Selecting Variables With Search Algorithms and Neural Networks to Improve the Process of Time Series Forecasting

Ivna Valença, Tarcísio Lucas and Teresa Ludermir

Informatics Center, Federal University of Pernambuco

Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, 50740-560, Recife-PE, Brazil

{icbv,tdpl,tbl}@cin.ufpe.br

Mêuser Valença

Department of Systems and Computer, University of Pernambuco

Benfica, 455, Madalena, Recife-PE, Brazil

meuserv@yahoo.com.br

Corresponding Author:

Ivna Valença

Informatics Center, Federal University of Pernambuco

Av. Jornalista Anibal Fernandes, s/n, Cidade Universitária, 50740-560, Recife-PE, Brazil

Tel: +55 81 8804 3472

icbv@cin.ufpe.br

Abstract

A time series is a sequence of observations of a random variable. Hence, it is a stochastic process. Forecasting time series data is important component of operations research because these data often provide the foundation for decision models. This models are used to predict data points before they are measured based on known past events. Researches in this subject have been done in many areas like economy, energy production, ecology and others. To improve the process of time series forecasting it is important to identify which of past values will be considered to be used in the models by eliminating redundant or irrelevant attributes. Two hybrid systems Harmony Search with Neural Networks (HS) and Temporal Memory Search with Neural Networks (TMS) are improved and a new one is proposed: the Temporal Memory Search Limited with Neural Networks (TMSL). The performance of the techniques is investigated through an empirical evaluation on twenty real-world time series.

Keywords: Harmony Search, Neural Networks, Temporal Memory Search, Time Series Forecasting, Variable Selection.

1 Introduction

Several real-world problems behave like a time series. Economy, ecology and energy production are some examples of research fields that face this kind of phenomenon [41]. To predict the behavior of time series is very important but it is not an easy task. The research related to time series forecasting has been an area of considerable interest in recent decades. Consequently, several models of time series forecasting have been developed over the years, taking into account their peculiarities. Artificial Neural Networks (ANNs) are one of them. Also, various hybrid systems associated with ANNs were proposed [42, 36, 19, 35, 32].

The process of time series forecasting is to predict future values based on past values [3]. Assuming the use of a series $Z(t) = (z_1, z_2, ..., z_t)$, the prediction of the future value z_{t+1} can be performed using past values z_t, z_{t-1}, z_{t-2}, z_{t-3},..., z_{t-p+1}, where p represents the number of past values considered (number of lags). In Fig. 1, two important steps in the process of time series forecasting are emphasized: the selection of input variables that will be included in the model and the prediction of the future value.

One of the major difficulties in applying models of time series forecasting is to determine which variables of the past are necessary to achieve the best accuracy [41]. The use of a large number of variables can make the accuracy of the model worst and also make it very complex. On the other hand, few variables can make the model simpler but also inaccurate. The main idea of attribute selection is to choose a subset of available attributes, eliminating features with few or no predictive information, and also redundant features that are strongly correlated. Proper selection of these factors that will be used for classification, clustering or prediction has an impact on the accuracy of the results achieved. Besides, when you have fewer variables as input values the time required to the prediction of the model is decreased. The optimal subset of attributes can be found by testing exhaustively all possible combination. However, as the number of possibilities increases, the computational cost grows exponentially (NP-complete problem) [1]. Thus, problems in large dimensionality domains are a major obstacle in machine learning and data mining.

The basic principle of hybrid systems is to join two or more intelligent computing techniques with the aim to unite potentials and eliminate individual limitations [13]. One limitation of selecting variables with ANNs is the computational cost of testing all possibilities. The use of a search algorithm to solve this problem is a classic application of hybrid intelligent systems [13]. Nowadays, ANNs with search algorithms have been widely used to select variables, where ANNs are used to evaluate the performance of the variables selected by the search algorithms.

In a previous paper we proposed the use of Harmony Search (HS), integrated with ANNs, to perform the selection of input variables for forecasting time series [37]. Moreover, we extend harmony search using correlations and memory. The extended method is called Temporal Memory Search (TMS). Our goal was to improve the speed without losing quality on results. Since the results of the two hybrid systems HS and TMS were satisfactory for most series tested we propose in this article to: increment the algorithm Harmony Search, including the assumption that there is not in HS all parts of the overall solution and including a mechanism for improving local search, create a new method call Temporal Memory Search Limited (TMSL) and double the number of experiments (time series).

This paper is organized as follows. In Section 2 a brief review of Variable Selection, also known as Feature Selection, is presented. Section 3 presents the models used in this work and is divided in five sections. Section 3.1 and 3.2 gives an introduction of two methods already proposed in literature. Sections 3.3, 3.4 and 3.5 describe the three proposed models. Section 4 presents the data used on the simulations and a comparative performance measurement of the results, this Section is divides in two: selecting variables (4.1) and evaluating the variables selected (4.2). Section 5 gives a summary and provides a conclusion and future works.

2 Variable Selection

More than eliminating useless variables for a given process, attribute selection involve: improving the performance of the model, the reduce of processing costs and facilitates the study of the phenomenon studied [26]. The most basic and complete form to select variables is to test exhaustively all possibilities. However, for each n attributes, it will be necessary $2^n - 1$ tests, a NP-complete problem [1]. If the number of variables is small it would be possible to test all possibilities, although would be computationally expensive. However, for most problems this number of n attributes is not small. In time series problems, for example, the optimal value for n is unknown, forcing the search into a large space of variables to guarantee a good solution. For these and other reasons the problem of feature selection is considerably difficult to solve [2] and in some cases intractable [15].

The high relevance of the attribute selection problem in several areas and the frequent inability to find the optimal solution by exhaustive search has been motivated the development of many methodologies, like Principal Component Analysis (PCA) [23], Simulated Annealing (SA) [25], Genetic Algorithms (GA) [18] Ant Colony Optimization (ACO) [6], Particle Swarm Optimization (PSO) [7], among others.

Some methodologies to solve attribute selection problem can be divided in two groups: filter and wrapper selection [22]. In the filter method, the process of variable selection is independent of the classification or prediction tool. In this case, it usually employs a measured gain of the filter information to guide the selection process. In the wrapper selection, the method uses a feedback from the classifier or predictor to guide the selection process. The feedback can be obtained by ANNs like Multi-Layer Perceptron (MLP) [39], Radial Basis Functions (RBF) [21] or Support Vector Machines (SVM) [33].

