Turkish Journal of Electrical Engineering & Computer Sciences

http://journals.tubitak.gov.tr/elektrik/

Research Article

Turk J Elec Eng & Comp Sci
(2017) 25: 677 – 688

doi:10.3906/elk-1503-118

Comparative study for identification of multiple alarms in telecommunication networks

Atila YILMAZ*
Department of Electrical and Electronics Engineering, Faculty of Engineering, Hacettepe University, Ankara, Turkey

Received: 15.03.2015 • **Accepted/Published Online:** 03.02.2016 • **Final Version:** 10.04.2017

Abstract: Telecommunication networks consist of communication units interconnected physically or by means of protocols in order to provide basic services like data, voice, or image transfers. In this study, a modeling frame for network units and their links in a topological frame is presented based on a real mobile communication network named TASMUS (TAktik Saha MUharebe Sistemi - Tactical Field Combat System). Alarm handling is one of the most critical features required in communication networks. Based on simulated single alarm and multiple (double) alarm scenarios, known powerful alarm estimation approaches, namely the coding method, neural networks, and knowledge-based systems, have been studied to assess their capabilities for identifying multiple faults that might occur simultaneously in real time. They have also been compared in order to evaluate the performance of alarms under different noise levels for specific TASMUS networks.

Key words: Telecommunications network, network management, neural networks, alarm correlation, fault identification, coding method, minimal distance decoder

1. Introduction

The basic management activations in telecommunications network units deal with configuration, performance tests, alarm handlings, security, and accounting managements [1,2]. Alarm management in general includes monitoring efforts of network alarms and taking diagnostic measures after identifying the faults causing alarms in the network. As the size of the telecommunications network increases, simultan[eous problems that may cause multiple alarms will also be inevitable. Thus, alarm monitoring and alarm handling are accepted as a serious matter. Alarm data should be processed carefully and the root cause/causes over the recorded alarms must be isolated and estimated with a high accuracy rate in order to take proper and corrective measures in time. The process to achieve this purpose begins with the alarm correlation along the alarm strings of the network. Possible causes are matched to those alarms to provide a fault-correlated alarm database with certain causes. Thus, one of the main applications for the network management system is employing the alarm manager and agents that define other useful manageable applications embedded on network elements in order to access unit resources [3]. Basically, agents send alarms to the manager whenever any abnormality occurs in its functional routines and routes.

There are various studies generally defining alarm management in a knowledge-based structure, and more rarely in statistical and neural basis (causal model, model-based reasoning, case-based reasoning, rule-based,

*Correspondence: ayilmaz@hacettepe.edu.tr

coding) [3–15]. The coding approach is the most popular, well-known, and relatively simple to implement as a main analysis tool in many studies [4,5]. In order to identify the fault alarms in correlations, their causes have to be clustered through recorded alarm data in time windows or using SOM topology [3]. In the coding approach, alarms are converted to vector codes and correlated with all elements of a matrix containing all observed/ recorded faults. These matching and correlation factors are generally evaluated by a minimum distance decoder (MDD). There is a study employing a neural network as an alternative tool to be used instead of MDD in the coding method [3,9]. It was reported that even in noisy environments, a neural network gives better results than the MDD for identifying single source-based faults [6].

In general, most of the studies published and shared in this area assume a single fault and associated time correlation. There is a lack of detailed analysis on neural models and their comparative performance reports, and a few analyses that consider the predictive potential of neural networks with multiple faults available [[8,9]. In this study, we begin with the modeling frame of a real telecommunication network and its components in order to obtain realistic simulation scenarios for alarm-providing signals [7]. Taking this network modeling frame as a basis for our case study, the coding method and MDD algorithms together have been examined for single and double alarms in various alarm scenarios. In the next stage, this standard method employing the coding approach was replaced by a neural network structure and a knowledge-based system while analyzing various design parameters for searching for the best possible outcome in a predictive purpose.

The first section of the article introduces the basic alarm concepts and the frame in which the research development is discussed. The problem development section reviews essentials for understanding the basic models in theory and simulations in terms of system topologies and their possible alarm scenarios. In this part, the article also presents fundamentals of building blocks of modeling tools and preparing the alarm data sets as close as possible to the real cases in the context of TASMUS configuration. The third section underlines the theory of well-known classification and correlation techniques under the problem development requirements and organizes these methods, namely MDD, multilayer perceptrons of an artificial neural network (ANN), and a knowledge-based system (KBS), separately for the case studies defined on the simulation models. The last section discusses all the results and performances for single and double faults generated by simulated models.

2. Problem development and modeling

2.1. Topology symbols

A telecommunication network is generally modeled by a set of radio link equipment and asynchronous transfer mode (ATM) switches. Each network element also has manageable agent software. As in the traditional way, the network elements are represented by nodes with their access points. The equipment for ATM switching, band III, and band IV radio links are abbreviated as S1, B1, and M1 as shown in Figures 1a, 1b, and 1c, respectively. Some typical connection types between units are shown in Figures 2a–2d. Band III radio links in general can be used to serve with a single connection while band IV can provide two radio links at the same time.

