Kondo and Ueno (2012) included three transfer functions; sigmoid, radial basis and polynomial functions in feedback GMDH algorithm. Srinivasan (2008) used GMDH-type neural network and traditional time series models to forecast energy demand prediction. It was shown that GMDH-type neural network was superior in forecasting energy demand prediction compared to traditional time series models with respect to mean absolute percentage error (MAPE). In another study, Xu et al. (2012) applied GMDH algorithm and ARIMA models to forecast the daily power load. According to the results, GMDH results were superior to the results of ARIMA models in terms of MAPE for forecasting performance.

There are some difficulties of applying GMDH - type neural network, since there is no free available software for the researchers to reach GMDH algorithm in the literature. We present an R package, **GMDH**, to make short term forecasting through GMDH - type neural network algorithms. The package includes two types of GMDH structures; namely, GMDH structure and revised GMDH (RGMDH) structure. Also, it includes a variety of options to use different transfer functions, sigmoid,

radial basis, polynomial, and tangent functions, simultaneously or separately. Data on cancer death rate of Pennsylvania from 1930 to 2000 are used to illustrate the implementation of GMDH package. We include ARIMA models and exponential smoothing (ES) methods for the comparison purpose.

Methodology

In this section, data preparation, two types of GMDH - type neural network structures, and estimation of regularization parameter in regularized least square estimation (RLSE) are given.

Data preparation

Data preparation has important role in GMDH - type neural network algorithms. To get rid of very big numbers in calculation and to be able to use all transfer functions in the algorithm, it is necessary for whole data to be in the interval of (0, 1). This necessity is guaranteed by the following transformation,

$$w_t = \frac{\alpha_t + \delta_1}{\delta_2} \tag{1}$$

with

$$\delta_1 = \begin{cases} |\alpha_t| + 1 & \text{if } \min(\alpha_t) \leq 0 \\ 0 & \text{if } \min(\alpha_t) > 0 \end{cases}$$

and

$$\delta_2 = \max(\alpha_t + \delta_1) + 1$$

where α_t is the actual time series dataset at hand. During the estimation and forecasting process in GMDH neural network algorithm, all calculations are done using the scaled data set, w_t. After all processes are ended; in other words, all predictions and forecasts are obtained, we apply the inverse transformation as follows,

$$\widehat{\alpha}_t = \widehat{w}_t \times \delta_2 - \delta_1 \tag{2}$$

Let's assume a time series dataset for t time points, and p inputs. An illustration of time series data structure in GMDH algorithms is presented in Table 1. Since we construct the model for the data with time lags, the number of observations, presented under subject column in the table, is equal to $t - p$ and the number of inputs, lagged time series, is p. In this table, the variable called z is put in the models as a response variable, and the rest of the variables are taken into models as lagged time series x_i, where $i = 1, 2, ..., p$. The notations in Table 1 are followed throughout this paper.

Table 1: An illustration of time series data structure in GMDH algorithms

Subject	z	x_1	x_2	...	x_p
1	w_t	w_{t-1}	w_{t-2}	...	w_{t-p}
2	w_{t-1}	w_{t-2}	w_{t-3}	...	w_{t-p-1}
3	w_{t-2}	w_{t-3}	w_{t-4}	...	w_{t-p-2}
⋮	⋮	⋮	⋮	⋱	⋮
$t-p$	w_{p+1}	w_p	w_{p-1}	...	w_1

A better model which explains the relation between response and lagged time series is captured via transfer functions. Mainly, sigmoid, radial basis, polynomial, and tangent functions, presented in Table 2, are used to explain the relation between inputs and output in GMDH-type neural network (Kondo and Ueno, 2012). We use all transfer functions, stated in Table 2, simultaneously in each neuron. In other words, we construct four models at each neuron, and then the model which gives smallest prediction mean square error (PMSE) is selected as a current transfer function at corresponding neuron.