Several studies on variable selection already have been done [34, 4, 43, 20, 33, 40, 5]. Some search methods, such as GA, need the simultaneous evaluation of many solutions at every iteration, which requires a large computational effort. Some works like in [12] propose to use GA associated with SA, for example, to decrease the computational cost.

A new search algorithm called Harmony Search was proposed [11]. Harmony Search is interesting because at each iteration, it generates only one new solution to be evaluated [31]. Researches on feature selection, indicates that the wrapper technique seems to have a better performance comparing to others but have limited applications due of the high computational complexity involved [29]. However, Harmony Search is considered easy to implement [8]. The method proposed to select variables in this paper belongs to the category wrapper [26]. Wrapper techniques can be slow [38] so in this paper we propose three methods to improve speed without losing quality on results.

3 Methods

A search algorithm to be considered good, it should provide: a good global search that allows the exploration on the solution space of new subsets without getting into local minima, a convergence to a near optimal quickly, a good local search and a low computational cost [5]. In this work the focus is to find an optimal feature subset with a low computational cost and maintain the quality of the generate feature subset solutions.

In this section, two methods already proposed on literature are described, the Linear Forward Selection with MLP (LFS) a wrapper approach and the Correlation-based Feature Selection (CFS) a filter approach. Then, three methods are proposed: Harmony Search with MLP (HS), Temporal Memory Search with MLP (TMS) and Temporal Memory Search Limited with MLP (TMSL). Primarily, the purpose to use Harmony Search is to improve the performance of prediction algorithms. Secondly, through this selection process it is possible to eliminate redundant or irrelevant variables. So, the goal is to simplify the models of time series forecasting and reduce the computational cost to execute these models.

3.1 Linear Forward Selection with MLP (LFS)

In the wrapper approach, the attribute sets are evaluated by using a learning scheme and the cross-validation is used to estimate the accuracy of the learning scheme for a set of attributes [26]. In this paper we used the Linear Forward Selection with MLP (LFS). The LFS is an extension of the Best First Search (BFS). BFS is a search algorithm that uses AI strategy. This algorithm allows backtracking along the search path, like greedy hill climbing, moves through the search space by making local changes to the current feature subset. Differently from hill climbing, if the path being explored begins to look less promising, the BFS can back-track to a more promising previous subset and continue the search from there. Given enough time, a BFS will explore the entire search space, so it is common to use a stopping criterion.

The Linear Forward Selection takes a restricted number of k attributes into account. Fixed-set selects a fixed number k of attributes, whereas k is increased in each step when fixed-width is selected. The search uses either the initial ordering to select the top k attributes, or performs a ranking. The search direction can be forward, or floating forward selection [14].

3.2 Correlation-based Feature Selection (CFS)

The Correlation-based Feature Selection (CFS) is a selection based on correlation. It is a method in which a subset of variables is considered good if it contains variables highly correlated with the class, but then the variables are not correlated between them. The basis of CFS is a heuristic method for evaluating subsets that considers not only the usefulness of individual variables, but also the degree of correlation between them. The method is associated with each subset S of k variables a measure of performance called score which is a weight, such that the subset with the greatest score will be selected by the heuristic found [17].

CFS begins with a empty set of variables and search forward, or start with the full set of attributes and search backward, or start at any point and search in both directions. The heuristic Best First searches the space of attribute subsets by greedy hillclimbing augmented with a backtracking facility, setting the number of consecutive non-improving nodes allowed controls the level of backtracking done. In this paper we used CFS and Best First with forward search. CFS uses the heuristic best-fisrt-search with a stopping criterion of five consecutive subsets that do not improve the scores.

3.3 Harmony Search with MLP (HS)

Harmony Search is one technique for search and optimization problems introduced in 2001 [11] that have been very successful in managing problems [24, 9, 27, 10, 28]. As the name suggests, HS is inspired by the construction of musical harmonies, by trying to mimic the process of improvising music players. It has some advantages when compared to traditional algorithms, such as the genetic algorithm, among them two stand out [28]:

1. Only generates a new individual to the population at each iteration so that only one individual needs to be evaluated;
2. The initial selection of the set of variables is done probabilistically.

The steps of the Harmony Search algorithm are as follows [28]:

1. Initialize the optimization problem and algorithm parameters;

2. Initialize the Harmony Memory (HM);
3. Improvise a new harmony from the HM;
4. Update the HM;
5. Repeat Steps 3 and 4 until the termination criterion is satisfied.

For example, a selection problem with four attributes, where for each chosen subset exists an associated error, calculated, for example, by an MLP. The goal is to choose the subset of x that minimizes the error. Three harmonies are inserted to initiate the HM (h_1, h_2, h_3). It is sorted by an estimation measure. So the rank is defined by the error, where r_1 is the best result and so on (Fig. 2).

In order to generate a new harmony (step 3 above), a note of each instrument is randomly chosen. Thus, a new harmony h_4 is improvised and its error is calculated. If exists a worst harmony in the HM, this harmony is replaced by the new generated harmony. In this case h_3 is replaced for h_4. Finally, the HM is updated ordering the harmonies by the rank (Fig. 3).

The definition of HS assumes that all parts of the global solution exist initially in HM. If this does not happen, in order to find the optimum global, the HS starts a parameter called Harmony Memory Considering Rate (HMCR) [11], with value between 0 and 1. If a random value between 0 and 1 is higher than HMCR, so the HS finds notes randomly within the range of possible notes without considering HM. The HMCR equals to 0.95, for example, means that the next step the algorithm chooses a value independent of HM with a 95% probability.

Another way to improve solutions and escape from local optima is the adjustment mechanism (pitch). It works by exchanging neighboring values within a range of possibilities. If there are six possible values, such as (1, 3, 4, 6, 7, 9), (6) can be exchanged by neighbor (4) or (7). The rate adjustment, Pitch Adjusting Rate (PAR) [11] of 0.10 means that the algorithm chooses a value of neighbors with 10% probability (greater than 5% or less than 5%). This option simulates the adjustment of each instrument to improve the whole.

Automatically, the Harmony Search is responsible for the selection of input variables according to the probability equals 50%. When a new harmony is created, the notes that were selected (when the note is set with 1) correspond to the neurons that will be activated in the input layer of the

MLP (Fig. 4). So, this Neural Network calculates the Mean Square Error (MSE). The objective function of optimization on Harmony Search is the error founded on the MLP, the MSE. The minimization of the objective function (MSE) is sorted in rank (r_1, r_2, ..., r_n), being the first rank the lowest error. The steps of the Harmony Search with MLP (HS) algorithm are:

1. Initialize the algorithm parameters;
2. Initialize at random, the Harmony Memory (HM) values equal to 0 or 1 (since the probability is 0.5);
3. Calculate through the MLP the MSE using the neurons that were set as 1 in the HM;
4. Update the HM putting smaller errors in the first rank;
5. Improvise a new harmony in HM;
6. Repeat steps 3-5 until the stopping criterion (e.g. maximum number of cycles or/and the MSE with small variation).

