FEATURE SELECTION FOR TIME-SERIES PREDICTION IN CASE OF NONDETERMINED ESTIMATION

Khmylovyy S.V. Skobtsov Y.A.
Chair of Automated Control Systems
Donetsk State Technical University. 58, Artema St., 83000 Donetsk, Ukraine,
hmelevoy@kita.dgtu.donetsk.ua , skobtsov@kita.dgtu.donetsk.ua
tel.+38-062-304-90-20, +38-050-614-29-40

Abstract.
The issues of factors selection are discussed in the article for the case when estimation of a set of factors is not stochastic. Here the quality comparison of two sets of factors is only possible with some probability, and modification of existing methods is required for their correct operation. For this purpose there is a proposal of CGA Compact Genetic Algorithms utilization the scheme of factor selection being indicated. For stochastic estimation of a set of factors the step of training is updated for genetic algorithms. Results are obtained for the standard benchmarks and Internet - traffic forecasting task.

Keywords: Data mining, Evolutionary computations, Forecasting, Time series.

1. The state of the feature selection problem

The Feature selection problem refers to the Data Preprocessing task in data Mining and Knowledge Discovery. In [21] the feature selection and the feature extraction tasks are being distinguished from the Data Preprocessing. Blum, Langley [5] divide the feature selection methods into embedded approach, filtering approach and wrapper approach.

Traditionally, for the feature selection task mathematical apparatus of correlation analysis, the detachment of linear dependences etc. were used. Baestans in [2] shows that for the dropping of insignificant input factor both, the presence of high level correlation between this and other input factors, and the lower one between this input variable and output variable are possible. Hattingh, Kruger [13] use mixed integer linear programming for the combined removal of unimportant factors and for the filtration of data that cause the most noise in the prediction. The significance of factors is defined with the help of the linear model. Ahmad, Dey [1] use probability based method to extract the significant attributes. F. Moerchen [18] offers the DWT and DFT based modified algorithm for the reduction of the set of data.

During the input factors interaction analysis Ezhov [7] uses Principal Component Analysis for the reduction of input dimension. It is based on the dropping of those inputs that have minor value of the covariance matrix, that takes into account only linear interaction. There one can observe the application of neural networks (NN), used for the realization of the nonlinear principal component method and allowing the high-order interactions. For the input factors and output variable interactions assessment it is proposed to use Box-counting algorithms based on calculating the Training Data examples occupation number of the boxes, into which the space of variables is being divided. The suboptimal algorithm of the serial addition of significant inputs, that on every stage of its work adds one more factor, which is the most significant together with selected ones, to the select set, is also suggested there.

The use of genetic algorithms (GA) for feature selection is justified and can compete with other methods in its efficiency. Such methods refer to wrapper-methods. Freitas [8], Vafaie, De Jong [23] suggest the classical approach to coding of individuals and to GA operators. The use of the classical approach is also shown at Guerra-Salcedo, Whitley [12], Vafaie, De Jong [23]. Minaei-

Bidgoli, Punch [17] suggest to use GA not only in factors selection, but also in determining their significance. In Hsu et al. [14] the GA are used not only in attribute selection, but in attribute partitioning too, that is creating the new attributes based on the group of old ones. In Oh et al. [19] like in Ezhov [7] the idea of sequential adding/removal of the most/ the least significant inputs is developed, having made the adding and removal of inputs a part of the hybrid genetic algorithm.

The idea of using GA as a set of factors selector combined with other methods as classifiers is also known enough. Bala et al. [4] uses GA as a “filter+wrapper” for a decision-tree learning algorithm, which directly carries out the classification. Ibid and also in Raimer et al. [21] GA is applied together with k-nearest neighbor algorithm. In Bala et al. [3] GA is used for the factor selection and the decision tree serves as a predictor.

The combination “GA+NN” can be observed, for instance, in Gruau, Whitley [11]. GA there is used for the creation of a grammar tree that yields both architecture and weights, specifying a particular neural network for solving specific Boolean functions. The neural networks for classification are used in Yang, Honawar [25]. The selection of significant factors for the neural network is carried out by genetic algorithm, which estimates a multi-criteria task (the precision of the prediction and the cost of factor value obtaining).

But there exist some conditions, by which an applicability of these methods has its limitations, and the results delivered are nonoptimal. In case the quality assessment of the specific data set has a stochastic nature, the application of the most methods is limited. For the qualitative assessment of every attribute subset its multiple estimation is essential, that requires an algorithm revision. Both, neural networks and genetic algorithms which carry out the prediction, can act as an environment giving the stochastic quality assessment of the factors set. While selecting the specific influencing factors it is reasonable to use genetic algorithms.