Table 2: Transfer functions

Sigmoid Function	$z = 1/(1 + \exp^{-y})$
Radial Basis Function	$z = \exp^{-y^2}$
Polynomial Function	$z = y$
Tangent Function	$z = \tan y$

GMDH algorithm

GMDH - type neural network algorithms are the modeling techniques which learn the relations among the variables. In the perspective of time series, the algorithm learns the relationship among the lags. After learning the relations, it automatically selects the way to follow in algorithm. First, GMDH was introduced by Ivakhnenko (1966) to construct high order polynomial. The following equation is known as the Ivakhnenko polynomial given by

$$y = a + \sum_{i=1}^{m} b_i \times x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} \times x_i \times x_j + \sum_{i=1}^{m}\sum_{j=1}^{m}\sum_{k=1}^{m} d_{ijk} \times x_i \times x_j \times x_k + \ldots \tag{3}$$

where m is the number of variables and $a, b, c, d, \ldots$ are coeffients of variables in the polynomial, also named as weights. Here, y is a response variable, x_i and x_j are the lagged time series to be regressed. In general, the terms are used in calculation up to square terms as presented below,

$$y = a + \sum_{i=1}^{m} b_i \times x_i + \sum_{i=1}^{m}\sum_{j=1}^{m} c_{ij} \times x_i \times x_j \tag{4}$$

GMDH algorithm considers all pairwise combinations of p lagged time series. Therefore, each combination enters each neuron. Using these two inputs, a model is constructed to estimate the desired output. In other words, two input variables go in a neuron, one result goes out as an output. The structure of the model is specified by Ivakhnenko polynomial in Eq. 4 where $m = 2$. This specification requires that six coefficients in each model are to be estimated.

GMDH algorithm is a system of layers in which there exist neurons. The number of neurons in a layer is defined by the number of input variables. To illustrate, assume that the number of input variables is equal to p, since we include all pairwise combinations of input variables, the number of neurons is equal to $h = \binom{p}{2}$. The architecture of GMDH algorithm is illustrated in Figure 1 when there exist three layers and four inputs.

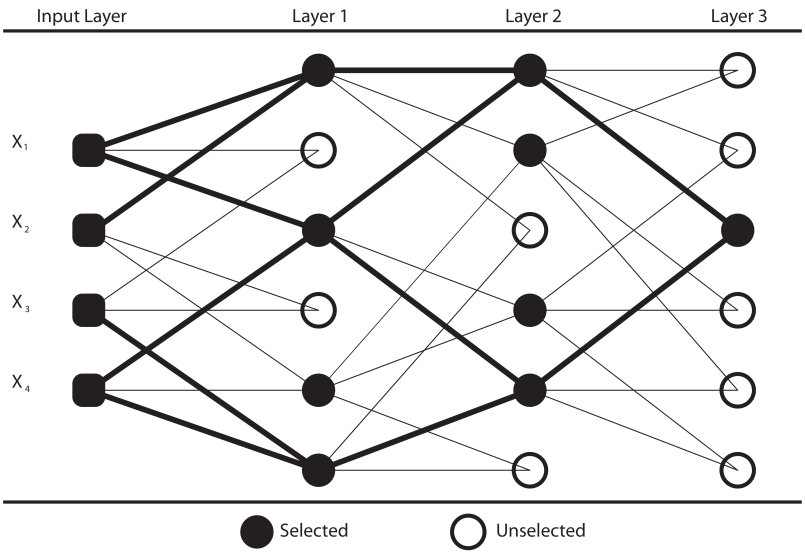

Figure 1: Architecture of GMDH algorithm

In GMDH architecture shown in Figure 1, since the number of inputs is equal to four, the number of nodes in a layer is determined to be six. This is just a starting layer to the algorithm. The coefficients of Eq. 4 are estimated in each neuron. By using the estimated coefficients and input variables in each neuron, the desired output is predicted. According to chosen external criteria, p neurons are

selected and $h - p$ neurons are eliminated from the network. In this study, prediction mean square error (PMSE) is used as external criteria. In Figure 1, four neurons are selected while two neurons are eliminated from the network. The outputs obtained from selected neurons become the inputs for the next layer. This process continues until the last layer. At the last layer, only one neuron is selected. The obtained output from the last layer is the predicted values for the time series at hand. The flowchart of the algorithm is able to be depicted in Figure 2.

