Methodology for the Model for Failure Prediction in a Digital Signal Distribution

José Cruz Ramos-Báez, María de Lourdes Martínez-Villaseñor, and
Dafne Rosso-Pelayo

Universidad Panamericana Campus México, México, D.F.,
Mexico

{jcramos,lmartine, drosso}@up.edu.mx

Abstract. In the case of Digital Signal Distribution (DSD), machine learning algorithms have contributed to elaborate better ways to enable failure prediction. In this work a nested model for predicting failures in the components involved in DSD failure is presented. The failure can be caused by multiple and different components and also due to correlations between them. We propose a clustering model to isolate component behavior, and subsequently apply predictive models to each cluster. With principal components analysis and cluster analysis we have been able to identify group of failures' causes in this way failures can be segmented and treated properly. We found seven significant features for classification to determine which part is failing. The clustering process generated two groups that allow us to predict if a general failure is going to occur, and the classification process permits us to forecast which component is probably going to present a failure.

Keywords: Digital signal distribution, failure prediction, predictive models

1 Introduction

Failure prediction has interested research communities from different areas for more than three decades. Different offline and online methods have been used to identify risk situations that can prevent the system of deliver the expected service. A survey of online failure prediction methods and propose taxonomy of online prediction methods is presented in [1]. They described four major categories: failure tracking, system monitoring, detected error reporting, and undetected error auditing. In this paper we focus on failure prediction based on Digital Signal Distribution (DSD) systems monitoring. Using undetected error auditing and cluster, determine a fewer of variables for detected failures.

"Online failure prediction is frequently confused with root cause analysis. Having observed some misbehavior in a running system, root cause analysis tries to identify the fault that caused it, while failure prediction tries to assess the risk that the misbehavior will result in future failure" [1].

pp. 91–101; rec. 2015-04-05; acc. 2015-07-15

In computer systems for Digital Sign Processing, there is "a need for real-time performance within the given operational parameters of a target system and, in most cases, a need to adapt to changing data sets and computing conditions"[2]. In complex Digital Signal Distribution (DSD) systems, the need is also to maintain real-time performance and avoid the interruption of system delivery. Maintaining quality service delivery in DSD is vital, given that phone and Internet service is delivered through fiber optic technology to homes and enterprises. When a system failure occurs, it is very important to maintain the system working as customers in residential homes and companies require fiber optics telephone and internet access, which depend on the DSD capacity.

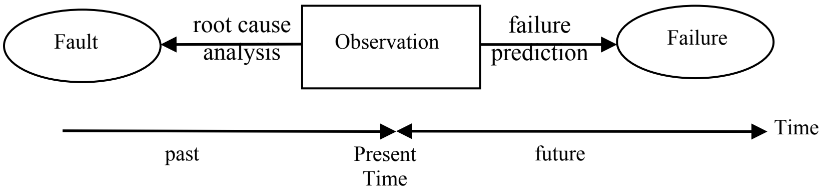


Fig. 1. Distinction between root cause analysis and failure prediction [1]

A methodology to detect and isolate failures in complex Digital Signal Distribution systems is presented, considering faults in: cards; router; VPN; link; FRSW (Finite Range Scattering Wave Function), IC (Integrated Circuit), IT (Information Technology), LANSM (Local Area Network Security Monitor), all of them named as Digital Signal Distribution. The methods included in this work are based in behavior detection and/or fault pattern recognition in big volumes of registers. This work is centered in proactive prediction and management: online failure prediction mainly in order to forecast faults and failures.

Most articles have referred to disk fault detection [3] software [4], using statistical models [5] and [8], and some other model with the results of Machine Learning [6], determine Method prediction [7] and make heuristics models [9].

Our research presents a different approach for DSD system failure detection. We aim to determine with the fewest relevant attributes, if a total failure is going to occur and which component fault is responsible for the failure. We performed principal component analysis to find the most relevant features that enable the failure prediction. Furthermore, we apply clustering processes to make data set segmentation in order to group failures by its behavior. Subsequently we performed a classification first in the whole data set, and afterwards in each of the identified cluster groups in order to find the rules that can describe failure patterns that allow us to detect and predict faults.