The remainder of this paper is organized as follows: Section 2 describes the formalizing of a factor selection problem. Section 3 discusses the scheme of factors selection. Section 4 presents the compact genetic algorithm application for factors selection. Finally, section 5 reports the practical results of presented system.

2. The formalizing of a factor selection problem.

The problem of factors selection includes the choice of a subset of d size from set of attributes of a total D number on the basis on the given optimization criterion. We shall indicate general source set of the data (the maximum possible number of tags) as $U = \{1, 2, \ldots, D\}$.

On the other hand, a subset of the selected factors shall be indicated through $X = \{1, 2, \ldots, d\}$ while the set of remained (moved off) tags through Y . Thus, $U=X\cup Y$.

When information transformation is carried out with its subsequent reduction then we operate not U, but some range $f(U)$ that is functional transformation from U. Accordingly, X and Y are sets of selected and remote attributes of this transformed set. Thus, $X\cup Y=f(U)$.

For example, for the case of time series data can be represented both in time and in frequency domain. Operating with time area we reduce a set of factors, while with frequency area we can also remove harmonics with low energy. One to another area transition is provided with Fourier transformation here playing a role of $f(U)$. Besides, there is, for example, a factor analysis which also receives some set of the factors describing the given situation using functional data conversion of time series.

A criterion of quality estimation X shall be indicated as $J(X)$ allowing evaluation of both the accuracy of a certain qualifier on a certain set of the data ("wrapper" approach) and a universal statistic unit ("filtering" approach). Anyway, the choice of J depends on a specific object.

The task of procedure of factor selection is to find the set of X to satisfy the following condition:

$$\begin{cases} J(X) \to \max \textit{ with} \\ \quad |X| \to \min \end{cases}, \qquad (1)$$

where $|X|$ - quantity of the attributes contained in *X*.

Thus, we have a multi-criteria problem which extremum may not be determined beforehand being dependent of *J(X)* and a minimum threshold for $|X|$, set by the user.

3. Scheme of factors selection.

There is a set of problems for which the result is not an exact estimation having been obtained with some probability. One of such problems is forecasting using artificial intelligence methods. The forecast, for example, with neural networks is not determined but contains some uncertainty. Also, evolutionary algorithms often find suboptimum solution only thus stochasticity of an issued solution can be spoken about. For example, various program launchings (including training) can provide different results. With the problem of factors selection at forecasting its value is that of quality. Containing uncertainty it is available in quality estimation of a data set.

As mentioned above, in case of the result stochasticity received with an estimation of some subset of attributes to find an optimum set it is recommended to use the algorithms repeatedly estimating each of such set, in particular, genetic algorithms (Goldberg [9]). Accordingly, the scheme of factor selection looks like that represented in Figure 1.