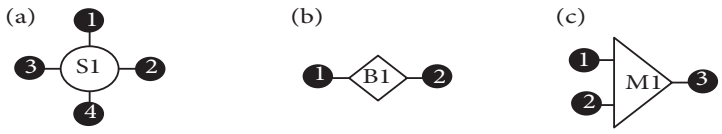

Figure 1. Network units: a) ATM switching, b) radio link band III, c) radio link band IV.

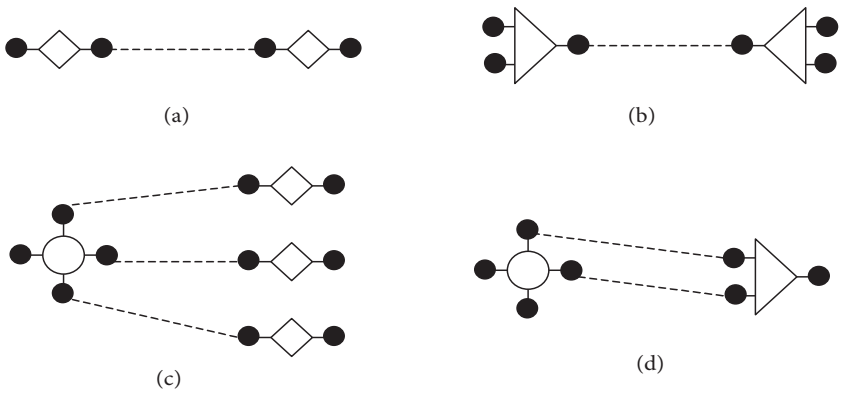

Figure 2. Some basic connection types: a) band III-band III, b) band I-band IV, c) switch-band III, d) switch-band IV.

In this study, a TASMUS network managed by ASELSAN (Turkey) as one of the projects was used to simulate a real network case. The TASMUS network, which combines ATM and ISDN technologies internally, is a communication network that provides secure, high-quality voice, video, and data communication to stationary and mobile users [7]. The system consists of three layers of WAN, LAN, and mobile area network (WLAN) and their associated links. In its mobile system structure, there are time division multiple access (TDMA)-based wireless radio receivers and mobile radio link providers basically for military uses. All those receivers use a data service provided by the ISDN central base to reach all network nodes. This briefly defined system may be a very complicated network with many links covering a military area. However, the modeling units in a topological frame will give us an accurate replication for other real configurations.

The following models shown in Figures 3a and 3b have been generated especially for simulating and understanding some of the real conditions in a similar TASMUS network. S1, S2, and S3 are ATM switching equipment and B1, B2, B3, B4, M1, and M2 are radio link devices physically connected to ATM switches. The numbered black circles show the access points of the equipment. Network elements are connected via links and can make connections as BB1, BB2, MM1, and SS1 indicating the connections between units. For example, BB2 refers the second band II to band II connection. Any equipment failure or link breakdown in the network can cause many possible faults in a chain response (some of them may be unpredictable according to the complexity of the scenario). For such cases the equipment sends alarms to an alarm manager warning that a fault has been observed at that instant. In this study, only link-related alarms are considered in order to make a simple analysis of the performance of the modeling methods outlined above.

Including device-related alarms requires more definitions on the alarm sets and is not necessary to explain the performance analysis of the alarm handling models in comparative ways in our model developing stage. From the topology given in Figure 3a, when a link is broken down or a network element is out of the service, multiple alarms even for a single cause may be generated from the network. For instance, when the MM1 link is broken down, M1, M2, S2-2 (2 refers to the terminal number), S2-3, S3-2, and S3-3 send alarms within a defined short time. Those alarms can also be correlated by using their time window relationships. However, the alarm operator cannot directly resolve the alarms and difficulties to find the root causes still remaining to be solved, e.g., the problem on the MM1 link for this case. As a result the error management system should sort out the problem by identifying the fault (MM1) from the alarm set and the correlated relations. This study extends the multiple alarms problem introducing the second root of causes as well. From the same model, it is also possible to consider another case when two links, BB1 and BB2, are broken down at the same time. In this case, S1-1,