The rest of this paper is organized as follows: in section 2, we describe the Digital Signal Distributor considered, and information obtained from system monitoring. We explain our proposed methodology in section 3. We describe our experiments, and discuss the results in section 4. Finally, we conclude and outline our future work in section 5.

2 Digital Signal Distribution

DSD are complex systems composed of multiple parts and elements related with each other. Each component and its relations are probable failure causes. The faults are recollected in a fault log in order to allow failure prediction in the DSD system. In this section we describe DSD system.

2.1 Digital Signal Distribution System

Digital Signal Distribution (DSD) consists in receiving and transmitting digital signals between multiple routers; VPN (Virtual Private Network); links; FRSW (Finite Range Scattering Wave Function); IC (integrated circuit); IT (Information Technology); LANSM (Local Area Network Security Monitor) and cards.

Components of a DSD system are numerous. A DSD system is a receiver and transmitter of digital signals to a certain customers, consisting of wires or fiber optics. Inside of it we can find cards, cables, filters, power supplies, cards memory, ports, etc. Therefor a DSD system is considered a complex system in which a fault in any component may cause that the DSD system suffer a general failure. At present it is not possible to predict what causes a failure to the DSD system, and certainly the one with a failure causes an economic loss by the lack of service.

2.2 Data Description

The data used in this work describe internal components of DSD involved in failures, just some of these failures were resolved, also the data set contains others parts that were involved on DSD system failure. The private data set used contains 11,000 instances detected with errors and has 86 attributes describing components such as card, memory, wires, etc. where the error was present; all data were collected daily during the period 2009 to 2012.

These attributes are numerical and nominal. All of them were taken from various DSD, all DSD consist of the same internal components. The difference is the usage time. We consider that the data collected represent all the possible components and circumstances that can produce a fault. This can be considered as a simplification of the real problem given that unknown causes are possible.

As first criteria in data transformation, from the 86 attributes considered to begin with, we ignored nominal features that are not relevant.

3 Methodology for Failure Prediction in a Digital Signal Distribution

The methods used for failure prediction vary from filter design specific fault, the use of statistical tests and innovations modeling algorithms, and others. We applied a different approach in order to get the best patterns and predict failure in the best way.

We propose a nested clustering and classification model in order to identify faults patterns and behavior to be able to generate failure prediction.

3.1 Data Preparation and Feature Selection

As first step in our methodology we performed a transformation and cleaning process over the data set, ignoring nominal irrelevant features, eliminating not existing values and repeated data. The class feature taken *Closure_code* contains the specific cause of failure, which is determined as the dependent variable *Y*. We use WEKA to find the correlation matrix with these attributes.

Table 1. Correlation matrix, using PCA

1	-0.02	0.03	-0.04	-0.02	0.97	-0.06	0.58	Organization
-0.02	1	-0.05	0.02	0.01	-0.03	-0.15	0.34	Cve_classification
0.03	-0.05	1	-0.05	-0.04	0.03	0.1	-0.53	Closure Code
-0.04	0.02	-0.05	1	0	-0.04	0.01	0.01	Failure time VPN
-0.02	0.01	-0.04	0	1	-0.02	0.01	0.02	Failure time IDE
0.97	-0.03	0.03	-0.04	-0.02	1	-0.05	0.57	Year
-0.06	-0.15	0.1	0.01	0.01	-0.05	1	-0.11	Hour
0.58	0.34	-0.53	0.01	0.02	0.57	-0.11	1	Inc Relation Service Call Id

As second step in our methodology we applied a Principal Component Analysis (PCA). PCA is used in order to emphasize variation and bring out the most significant variables with strong patterns in the dataset. PCA allows better visualization and management of the dataset simplifying the dataset in only 7 attributes with $x_{i,}$ $Y \in X$ and i=1..7. Attributes' correlations with the dependent variable resulted from PCA, are shown in Table 1. This table reveals relationships between our data set attributes, we obtained a small set of independent principal components from our larger 86 set of related original attributes. In general, higher values are more useful, and we consider excluding low values from the analysis. Table 2 shows the variables.