The algorithm returns for each simulated population all the settings of the HM. The last setting is taken into account. The harmony of the first rank has the lowest error found. Thus, the number of variables and cycles needed are found.

3.4 Temporal Memory Search with MLP (TMS)

Based on the idea of the Harmony Search, a variant of this search technique is proposed, the Temporal Memory Search (Fig. 5) that is also responsible for the variable selection. The idea to use TMS comes from the fact that the process to define the past values (lags) can be viewed as a temporary memory, that is, until which lag should be considered to solve the problem. In general, values more distant in time are less important than the latest values. The TMS algorithm has the following steps:

1. Initially it is generated a set of Temporal Memories (TM);
2. Each TM is composed of a set of neurons where each neuron represents a past value of the variable to be considered as input variable on the MLP;

3. When this neuron is selected to be a input variable it will be activated with the value one (1) and will assume the value zero (0) when it will not be selected.

The Autocorrelation Function (ACF) is a statistical technique easy to use and it has been applied extensively to determine the number of *lags* in time series analysis. It is a measure of linear dependence between the random variable at time t and its value at time $t-k$, where k is the lag (delay). The autocorrelation is the auto covariance divided by the standard deviation. The auto covariance ρ_k between Z_t and Z_{t-k} can be defined by:

$$\rho_{k,m} = \frac{1}{N}\sum_{t=1}^{N}(Z_t - \mu_m)(Z_{t-k} - \mu_{m-k}) \tag{1}$$

where N is the number of observations; k is the lag; μ_m is a periodical estimative of average and m is period used.

Therefore, the autocorrelation $r_{k,m}$ can be defined by:

$$r_{k,m} = \frac{1}{N}\sum_{t=1}^{N}\frac{(Z_t - \mu_m)(Z_{t-k} - \mu_{m-k})}{\sigma_m \sigma_{m-k}} \tag{2}$$

where σ_m is an estimate of standard deviation for the period m.

However, the linear autocorrelation function measures only the degree of linear time dependence between two variables and does not capture the non-linear relations [3]. The correlation coefficient can be used, when working with time series, as the measure to characterize the linear dependence between past and present values. When dealing with complex phenomena that exhibit non-linear behaviour, the correlation coefficient serves only as estimation. Moreover, the higher the correlation coefficient the greater the temporal dependence between variables. Therefore, the proposed initial selection for the activation of neurons in the TMS algorithm is the probability of activation, which is directly proportional to the correlation coefficient. Based on this principle, the past variables with the greatest gap will have less chance of being selected.

The basic difference between HS and TMS is in the form of creation of the initial population of individuals. In Harmony Search, the harmonics are constructed randomly and each note has the same probability of being selected (50%). In Temporal Memory Search, the temporal memories are constructed in a probabilistic way such that the latest values are more likely to be selected (probability varies between 0%-100%). In other words, in HS all variables have the same probability of being randomly selected while in the TMS the variables have a probability of being selected based on the correlation coefficient.

3.5 Temporal Memory Search Limited with MLP (TMSL)

After calculate the probabilities of correlation between the variables it is possible to detect that for some cases the probabilities used on the model TMS were too small (e.g. 0.01%) or too high (e.g. 0.99%). This make the model rejects or accepts a variable without giving the variable a real chance of being or not being selected. In this case, we used another model (TMSL) with limited probabilities. For values smaller than 20%, we assume 0.20 (20%) and for values bigger than 80% we used 0.80 (80%).

4 Experiments and results

In order to verify the proposed models (HS, TMS and TMSL), twenty time series (TS) with distinct characteristics and complexities were selected[1]:

- Consumption: Time Series of consumption (shipments)
- CPI: Time Series of Cost-of-living Index
- Energy: Time Series of electric power consumption
- Flow FA: Time Series of daily flow in the reservoir of *Foz do Areia*
- Flow Sobradinho: Time Series of annual flow of the reservoir of *Sobradinho*
- GDP: Time Series of monthly Gross Domestic Product (GDP) from the Brazilian economy
- Humidity: Time Series of relative humidity
- IPI: Time Series of Industrial Production Index (IPI)
- Ozone: Time Series of monthly values of ozone

[1]The series can be found at `www.stat.duke.edu/~mw/ts_data_sets.html` and `www.ime.usp.br/~pam/ST`

- Petrobras: Time Series of daily price of Petrobras (stock market)[2]
- PFI: Time Series of Private Finance Initiative (PFI)
- Pollution CO: Time Series of pollution of carbon monoxide (CO)
- Pollution SO2: Time Series of pollution of sulfur dioxide (SO2)
- Rain Fortaleza: Time Series of atmospheric precipitation (rain) of *Fortaleza* city
- Rain Lavras: Time Series of atmospheric precipitation (rain) of *Lavras* city
- Sea-level: Time Series of monthly values of the Darwin sea level pressures
- SOI: Time Series of monthly values of the The Southern Oscillation Index (SOI)
- Sunspot: Time Series of sunspots
- Temp. Cananéia: Time Series of temperature of *Cananéia* city
- Temp. SP: Time Series of temperature of *São Paulo* city

In Table 1, the characteristics (period of observation, presence of stationarity, trend, seasonality and length) of each time series are displayed. The real-world time series selected for the experiments seek to cover most possible features that can be found in the series. These analysis were performed using the software R (a programming language and software environment for statistical computing and graphics). The presence of stationarity was detected by the t test between average of the first half and the second half part of the series. To check the trend of the series it was held the Cox and Stuart's test. The seasonality could be detected by the graphs.

The experiments were divided in two stages: the selection of the variables using the methods and the use of these variables in the predictor (MLP). In both phases, for each experiment using MLP-BP (trained by the backpropagation algorithm), the data of each base were divided in training (50%), validation (25%) and tests (25%). The network weights were determined randomly between -1 and 1. The network training stopped when the validation error increased five times in succession or when the maximum number of cycles was reached.

[2]Petróleo Brasileiro S. A., Brasil.

