International Journal of Engineering Research and Technology. ISSN 0974-3154, Volume 12, Number 7 (2019), pp. 1075-1080

A Tool for Predicting the Decline in a Supply Chain during the Life Cycle of a Medicament

Holman Montiel Ariza[1], Fernando Martínez Santa[2] and Fredy H. Martínez S.[3]

[1] *Associate Professor, Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia.*
[1]*ORCID:* 0000-0002-6077-3510

[2,3]*Assistants Professor, Facultad Tecnológica, Universidad Distrital Francisco José de Caldas, Bogotá D.C, Colombia.*
[2]*ORCID:* 0000-0003-2895-3084 [3]*ORCID:* 0000-0002-7258-3909

Abstract

The decline is used to estimate the losses that suppliers may have in different supply chains during the life cycle of a product. In the Colombian pharmaceutical industry, predicting the level of loss during the life cycle of a drug is a complex task, because information during the sales and invoicing process in the different supply chains is managed in decentralized databases. Therefore, the different actors involved during the life cycle of a product cannot properly estimate the profits and losses of certain products. This article proposes a partial solution to this problem through the development of a tool based on support vector machines.

Keywords: Support vector machines, stochastic methods, deterministic, decline, Markov chain.

1. INTRODUCTION

The health sector in Colombia is regulated by various actors, among which are: current regulations, providers, IPS (Institutes that Provide Services), intermediaries (e.g., pharmacies or health care providers), and clients. Although these actors are articulated through a functional structure, this articulated structure or supply chain does not allow the available financial resources to be executed in an appropriate manner [1-4].

Incorrect execution of financial resources does not allow IPS to timely provide intermediaries with inputs or financial resources for their operation. Because these financial resources are not available in certain periods of time, the cash flow of the intermediaries is not constant or high enough to pay salaries, buy inputs or maintain their infrastructure. Although there have been attempts to improve the supply chain process by merging IPS, these attempts do not yet show a significant improvement over the conventional scheme [1-4].

Supply chain problems have been widely studied using approximate mathematical models that attempt to reproduce the movement of resources and utilities over the life of a product. These models are based on flow diagrams, block diagrams or graphs, as these diagrams represent the process in general and allow an abstraction of its structure quickly. Similarly, non-deterministic models roughly solve the problem by estimating the cost of transactions over the useful life of a product at certain time intervals [5, 6].

Non-deterministic models are based on stochastic methods that attempt to optimize the function used to represent the process or product. These stochastic methods or processes are used to characterize random variables that evolve as a function of a mathematical expression or a variable. Normally, these methods show the progress of the process over time through sets of discrete variables, that is, this type of technique does not allow knowing the state of the product in any instant of time, but for which the designer has defined the model [6-9].

An example of stochastic methods are the Markov chains that are used to model the evolution of a variable from the immediately previous event, in other words, it is a cumulative multiplication whose actual result depends on the previous value obtained from the operation. In addition, changes in outcome depend on the probability assigned to each event that makes up the process [10].

As can be seen, there are several techniques for estimating the behavior of the different actors involved in a process and their influence during the life cycle of a product [6-10]. However, the proposed models depend on information that is sometimes unavailable or difficult to access, which increases the margin of error between the expected results and those obtained. Therefore, this article proposes an automated way to know the behavior of the different actors involved in the supply chain of a medicine in Colombia. This technique is described in detail in the following sections, which are organized as follows; sections 2 and 3 present the general definitions and methodology used respectively, and section 4 presents the results obtained.

2. MATERIALS AND METHODS

The algorithm developed is based on a support vector machine, which was implemented entirely in MATLAB. The training of the algorithm was carried out with a group of probabilities belonging to an IPS, where the life cycle of the drug ADEFOVIR is monitored. This is followed by a detailed description of the proposed model and the definitions needed to understand how it works.

2.1. Markov Chain

A Markov chain is a discrete, non-deterministic method that estimates the state of a process over time, based on the probability assigned to each of its component events. This set of probabilities in the current state makes it possible to estimate the next state, i.e. Markov is an iterative process that by means of a cumulative multiplication multiplies to the initial state by itself up to the given number of iterations [10].