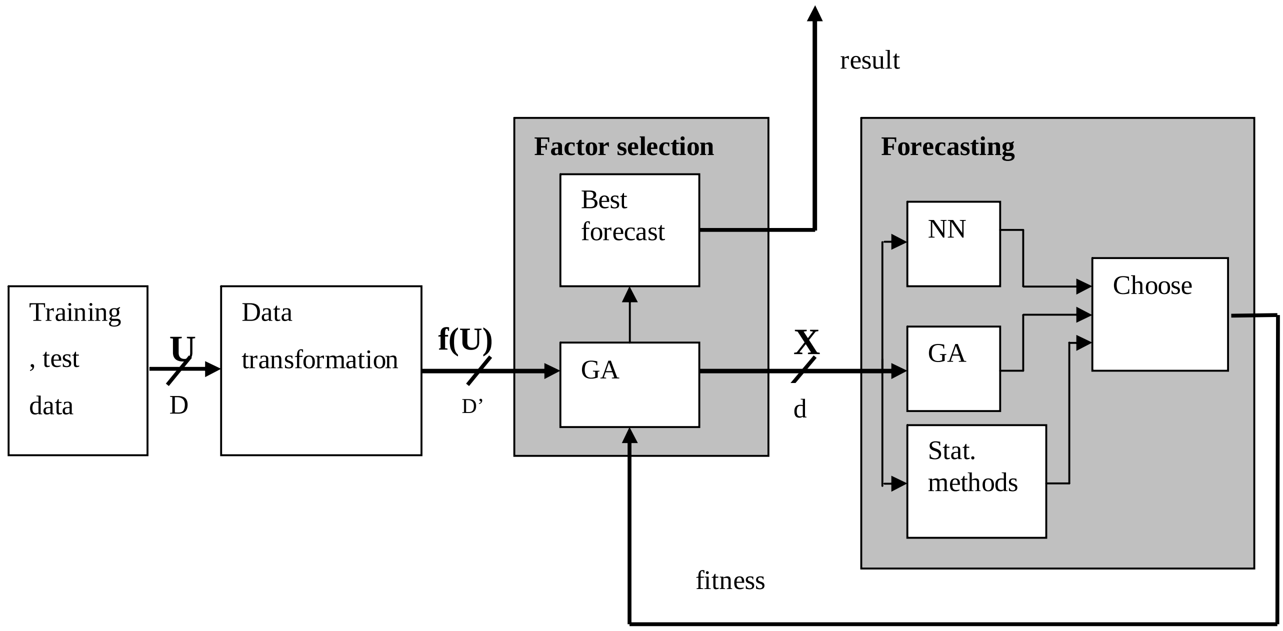

Fig.1. Scheme of Factors Selection.

In general terms the set of data U including training and testing data the total number of factors being *D* are subject to transformation (feature extraction task). The *f(U)* transformed set of data with *D`* number of factors shall be reduced using genetic algorithms. A reduced set of data *X* with $d = |X|$ is used for the forecast obtainments. So one can use any methods, both statistical and those of an artificial intelligence like neural networks or genetic algorithms. The choice of a method to be used can be done either on the ground of the obtained forecast optimality or by a user manually. One should notice however that obtaining a forecast by means of statistical methods its result be not a stochastic value thus the idea of the given scheme gets lost. The forecast value obtained is used as fitness - function for genetic algorithm of factors selection. The forecast best value as well as the set of factors corresponding to this forecast shall be reserved being the result of factors selection system.

4. Compact genetic algorithm application for factors selection.

In Fig.1 a genetic algorithm application for the factor selection is offered. In this case a definite kind of genetic algorithm should be selected considering a simplicity and a briefness of the algorithm as a fitness-function value shall be calculated for each chromosome in each epoch in any case.

In (Haric et al. [10]) a CGA (Compact Genetic Algorithm) is proposed with its application efficiency investigation for optimization purposes. One should note the given approach permits an extremely simple soft- and hardware embodiment the results being comparable with the classical genetic algorithm. Compact genetic algorithm scheme is represented in Fig. 2.

In CGA the population of binary individuals is substituted by a probabilities vector for each bit of chromosomes the probability of its zero (single) value is being presented. It allows the compact representation of binary chromosomes initial population. For example, see table1. The population (first four lines) can be represented in CGA by the following vector of probabilities (Pcga, the last line).

Table 1. Vector of probabilities in CGA

X1	X2	X3	X4	X5
1	0	1	1	0
1	0	0	1	0
1	0	1	1	0
0	0	0	1	1
0.75	**0**	**0.5**	**1**	**0.25**

The essence of compact genetic algorithm is that each epoch makes its own new micro-population. It is generated based upon the pointed vector of probabilities. Then the tournament selection is carried out where fitness-functions of all individuals of a micro-population are compared. The vector of probabilities is corrected after each comparison. The correction is done as follows: if alleles of a victor and a vanquished are of different values (0 and 1) the probability of the further generation of number positioned in the allele of the victor increases. In an original (Haric et al. [10]) the step of change equals to $1/n$, where n is the full size of algorithm population. After the tournament selection gets finished individuals are liquidated to be and formed again in the next epoch. Transition of all probabilities of a probability vector to stationary statuses (0 or 1) preconditions the algorithm work termination to be understood as complete determinacy of individuals' generation.

When we utilize genetic algorithms for factors selection an individual is represented by a classical characteristic vector (where 1 in $\boldsymbol{i}$ position corresponds to $\boldsymbol{i}$ factor entrance to in training data while 0 is its absence, the length of an individual is equal to the total number of all significant factors).