4.1 Selecting Variables

In the variable selection each technique began with a search space of 30 past values. By analyzing the time series, the initial number of the search space was chosen by the Partial Autocorrelation Function (PACF). The PACF determines the order p (lags) of the autoregressive process (AR) [3]. For all models we used the same initial set.

The models LFS and CFS were simulated using the Weka [16]. In this case, each time series was divided in two sets: training (75%) and test (25%). We used training set and the default parameters, as follows:

1. LFS: classifier = MLP, seed = 1, threshold = 0.01; with Foward Selection: forwardSelectionMethod = Forward Selection, lookupCacheSize = 1, numUsedAttributes = 50, performRanking = True, searchTermination = 5, startSet = blank, type = Fixed-set, verbose = False; and,

2. CFS: locallyPredictive = True, missingSeparate = False; with Best First: direction = Forward, lookupCacheSize = 1, searchTermination = 5, startSet = blank.

The proposed models were implemented in Java. The fitness of each harmony was the error generated by an MLP-BP with fixed parameters (learning rate = 0.3 and momentum = 0.2) for online learning. It is important to remember that at this phase of the experiments the error (Mean Square Error) generated by the MLP-BP aims only to differentiate the quality of the harmonies generated during the search process. The settings used by the series in the hybrid model HS are shown in Table 2. For the model TMS the settings are the same as HS, the difference is only the probability. In the HS, they are all set at 50%, and in the TMS, the probabilities are based on the Autocorrelation Function (0%-100%). The probabilities used on TMSL were limited at 20%-80%. The experiments were carried out with different population size because when we have a small number of harmonies (or memories) the algorithm is not able to find the optimal value. In contrast, the computational cost is greatly reduced when compared with larger populations.

Each set of experiment was repeated 30 times for each database. The best harmonies (or memories) were those that had the lowest validation error. In case of finding the same error, the set, which had the lowest number of variables was choose. The same happened for the TMS and

TMSL. In Table 3 and Table 4, the results of the five models (LFS, CFS, HS, TMS and TMSL) are displayed for the twenty real-world time series chosen.

The runtime of the algorithm HS was always twice comparing to the TMS algorithm. The model TMSL converges faster than the models HS and TMS. The model TMS selected the same number of variables or in many cases less variables comparing to the model HS. With the selection of few variables the experiment becomes faster. For the Flow FA, a series with a long memory, the models proposed were better. The model TMSL proposed behaves like the TMS when the probabilities are not too high and like HS in otherwise.

4.2 Evaluation of selected variables

Variables selected by different techniques were placed into the MLP-BP. In this phase we used a new architecture and parameters for training the MLP-BP (learning rate = 0.3, momentum = 0.1, input neurons = see Table 3 and Table 4, hidden neurons = see Table 2). The maximum number of training cycles was 10000. Each simulation was repeated 30 times and the error found for each time series can be seen in Table 5. The results are displayed as: the average of the Mean Absolute Percentage Error and in parentheses the corresponding standard deviation. The Mean Absolute Percentage Error (MAPE) usually expresses accuracy as a percentage and is defined by the formula:

$$MAPE = \frac{1}{n}\sum_{t=1}^{n}|\frac{A_t - F_t}{A_t}| \tag{3}$$

Where A_t is the actual value and F_t is the forecast value. The difference between A_t and F_t is divided by the actual value A_t again. The absolute value of this calculation is summed for every fitted or forecast point in time and divided again by the number of fitted points n. This makes it a percentage error so one can compare the error of fitted time series that differ in level. When having a perfect fit, MAPE is zero.

The Student's t-test, or simply t-test, is a statistical hypothesis test in which the test statistic follows a Student's t distribution if the null hypothesis is supported [30]. The t-test were performed to compare the two averages of error (MAPE) to evaluate if there is a statistically significant difference between the models: LFS, CFS, HS, TMS and TMSL. The values considered different and statistically significant (with 95% of confidence) were those that had values bigger or equal than 1.96 (critical point). Thus, it was possible to determine the best model or models for each series. In Table 5 these results are highlighted in bold. The Fig. 6 shows how many times a model had a better performance comparing to the other models. It is possible to see that for the proposed models TMS and TMSL presented good results for these time series.

To evaluate the process of variable selection is not an easy task. We have to look for all kinds of results: e.g. a trade-off between the error of forecasting and the number of variables selected (in search of a parsimonious model). Also, we need to look for the runtime, among other things.

5 Summary and conclusions

This study proposed three prediction models based on Multi-Layer Perceptron with a hybrid variable selection method: Harmony Search with Neural Network (HS), Temporal Memory Search with Neural Network (TMS) and Temporal Memory Search Limited with Neural Network (TMSL). The models proposed here seek to increase the speed of runtime without loosing quality on the results. We compare our work against the following algorithms: Linear Forward Selection with MLP (LFS) and Correlation-based Feature Selection (CFS). Then, we study the performance of the algorithms for twenty real-world time series. The performance of the proposed models is highly encouraging.

Keeping in mind that there is not a single learning algorithm superior to all others for all problems. Researches in machine learning try to provide insight into the strengths and limitations of these different algorithms. With this and background knowledge for a particular problem, it is possible to choose which algorithms will be used to solve that particular problem.

As a future research line, it is necessary the study of others metrics to be used to establish the probabilities on the Temporal Memory for the TMS and TMSL methods, e.g. Mutual Information. Also, would be interesting to investigate another machine learning, e.g. Support Vector Machine.

Acknowledgements

The authors would like to thank CNPq, FACEPE and CAPES (Brazilian research agencies) for their financial support.

References

[1] A. Albrecht. Stochastic local search for the feature set problem, with applications to microarray data. *Applied Mathematics and Computation*, 183(2):1148–1164, 2006.

[2] A. L. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97:245–271, 1997.

[3] G. E. P. Box, G. M. Jenkins, and G. C. Reinsel. *Time series analysis: forecasting and control*. Wiley, 4th Edition, San Francisco, 2008.

[4] A. Campoccia, L. Dusonchet, A. Augugliaro, E. Sanseverino, and M. Di Silvestri. Ga-based feature selection for faults identification in electrical distribution systems. In *Electric Power Engineering, 1999. PowerTech Budapest 99. International Conference on*, page 186. IEEE, 2002.

[5] S. F. Crone and N. Kourentzes. Feature selection for time series prediction: a combined filter and wrapper approach for neural networks. *Neurocomputing*, 73(10-12):1923 – 1936, 2010. Subspace Learning / Selected papers from the European Symposium on Time Series Prediction.

