Fault Prediction in Self-Healing Telecommunication Systems

Ahmad S. Kazmi [1]

Abstract

Telecommunication systems are heterogeneous networks with parts supplied by many vendors. Such complex systems face a number of faults that may deny services to the end users resulting in revenue losses to the telecommunication companies. Best case scenario is to avoid these faults completely, or failing that, correct the faults as soon as possible. Therefore there is a need for self-healing networks that can proactively predict and correct faults automatically. In this paper a fault prediction technique is presented that is useful in a self-healing network. The proposed technique first trains an artificial intelligence technique on the historical alarm data to find correlations and then uses these correlations to predict future alarms. The artificial intelligence techniques being used are, Artificial neural network, support vector machine, Kalman filter and hidden Markov model. In this paper we reported on artificial neural network. The proposed technique is applied on the alarm data from a real telecommunication company and prediction accuracies of the proposed technique are calculated. The details of the proposed fault prediction technique and results that suggest optimal parameters are presented. The proposed technique is effective in a proactive self-healing network.

Keywords: Self-healing Networks; Alarm; Faults; Neural Networks; Telecommunication system

1 Introduction

Today's telecommunication networks are complex heterogeneous systems consisting of components from multiple vendors. Recurring faults in a telecommunication network is a major reason for degradation and refusal of service. A faster fault resolution results in quick resumption of service. Therefore the need for self-healing networks have arisen [1] [2] [3] [4]. In a self-healing network faults are detected and corrected automatically. In a more proactive self-healing network faults are predicted and corrected before happening resulting in less network failures and degradation of the service. There are two parts to a proactive self-healing network: one is prediction of impending faults and second is the root cause analysis of the impending faults [5]. This paper deals with the first part, fault prediction, and suggests how to do the second part, root cause analysis. The reported self-healing networks [1] [2] use current state of the network to make a prognosis of an impending fault. One approach models the degrading factor as additive faults and fault estimate sequence is treated as time series [3]. Another approach uses compensating service degradation by readjustment of parameters using fuzzy logic in a wireless healing

network [4]. We have taken a different approach of using historical alarms to find correlations between components of a telecommunication network and then predicting faults.

[1] *University of Central Punjab, Lahore | ahmad.kazmi@ucp.edu.pk*

Any telecommunication network management system must have a mechanism of collection and reporting of network faults in the form of alarms. These alarms contain a lot of information about the health of the system over time. Unfortunately this valuable information is hidden in a large volume of alarms. Therefore use of an automated tool is a must to make sense of the alarm data and develop algorithms to find hidden patterns. There are two types of correlations that exist in the alarm data: one is spatial relation between network components and second is the temporal relation between sequence of alarms that can be considered as a time series. ANN has been proven to be valuable in recognizing patterns not found by other techniques in a time series [10] [11].Various applications of ANN have been reported for pattern recognition [12] [13]. ANN has the inherent property of making sense of the correlations found in nonlinear data. The addition of the hidden neuron layer and positive or negative feedback mechanism is the main reason for ANN's success. The fault prediction, from historical alarm data, can be formulated as a pattern recognition (or a classification) problem.

In the proposed technique the artificial neural network (ANN) is trained to learn to find these correlations and use this learning to predict future alarms. The proposed ANN based technique predicts faults by formulating an input matrix based on historical alarm data. The input (feature) matrix contains spatial and temporal features that help ANN in finding patterns and correlations. We believed that the formation of the input matrix and our application of the ANN are new, unique and do better fault predictions. In this paper we reported the details of the proposed ANN based fault prediction technique. A number of experiments have been conducted to calculate prediction parameters and accuracy of the prediction. The accuracy of prediction is calculated by comparing the predicted fault occurrences with the actual fault occurrences. The optimal parameters are reported that produce the best prediction accuracies of a single event. It is believed that the same approach is useful in any network to find optimal parameters for accurate faults prediction using ANN. Furthermore an algorithm is suggested on how to proactively selfheal a telecommunication network.

This paper is arranged in five sections as follows. First section is the introduction. Second section is about the related work. Third section presents the proposed ANN based fault prediction technique in detail. 4th section presents the results and optimal parameters. Last section is about conclusions and possible future research directions.

2 Related Work

A Self-healing Networks

Telecommunication networks are becoming more complex as more functionalities are being added. This complexity is further enhanced due to multivendor components servicing highly heterogeneous networks. It is very common that a typical telecommunication network supports various communication channels (wired, wireless, and a combination) and technologies (SONET, GSM, CDMA, 3G, 4G etc.). The Telecommunication Management Network paradigm of FCAPS (what is FCAPS) needs changes to become more autonomous. An automatically managed network will be self-organizing requiring self-configuration, self-optimization and self-healing

[2]. Such an automated network is called a Self-Organizing Network (SON) (www.fp7-socrates.org). An important feature of SON is self-healing.

A self-healing network will proactively and automatically monitor, diagnose and correct anomalies and faults. A self-healing network can be considered as one that can automatically manage faults. Although that automatic fault management must also include a way to look at the system as a whole and not just rely on alarms from network elements. A number of techniques have been proposed on how to achieve this holistic approach to automatic fault management.

[1] proposes a hybrid fault prediction model that supports automatic self-healing networks. Based on the system situation the fault prediction model selects one of these methods: ID3 algorithm, fuzzy interference, fuzzy neural network and Bayesian network. The proposed self-healing system consists of five modules: system monitoring module collects system data, resource level evaluator module decides the level of detail in the system, prediction model selector module selects a suitable algorithm that can be applied for the current state, prediction model algorithm executor applies the selected prediction algorithm and model updater module feedbacks the resulting prediction to the four modules.