A Markov process depends on a network and a transition matrix. On the one hand, the network is used to raise the probability that an event will occur or that one of the components of the network will pass from one state to another. On the other hand, the transition matrix is constructed from the resulting network and is used as the time parameter of the model [12, 13].

In this case, a graph representing the interactions between the various actors involved in the supply chain of a drug (ADEFOVIR) from an IPS to the client is considered. Each of the actors is assigned a probability, which indicates the possibility of the drug circulating between one actor and another, this circulation is bi-directional in all cases except in the case of clients. In other words, there is the possibility that one actor may return the medicine to another, but the client is the only one who does not return it, since it is the consumer (see Fig. 1) [13].

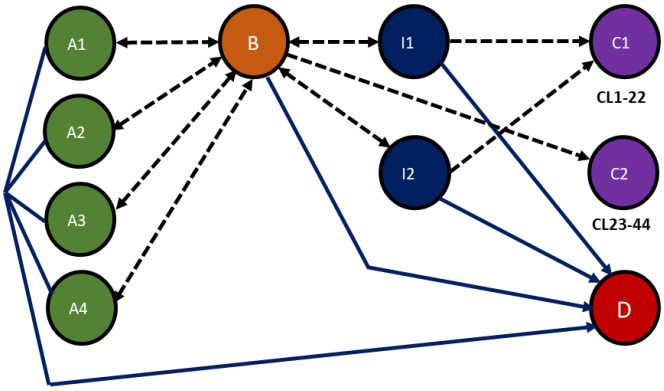


Fig. 1. Markov chain representing the supply chain of a medicine (Based on [1]).

The letter A represents suppliers (four in total), the letter B represents IPS, the letter I represents intermediaries, the letter C represents customers and the letter D represents waste. In addition, the arrows connected to each state (circle) are the probability distribution that indicate the strength of the relationship between one actor and another. The sum of probabilities in each state should equal one and are presented in more detail in Table 1.

Table 1 shows the "Actor-Actor" scheme, i.e. the name of the cell is indicated at the top and refers to the probability assigned to the state connection. For example, "P1-IPS" marked in red indicates the probability that the product will go from supplier one (1) to IPS. These probabilities are arranged in matrix form to estimate the temporal result, where the zeros indicate that there is no kind of relationship between the actors. As shown in the segment in Table 2.

Table 1. Probabilities of the Markov chain (Based on [1]).

P1-IPS	P1-Waste	IPS-P1	P2-IPS	P2-Waste	IPS-P2	P3-IPS	P3-Waste	IPS-P3
0.951	0.049	0.0273	0.965	0.035	0.0299	0.955	0.045	0.0295
P4-IPS	**P4-Waste**	**IPS-P4**	**IPS-I1**	**IPS-I2**	**I1-IPS**	**I2-IPS**	**I2-CL4**	**I2-CL5**
0.98	0.02	0.03	0.0999	0.1501	0.015	0.02	0.05	0.05
I2-CL6	**I2-CL7**	**I2-CL8**	**I2-CL9**	**I2-CL10**	**I2-CL11**	**I2-CL12**	**I2-CL13**	**I2-CL14**
0.04	0.03	0.05	0.05	0.05	0.06	0.03	0.03	0.03
I2-CL15	**IPS-CL21**	**IPS-CL22**	**IPS-CL23**	**IPS-CL24**	**IPS-CL25**	**IPS-CL26**	**IPS-CL27**	**IPS-CL28**
0.04	0.019	0.0211	0.0189	0.0178	0.0222	0.0212	0.0188	0.0214
IPS-CL29	**IPS-CL30**	**I1-CL1**	**I2-CL17**	**I2-CL18**	**I2-CL19**	**I1-Waste**	**I2-CL2**	**I2-CL3**
0.0199	0.0255	0.935	0.05	0.04	0.08	0.05	0.06	0.15
IPS-CL38	**IPS-CL19**	**IPS-CL20**	**I2-CL16**	**IPS-CL32**	**IPS-CL33**	**IPS-CL34**	**IPS-CL35**	**IPS-CL36**
0.02	0.1	0.021	0.05	0.0205	0.0195	0.0193	0.0207	0.02
IPS-CL37	**IPS-Waste**	**IPS-CL31**	**IPS-CL39**	**IPS-CL40**	**IPS-CL41**	**IPS-CL42**	**IPS-CL43**	**IPS-CL44**
0.02	0.032	0.0145	0.02	0.02	0.02	0.02	0.02	0.02

Table 2. Markov Transition Matrix Segment (Based on [1]).