[6] M. Dorigo and C. Blum. Ant colony optimization theory: a survey. *Theoretical Computer Science*, 344(2-3):243–278, 2005.

[7] R. Eberhart and Y. Shi. Particle swarm optimization: developments, applications and resources. *Proceedings of the 2001 congress on evolutionary computation*, 1:81–86, 2001.

[8] Z. Geem. Optimal cost design of water distribution networks using harmony search. *Engineering Optimization*, 38(3):259–277, 2006.

[9] Z. Geem, J. Kim, and G. Loganathan. Harmony search optimization: application to pipe network design. *International journal of modelling & simulation*, 22(2):125–133, 2002.

[10] Z. Geem, C. Tseng, and Y. Park. Harmony search for generalized orienteering problem: best touring in china. *Lecture Notes in Computer Science*, 3612:741–750, 2005.

[11] Z. W. Geem, J. H. Kim, and G. V. Loganathan. A new heuristic optimization algorithm: harmony search. *SIMULATION*, 76(2):60–68, 2001.

[12] I. Gheyas and L. Smith. Feature subset selection in large dimensionality domains. *Pattern Recognition*, 43(1):5–13, 2010.

[13] S. Goonatilake and S. Khebbal. Intelligent hybrid systems: issues, classifications and future directions. *Intelligent Hybrid Systems*, pages 1–20, 1995.

[14] M. Gutlein, E. Frank, M. Hall, and A. Karwath. Large-scale attribute selection using wrappers. In *Computational Intelligence and Data Mining, 2009. CIDM'09. IEEE Symposium on*, pages 332–339. IEEE, 2009.

[15] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *J. Mach. Learn. Res.*, 3:1157–1182, 2003.

[16] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *ACM SIGKDD Explorations Newsletter*, 11:10–18, 2009.

[17] M. A. Hall. *Correlation-based feature selection for machine learning*. Tese de doutorado, University of Waikato (Department of Computer Science), 1999.

[18] J. Holland. *Adaptation in natural and artificial systems*. MIT press Cambridge, MA, 1992.

[19] A. Jain and A. M. Kumar. Hybrid neural network models for hydrologic time series forecasting. *Applied Soft Computing*, 7(2):585–592, 2007.

[20] R. Jensen. Performing feature selection with aco, swarm intelligence and data mining. *Studies in Computational Intelligence*, 34:45–73, 2006.

[21] P. Jia and N. Sang. Feature selection using a radial basis function neural network based on fuzzy set theoretic measure. *Third International Symposium on Multispectral Image Processing and Pattern Recognition*, 5286(1):109–114, 2003.

[22] G. John, R. Kohavi, and K. Pfleger. Irrelevant features and the subset selection problem. In *Proceedings of the eleventh international conference on machine learning*, volume 129, pages 121–129. Citeseer, 1994.

[23] I. Jolliffe. *Principal component analysis.* Springer, 2nd Edition, 520 pages, New York, 2010.

[24] J. Kim, Z. Geem, and E. Kim. Parameter estimation of the nonlinear muskingum model using harmony search. *Journal of the American Water Resources Association*, 37(5):1131–1138, 2001.

[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi. Optimization by simulated annealing. *Science, New Series*, 220(4598):671–680, 1983.

[26] R. Kohavi and G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[27] K. Lee and Z. Geem. A new structural optimization method based on the harmony search algorithm. *Computers and Structures*, 82(9-10):781–798, 2004.

[28] K. Lee and Z. Geem. A new meta-heuristic algorithm for continuous engineering optimization: harmony search theory and practice. *Computer methods in applied mechanics and engineering*, 194(36-38):3902–3933, 2005.

[29] M. Lee. Using support vector machine with a hybrid feature selection method to the stock trend prediction. *Expert Systems with Applications*, 36(8):10896–10904, 2009.

[30] E. Lehmann and J. Romano. *Testing statistical hypotheses.* Springer Verlag, 2005.

[31] L. Li, G. Yu, X. Chu, and S. Lu. The harmony search algorithm in combination with particle swarm optimization and its application in the slope stability analysis. In *Computational Intelligence and Security, 2009. CIS'09. International Conference on*, volume 2, pages 133–136. IEEE, 2010.

[32] P.-F. Pai, K.-P. Lin, C.-S. Lin, and P.-T. Chang. Time series forecasting by a seasonal support vector regression model. *Expert Systems with Applications*, 37(6):4261–4265, 2010.

[33] T. Pfingsten, D. Herrmann, T. Schnitzler, A. Feustel, and B. Scholkopf. Feature selection for troubleshooting in complex assembly lines. *IEEE Transactions on Automation Science and Engineering*, 4(3):465–469, 2007.

[34] P. Pudil, J. Novovičová, and J. Kittler. Floating search methods in feature selection. *Pattern Recogn. Lett.*, 15(11):1119–1125, 1994.

[35] N. Spanoudakis, K. Pendaraki, and G. Beligiannis. Portfolio construction using argumentation and hybrid evolutionary forecasting algorithms. *International Journal of Hybrid Intelligent Systems*, 6(4):231–243, 2009.

[36] K. Srinivasa, K. Venugopal, and L. Patnaik. An efficient fuzzy based neuro-genetic algorithm for stock market prediction. *International Journal of Hybrid Intelligent Systems*, 3(2):63–81, 2006.

[37] I. C. B. Valena, T. B. Ludermir, and M. J. S. Valena. Hybrid systems to select variables for time series forecasting using mlp and search algorithms. *2010 Eleventh Brazilian Symposium on Neural Networks (SBRN)*, pages 247–252, 2010.

[38] G. Van Dijck and M. Van Hulle. Speeding up the wrapper feature subset selection in regression by mutual information relevance and redundancy analysis. *Artificial Neural Networks–ICANN 2006*, pages 31–40, 2006.

[39] A. Verikas and M. Bacauskiene. Feature selection with neural networks. *Pattern Recogn. Lett.*, 23(11):1323–1335, 2002.

[40] W. Xiong and C. Wang. Feature selection: a hybrid approach based on self-adaptive ant colony and support vector machine. *Proceedings of the 2008 International Conference on Computer Science and Software Engineering*, 4:751–754, 2008.

[41] G. Zhang, B. E. Patuwo, and M. Y. Hu. Forecasting with artificial neural networks: The state of the art. *International Journal of Forecasting*, 14(1):35–62, 1998.

[42] G. P. Zhang. Time series forecasting using a hybrid arima and neural network model. *Neurocomputing*, 50:159–175, 2003.