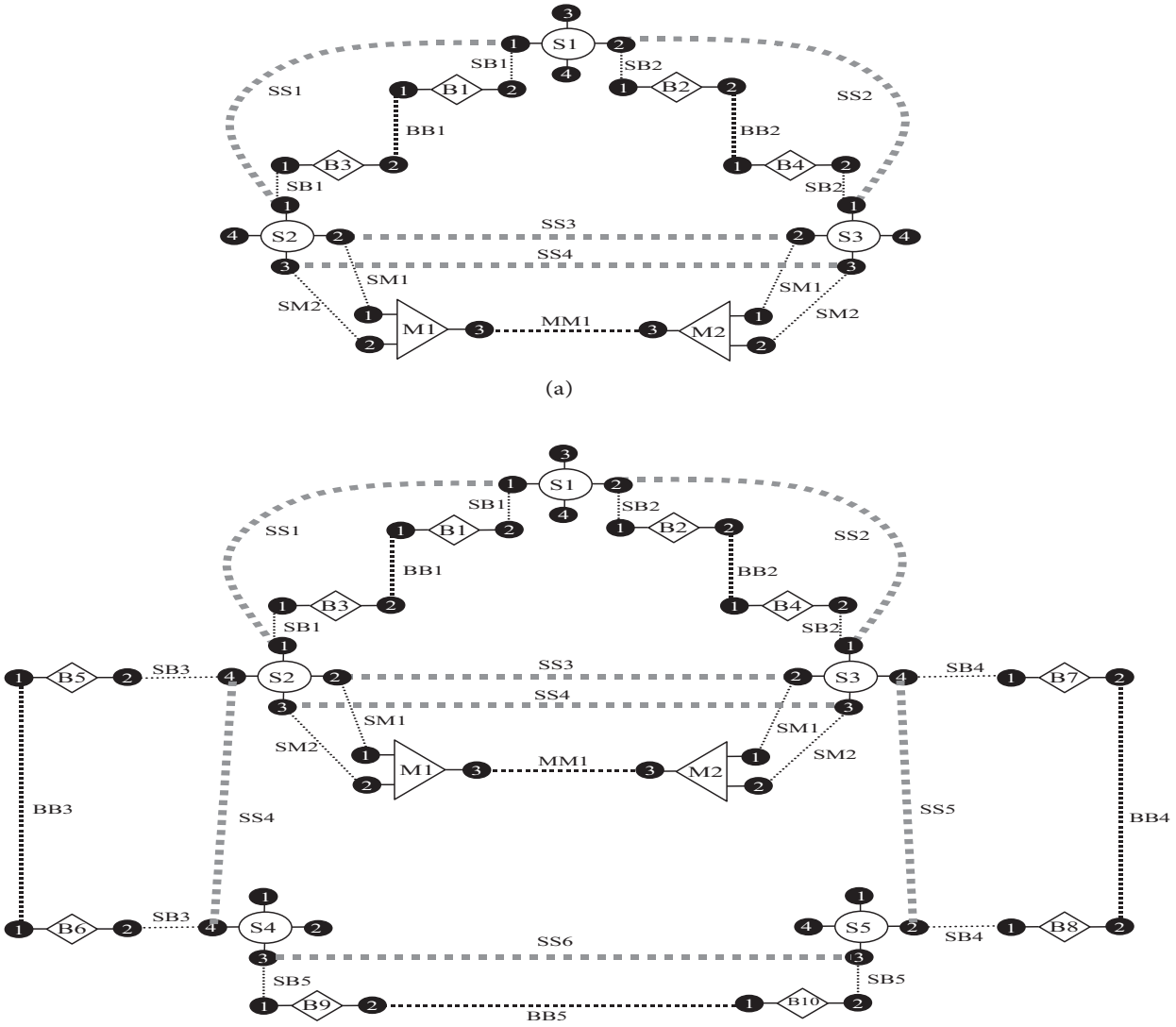

Figure 3. Network topologies used in simulations: a) first case, b) second case.

S2-1, B1, B3, S1-2, S3-1, B2, and B4 related alarms will be observed. Note that, for this case, time correlation on received alarms may not correctly cluster the alarms since correlated alarms contain multiple faults.

2.2. Correlation matrix

For identifying the cases in the correct tracks of alarms, a correlation matrix, introduced as a simple solution by including all the faults and alarms in the network, is tabulated in Figure 4. This is the corresponding matrix for the previously defined topology given in Figure 3a as an example. All of the network elements and links are defined together with all of the link-related alarms that can possibly be received. In the correlation matrix the row headings show the access points or network elements that send the alarms and column headings show the probable problems. The top of each column vector in the matrix defines a single symptom with the associated (alarm) vector. The correlation matrix for the second case given in Figure 3b is inevitably more complicated

due to its size and connections. However, in the simulations both cases will be considered to test our chosen methods for comparison. The creation of the correlation matrix is straightforward and easy to build in practice. If the network topology is available and modeled then all scenarios can be conveyed into the table. In terms of computer simulation, the associated correlation matrix can easily be generated by using a suitable compiler. One should note that the relevant codebook defined by the method is the same as the correlation matrix since there is no redundant alarm information.

	s1	s2	s3	b1	b2	b3	b4	m1	m2	bb1	bb2	sb1	sb2	mm1	sm1	sm2
s1-1	1		1			1				1		1				
s1-2	1	1		1	1		1				1		1			
s2-1				1		1				1		1				
s2-2			1					1	1					1	1	
s2-3			1					1	1					1		1
s3-1					1		1				1		1			
s3-2		1						1	1					1	1	
s3-3		1						1	1					1		1
b1						1				1						
b2				1			1				1					
b3										1						
b4					1						1					
m1									1					1		
m2								1						1		

Figure 4. Correlation matrix for case 1.

3. Theory for the tools used for fault detection

This study uses the same topology with different alarm scenarios for generating experimental data sets for implemented modeling approaches for the detection task. Thus, there will be brief explanations in the following sections about how these modeling tools utilize the data set in the problem development stage.