	P1	P2	P3	P4	IPS	I1	I2	CL1
P1	0.0000	0.0000	0.0000	0.0000	0.9510	0.0000	0.0000	0.0000
P2	0.0000	0.0000	0.0000	0.0000	0.9650	0.0000	0.0000	0.0000
P3	0.0000	0.0000	0.0000	0.0000	0.9550	0.0000	0.0000	0.0000
P4	0.0000	0.0000	0.0000	0.0000	0.9800	0.0000	0.0000	0.0000
IPS	0.0273	0.0299	0.0295	0.0300	0.0000	0.0999	0.1501	0.0000
I1	0.0000	0.0000	0.0000	0.0000	0.0150	0.0000	0.0000	0.9350
I2	0.0000	0.0000	0.0000	0.0000	0.0200	0.0000	0.0000	0.0000
CL1	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	1.0000

Finally, in order to know the result in a time interval (months), the transition matrix is multiplied in a cumulative way by how many months it has been indicated (Eq. 1).

If A=Translation matrix at time zero and C[0]=A, then

$$C[i] = \sum_{i=1}^{\text{Months}} A * C[i-1] \qquad (1)$$

2.2. Support Vector Machine

Support Vector Machines (SVMs) are a group of supervised learning algorithms based on regression and classification techniques. These techniques assign a label to each class present in the group of training samples in order to construct an approximate model that allows predicting a new sample [14].

An SVMs predicts a new sample by modeling the set of samples into two different categories and the model created indicates whether this sample belongs to one category or another. These categories are generated by finding a hyperplane that separates points between one class and another, this hyperplane is generated by projecting the domain of each point in a space of superior dimensionality. The main characteristic of this hyperplane is that it must cross as close as possible to the projections made by each point of the sample group (see Fig. 2) [15].

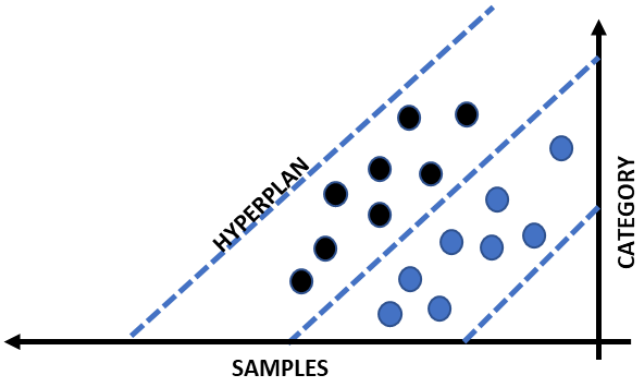


Fig. 2. Operation of a two-dimensional SVM.

Learning models based on SVMs are very similar to neural networks, since SVMs can be built with training methods like those implemented in polynomial neural networks, radially based or perceivers [15]. The features used to implement the support vector machine and predict the Markov transaction matrices are described in detail in the following section.

3. IMPLEMENTATION

As mentioned, the technique developed is based on Markov transition matrices to predict the loss of a medically during its circulation between the different actors in the supply chain (see Fig. 1). The matrix of probabilities in the initial time or time zero is a set of values proposed by Cardona [1] and by means of the Markov chains its temporal evolution is estimated, whose structure is a cumulative multiplication as shown in Eq. 1. This structure is shown in detail in Algorithm 1 and was used to estimate the SVM training matrices.