[43] S. Zhou and J. Jin. Automatic feature selection for unsupervised clustering of cycle-based signals in manufacturing processes. *Institute of Industrial Engineers (IIE Transactions)*, 37(6):569–584, 2005.

Table 1: Statistical characteristics of time series

Time Series	Period	Stationary	Trend	Seasonality	Length
Consumption	Monthly	Yes	No	Yes	154
CPI	Monthly	No	Yes	No	126
Energy	Monthly	No	Yes	No	141
Flow FA	Daily	No	Yes	Yes	4017
Flow Sobradinho	Annual	Yes	Yes	Yes	76
GDP	Monthly	Yes	Yes	Yes	216
Humidity	Daily	Yes	No	Yes	365
IPI	Monthly	No	Yes	Yes	187
Ozone	Monthly	Yes	No	Yes	180
Petrobras	Daily	Yes	No	No	1499
PFI	Monthly	No	Yes	Yes	115
Pollution CO	Daily	Yes	Yes	Yes	365
Pollution SO2	Daily	Yes	Yes	Yes	365
Rain Fortaleza	Annual	No	No	Yes	149
Rain Lavras	Monthly	Yes	No	Yes	384
Sea-level	Monthly	Yes	No	Yes	1400
SOI	Monthly	Yes	Yes	Yes	540
Sunspot	Annual	Yes	Yes	Yes	175
Temp. Cananéia	Monthly	Yes	No	Yes	120
Temp. SP	Daily	Yes	No	Yes	365

Table 2: Configuration parameters for the models HS + MLP

Time Series	Input	Hidden	Output	Population	Probability	Cycles
Consumption	12	4	1	10, 20, 30	0.5	1000
CPI	12	6	1	10, 20, 30	0.5	1000
Energy	12	6	1	10, 20, 30	0.5	1000
Flow FA	30	10	1	10, 20, 30	0.5	1000
Flow Sobradinho	15	4	1	10, 20, 30	0.5	1000
GDP	30	5	1	10, 20, 30	0.5	1000
Humidity	10	6	1	10, 20, 30	0.5	1000
IPI	30	5	1	10, 20, 30	0.5	1000
Ozone	30	5	1	10, 20, 30	0.5	1000
Petrobras	12	4	1	10, 20, 30	0.5	1000
PFI	30	5	1	10, 20, 30	0.5	1000
Pollution CO	10	6	1	10, 20, 30	0.5	1000
Polluition SO2	30	5	1	10, 20, 30	0.5	1000
Rain Fortaleza	30	5	1	10, 20, 30	0.5	1000
Rain Lavras	12	8	1	10, 20, 30	0.5	1000
Sea-level	30	5	1	10, 20, 30	0.5	1000
SOI	30	5	1	10, 20, 30	0.5	1000
Sunspot	14	4	1	10, 20, 30	0.5	1000
Temp. Cananéia	30	5	1	10, 20, 30	0.5	1000
Temp. SP	30	5	1	10, 20, 30	0.5	1000

Table 3: Results for variable selection (1/2)

Time Series	LFS	CFS	HS	TMS	TMSL
Consumption	6(t-11,t-10, t-5,t-3,t-1 ,t)	3(t-11,t-5,t)	5(t-7,t-5,t-4, t-1,t)	1(t-11)	1(t-11)
CPI	5(t-10,t-7,t-6, t-1,t)	1(t)	6(t-11,t-9,t-6, t-5,t-1,t)	11(t-11,t-10,t-8, t-7,t-6,t-5,t-4,	6(t-11,t-9,t-6, t-5,t-1,t)
Energy	3(t-2,t-1,t)	4(t-10,t-6, t-1,t)	4(t-4,t-3, t-2,t)	4(t-6,t-3, t-1,t)	4(t-4,t-3, t-1,t)
Flow FA	5(t-9,t-6,t-3 ,t-1,t)	1(t)	16(t-29,t-28, t-27,t-25,t-24, t-18,t-16,t-15, t-9,t-8,t-6, t-5,t-3,t-2, t-1,t)	10(t-27,t-26, t-10,t-9,t-5, t-4,t-3,t-2, t-1,t)	10(t-27,t-26, t-10,t-9,t-5, t-4,t-3,t-2, t-1,t)
Flow Sobradinho	3(t-10,t-4,t)	4(t-9,t-6, t-1,t)	6(t-13,t-11, t-9,t-5,t-3 ,t)	3(t-5,t-1,t)	7(t-14,t-13, t-4,t-3,t-1, t)
GDP	6(t-16,t-14, t-11,t-3, t-1,t)	2(t-11,t)	14(t,t-2,t-3, t-6,t-7,t-9, t-10,t-14,t-18, t-23,t-24,t-25, t-28,t-29)	12(t-4,t-6,t-7, t-8,t-14,t-15, t-18,t-19,t-23, t-25,t-28,t-29)	12(t-5,t-6, t-14,t-15,t-18, t-19,t-21,t-23, t-24,t-25,t-28, t-29)
Humidity	3(t-8,t-1,t)	2(t-6,t)	3(t-6,t-3,t)	2(t-2,t)	2(t-2,t)
IPI	6(t-29,t-18, t-12,t-11, t-8,t)	1(t)	14(t-1,t-6,t-7, t-13,t-14,t-18, t-19,t-21,t-22, t-24,t-25,t-26, t-28,t-29)	9(t-3,t-6,t-7, t-9,t-18,t-19, t-21,t-25,t-29)	8(t-2,t-6,t-7, t-9,t-13,t-18, t-19,t-20)
Ozone	6(t-23,t-11, t-9,t-6,t-5, t)	5(t-23,t-22, t-11,t-5,t)	16(t-2,t-3,t-4, t-5,t-9,t-12, t-13,t-15,t-17, t-18,t-19,t-20, t-21,t-26, t-27,t-28)	11(t-2,t-5,t-7, t-12,t-13,t-17, t-18,t-19,t-24, t-25,t-26)	10(t-2,t-5,t-6,) t-12,t-13,t-14, t-17,t-19, t-26,t-28
Petrobras	2(t-11,t-8)	6(t-8,t-7,t-5, t-4,t-2,t)	5(t-5,t-4,t-3, t-2,t-1)	6(t-8,t-7,t-5, t-4,t-2,t)	3(t-7,t-5,t-1)
PFI	4(t-23,t-11, t-4,t)	5(t-29,t-23, t-11,t-10,t)	15(t-1,t-3,t-4, t-5,t-6,t-10, t-13,t-15,t-18, t-19,t-21,t-22, t-23,t-28,t-29)	6(t-1,t-6,t-7, t-18,t-19,t-29)	7(t-1,t-6, t-12,t-18,t-19, t-26,t-28)