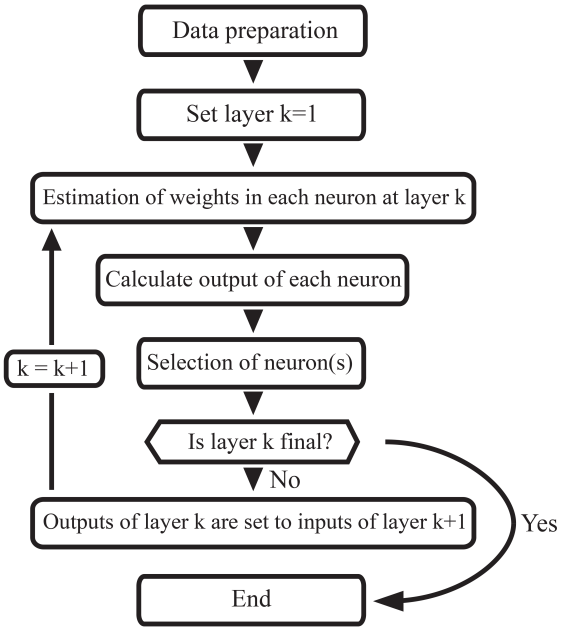

Figure 2: Flowchart of GMDH algorithms

In GMDH algorithm, there exist six coefficients to be estimated in each model. Coefficients are estimated via RLSE.

RGMDH algorithm

GMDH - type neural network constructs the algorithm by investigating the relation between two inputs and the desired output. Architecture of revised GMDH (RGMDH) - type neural network does not only consider this relation, but it also considers individual effect on desired output (Kondo and Ueno, 2006b). There are two different types of neurons in RGMDH - type neural network. In the first type of neuron, it is same as in GMDH - type neural network, given as in Eq. 4. That is, two inputs enter the neuron, one output goes out. In the second type of neuron, r inputs enter the neuron, one output goes out. Second type neuron is given by

$$y = a + \sum_{i=1}^{r} b_i \times x_i \quad , \quad r \leq p \tag{5}$$

where r is the number of inputs in the corresponding second type neuron.

As mentioned above, there exist $h = \binom{p}{2}$ neurons in one layer in GMDH-type neural network. In addition to this, with the p neurons from the second type of neuron, the number of neurons in one layer becomes $\eta = \binom{p}{2} + p$ in RGMDH - type algorithm. The architecture of RGMDH algorithm is shown in Figure 3 when there exist three layers and three inputs. In this architecture, since the number of inputs is three, the number of nodes in a layer is determined to be six. Here, three of six nodes are first type of neurons in which all pairwise combinations of lagged time series are already used as in the GMDH algorithm. The rest of the three nodes are second type of neurons where the individual effect of the lagged time series are sequentially added to the layer starting from lag 1. In each neuron, coefficients of models are calculated by using the corresponding models in Eqs. 4 and 5. For instance, in Figure 3, there are six coefficients to be estimated as given Eq. 4 for the first type of neurons, and two, three and four coefficients are estimated as given in Eq. 5 for the second type of neurons when r equals to 1, 2 and 3, respectively. The desired output is predicted by utilizing estimated coefficients and input variables in each neuron. p neurons are selected as living cells and $\eta - p$ death cells are eliminated from the network according the external criteria. The rest of the algorithm is same with GMDH.