Algorithm 1. Calculation of time matrices

```
Function Markov Chains ()
Start vector Samples.
Start Label vector.
A ← Initial probabilities.
Month ← Assign number of months.
   For i=1 to length (Months)
      Transition matrix=A* Transition matrix
      Samples[i]= Transition Matrix
      Labels [i]= i
   End for
   Return (Samples, Labels)
```

Each of the matrices is assigned a label indicating the month in which it was estimated using the Markov chain. These data are then normalized so that the matrix values take a range between zero (0) and one (1), to become the training data. Training data are divided into two groups; the first group are predictors that are probability nuances and the second are expected responses that correspond to the label assigned to each matrix. Once the label is assigned to the training data, the SVM model is estimated and its performance evaluated with a cross-correlation validation with 10% of the training data (see Fig. 3).

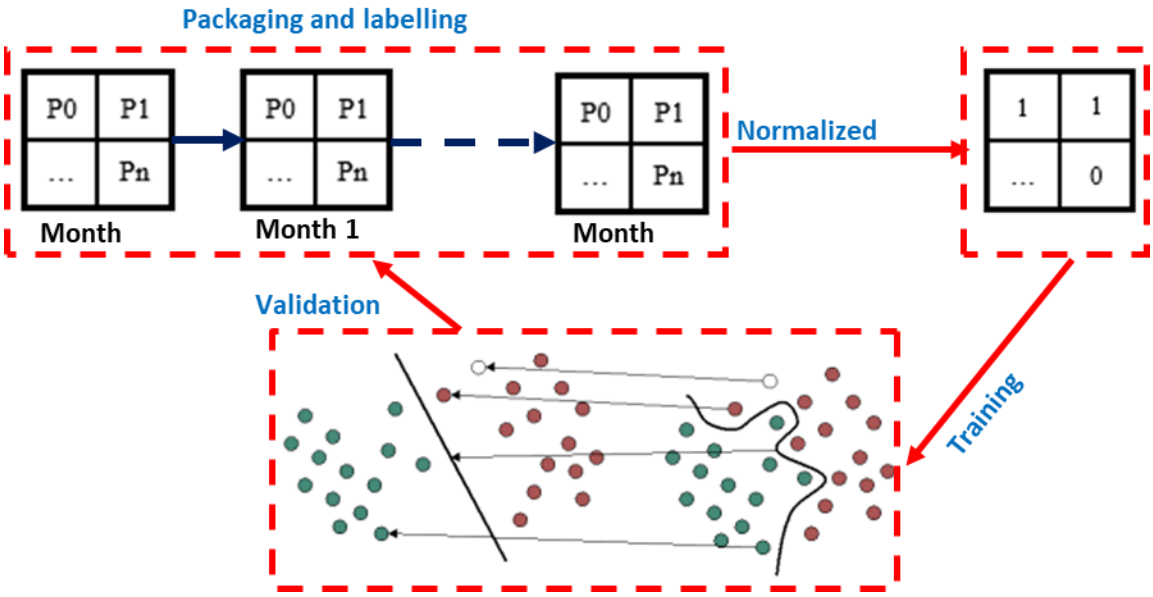


Fig. 3. SVM training scheme.

At the end of the SVM training, a cluster is exported that stores the coefficients and other characteristics of the model found. The model is a mathematical abstraction characterized by a function of at least 100 dimensions, since, the training arrangements are matrices of 51X51X100, where 100 is the number of months given to the function that estimates the Markov chain. This mathematical function was used to carry out an application, which allows to estimate the decrease by means of the Markov chain and the SVM, in order to compare the results estimated by both strategies. The result in each case is shrinkage, which is estimated by multiplying a value of units (drug value) by the probability assigned to each provider. Once this multiplication has been carried out, the columns associated with each actor are extracted from the resulting matrix, since the cumulative multiplication propagates the value of the decrease through the time matrix and thus the value corresponding to each actor is automatically estimated. Finally, the results were compared by calculating the margin of error between the estimated loss using the Markov chain and the SVM. This application is described in detail in Algorithm 2.