Table 4: Results for variable selection (2/2)

Time Series	LFS	CFS	HS	TMS	TMSL
Pollution CO	5(t-5,t-4, t-3,t-1,t)	2(t-9,t)	5(t-8,t-7, t-4,t-2,t)	3(t-9,t-8,t)	3(t-9,t-8,t)
Pollution SO2	2(t-1,t-2)	8(t-28,t-21, t-17,t-10,t-4, t-2,t-1,t)	17(t,t-1,t-2, t-3,t-4,t-5, t-9,t-11,t-12, t-13,t-14,t-16, t-19,t-25,t-26, t-28,t-29)	10(t-2,t-3, t-5,t-9,t-11, t-15,t-21,t-25, t-26,t-29)	14(t,t-2,t-3, t-5,t-9,t-11, t-13,t-16,t-20, t-24,t-25,t-26, t-28,t-29)
Rain Fortaleza	3(t-24,t-5 ,t)	9(t-26,t-25, t-24,t-23, t-22,t-19, t-7,t-5,t)	15(t,t-1,t-5, t-14,t-17,t-20, t-21,t-22,t-23, t-24,t-25,t-26, t-27,t-28,t-29)	10(t,t-4,t-5, t-6,t-9,t-11, t-22,t-23, t-24,t-25)	7(t-4,t-5, t-14,t-23,t-24, t-25,t-26)
Rain Lavras	4(t-6,t-5, t-4,t)	6(t-11,t-10, t-7,t-6,t-5 ,t-4)	7(t-11,t-10, t-9,t-8,t-7, t-6, t-2)	7(t-11,t-10, t-9,t-7,t-6, t-5, t)	7(t-11,t-10, t-9,t-8,t-7, t-6, t)
Sea-level	8(t-29,t-22, t-18,t-16,t-15, t-11,t-1,t)	9(t-28,t-23, t-18,t-17,t-16, t-11,t-10, t-1,t)	12(t-1,t-2, t-4,t-7,t-9, t-13,t-14,t-18, t-19,t-20,t-28, t-29)	15(t-1,t-7, t-11,t-12,t-13, t-14,t-18,t-19, t-20,t-21,t-24, t-25,t-27,t-28, t-29)	11(t-1,t-7, t-13,t-14,t-18, t-19,t-20,t-24, t-25,t-28, t-29)
SOI	4(t-13,t-4, t-1,t)	4(t-18,t-4, t-1,t)	18(t,t-1,t-2, t-3,t-4,t-5, t-7,t-8,t-10, t-11,t-12,t-13, t-16,t-17,t-18, t-20,t-21,t-22)	7(t,t-1, t-5,t-21, t-22,t-26, t-29)	5(t,t-17, t-22,t-26, t-29)
Sunspot	4(t-8,t-5, t-1,t)	3(t-10, t-4, t)	3(t-7, t-6, t)	3(t-7, t-6, t)	4(t-12,t-9, t-2, t-1)
Temp. Cananéia	3(t-29,t-5, t-1)	8(t-29,t-28, t-18,t-12,t-11, t-5,t-4,t)	11(t-2,t-9, t-1,t-11,t-16, t-17,t-18, t-20,t-27, t-28,t-29)	14(t,t-1,t-2, t-8,t-11,t-12, t-13,t-14,t-18, t-19,t-20,t-25, t-26,t-27)	13(t-2,t-12, t-13,t-14,t-18, t-19,t-20,t-22, t-24,t-25,t-26, t-27,t-28)
Temp. SP	3(t-23, t-13,t)	3(t-15, t-1,t)	14(t,t-1, t-2,t-4,t-5, t-8,t-10,t-11, t-13,t-16,t-23, t-24,t-28, t-29)	12(t-1,t-4, t-8,t-12,t-13, t-15,t-16,t-18, t-23,t-24, t-28,t-29)	14(t-1,t-2, t-3,t-4,t-5, t-8,t-11,t-12, t-13,t-15,t-16, t-23,t-28, t-29)

Table 5: Results for the forecasting using variable selection (values displayed in percent)

Time Series	LFS	CFS	HS	TMS	TMSL	MLP
Consumption	**1.12** **(0.103)**	1.54 (0.240)	2.00 (0.337)	1.19 (0.134)	1.20 (0.118)	1.64 (0.173)
CPI	0.06 (0.011)	0.07 (0.002)	0.06 (0.011)	0.07 (0.014)	0.06 (0.012)	**0.02** **(0.005)**
Energy	0.09 (0.006)	0.08 (0.002)	0.08 (0.001)	0.09 (0.002)	0.07 (0.003)	**0.06** **(0.005)**
Flow FA	1.05 (0.016)	1.09 (0.011)	**0.11** **(0.003)**	0.26 (0.024)	0.27 (0.024)	0.12 (0.010)
Flow Sobradinho	0.21 (0.020)	**0.19** **(0.008)**	0.20 (0.007)	**0.19** **(0.005)**	0.21 (0.002)	0.21 (0.022)
GDP	0.06 (0.001)	0.08 (0.001)	0.03 (0.001)	**0.02** **(0.001)**	**0.02** **(0.001)**	**0.02** **(0.001)**
Humidity	0.08 (0.001)	0.08 (0.001)	0.08 (0.001)	0.08 (0.001)	0.08 (0.001)	0.08 (0.002)
IPI	0.05 (0.001)	0.10 (0.001)	0.05 (0.003)	**0.04** **(0.002)**	**0.04** **(0.003)**	0.05 (0.004)
Ozone	0.22 (0.005)	0.24 (0.008)	0.16 (0.013)	0.17 (0.010)	**0.15** **(0.009)**	0.21 (0.012)
Petrobras	**0.90** **(0.093)**	**0.92** **(0.155)**	0.96 (0.184)	**0.85** **(0.155)**	**0.86** **(0.222)**	**0.88** **(0.159)**
PFI	0.05 (0.001)	0.04 (0.001)	**0.03** **(0.001)**	**0.03** **(0.001)**	**0.03** **(0.002)**	**0.03** **(0.003)**
Pollution CO	0.32 (0.003)	0.30 (0.002)	0.31 (0.003)	0.30 (0.002)	0.30 (0.002)	**0.23** **(0.013)**
Pollution SO2	0.50 (0.007)	0.38 (0.007)	**0.28** **(0.006)**	0.29 (0.006)	**0.28** **(0.005)**	0.30 (0.008)
Rain Fortaleza	**0.29** **(0.006)**	**0.29** **(0.007)**	**0.29** **(0.009)**	**0.29** **(0.007)**	**0.29** **(0.011)**	0.30 (0.009)
Rain Lavras	2.93 (0.442)	2.40 (0.362)	2.39 (0.290)	**2.09** **(0.173)**	**2.10** **(0.178)**	**2.09** **(0.541)**
Sea-level	0.09 (0.007)	0.10 (0.002)	**0.08** **(0.004)**	**0.08** **(0.004)**	**0.08** **(0.005)**	0.09 (0.004)
SOI	**0.84** **(0.075)**	**0.84** **(0.093)**	1.04 (0.060)	0.94 (0.029)	0.91 (0.015)	1.08 (0.038)
Sunspot	17.88 (0.269)	15.31 (0.269)	18.63 (0.415)	18.63 (0.483)	16.90 (0.278)	**4.69** **(0.903)**
Temp. Cananéia	0.06 (0.001)	**0.04** **(0.003)**	**0.04** **(0.003)**	**0.04** **(0.004)**	**0.04** **(0.004)**	**0.04** **(0.003)**
Temp. SP	0.12 (0.001)	0.13 (0.003)	0.09 (0.001)	**0.08** **(0.001)**	**0.08** **(0.001)**	0.10 (0.002)