3.1. Coding approach

The coding approach uses the terms "problem", "symptom", and "correlation" instead of "fault", "alarm", and "identification", respectively. The set of symptom events caused by a problem is treated as a "code" that underlines the problem. This approach uses the MDD to match alarm patterns in the run-time. Correlation is simply the process of decoding the set of observed symptoms by determining which problem has the observed symptoms as its code. The coding approach consists of two phases [4,10]. The first phase is the codebook selection phase. In this phase, all of the problems and symptoms are organized using their cause and effect relationships. A causality graph is obtained with these relationships. Using the causality graph, a correlation matrix including all sets of symptoms and problems is formed. Finally, the correlation matrix is reduced to a codebook. The codebook is an optimal subset of events that must be monitored to distinguish the problems of interest from one another while ensuring the desired level of noise tolerance. The preprocessing algorithm can be used to build the optimal codebook [11].

The following phase is the decoding phase, in which the events in the codebook are monitored and analyzed in real-time by finding problems whose codes optimally match an observed symptom vector. For binary vectors with the same number of elements, the number of locations in which their respective elements differ is called the Hamming distance between these vectors [16]. The MDD calculates the Hamming distance between the observed symptom vector and the code. The cause represented by the codebook vector with the

smallest Hamming distance is proposed to be the problem. Missing and additional alarms are rated as noise. Minimal distance decoding can resolve problems in the case of noise depending on the codebook selection. In the experimental study, a data set is prepared by concatenating 1000 vectors picked up randomly from the matrix. In the first topology there are 16 different alarm vectors while in the second topology there 30 different vectors replicated in a 1000-samples-long data matrix for each case as shown in Figure 5.

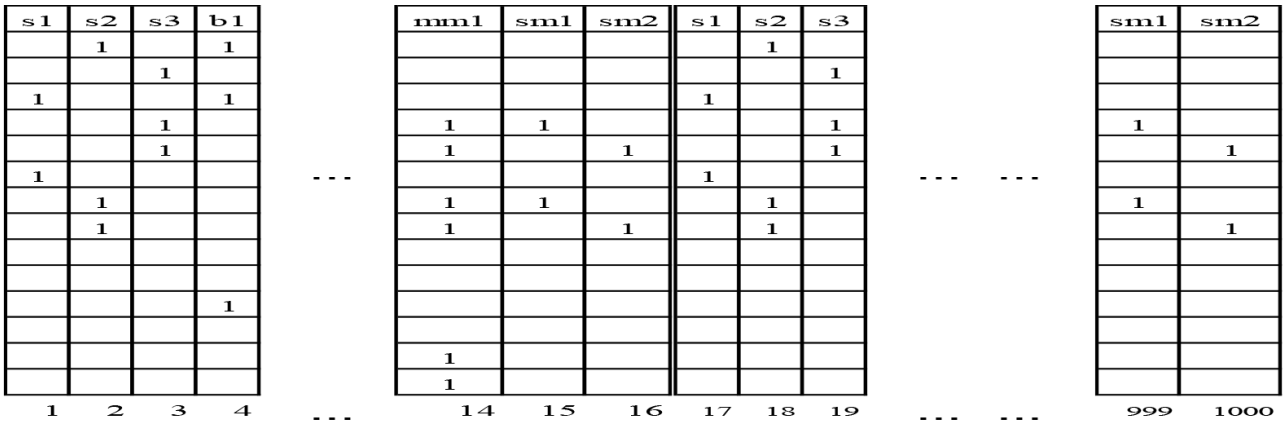

Figure 5. Concatenated alarm vectors.

For the first topology the data matrix is 14 $\times$ 1000 in dimension, whereas the second one is 26 $\times$ 1000. It is also possible that alarm vectors can carry their own structural errors in real time due to misreading, transmission, and alarm correlation problems. In order to minimize the effect of this actual problem, all possibilities are covered by introducing different level of noises, i.e. replacing digits with their complementary ones. For example, in the first data set, if 10% noise is required to be added to the data set, 10% noise will mean replacing 1400 digits with its complementary digits out of the total 14,000 elements in the matrix. For these two topologies the performances of the MDD are presented as correct identification of single faults versus the percentage of noise introduced in Figure 6a. If the time window is not selected properly or two alarms occur at the same time, naturally alarm vectors will obtain indications defined by two alarms. This situation introduces a more difficult case to be handled by the alarm management unit. In Figure 6b, this case with a similar process is presented for two possible causes that can be detected in two different topologies with the same noise-adding schedule. In order to prepare the data set including two faults in one vector, the combinations of two vectors are used in a random manner. The samples that have the same elements are not included in the set and the alarm vectors through the combination process are concatenated to get 1000 samples as before. The best results indicating two faults are accepted as the sources of the alarms by using the MDD method utilizing the Hamming distance on the noisy data. As we can conclude from the figures, the MDD method works fine when there is no noise introduced. However, as expected, the performance drops gradually as the noise level is increased. The performance of the MDD for the double faults becomes poor considering that only 480 alarm vectors are recognized correctly with no noise presence since there is a difficulty to distinguish different alarms from generated similar alarm vectors. The next part will assess the performance of a neural network approach, specifically a multilayer perceptron structure.