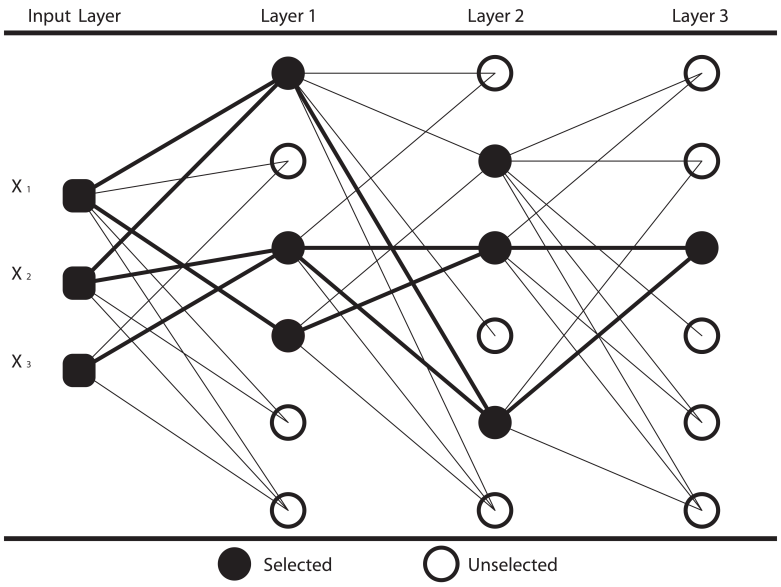

Figure 3: Architecture of RGMDH algorithm

Estimation of regularization parameter in RLSE

In each estimation step, there exist the coefficients to be estimated. While we are estimating these coefficients, we use regularized least square estimation method. It is stated that regularized least square estimation is utilized when there is the possibility of multi-collinearity problem by integrating regularization parameter, λ, into the estimation step. It is important to note that regularized least square estimation differs from the least square estimation when regularization parameter is not zero.

We integrate the estimation of regularization parameter (penalizing term) via cross-validation in GMDH algorithms. For this purpose, we divide the data into two parts as a learning set and a testing set. In GMDH package, 70% of the time series is taken for learning set as default. Since the data set is time dependent, order of data is saved in division process. In other words, first 70% of the data is used for learning set and the last 30% of the data is utilized for testing set. This whole process is applied for each model constructed in each neuron. The algorithm of regularization parameter estimation is as follows,

i) Clarify the possible regularization parameter, $\lambda = 0, 0.01, 0.02, 0.04, 0.08, \ldots, 10.24$. Note that, when $\lambda = 0$, RLSE is converted to LSE.

ii) For each possible λ value, coefficients are estimated via RLSE by using learning set.

iii) After calculation of coefficients, calculate the predicted values by utilizing test set and obtain MSE for each regularization parameter.

iv) Select the regularization parameter which gives minimum MSE value.

Implementation of GMDH package

Data used in this application are yearly cancer death rate (per 100,000 population) of Pennsylvania between 1930 and 2000. The data were documented in Pennsylvania Vital Statistics Annual Report by the Pennsylvania Department of Health in 2000 (Wei, 2006). This dataset is also available as a demo dataset in our R package **GMDH**. After installing package **GMDH**, it can be loaded in R workspace by

```
# load GMDH package
R> library("GMDH")

# load cancer data
R> data("cancer")
R> cancer
```

After the cancer death rate data set is loaded, one may use `fcast` function in **GMDH** package to make short term forecasting. To utilize GMDH structure for forecasting, method is set to "GMDH". One should set method to "RGMDH" to use RGMDH structure.

```
R> out = fcast(cancer[1:66], method = "GMDH", input = 15, layer = 1, f.number = 5,
level = 95, tf = "all", weight = 0.70, lambda = c(0, 0.01, 0.02, 0.04, 0.08, 0.16,
0.32, 0.64, 1.28, 2.56, 5.12, 10.24))

   Point Forecast    Lo 95    Hi 95
67       249.5317 244.9798 254.0836
68       249.6316 244.4891 254.7741
69       248.9278 243.0318 254.8239
70       247.0385 240.7038 253.3731
71       244.7211 237.1255 252.3168

# display fitted values
R> out$fitted

# return residuals
R> out$residuals

# show forecasts
R> out$mean
```

In this part, we divided the data into two parts for the aim of observing the ability of methods on prediction ($n = 66$) and forecasting ($n = 5$). We include ARIMA models and ES methods for the comparison purpose. For the determination of the best order of ARIMA models and the best method of ES techniques, there are two functions in R package **forecast** (Hyndman and Khandakar, 2008). These functions, `auto.arima` and `ets`, which use grid search, select the best model according to the criteria of either AIC, AICc or BIC. The functions suggested the model ARIMA (1, 1, 0) with intercept and ES method with multiplicative errors, additive damped trend and no seasonality (M, Ad, N), respectively. We also added the model ARIMA (0, 1, 0) with intercept for this data set suggested by Wei (2006). For all models, prediction mean square error (PMSE) and forecasting mean square error (FMSE) are stated in Table 3.