Table 2. Relevant attributes obtained with PCA

Closure_Code.- the specific component that failed,	Y
Organization- It is used to know the organization that created the incident.	x_1
cve_classification.- The classification of where the incident occurred example: Hardware, software, configuration, etc.	x_2
failure time VPN.- The service downtime.	x_3
failure time IDE.- Downtime in the IDE service, Internet Business Manager.	x_4
Year.- year failure	x_5
Hour.- failure time	x_6
Inc Relation Service Call Id.- It is the relationship you have with another ticket raised with or without affectation.	x_7

3.2 Clustering Process

The third step consists in making data set segmentation in order to group failures by its behavior, the procedure used is a clustering method. We applied K-means with WEKA obtaining two clusters shown in Fig. 2. Clusters C_1 y C_2 represent two separation groups of variable Y which is a nominal feature with $y \in Y$; $x_i, y_i \in C_1$ and $xj, yj \in C_2$ where $x_i \in X$. These clusters help to determine if types of failures can be grouped according to their components and behavior.

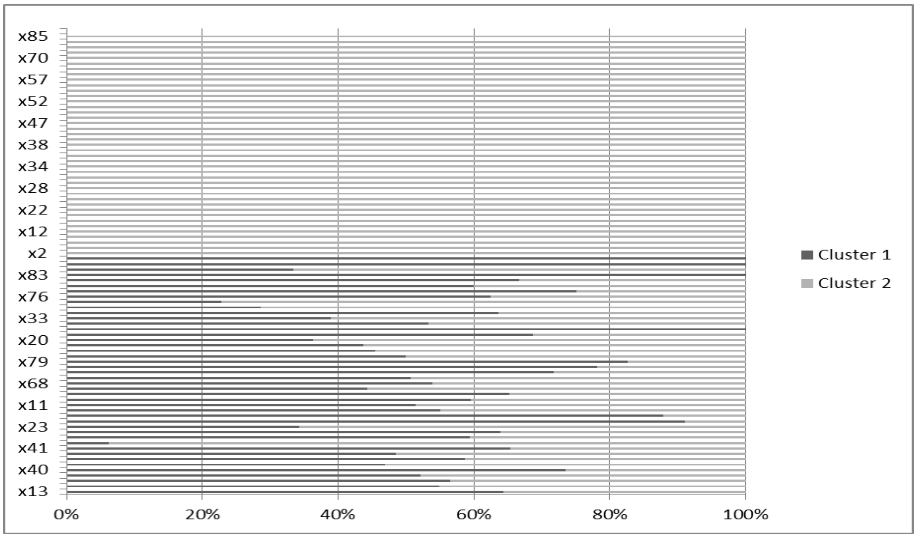


Fig. 2. Cluster using Kmeans

Table 3. Nominal values of *Closure_code*

Variable	Y	Variable	Y
Infrastructure Company	y_{13}	Team strength	y_{65}
Memory card	y_6	Ventilator	y_{60}
Way transmission	y_5	Port	y_{20}
Cabling	y_3	Autoreset	y_{71}
Memory error	y_{16}	Planning hardware	y_{19}
Transmission equipment	y_{41}	Incorrect statement to a third party	y_{33}
support / validation	y_4	Designing	y_{44}
Error in entering commands	y_8	Chassis	y_{76}
Bug	y_{23}	Routing	y_{24}
Air filter	y_{67}	Equipment	y_2
In investigation the root cause	y_{58}	SUCCESSFUL INTERVENTION	y_7

Variable	Y	Variable	Y
Inclusión de actividad tarea instrucción	y_{11}	EQUIPMENT successfully received	y_9
Power supply	y_{62}	Functionality	y_{12}
Equipment climate	y_{59}	Damaged processor board	y_{25}
Incorrect request for a change	y_{46}	Transitory Crash	y_{30}
Error running	y_{68}		

Given the two resulting clusters, we observe that C_1 has the majority of instances that caused a system crash (general failure of the system). C_2 contains very few of this type of instances. This grouping allows us to proceed to further analysis of failure behavior. The nominal values of *Closure_code* presented in table 3 represent those with more relevance to cluster determination.