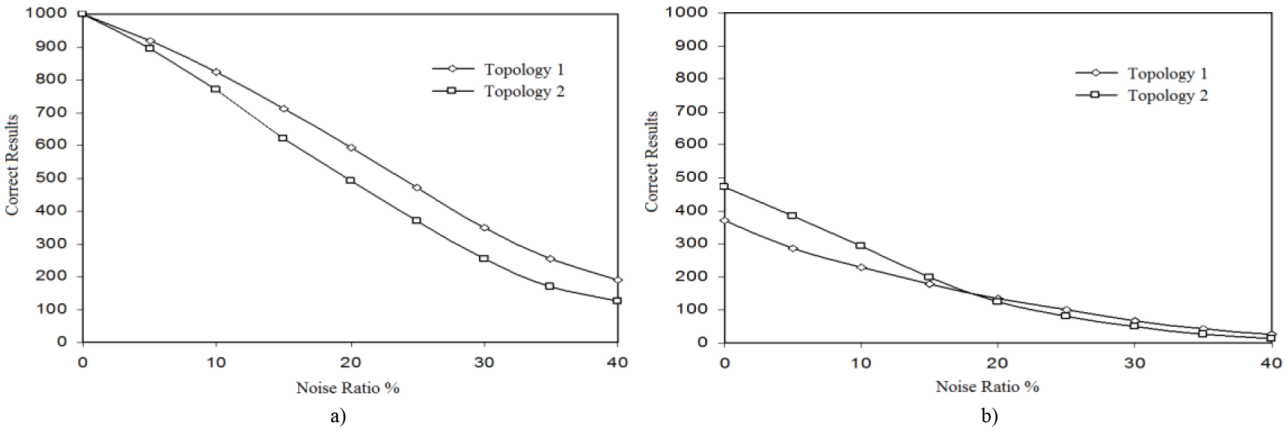

Figure 6. The effect of noise on MDD for a) single fault, b) double faults.

3.2. Neural network as a nonlinear modeling tool

Multilayer perceptrons (MLPs) are proven to be universal approximators for a nonlinear input output mapping. It has long been a problem in nonlinear modeling to find an efficient set of nonlinear functions to approximate the nonlinearity of problems for which the system architecture is not necessarily known [16,17]. Modeling a process can simply be defined as follows:

$$Y_P = \Phi(X_p) + \upsilon, \tag{1}$$

where $X_p = \{x_1, x_2, \ldots x_n\}$ stands for input parameters for the process, while $Y_p = \{y_1, y_2, \ldots y_n\}$ refers to the outputs of the process and υ is an additive noise (it is assumed that the process to be modeled obeys Eq. (1) given above). Under these assumptions, an associated neural network (NN) model can be given by:

$$Y_n = \varphi(X_P). \tag{2}$$

$Y_n = \{y_{n1}, y_{n2}, \ldots . y_{nm}, \ldots y_{nM}\}$ is the output of the NN model and m is the number of the outputs defined by $m = \{1, \ldots, M\}$. $\Phi(.)$ in Eq. (1), and $\varphi(.)$ in Eq. (2) are nonlinear functions that describe input-output relations for the process and NN-based model, respectively. This MLP structure with hidden layers together with a popular learning algorithm known as the error backpropagation algorithm shows an ability to learn from experience through adjusting weight parameters in the training phase. After sufficient training, the model is said to generalize the physical model well on the test data, which had never been used before in the testing stage of the NN model. For a valid generalization, it is important to avoid a phenomenon known as overfitting (overtraining), as well as undertraining, which stands for using fewer epochs than required for the optimal result of the generalization problem [16]. In both cases, the network loses the ability to generalize between similar input-output pairs. The performance of this structure is also strongly related to the training time along with the training set, which consists of carefully chosen pairs of inputs and their corresponding outputs. The practical network used in our study is a feedforward neural network with 100 hidden neurons in order to cover the complicacy that the problems demand. For the backpropagation algorithm, the learning rates are fixed to $\eta = 0.5$ and the momentum term is chosen as $\alpha = 0.6$. The input and output neurons are selected based on the problem as usual. The output neurons, for instance, are limited by the number of causes that might occur in the network. Hidden and output neurons use "sigmoid" functions as a nonlinear unit. It is well known that there are many parameters that affect the performance of the neural network in the learning and the testing stages. Many important parameters were analyzed in detail and reported with a sufficiently comparative manner

[18]. This study refines many possibilities from parameter selection to model selection along with determining a proper learning algorithm selection. Based on the best topology and the parameter selections the following success rate for the fault recognition task is recorded and presented in Figure 7. The same vector sets for two cases have been used again for the neural networks solution. As a standard phase, the set has been divided into two sets to be used in training and testing stages. In the testing stage we have used all available data (1000 vectors) to cover all possible alarm scenarios.