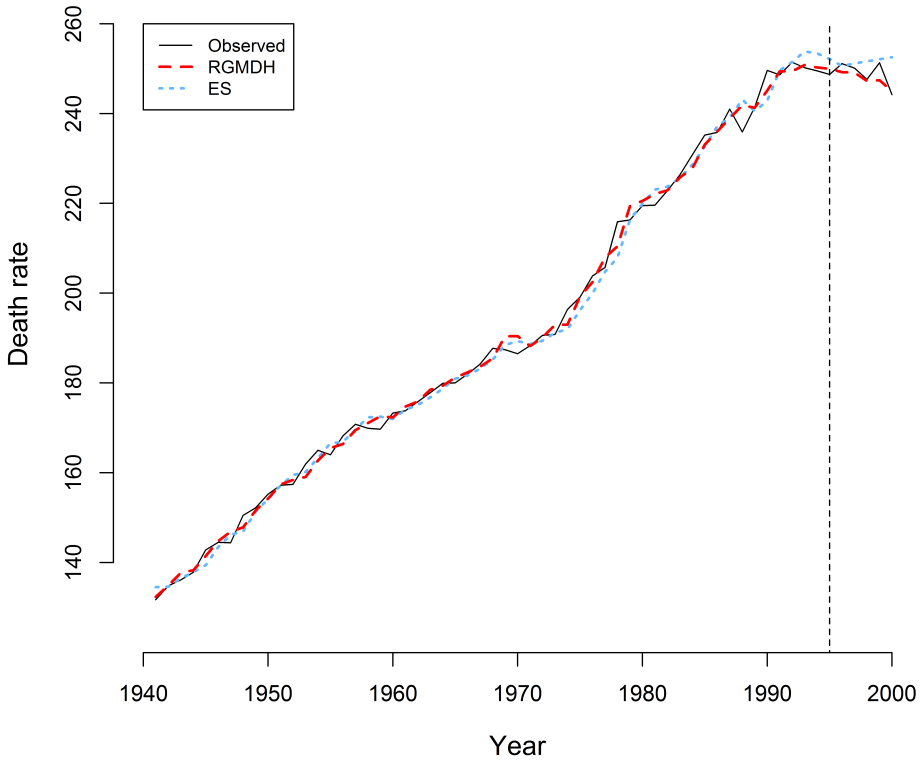

Figure 4: Yearly cancer death rate (per 100,000 population) in Pennsylvania between 1941 and 2000 with predictions and forecasts obtained via RGMDH and ES(M,Ad,N)

Table 3: Comparison of GMDH algorithms with other models on cancer death rate in terms of prediction mean square error (PMSE) and forecasting mean square error (FMSE)

	PMSE	FMSE
GMDH	4.985	4.575
RGMDH	4.287	4.102
ARIMA(1, 1, 0) with intercept	5.995	81.874
ARIMA(0, 1, 0) with intercept	6.324	73.756
ES (M, Ad, N)	6.153	17.508

The best forecasting performance belongs to RGMDH algorithm and its prediction accuracy also yields better results compared GMDH, ARIMA and ES models. Moreover, GMDH algorithm outperforms ARIMA and ES models in prediction and forecasting. To avoid visual pollution in Figure 4, we include only the predictions and forecasts of RGMDH algorithm and ES (M, Ad, N).

Conclusion

In this study, we used GMDH - type neural network algorithms, the heuristic self-organization method for modelling the complex systems, to make forecasts for time series data sets. We primarily concentrated to develop a free software. Concretely, we developed an R package called **GMDH** to make forecasting in short term via GMDH - type neural network algorithms. Also, we included different transfer functions; sigmoid, radial basis, polynomial, and tangent functions, into **GMDH** package. Our R package proposed that these functions are able to be exerted simultaneously or separately depending on the desire.

In estimation of coefficients, since we construct the model for the data with lags, there exists high possibility of occurring multi-collinearity problem. Therefore, we utilized regularized least square estimation to handle such a problem. It is important to note that estimation of regularization parameter is the question of interest. Cross validation was applied in order to estimate regularization term. After selection of regularization term, coefficients were estimated by the help of all observations and regularization parameter.

Application of the algorithms on a real life dataset suggests improved performance of GMDH - type neural network algorithms over ARIMA and ES models in prediction and short term forecasting. Researchers are able to reach GMDH algorithms easily since our proposed R package **GMDH** is available on Comprehensive R Archive Network (CRAN) at http://CRAN.R-project.org/package=GMDH.