As a result of K-means clustering we obtained the following clustered instances: cluster C_1 has 4729 (43%) and cluster C_2 contains 6343 (57%) instances. Cluster C_1 is identified mainly with *Infrastructure Company*, and cluster C_2 with *support / validation*. It is interesting to observe that the values, y_{30} which is *Transitory Crash*, which refers to a total or general failure of the DSD system, can just be found in C_2 opening the possibility of relating these failures with the other features.

In order to verify our clustering results, we applied the Expectation Maximization (EM) algorithm. We also applied cross validation in order to obtain better results. The classification results with the whole data set are presented in Fig. 3.

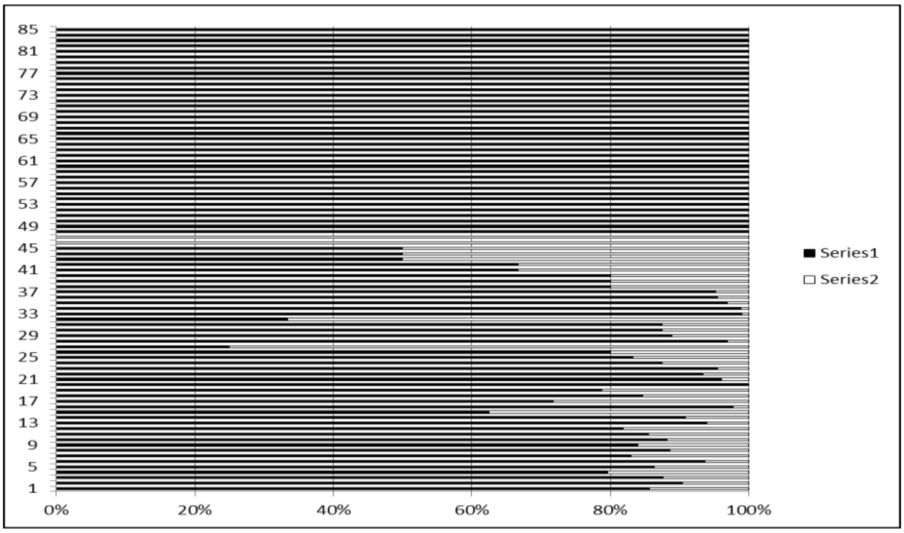


Fig. 3. Clusters with EM

The K-means and EM process used the same features for each cluster, but EM gives a better explanation of the clusters percent. The cluster classifications were 92% and 8% for two clusters.

The following plot graphics has shown the visual representation of cluster 1 data set. We can observe how the cluster items are related to each other; Fig. 4 shows the relation between *Closure_Code* and *Classification*, and Fig. 5 shows the relation of *Closure_code* with the *Service Call Id*. Fig. 6 shows the relation of cluster 2 with the *Service Call Id*, all behavioral differences are shown in clusters. The graphics were developed with K-means.