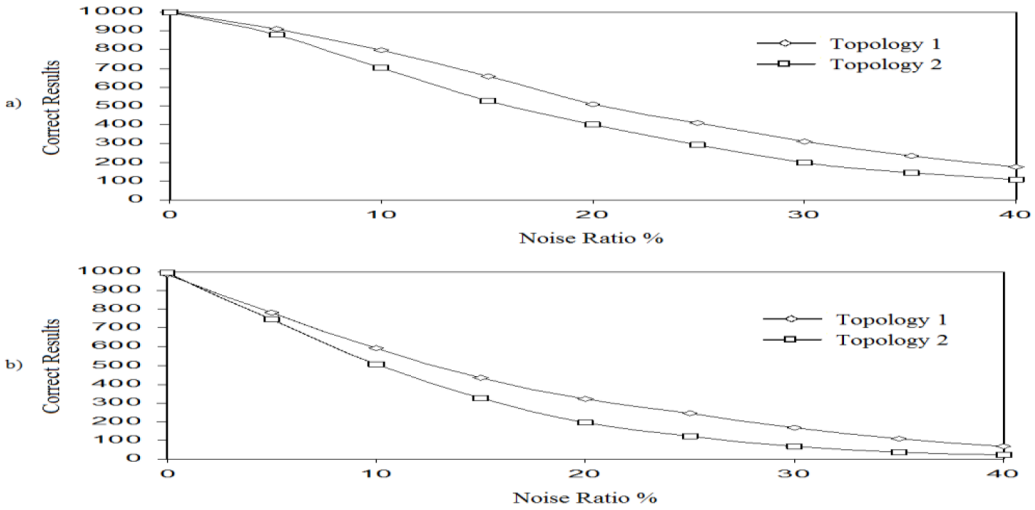

Figure 7. Neural network results under noise: a) for a single fault, b) double faults ($\alpha = 0.6$, $\eta = 0.5$, neurons in single hidden layer = 100, epochs = 500).

More detailed analysis has been carried out in order to propose the best parameter set for the optimal neural network considering the effects like number of epochs, momentum terms, learning rates, number of neurons, and hidden layers [18]. As presented for the MDD, the results obtained from the best neural structure are given in Figures 7a and 7b for single and double faults, respectively. Due to the inherent nonlinearity of the neural model, the performance is much better if it is compared with the results of the MDD, especially for scenarios including double faults. In both approaches, the second case has proven itself as a more challenging topology to identify the alarms. Under 10% noise addition, single and double fault recognitions will be in the bands of 800–700 and 600–500 correct identification rate, respectively. These figures show that the neural network proposed performs well even if noise is extremely effective through communication channels. If there is not any noise challenge, the figures show that the NN model picks up all alarms and interprets correctly in the given time window. When the noise (number of bits complemented) is increased, the performance of identification drops gradually. The last method to be compared is a rule-based model, which shows another paradigm toward a similar task.

3.3. Simulations by rule-based model

In previous parts of the analysis, practical difficulties were observed in generation of the correlation matrix based on cause-and-effect graphs. This part will use the first phase of experience obtained by both the MDD and NN models and redefine the same problem in the frame of knowledge bases in order to develop a better working mechanism. Facts and rules that can be used in constructing correlation matrices in PROLOG realization will

be reviewed for the implementation of the rule-based solution when the alarms from the network are used as inputs to a similar problem.

3.3.1. Generating the correlation matrix

Due to its inherent solution structure with rules and facts, the knowledge-based approach should allow the construction of the correlation matrix easily when the facts are accepted as network devices (elements) and rules refer to the obvious relations between them. In PROLOG, an ATM switch can be described as "central_node (s1_1, s1)", implying that it is the first node of unit s1. In a similar manner, devices like band III are defined as rlA(b1), rlA(b2), rlA(b3), rlA(b4), whereas "rlA(b1)" signifies that band III, named as the b1 device, is in the network. Band IV devices in the network are presented in the form of rlB(m1), rlB(m2) in the same manner. For example, two band III devices can make a radio-link connection and this link is formulized as "linkA(bb1, b1, b3)". In this representation, it is understood that two devices, namely b1 and b3, have a radio-link connection through bb1. Similarly, the expression "bagC(mm1, m1, m2)" means that there is a radio-link connection called mm1 between m1 band IV e and m2 band IV devices [18]. For formulation of realistic scenarios, Table 1 includes some of the facts and associated PROLOG expressions. Rule-based systems use some stated rules to deduce possibilities of solutions by evaluating the listed facts for predefined topologies. Therefore, there should be rules as well in order to underline the dynamic structure that may likely produce a chain up to possible fault sources. These observed rules are summarized in Table 2.

Table 1. Facts and codes of PROLOG defined by topologies.