Algorithm 2. Application developed

```
A ← Initial probabilities.
Month ← Assign number of months.
   For i=1 to length (Months)
      Transition matrix=A* Transition matrix
   End for
Transition Matrix SVM=SVM(Months)
M1= Extract merma (Transition Matrix)
M2= Extract merma (SVM Transition Matrix)
C=error (M1, M2)
Publish (Correlation)
```

4. RESULTS AND DISCUSSION

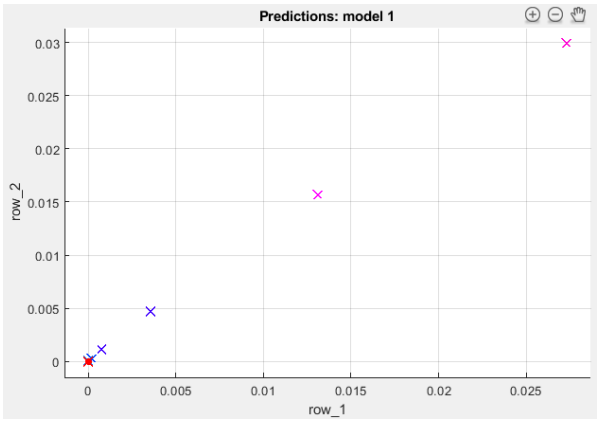
This technique was implemented in a computer with 250GB of hard disk, 8 GB of RAM memory and a Core I3 n Series processor. The SVM training took about fifteen (15) minutes, using the *MATLAB SVM CLASSIFICATION LEARNER TOOLBOX* and the CUBIC SVM classifier. The input data was one hundred (100) 52X52 matrices whose labels are the month for which they were estimated. These matrices are arranged as shown in Table 2 and the resulting model allows you to enter the month as input to predict the decline of each actor.

SVM performance was initially measured by reviewing the level of accuracy achieved (86.5%) when comparing training and validation data. Other features related to the performance of the SVM became visible at the end of its training, such as the scatter plot (Fig. 4a), the ROC curve (Fig. 4b, receiver operating characteristic curve) and the parallel coordinate plot (Fig. 4c). In this case the scatter plot shows the closeness between the categories found and the hyperplane. The ROC curve shows the rate of false positives and negatives and the parallel coordinate graph presents the expected behavior of each actor with a trend graph.

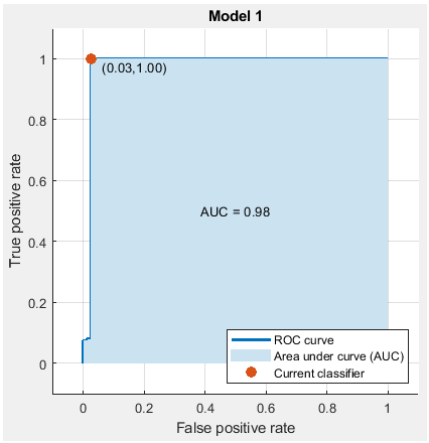
Finally, the estimated loss for each actor was used as a parameter to validate the performance of the SVM, by estimating the margin of error between the data provided by the Markov chain and the SVM in different months of the year (Table 3, considering that an average of the loss was taken from clients 2-44).

Table 3. Estimated error margin in different input configurations (P1 = 100 Units, P2=200 Units, P3=100 Units, P4=200 Units).