Future studies are planned in the direction of transfer functions. In this study, we used four different transfer functions - sigmoid, radial basis, polynomial, and tangent functions - into GMDH algorithms. We plan to integrate Box-Cox transformation into GMDH algorithms. GMDH algorithms with four transfer functions and GMDH algorithms with Box-Cox transformation are going to be performed on real data applications to compare the prediction and short term forecasting. After well-documented, the related R function of GMDH algorithms with Box-Cox transformation are going to be released under our proposed R package **GMDH**.

Acknowledgment

We thank the anonymous reviewers for their constructive comments and suggestions which helped us to improve the quality of our paper.

Bibliography

W. T. Dunsmuir and D. J. Scott. The glarma package for observation driven time series regression of counts. *Journal of Statistical Software*, 67(7):1–36, 2015. [p1]

S. J. Farlow. The gmdh algorithm of ivakhnenko. *The American Statistician*, 35(4):210–215, 1981. [p1]

C. Fraley, A. E. Raftery, T. Gneiting, J. Sloughter, and V. J. Berrocal. Probabilistic weather forecasting in r. *The R Journal*, 3(1):55–63, 2011. [p1]

E. E. Holmes, E. J. Ward, and K. Wills. Marss: Multivariate autoregressive state-space models for analyzing time-series data. *The R Journal*, 4(1):11–19, 2012. [p1]

R. Hyndman and Y. Khandakar. Automatic time series forecasting: the forecast package for r. *Journal of Statistical Software*, 27(3):1–22, 2008. [p1, 6]

A. Ivakhnenko. The group method of data handling-a rival of the method of stochastic approximation. *Soviet Automatic Control*, 13(3):43–55, 1966. [p1, 3]

A. Ivakhnenko. Heuristic self-organization in problems of engineering cybernetics. *Automatica*, 6(2): 207–219, 1970. [p1]

T. Kondo. Gmdh neural network algorithm using the heuristic self-organization method and its application to the pattern identification problem. In *SICE'98. Proceedings of the 37th SICE Annual Conference. International Session Papers*, pages 1143–1148. IEEE, 1998. [p1]

T. Kondo and J. Ueno. Medical image recognition of the brain by revised gmdh-type neural network algorithm with a feedback loop. *International Journal of Innovative Computing, Information and Control*, 2(5):1039–1052, 2006a. [p1]

T. Kondo and J. Ueno. Revised gmdh-type neural network algorithm with a feedback loop identifying sigmoid function neural network. *International Journal of Innovative Computing, Information and Control*, 2(5):985–996, 2006b. [p1, 4]

T. Kondo and J. Ueno. Logistic gmdh-type neural network and its application to identification of x-ray film characteristic curve. *JACIII*, 11(3):312–318, 2007. [p1]

T. Kondo and J. Ueno. Feedback gmdh-type neural network and its application to medical image analysis of liver cancer. In *42th ISCIE international symposium on stochastic systems theory and its applications*, pages 81–82, 2012. [p1, 2]

J. A. Mueller, A. Ivachnenko, and F. Lemke. Gmdh algorithms for complex systems modelling. *Mathematical and Computer Modelling of Dynamical Systems*, 4(4):275–316, 1998. [p1]

H. L. Shang. ftsa: An r package for analyzing functional time series. *The R Journal*, 5(1), 2013. [p1]

D. Srinivasan. Energy demand prediction using gmdh networks. *Neurocomputing*, 72(1):625–629, 2008. [p1]

W. W. S. Wei. *Time series analysis: univariate and multivariate methods*. Addison-Wesley publ, 2006. [p5, 6]

H. Xu, Y. Dong, J. Wu, and W. Zhao. Application of gmdh to short-term load forecasting. In *Advances in Intelligent Systems*, pages 27–32. Springer-Verlag, 2012. [p1]

Osman Dag
Department of Biostatistics
Faculty of Medicine
Hacettepe University
06100 Ankara, Turkey
osman.dag@hacettepe.edu.tr

Ceylan Yozgatligil
Department of Statistics
Faculty of Arts and Sciences
Middle East Technical University
06531 Ankara, Turkey
ceylan@metu.edu.tr