Figure captions

Fig. 1: Process of time series forecasting: input, model and output.

Fig. 2: Harmony Search: improvise a new harmony from the HM.

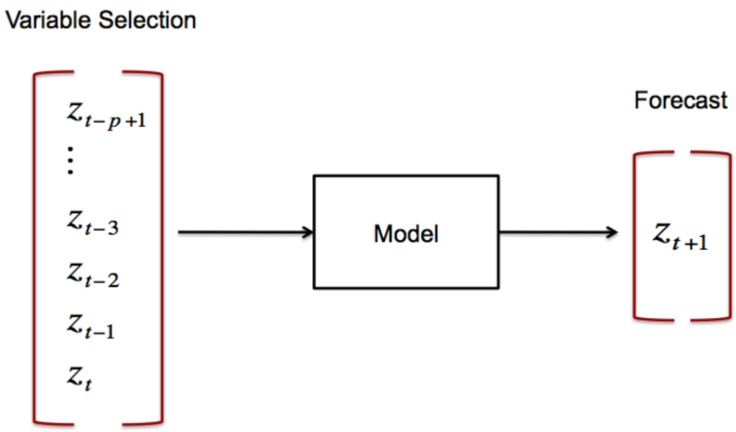
Fig. 3: Harmony Search: update the HM.

Fig. 4: Structure of the HS model.

Fig. 5: Structure of the TMS model.

Fig. 6: Comparison of model results (LFS, CFS, HS, TMS, TMSL and MLP).

Fig. 1



Variable Selection
z_{t-p+1}
$\vdots$
z_{t-3}
z_{t-2}
z_{t-1}
z_t
Model
Forecast
z_{t+1}

Fig. 2

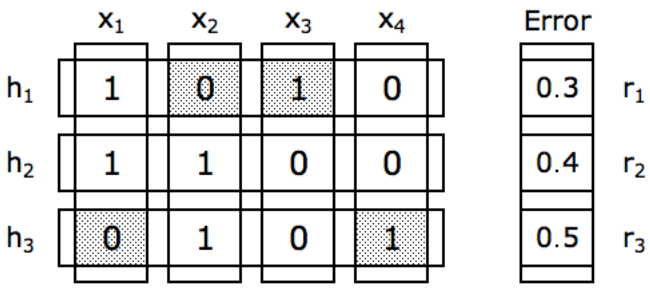


Fig. 3

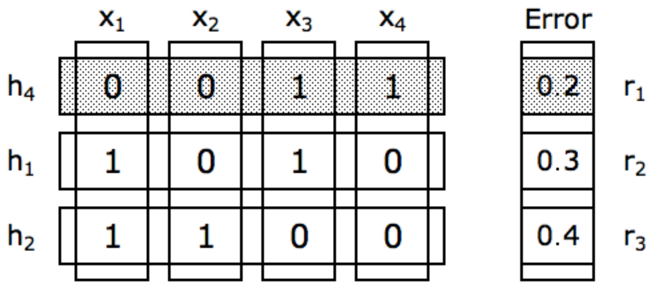


Fig. 4

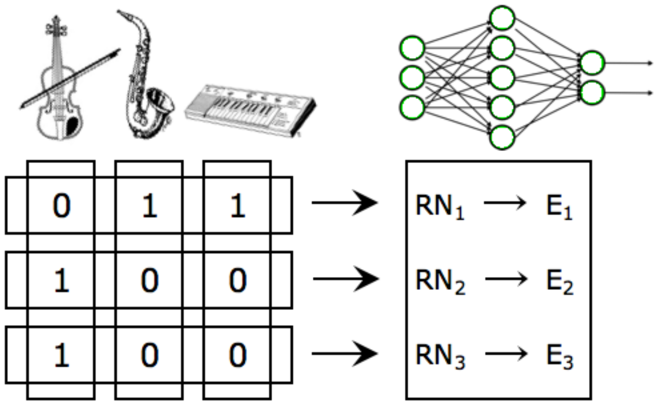


Fig. 5

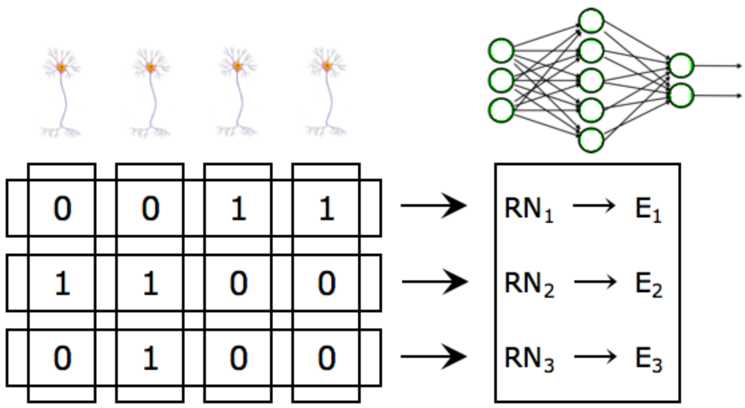


Fig. 6