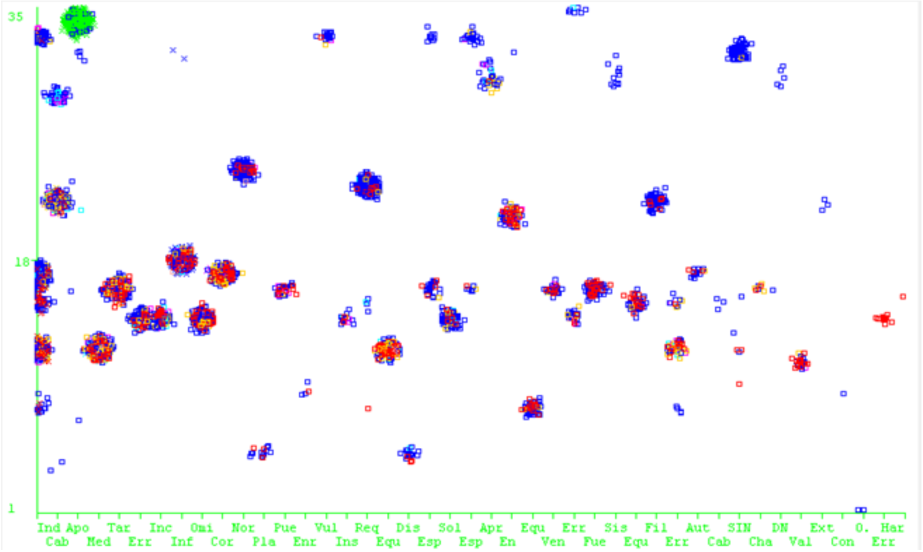


Fig. 4. Closure Code vs Classification in cluster C_1

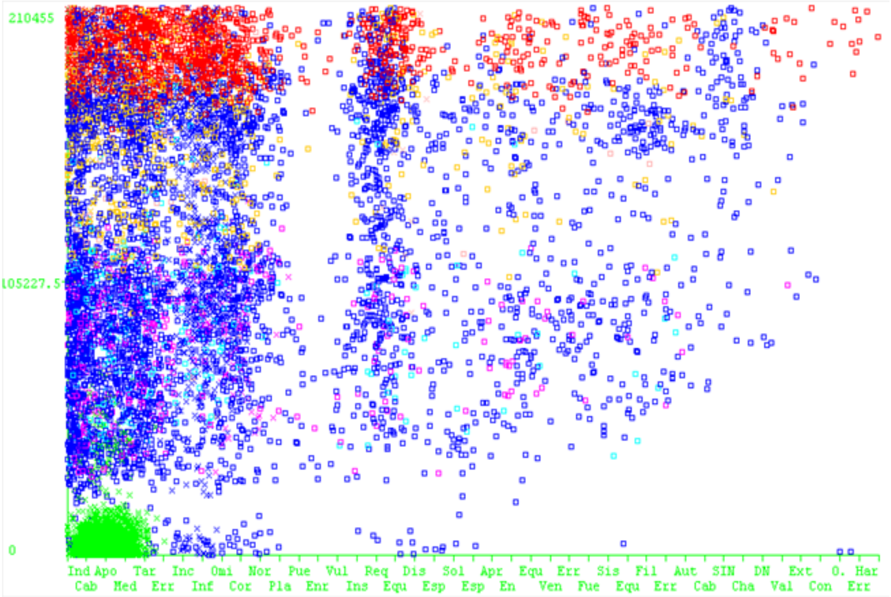


Fig. 5. Closure_code vs Service Call Id in cluster C_1

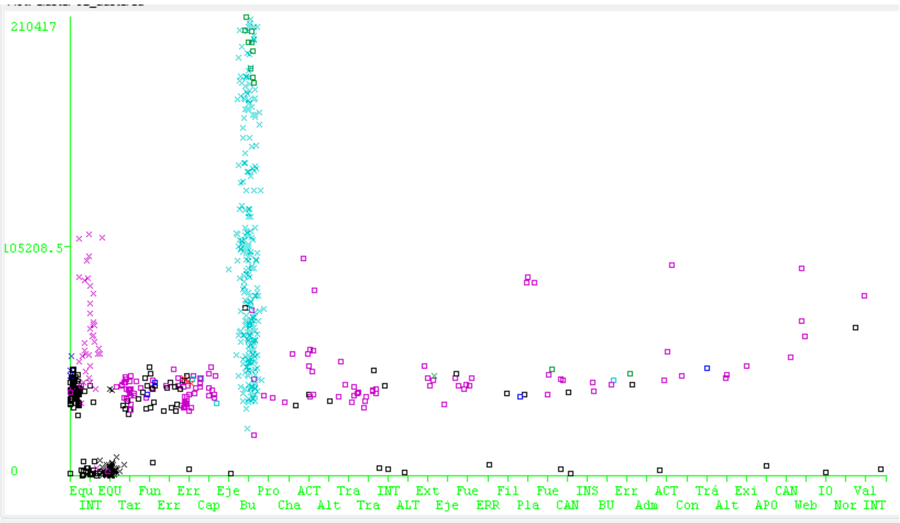


Fig. 6. Closure_code vs Service Call Id in cluster C2

In fig. 5 and 6 we can observe different behaviors in the relation of the features. *Closure_code* vs *Service Call Id* for each cluster.

3.3 Classification Process

The fourth step is performing a classification first in the whole data set, and afterwards in each of the identified cluster groups in order to find the rules that can describe failure patterns that allow us to detect and predict faults. We used the dataset with seven relevant features described in table 2 in addition to *Closure_code* as the supervised class feature to apply the following algorithms: C4.5, J48, Random Forest and Table Decision.