Fact number	Explanation	PROLOG codes for the example topology
1	There are some ATM switch nodes connected to some other ATM switch nodes	santral_uc(s1_1,s1), santral_uc(s1_2,s1) santral_uc(s2_1,s2), santral_uc(s2_2,s2) santral_uc(s2_3,s2), santral_uc(s3_1,s3) santral_uc(s3_2,s3), santral_uc(s3_3,s3)
2	There are some band III units in the network	rIA(b1), rIA(b2) rIA(b3), rIA(b4)
3	There are some band IV units in the network	rIB(m1) rIB(m2)
4	Some band IIIs are connected to each other	bagA(bb1,b1,b3) bagA(bb2,b2,b4)
5	Some band IVs are connected to each other	bagC(mm1,m1,m2)
6	Some band IIIs are connected to the ATM switch in the network	bagB(sb1,s1_1,b1), bagB(sb1,s2_1,b3) bagB(sb2,s1_2,b2), bagB(sb2,s3_1,b4)
7	Some band IVs are connected to the ATM switch in the network	bagD(sm1,s2_2,m1), bagD(sm1,s3_2,m2) bagD(sm2,s2_3,m1), bagD(sm2,s3_3,m2)

It is possible to obtain a correlation matrix by using these rules and facts. If the program is activated based on all required possible alarms of the first scenario, the following table is generated as an output as given in Figure 8.

The first element gives the fault; the other shows all possible alarms that should be observed for this fault as assigned by 1. This is of course a perfect basis to construct the correlation matrix for the data set to be used for any method named in this study. If the alarm vector includes more than one fault than the PROLOG

then codes generate more than one fault source as a list. For this reason, two different analyses are presented in Figure 9 as a forced single answer or the answer available among possibilities in the list for each single and double fault cases. Figures 9a and 9b also summarize the success rate for different levels of noise effects on results. The rule-based system is forced to produce a single answer for a presented single fault, whereas it produces two likely sources of double faults for searching the exact results capability. Listed results, on the other hand, mark the exact fault sources from all possible answers in the list considering to evaluate whether correct answers are in the list or not.

Table 2. Rules for faults and alarm generations.

Rule number	Explanation
1.1	If a fault occurs in the band III unit, then all other connected band IIIs and connected ATM nodes send alarms
1.2	If a fault occurs in the band IV unit, then all other connected band IVs and connected ATM nodes send alarms
1.3	If a fault occurs in the ATM device, then connected all ATM nodes send alarms
1.4	If a disconnection occurs between band III units, then all connected band III units and connected ATM nodes send alarms
1.5	If a disconnection occurs between a band III unit and ATM node, then all connected ATM nodes send alarms
1.6	If a disconnection occurs between band IV units, then all connected band IV units and connected ATM nodes send alarms
1.7	If a disconnection occurs between a band IV unit and ATM node, then all connected ATM nodes send alarms

```
?- tum_hata(X).

X = [[b4, b2, s1_2, s3_1], [b3, b1, s1_1, s2_1], [b2, b4, s3_1, s1_2], [b1,
b3, s2_1, s1_1], [m2, m1, s2_3, s2_2, s3_3, s3_2], [m1, m2, s3_3, s3_2,
s2_3, s2_2], [s3, s2_3, s2_2, s1_2], [s2, s3_3, s3_2, s1_1], [s1, s3_1,
s2_1], [bb2, b2, b4, s3_1, s1_2], [bb1, b1, b3, s2_1, s1_1], [sb2, s3_1,
s1_2], [sb1, s2_1, s1_1], [mm1, m1, m2, s3_3, s3_2, s2_3, s2_2], [sm2, s3_3,
s2_3], [sm1, s3_2, s2_2]] .
yes
?-
```

Figure 8. PROLOG fault possibilities for the first scenario.

4. Discussion and conclusion

This study has concentrated on presenting three methods in a comparative manner in order to assess dynamic alarm handling capabilities. In simulation results, it is seen that the ANN provides better performance in comparison with the other implemented methods. It is also observed that performances of both the MDD and ANN are strongly affected by the complexity of the topology or dynamic variations introduced by considering new connections. The basic reason for this common disadvantage is that the correlation matrix required by both methods has to be reorganized for any change in the topology as the ANN also demands a new learning phase with a newly defined correlation matrix. On the other hand, for the KBS type fault prediction scheme, introducing only a new device and associated connection knowledge will be sufficient enough to provide the expected performance with minimum disturbance of the whole process. This experience obtained by this study will provide a unified frame to analyze the network in terms of fault sources and adaptive network modifications, as will be reported in future work.