The fitness-function of forecasting problem is the forecasting error in the neural network for sampling composed of factors encoded by an individual. The process of the fitness-function obtainment is as follows: training and testing data is made including the factors encoded by the given individual. Further training data is used for neural network training. We present input parameters of test data after training with NN and get an error of forecasting as a result of comparison of the NN parameters obtained on an output and calibrated values of test data. This error is used as a fitness-function. A peculiarity of the given of CGA application is that the error of forecasting issued by the neural network is unstable and differs for same training data and test data at various sessions of training of the same NN. Thus, the fitness-function of an individual in CGA is a random quantity. Accordingly comparing two fitness-functions values one can speak about prevailing of one over another just with a certain probability. Calculating the individual fitness-function and consequently errors in forecasting this defect can be minimized in every epoch.

```
1.  Initialization of a probability vector
    for i:=1 to l do p[i]:=0.5;

2.  Generation of m individuals based on a probability vector
    for i:=1 to m do
    M[i]:=generate(p);

3.  Circular tournament execution
    for j:=1 to m-1 do
        for k:=j+1 to m do
        begin
          victor, vanquished:=evaluate(M[j],M[k])
4.              Updating a probability vector
                for l:=1 to length(M) do
                    if winner[l]<>loser[l] then
                    if winner[l]=1
                    then Pcga[l]:=Pcga[l]+chag
                    else Pcga[l]:=Pcga[l]-chag
                end
        end
    end

5.  Check a vector for convergence
    for i:=1 to l do
      if p[i]>0 and p[i]<1 then
       return to step 2;

6.  p represents final solution

CompactGA parameters:
n: population size
l: length of a chromosome
```

Fig. 2. The scheme of compact genetic algorithm.

In algorithm an *n* parameter is present as the scale of the initial population remained as “an inheritance” of standard genetic algorithms and not based on structure of CGA itself. The size of change step of probabilities vector is based upon this parameter. Generally the given parameter can be taken absolutely arbitrary.

Assume the value of the fitness-function is calculated correctly. We shall determine the step size for achieving an optimum in 1 epoch. Then with initial probability of each element of probabilities vector of *p[i]* = 0.5 and the size of micro-population m we have a set of equally probable states (0 or 1). The probability of choice of *k1* ciphers and (*k*-*k1*) unities from the set containing *m1* ciphers and (*m*-*m1*) unities is equal (Chernova [6]).

$$P = \frac{C_{m1}^{k1} * C_{m-m1}^{k-k1}}{C_m^k}. \quad (2)$$

Comparing two values the probability of comparison between different notions of 0 and 1 for the set size=*n* (only in this case the modification of probability factor is available) is equal provided there is an updating of a probability vector for CGA:

$$P_h = \frac{C_{n/2}^1 * C_{n/2}^1}{C_n^2}. \quad (3)$$

The total amount of C_n^2 comparisons is executed so to achieve an optimum for one CGA epoch the size of step should comprise

$$h = \frac{1}{C_n^2 * P_h} = \frac{1}{(C_{n/2}^1)^2}. \quad (4)$$

This statement is true for even number of chromosomes in population what is quite easily performed.

In case of incorrectness the fitness-function is described as a random number with mean value μ and deviation of σ. Then in tournament a comparison is executed of the mean values and random quantities $\mu 1$ and $\mu 2$. When $\mu 2 > \mu 1$, the probability of an error equals to (Chernova [6]):

$$P_{err} = P(\mu_1 - \mu_2 < 0) = 1 - \Phi\left(\frac{\mu_1 - \mu_2}{\sqrt{\sigma_1^2 + \sigma_2^2}}\right), \quad (5)$$

where $\Phi(z) = \frac{1}{2\pi} \int_0^z e^{-\frac{z^2}{2}} dz$ – Laplace cumulative distribution function.

Thus the size of step shall be corrected by the probability of wrong comparison of error. The size of the step is to be multiplied by *P*=1-2**Perr* coefficient.

Then the next step change formula of vector of probabilities in CGA looks as follows:

$$h_m = h * (1 - 2 * P(\mu_1 - \mu_2 < 0)). \quad (6)$$

In the process of algorithm work the number of one and zero is changed being determined by Pcga probability vector. An average number of unities in chromosome shall equal to

$$SumPcga = \sum_{i=1}^{length(P_{CGA})=l} P_{CGA}[i]. \quad (7)$$

The probability of various values' comparison is reduced:

$$P_h = \frac{C^1_{SumPcga} * C^1_{l-SumPcga}}{C^2_l}. \quad (8)$$

Accordingly, the step of probability vector alteration must increase:

$$h = \frac{1}{C^2_n * P_h} = \frac{C^2_l}{C^2_n * C^1_{SumPcga} * C^1_{l-SumPcga}} = \frac{C^2_l}{C^2_n * SumPcga * (l - SumPcga)}. \quad (9)$$

The final modified step looks as follows:

$$h_m = \frac{C^2_l}{C^2_n * SumPcga * (l - SumPcga)} * (1 - 2 * P(\mu_1 - \mu_2 < 0)). \quad (10)$$

Thus the algorithm is developed as for optimum attributes subset finding for the case of stochastic quality estimation of this subset. As the quality estimation unit the value of neural network output is taken used in time series forecasting. CGA (Compact Genetic Algorithms) are used as an adjustment algorithm of attributes subset. The change step for CGA vector of probabilities is modified considering estimation stochasticity.