The results obtained with four applied classification algorithms, C4.5, J48, Random Forest and Decision Table, are very similar. The percentages of correctly classified instances are in the range of 78 to 79% which is a fair classification. They also show a root mean squared error around 0.06. Results are presented in table 4.

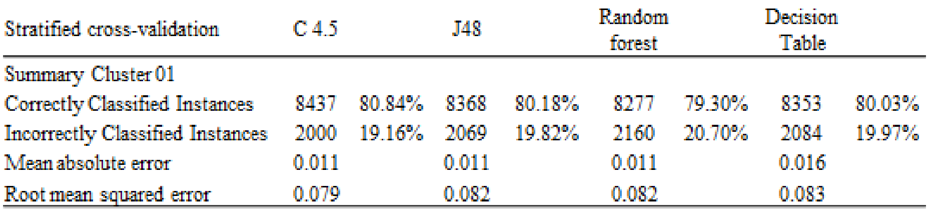
Table 4. Classification algorithms results with the whole data set.

Stratified cross-validation	C 4.5		J48		Random forest		Decision Table	
Summary								
Correctly Classified Instances	8774	79.24%	8790	79.39%	8676	78.36%	8671	78.31%
Incorrectly Classified Instances	2298	20.76%	2282	20.61%	2396	21.64%	2401	21.69%
Mean absolute error	0.006		0.006		0.006		0.01	
Root mean squared error	0.061		0.058		0.059		0.063	

Table 5 shows the results obtained from the classification performed only on the dataset of cluster C_1 using the same classifiers: C4.5, J48, Random Forest, and

Decision Table. We can observe 80% of correctly classified instances and a root mean squared error of 0.08 which is considered low.

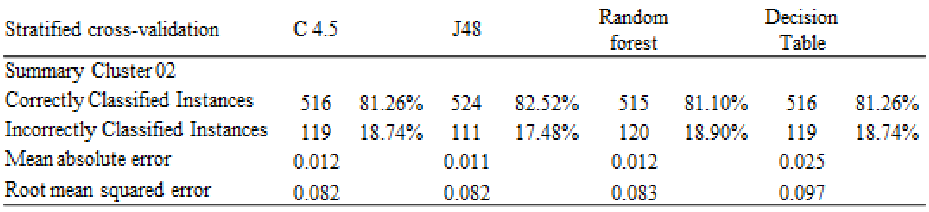
Table 5. Classification algorithms results cluster C_1



Stratified cross-validation	C 4.5		J48		Random forest		Decision Table	
Summary Cluster 01								
Correctly Classified Instances	8437	80.84%	8368	80.18%	8277	79.30%	8353	80.03%
Incorrectly Classified Instances	2000	19.16%	2069	19.82%	2160	20.70%	2084	19.97%
Mean absolute error	0.011		0.011		0.011		0.016	
Root mean squared error	0.079		0.082		0.082		0.083	

Another classification on *Closure_code* feature was performed in the same way with the dataset of cluster C_2 using the same classifiers: C4.5, J48, Random Forest, and Decision Table. We obtained 81% of correctly classified instances and a root mean squared error of 0.08 which is also considered low. Results are shown in table 6.

Table 6. Classification algorithms results C_2



Stratified cross-validation	C 4.5		J48		Random forest		Decision Table	
Summary Cluster 02								
Correctly Classified Instances	516	81.26%	524	82.52%	515	81.10%	516	81.26%
Incorrectly Classified Instances	119	18.74%	111	17.48%	120	18.90%	119	18.74%
Mean absolute error	0.012		0.011		0.012		0.025	
Root mean squared error	0.082		0.082		0.083		0.097	

4 Discussion of Results

In this section, we interpret and discuss the results described in section 3. We described a nested model for predicting failures in the components involved in a DSD system. Firstly we presented a clustering process. Processes K-means and EM process have differences in percentage of the two clusters; however K-mean is more representative in the separation of variables, since K-mean separates almost 50% instances in each cluster. We can also observe different behavior in each cluster comparing the relation of two features in figures 4, 5 and 6. With the K-means clustering process we obtained two interesting groups. Cluster C_1 represent the instances that produce a general failure of the DSD system, and cluster C_2 represent failure that don't produce a total failure of the system.