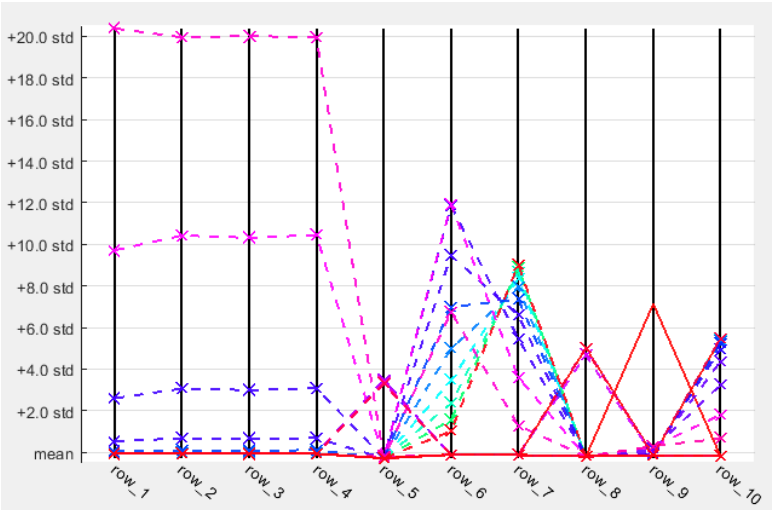
	P1	P2	P3	P4	I1	I2	CL1	CL 2-22	CL 23-44
Months = 2									
MARKOV	4.9	7	4.5	4	2.89	3.47	54.13	18.4	11.44
SVM	5	6.6	4.7	4.2	2.7	3.3	55	20.2	12
ERROR	0.1	-0.4	0.2	0.2	-0.19	-0.17	0.87	1.8	0.56
Months = 4									
MARKOV	0.2648	0.26	0.25	0.833	0.29	5.91	104.44	10.74	4.2
SVM	0.28	0.3	0.2	0.7	0.3	6	105	10	4
ERROR	0.0152	0.04	-0.05	-0.133	0.01	0.09	0.56	-0.74	-0.2
Months = 6									
MARKOV	0.137	0.095	0.125	0.0017	0.160	4.08	104.60	3.43	0.83
SVM	0.14	0.1	0.1	0.002	0.2	4.06	105	3	0.86
ERROR	0.003	0.005	-0.025	0.0003	0.04	-0.02	0.4	-0.43	0.03
Months = 8									
MARKOV	7*10^-4	3*10^-4	4*10^-4	3*10^-4	6*10^-4	1.72	83.53	0.87	0.13
SVM	7*10^-4	2*10^-4	2*10^-4	3*10^-4	6*10^-4	1.7	85	0.5	2
ERROR	0	1*10^-4	2*10^-4	0	0	0.02	0.53	-0.13	1.87
Months = 10									
MARKOV	3*10^-5	1*10^-5	2*10^-4	7*10^-5	2*10^-6	0.58	61.96	0.20	0.01
SVM	2*10^-5	1*10^-5	2*10^-4	7*10^-5	2*10^-6	0.6	62	0.50	0.01
ERROR	1*10^-5	0	0	0	0	-0.02	-0.04	-0.30	0



a) Scatter plot



b) ROC



c) Parallel coordinates

Fig. 4. Behavior of the SVM.

5. CONCLUSIONS

As the graphs in Fig. 4 were presented, it is possible to create a prediction strategy for values contained in matrices up to 52X52 based on SVM. The advantage of using this type of strategy is that it has a relatively short training time compared to other strategies such as neural networks or conventional optimization algorithms, which may require entire hours of training when trying to solve this type of problems. Another advantage of the model is the reduction of computational resources, which is observed when estimating the value of the decrease in months. Because, when implementing the Markov chain, the application takes about 10 seconds to estimate the parameters of the time matrices and when implementing SVM, the application takes 5 seconds.

As can be seen from the values in Table 3, the trained SVM finds values close to the estimated depletion values with the Markov chain at the same time intervals. This indicates that in this case, the Markov transition matrix model is replicable using SVM-based strategies with an average error margin of 0.0045±1.5.

Finally, the relationship between suppliers and customers shown in Table 3 is decreasing, indicating that under ideal conditions the value of the decline decreases in suppliers and increases in customers. This means that absorbent states (clients) over time can access the drug through intermediaries more efficiently than in cases where it is supplied by the provider or IPS directly to the client.

REFERENCES

[1] Cardona Orozco Valentina, "Modelamiento Matemático para la reducción de las mermas logísticas en la operación de distribución de medicamentos en una IPS en la ciudad de Bogotá", 2019, Universidad de la Salle.

[2] L. Wei-Qiang, G. Xiao-Miri, Z. Zong-Liang and W. Kai, "Research on Drug Quality Risk Early Warning Model Based on Multiple Improved Optimizing Algorithm," 2018 2nd IEEE Advanced Information Management, Communicates, Electronic and Automation Control Conference (IMCEC), Xi'an, 2018, pp. 2244-2248.