5. Approbation and practical results

Approbation of CGA modified step results was done on benchmarks from Proben1 [20], UCI Library [22], and on real forecasting data flows of in- and outcoming channels of Internet Service Provider, ISP).

5.1. Testing with standard Benchmarks

In each testing task the data distribution was reduced to an even form thus improving the resulting accuracy. Classification results were obtained by NN on usual non transformed data (*Untransform*), on data transformed to uniform distribution (*Uniform*), on data with the reduced classic CGA factors set (*CGA*) as well as reduced CGA with modified CGA step (*ModCGA*). For the set of Glass 1 (Proben1, [20]) NN was trained on training data, test and validity data, as for the rest sets (Hepatitis and Ionosphere, UCI, [22]) 3-fold cross-validation was performed: three sets of training data were created occupying the one–third of the sampling and being taken from the beginning, the middle and the end of data file. MSE results were obtained on training data (*Train Error*) and test data (*Test Error*) together with classification errors on train data (Train Class Err). For training data not an error of classification was determined but classification accuracy (Test Class) due to the fact that in all sources such data were offered for those sets. A separate CGA was introduced for each of the sets its results being tested on all versions of the set (three for 3-fold cross validation). To prove the quality of factors' set NN was trained 60 times for each set, for each class separately (for example, for Glass 1 total 60*6=360 trainings of NN). The result was mean value and standard deviation of classification accuracy (*Test Class*) of all experiments for one set. Besides, for CGA the number of epochs was introduces for algorithm execution (*Ep*) and factors number in the set (*Fact*). The fitness function of CGA was classification accuracy on training data. Those components responsible for the set briefness were not utilized. The results obtained were compared with those by NN on (*PCA+NN*) sampling converted by means of principal component analysis as well as with those represented in Kwedlo, Kretowsky [16], Oh [19], Yang, Honavar [25]. In Oh [19], Yang, Honavar [25] the combination of GA + NN was also used, while in Kwedlo, Kretowsky [16] - C4.5 and EDRL.

Table 2. Results on Glass1 benchmark.

Glass1 9+6, 214rec	Train error	Test error	Train Class Err	Test Class	Ep	Fact
Untransform	2.83 (2.51)	12.26 (10.31)	2.88 (3.2)	**88.13 (11.15)**		
Uniform	4.59 (4.51)	11.01 (9.4)	5.50 (6.17)	**88.6 (10.69)**		
CGA	5.7 (4.13)	10.74 (10.47)	8.16 (6.11)	**90.32 (8.29)**	79	3
CGA Valid	5.71 (4.16)	10.59 (11.01)	8.15 (6.17)	**90.21 (8.13)**	79	3
ModCGA	4.22 (3.79)	10.04 (7.85)	5.39 (5.14)	**90.30 (7.35)**	78	4
ModCGA Valid	4.20 (3.75)	10.39 (9.35)	5.36 (5.05)	**90.13 (7.89)**	78	4
PCA+NN	0.77 (0.65)	17.47 (10.12)	0.34 (0.61)	**84.78 (10.8)**		6
Proben1 linear	8.83 (0.01)	9.98 (0.1)		**53.96 (2.21)**		
Proben1 mult 8+0+l			9.184	**67.92**		
Proben1 pivot	7.68 (0.79)		9.75 (0.41)	**61.97 (8.14)**		
in Yang, Honavar (1998)				**70.5 (8.5)**		
in Yang, Honavar GA. (1998)				**80.8 (5.0)**		
in Oh (2004)				**100**		
in Kwedlo, Kretowsky (1998) C4.5				**67.5 (0.8)**		
in Kwedlo, Kretowsky (1998) EDRL				**66.7 (1.0)**		

Table 3. Results on Ionosphere benchmark.