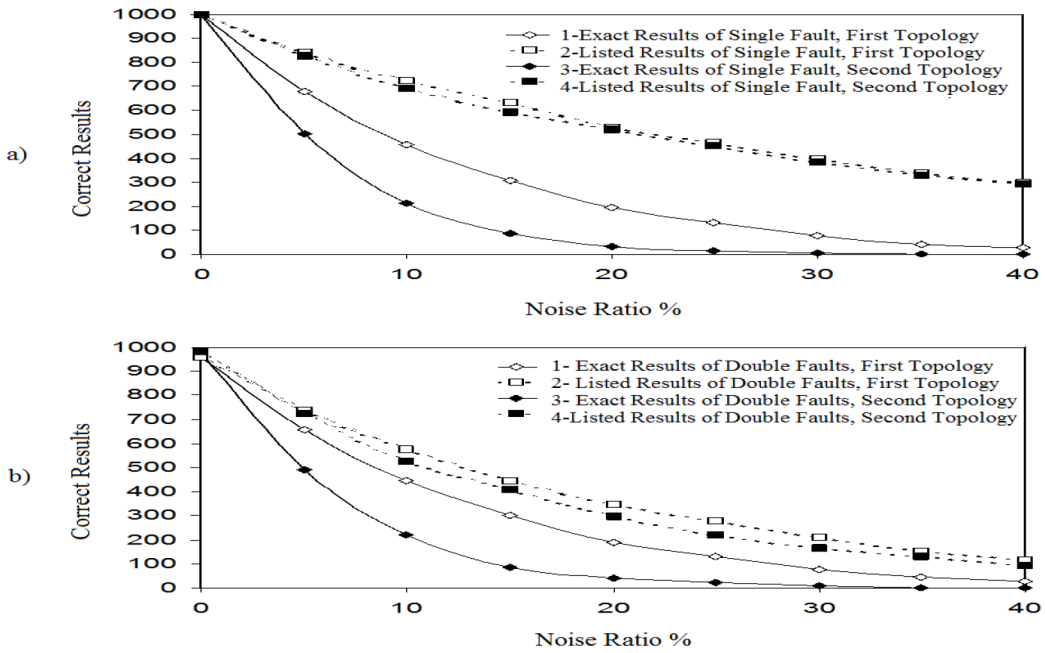

Figure 9. Correct answers from exact and listed results for: a) single, b) double faults.

Acknowledgment

The author would like to thank Associate Professor Karl Shoemaker and Tuna Orhanlı for their comments and contributions in the editing process.

References

[1] International Telecommunication Union. ITU-T Recommendation M.3400 TMN Management Functions. Geneva, Switzerland: ITU, 2000.

[2] International Telecommunication Union. ITU-T Recommendation X.701, System Management Overview. Geneva, Switzerland: ITU, 1997.

[3] Gardner RD, Harle DA. Methods and systems for alarm correlation. In: Proceedings of the 1996 IEEE Global Telecommunications Conference; 18–22 November 1996; London, UK. New York, NY, USA: IEEE. pp. 136-140.

[4] Yemini SA, Kliger S, Mozes E, Yemini Y, Ohsie D. High speed and robust event correlation. IEEE Commun Mag 1996; 34: 82-90.

[5] Mohamed A. Fault detection and identification in computer networks?: a soft computing approach. PhD, University of Waterloo, Waterloo, Canada, 2009.

[6] Wietgrefe H, Tuchs K, Jobmann K, Carls G, Frohlich P, Nejl W, Steinfeld S. Using neural networks for alarm correlation in cellular phone networks. In: Alspector J, Goodman R, Timothy XB, editors. Proceedings of the International Workshop on Applications of Neural Networks in Telecommunications. Hoboken, NJ, USA: Psychology Press, 1997. pp. 248-255.

[7] Uzun Ş, Çiftçibaşi Erkan E. User services of TASMUS in the digital battlefield. In: Military Communications, Meeting Proceedings RTO-MP-IST, 2006. pp. 1-12.

[8] Zhao C. Fault subspace selection and analysis of relative changes based reconstruction modeling for multi-fault diagnosis. In: 26th Chinese Control and Decision Conference; 31 May–2 Jun 2014, Changsha, China. New York, NY, USA: IEEE. pp. 235-240.

[9] Angeli ACC. On-line fault detection techniques for technical systems: a survey. International Journal of Computer Science & Applications 2004; 1: 12-30.

[10] Klinger S, Yemini S, Yemini Y, Ohsie D, Stolfo S. A coding approach to event correlation. In: Integrated Network Management IV, Proceedings of the Fourth International Symposium on Integrated Network Management; May 1995; Santa Barbara, CA, USA. pp. 266-277.

[11] Gupta M, Subramanian M. Preprocessor algorithm for network management codebook. In: Workshop on Intrusion Detection and Network Monitoring; 1999; Berkeley, CA, USA. pp. 93-102.

[12] Towell GG, Shavlik JW. Knowledge-based artificial neural networks. Artif Intell 1994; 70: 119-165.

[13] Wen F. Intelligent alarm processing and fault diagnosis in digital substations. In: 2010 International Conference on Power System Technology; 24–28 October 2010; Hangzhou, China. New York, NY, USA: IEEE. pp. 1-5.

[14] Nygate TA. Event correlation using rule and object based techniques. In: 2008 Fourth International Symposium on Integrated Network Management; May 1995; Santa Barbara, CA, USA. pp. 278-289.

[15] Jang JSR, Sun CT, Mizutani E. Neuro-Fuzzy and Soft Computing: A Computational Approach to Learning and Machine Intelligence. Upper Saddle River, NJ, USA: Prentice-Hall, 1997.

[16] Haykin S. Communication Systems. 3rd ed. New York, NY, USA: Wiley Publishing, 2004

[17] Haykin S. Neural Networks: A Comprehensive Foundation. Upper Saddle River, NJ, USA: Prentice-Hall, 2007.

[18] Kılınç I. Analysis of fault management in telecommunication networks. MSc, Hacettepe University, Ankara, Turkey. 2002.