In the second phase we applied four different classification processes in order to find which component is failing described in the feature *Closure_code*. Classifiers show fair performance for each cluster: for cluster 1 with the classifier J48 it shows a 80.18% of instances classified correctly with a square error of 0.082, and cluster 2 present 82.52% (Tables 5 and 6) of instances classified correctly and the error square is 0.082.

We found that the seven relevant features obtained with the PCA analysis can predict which component of the DSD system will probably produce a failure with a percentage and error determined by the model.

5 Conclusions

We presented a nested model for failure prediction in a DSD system. Our approach firstly separated the instances in two groups that represent the instances that produced general failure of the DSD system, and those which produce other types of failure. Later on, we determined how to predict which part of the system is possibly failing by using seven relevant features. If the classification process indicates which part is possibly failing, we can predict if the system will produce a total failure or not.

We observed that our proposed methodology obtained significant results in DSD system failure detection and prediction. The nested model included in our research is a novel approach in this field.

We found seven significant features for classification on *Closure_code* class that determine which part is failing. The clustering process generated two groups that allow us to predict if a general failure is going to occur, and the classification process permits us to forecast which component is probably going to present a failure.

As a future work, the prediction model can be improved with other clustering and classification algorithms, and we will prove these nested models with real data.

We are also trying to forecast failures considering time intervals, using time series methods.

References

1. Salfner, F., Lenk,M., Malek, M.: A survey of online failure prediction methods. ACM Comput. Surv. vol. 42, no. 3, (2010) DOI=10.1145/1670679.1670680
2. Tessier, R., Burleson, W.: Reconfigurable Computing for Digital Signal Processing: A Survey. Journal of VSLI signal processing systems for signal, image and video technology, vol. 28, no. 1-2, pp. 7–27 (May 2001).
3. Salfner Felix: Event-based Failure Prediction: an Extended Hidden Markov Model Approach. Dissertation. Humboldt Universitat zu Berlin (Feb. 2008)
4. Andrzejak, A., Silva, L.: Deterministic models of software aging and optimal rejuvenation schedules. In: 10th IEEE/IFIP International Symposium on Integrated Network Management (IM'07) pp. 159–168 (2007)
5. Blischke, W. R., Murthy, D. N. P.: Reliability: Modeling, Prediction, and Optimization. Probability and Statistics. John Wiley and Sons (2000)
6. Candea, G., Kiciman, E., Kawamoto, S., Fox, A.: Autonomous recovery in componentized internet applications. Cluster Computing 9(2):175–190 (2006)
7. Csenki, A.: Bayes predictive analysis of a fundamental software reliability model. IEEE Transactions on Reliability 39(2):177–183 (1990)

8. Guyon, I., Elisseeff, A.: An introduction to variable and feature selection. Journal of Machine Learning Research 3:1157–1182, Special Issue on Variable and Feature Selection (2003)
9. Muthumani, N. et al.: A Survey on Failure Prediction Methods. International Journal of Engineering Science and Technology (IJEST) 3(2):1400–1404 (2011)
10. Liang, Y., Zhang, Y., Sivasubramaniam, A., Jette, M., Sahoo, R.: BlueGene/L, Failure Analysis and Prediction Models. In: IEEE Proceedings of the International Conference on dependable Systems and Networks (DSN 2006), pp. 425–434 (2006)
11. Ting-Ting, Y., Lin, D., Siewiorek, P.: Error Log Analysis: Statistical Modeling and Heuristic Trend Analysis. IEEE Transactions on reliability 39(4): 419–432 (1990)
12. Breuer, M. A.: Testing for intermittent faults in digital circuits. IEEE Trans. Compurers, vol. C-22, pp. 241–246 (1973)
13. Dolev, D. et al.: Fault-tolerant Distributed Systems in Hardware. Hebrew University of Jerusalem, Distributed Computing Column, Stefan Schmid TU Berlin & T-Labs, Germany (2015)
14. Kerkhoff, H. G., Ebrahimi, H.: Intermittent Resistive Faults in Digital CMOS Circuits. University of Twente, Centre for Telematics and Information Technology (CTIT) (2015)