Ionosphere 34+2, 351rec	Train error	Dev	Test error	Dev	Train Class Err	Dev	Test Class	Dev	Ep	Fact
Uniform	0.52 (0.38)	0.38	3.84	1.51	0.00	0.04	97.89	2.28		
CGA	0.30	1.09	1.40	3.17	0.35	1.52	99.39	2.49	168 (49.38)	8.66 (6.26)
ModCGA	0.46	0.89	2.54	3.66	0.00	0.20	98.82	3.22	70.93 (51.99)	8.73 (3.96)
PCA+NN	0.37	0.12	8.08	2.72	0.03	0.11	94.43	1.6		34
in Yang, Honavar (1998)							94.30	5.00		
in Yang, Honavar (1998)							98.60	2.40		
in Oh (2004)							91.45			

Table 4. Results on Hepatitis benchmark.

Hepatitis 19+1, 155rec	Train error	Dev	Test error	Dev	Train Class Err	Dev	Test Class	Dev	Ep	Fact
Uniform	1.21	2.32	20.00	8.57	0.31	1.37	79.96	7.50		
CGA	4.50	1.56	16.62	13.86	4.90	3.89	82.53	9.83	160.67 (61.12)	8.67 (1.32)
ModCGA	3.49	12.14	18.27	23.07	3.40	10.02	82.04	12.30	15.33 (6.08)	10.50 (1.64)
PCA+NN	0.03	0.07	40.94	12.56	0.02	0.12	64.09	6.15		19
in Yang, Honavar (1998)							84.70	9.50		
in Yang, Honavar (1998)							97.1	4.30		
in Kwedlo, Kretowsky (1998) C4.5							79.6	0.60		
in Kwedlo, Kretowsky (1998) EDRL							81.2	1.80		

As we can see, the results obtained by CGA+NN combination with data reduced to the uniform condition actually exceed all results of similar works. A modified step introduction practically does not reduce the accuracy while the time for search is often reduced by an order, up to 90.6 percent.

5.2. ISP data forecasting task description

Every provider has incoming and outgoing channels of data communications. Depending on the topology of commutation of data flows on the ISP equipment the same flow can be incoming for one site and outgoing for another one. Besides, it can multiplex that is to become a part of a larger data flow including some logical flows in a physical one.

Fig.3 shows dependences of two resulting flows data which are the main numeral representation of provider channels loading. The In channel (general input flow shown green) and Out channel (general output stream shown violet) have already included all incoming and outgoing data flows of ISP considering their multiplexing and unite some logical flows into a physical one. This information as well as some other data is stored in ISP database.

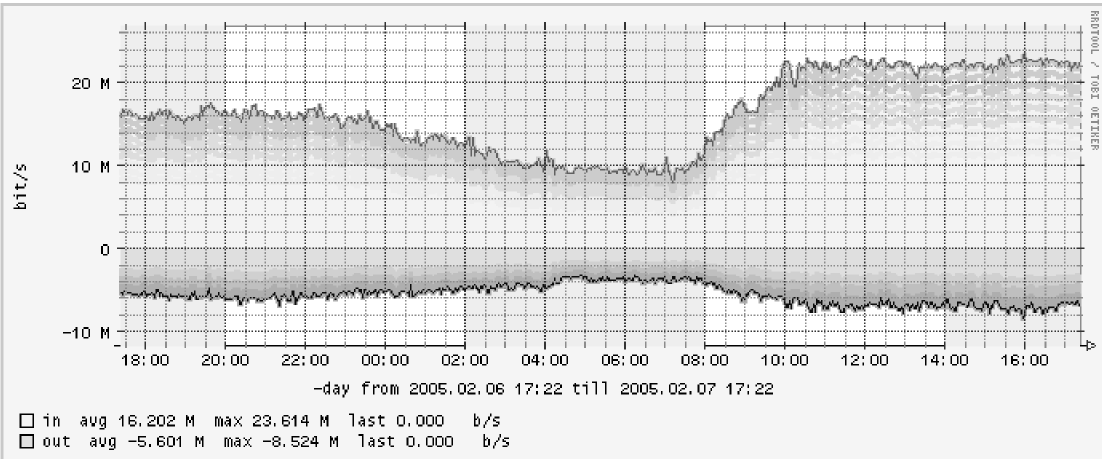

Figure 3. General ISP input and output channels

From time to time during ISP functions the so-called critical situations occur. They are the spontaneous jumps and falls of information volume passing through In or Out channels occurring both due to ISP and by the reasons regardless of a provider. To liquidate such situations an intervention by maintenance staff is required. ISP must detect such situations in proper time as well as to predict the channels' loading like the volume of information passing through channels per time unit in subsequent time intervals.

The more substantial task description and affecting factors selection is presented in the work (Khmilovyy [15]). On the ground of task analysis the surplus row of influencing factors is selected. Superfluity does confirm the fact some affecting factors worsen the general forecasting accuracy. The further optimum reduction of the affecting range is one of the tasks of this work.