[3] Z. Wang, L. Yang, D. He, S. Ding and B. Du, "Modeling and Optimizing Control of Particle Quality for Drug Fluidized Bed Granulation Process," 2016 International Conference on Industrial Informatics - Computing Technology, Intelligent Technology, Industrial Information Integration (ICIICII), Wuhan, 2016, pp. 195-198.

[4] T. Ban, M. Ohue and Y. Akiyama, "Efficient hyperparameter optimization by using Bayesian optimization for drug-target interaction prediction," 2017 IEEE 7th International Conference on Computational Advances in Bio and Medical Sciences (ICCABS), Orlando, FL, 2017, pp. 1-6.

[5] C. Tsai, H. Lin, J. Taur and C. Tao, "Iris Recognition Using Possibilistic Fuzzy Matching on Local Features," in IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics), vol. 42, no. 1, pp. 150-162, Feb. 2012. doi: 10.1109/TSMCB.2011.2163817

[6] K. Tsuji et al., "Use of Library Loan Records for Book Recommendation," 2012 IIAI International Conference on Advanced Applied Informatics, Fukuoka, 2012, pp. 30-35. doi: 10.1109/IIAI-AAI.2012.16

[7] X. Guo, W. Du, R. Qi and F. Qian, "Minimum time dynamic optimization using double-layer optimization algorithm," Proceedings of the 10th World Congress on Intelligent Control and Automation, Beijing, 2012, pp. 84-88. doi: 10.1109/WCICA.2012.6357844

[8] S. Bechikh, A. Chaabani and L. Ben Said, "An Efficient Chemical Reaction Optimization Algorithm for Multiobjective Optimization," in IEEE Transactions on Cybernetics, vol. 45, no. 10, pp. 2051-2064, Oct. 2015. doi: 10.1109/TCYB.2014.2363878

[9] P. Kurgan, S. Koziel and Q. S. Cheng, "Multi-objective EM-based design optimization of compact branch-line coupler," 2017 IEEE MTT-S International Conference on Numerical Electromagnetic and Multiphysics Modeling and Optimization for RF, Microwave, and Terahertz Applications (NEMO), Seville, 2017, pp. 215-217. doi: 10.1109/NEMO.2017.7964238

[10] M. Kohlík and H. Kubátová, "Reduction of complex safety models based on Markov chains," 2012 IEEE 15th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS), Tallinn, 2012, pp. 183-186.

[11] V. M. Zakharov, B. F. Eminov and S. V. Shalagin, "Representation of Markov's chains functions over finite field based on stochastic matrix lumpability," 2016 2nd International Conference on Industrial Engineering, Applications and Manufacturing (ICIEAM), Chelyabinsk, 2016, pp. 1-5.

[12] H. Kang, K. Cho, X. D. Zhang, T. Zeng and L. Chen, "Inferring Sequential Order of Somatic Mutations during Tumorgenesis based on Markov Chain Model," in IEEE/ACM Transactions on Computational Biology and Bioinformatics, vol. 12, no. 5, pp. 1094-1103, 1 Sept.-Oct. 2015.

[13] G. Han and B. Marcus, "Analyticity of Entropy Rate of Hidden Markov Chains with Continuous Alphabet," in IEEE Transactions on Information Theory, vol. 61, no. 6, pp. 3013-3028, June 2015.

[14] S. K. Sahu, A. K. Pujari, V. R. Kagita, V. Kumar and V. Padmanabhan, "GP-SVM: Tree Structured Multiclass SVM with Greedy Partitioning," 2015 International Conference on Information Technology (ICIT), Bhubaneswar, 2015, pp. 142-147

[15] X. Wu, W. Zuo, L. Lin, W. Jia and D. Zhang, "F-SVM: Combination of Feature Transformation and SVM Learning via Convex Relaxation," in IEEE Transactions on Neural Networks and Learning Systems, vol. 29, no. 11, pp. 5185-5199, Nov. 2018.