5.3. Selection of minimum significant factors number

Comparison of forecasting accuracy results is done issued by a neural network on the full redundant number of factors with a prognosis on the great number of factors with the reduced length of a time range up to 1 lag (see Baestens et al. [2]) as well as on the factors set reduced using correlation analysis, standard CGA with n=50 and CGA with the modified step.

With the simple length reduction of time range 4 significant factors of 20 were removed. The result of forecasting got improved by 5% compared with a complete set of factors.

Ten factors were used for correlation analysis in forecasting. The result of forecasting got improved by 11% compared with the result on a complete training data set. Correlation analysis showed as follows:

- Great interconnection between many variables. For each input factor 2-3 factors exist, the correlation coefficient with them not exceeding 0.5.
- Small influence of factors on the predictable values. For 40% of affecting factors the coefficient of correlation with predictable values did not exceed 0.05.
- Different connection strength of different affecting factors with the different predictable values. Thus one can state that three values under forecasting three different sets of affecting factors are required to provide an optimum prognosis development.

Standard CGA utilization with the population size of n=50 for affecting factors set minimizing has decreased the forecasting error by 16% compared with a result on a complete teaching set. After CGA step modification a search time of decision got reduced by 51%. The forecasting error compared with the results represented by standard CGA did not change seriously. The results obtained are presented in Table 5.

Table 5. Investigations results on Internet traffic Data

Compared to: Modification	Complete set of factors	CGA with n=50
Small series length	Error reduction by 5%	
Correlation analysis	Error reduction by 11%	
CGA c n=50	Error reduction by 16%	
CGA with modified step	Error reduction by 16%	Calculation time decrease by 51%
Classification and forecasting tasks separation	Error reduction by 4%	

5.4. Classification and forecasting tasks separation

Two tasks were initially considered: that of critical situations classifications and time series forecasting. As soon as it was proved that different sets of factors are important for those tasks an attempt was made to separate those tasks through two-step data processing. Thus on the first stage critical situations are determined and deleted while on the second stage a time series forecasting is made. As a result of two-step forecasting the forecasting error got reduced by 4% as compared to an initial one. Thus the importance of separate tasks fulfillment is demonstrated as for specific situations classification and time series forecasting with the individual search of optimum affecting factors set for every task.

6. Conclusions

1. The conditions are determined in which classic methods of factors selection are insufficient and require modifications the stochasticity of estimation of subset factors being one of them. The optimum group of methods is chosen which are maximally suitable for work in such conditions. They are characterized by the repeated procedure of factors estimation. Genetic algorithms are mostly suitable for this purpose.
2. The conditions are stated when the got estimation of attributes subset is stochastic. Such estimation is possible for the output of neural network or genetic algorithm utilized for time series forecasting
3. The scheme of factors selection is proposed. Its specific feature is that some data extraction is possible beforehand, for example, based on a factor analysis or Fourier transformation followed by the direct factors selection using a genetic algorithm the forecasting being developed on the ground of the resultant reduced factors set. The forecasting obtained is used as fitness-function for the genetic algorithm of factor selection.
4. The type of genetic algorithm is chosen as most suitable for such class of tasks fulfillment. These are compact genetic algorithms. It is characterized by an outstanding simplicity of program realization together with considerable efficiency of work with such type of tasks.
5. For the condition of stochastic estimation the step of probability vector change for CGA got modified. Modification is based on determination of size dependence of probability vector change of the authenticity degree of two values comparison and namely estimation units of attributes subsets.

7. References

1. Ahmad Amir, Dey Lipika, 2005. A feature selection technique for classificatory analysis. Sciencedirect. Pattern Recognition Letters, 26, pp 43-56.
2. Dirk-Emma Baestaens, Douglas Wood, Max Van De Bergh 1994. Neural Network Solutions For Trading In Financial Markets. Financi Times/ Prentice Hall.
3. Bala, J., K. DeJong, J. Huang, H. Vafaie, and H. Wechsler, 1995. Hybrid Learning Using Genetic Algorithms and Decision Trees for Pattern Classification, 14th Int. Joint Conf. on Artificial Intelligence (IJCAI), Montreal, Canada.
4. Bala J, De Jong K, Huang J, Vafaie H, and Wechsler H., 1997. Using learning to facilitate the evolution of features for recognizing visual concepts. Evolutionary Computation 4(3) - Special Issue on Evolution, Learning, and Instinct: 100 years of the Baldwin Effect.
5. Blum, L.A., Langley, P., 1997. Selection of relevant features and examples in machine learning. Artifical Intelligence; 97; pp.245-271.
6. Chernova N.I., 1999. Theory of chances. http://text.marsu.ru/books_edu/11/lec.html
7. Ezhov A.A., Shumskiy S.A., 1998. Neurocomputing and its using in economy and business. Moskow.
8. Freitas Alex, 2002. A survey of evolutionary algorithms for data mining and knowledge discovery. Springer-Verlag. In A. Ghosh and S. Tsutsui *editors*, Advances in Evolutionary Computation, chapter 33, pages 819-845.
9. Goldberg D.E., 1989. Genetic Algorithms in search, optimization and machine learning; Addison-Wesley.
10. Georges R. Harik, Fernando G. Lobo, David E. Goldberg, 1998. The compact Genetic Algorithm //http://citeseer.ist.psu.edu/harik98compact.html

11. Gruau F., and D. Whitley, 1993. Adding Learning to the Cellular Development of Neural Networks: Evolution and the Baldwin Effect, Evolutionary Computation, Vol.1, No.3, pp. 213-234.
12. Guerra-Salcedo C. and Whitley D., 1999. Feature selection mechanisms for ensemble creation: a genetic search perspective. Data Mining with Evolutionary Algorithms: Research Directions, pp. 13-17, AAAI Press, 18 July.
13. Hattingh, J.M. & Kruger, H.A. 2001. Robust linear models by discarding data and regressors simultaneously, Research Report No FABWI-N-RKW:2001-56.
14. Hsu William H. and Pottenger William M., Weige Michael, Wu Jie. and Yang Ting-Hao, 1999. Genetic algorithms for selection and partitioning of attributes in large-scale data mining problems. Data Mining with Evolutionary Algorithms: Research Directions, pp. 1-6, AAAI Press, 18 July.
15. Khmilovyy S.V., 2005. Internet-traffic forecasting with neural approach. Ukraine: Donetsk; Naukovi Pratsi DONNTU; DONNTU Publishing House.
16. Kwedlo Wojciech and Kretowski Marek, 1998. Discowery of Decision Rules from Databases: An Evolutionary Approach. Proc. 2nd European Symp. on principles of Data Myning and Knowledge Discovery (PKDD-98). Lecture Notes in Artificial Intelligence 1510, 371-378. Springer-Verlag.
17. Minaei-Bidgoli Behrouz, Punch William F. III, 2003. Using Genetic Algorithms for Data Mining Optimization in an Educational Web-based System. GECCO 2003 Genetic and Evolutionary Computation Conference, Springer-Verlag 2252-2263, Chicago, IL.
18. Moerchen Fabian, 2003. Time Series Feature extraction for data mining using DWT and DFT. In Technical Report No. 33, Departement of Mathematics and Computer Science Philipps-University Marburg.
19. Oh II-Seek, Lee Jin-Seon, And Moon Byung-Ro, 2004. Hybrid Genetic Algorithms for Feature Selection. IEEE Transactions on pattern analysis and machine intelligence, vol 26, no.11.
20. PROBEN1 - A Set of Neural Network Benchmark Problems and Benchmarking Rules, 1994. Lutz Prechelt, Fakultaet fuer Informatic, Universitaet Karlsruhe.– Karlsruhe, Germany. ftp.ira.uka.de/pub/neuron.
21. Raimer Michael L., Punch William F., Goodman Erik D., Kuhn Leslie A. and Jain Anil K., 2000. Dimensionality Reduction Using Genetic Algorithms. IEEE Transactions on Evolutionary Computation, ISSN 1089-778X CODEN ITEVF5 , vol. 4, no 2, pp. 164-171 (42 ref.)
22. UCI Machine Learning Data: - University of California, Irvine . – http://mlearn.ics.uci.edu/databases
23. Vafaie, H., and De Jong, K., 1992. Genetic Algorithms as a Tool for Feature Selection in Machine Learning. Proceeding of the 4th International Conference on Tools with Artificial Intelligence, Arlington, VA, pp.200-204. http://citeseer.ist.psu.edu/vafaie92genetic.html
24. Vafaie H and De Jong K., 1993. Robust feature selection algorithms. Proc. 1993. IEEE Int. Conf on Tools with AI, 356-363. Boston, Mass., USA. Nov.
25. Yang J.H., Honawar V., 1998. Feature Subset Selection Using a Genetic Algorithms. IEEE Intelligent Systems, vol. 13, no. 2, pp. 44-49.