[2] presents a unified self-healing network model that identifies tasks to be performed automatically. First task is automatic information collection from these sources: configuration parameters, alarms, network counters, network traces, real time monitoring, drive tests, Key Performance Indicators (KPI) and context information. The second task is fault detection that includes identification of out of service components and service degradation components. Of course identification of out of service components is easy, but the identification of degrading components will require looking at alarms and KPIs. The third task is diagnosis requiring identification of faults and then the corresponding remedial actions. The fourth task is fault compensation that requires changing system parameters for graceful degradation of service rather than abrupt drop of service. The last task is reporting and storing the current steps for future use.

[3] Presents a fault prediction based self-healing approach. First equation for a discrete time dynamic system is provided and then any degradation of the system is modeled as another input to the system, basically a fault. Based on this model, fault prediction and estimation is made. After fault prediction system reliability is predicted. Particle filtering is used to estimate faults. The fault estimates result in a time series that is processed using exponential smoothing.

[4] Uses Mamdani [14] type fuzzy logic to adjust system parameters to compensate the degrading network. For example antenna downward tilt can be controlled by a bisector defuzzyfication. The proposed system consists of three types of modules: Fuzzy Forward Module that applies the changes in the forward (increase) direction, Fuzzy Backward Module that applies the changes in the reverse (decrease) direction and Monitor module that monitors the wireless network performance. The proposed algorithm keeps applying changes via Fuzzy Forward module and monitors the performance. As soon as the performance is targeted degrading the Fuzzy Backward module acts in the opposite direction until optimal values are achieved.

[5] Targets the self-healing part of a SON and presents a framework for automatic detection (of what) and diagnosis of mobile communication systems. The detection part of the framework uses KPI from the network by a unified KPI interface that compares a KPI with a reference or profile and returns a level (0,1) indicating conformance. The KPI profile is built using sampled values of the KPI and using Cumulative Distribution Function (CDF). The diagnosis part of the framework targets either a root cause or a corrective action. In order to avoid false alarm/wrong diagnosis, a built in null target is used. The heuristic knowledge from an operator is stored in the form of KPI level deviation that indicates a fault and corrective steps that will restore the desired KPI level. The relationship between a KPI level and target root cause is stored in a structure called Report. A report consists of a KPI subset and diagnosis target of the report. A scoring system is used to identify the most suitable target from a list of targets (root causes) contained in the reports. The scoring system assigns a score to a target based on reports that best match deviations from KPI levels. Since a target may have different reports containing KPI subsets, a particular likelihood value is used for the KPI. Basically expert knowledge is used to find out the high level of KPI against a given target. A KPI may consistently happen with a target, therefore a consistency score between (0,1) is used.

This paper addresses the issue of fault detection in a self-healing network. The approach is to go one step further and predict a fault before happening. Furthermore, historical alarm data is used for fault prediction rather than other information sources. The reason is that active information collection from various resources of an active real network impacts on network performance and overhead of the mechanism needed for information collection. Furthermore, historical alarms over a period of time contain a lot of information about the network health and there is no need to collect information from other sources. The main problem with historical alarms is that the network health information is not easily accessible and identifiable. Fortunately the information in historical alarms can be considered as a classification or pattern matching problem and a number of artificial intelligence techniques are available for pattern matching or classification. Furthermore features in the historical alarm data are used such a way that leads to better fault predictions.

B Artificial Intelligence Techniques for Fault Prediction

There is a wealth of information hidden in the historical alarm data. The alarm data collected at various network points contain the state of the system health at any point in time. Any number of simple statistical techniques can reveal valuable facts about a system e.g. the most occurring faults and where and how these happen; how these faults are caused and how to eliminate these causes; average down time due to these faults; order of redundancy needed to keep the system running etc. Artificial intelligence techniques are needed to find or classify various patterns available in the alarm logs.The fault prediction can be considered as an event prediction problem. More specifically an event prediction can be considered as a pattern recognition or classification problem for a time series. A number of event prediction techniques, in general and fault prediction in particular, are reported in literature.

Intelligent prediction systems are expert systems that use a variety of different prediction (AI and non AI) techniques to make a hybrid system that can predict more accurately than individual prediction techniques. One of the studies [1] presents a hybrid intelligent system for learning about data. The architectures for Neural Networks (ANN), Fuzzy Intelligent systems (FIS), Evolutionary Computing (EC) and probabilistic reasoning (PR) are discussed along with the integration issues of these techniques.

Another research [2] presents a four layered FMAS (Fuzzy multi agent system) for stock market prediction. These four layers use ANN (Artificial Neural Networks), GA (Genetic Algorithm) and SOM (Self Organizing Maps) to develop a hybrid prediction model. The model is based on the co-ordination of intelligent agents that perform the functions of data processing and learning. GA has been used to automate the design process of ANN in some studies [3] and the process is proved to give better time series forecasting results than statistical models like ARIMA. Hence there has been huge focus on designing this kind of hybrid prediction systems for various predictions in different industries. Time series prediction systems have also been explored many times in the past for efficient and intelligence based forecasting of time series data.

3 Proposed Fault Prediction Using ANN

We present a new fault prediction technique that uses ANN on real historical alarm data of a telecommunication company. The proposed technique first statistically analyzes the alarm data and then formulates an input matrix for the ANN system. The formulation of input matrix is central to our approach. Therefore here we have presented some basis and basic terminologies that will be helpful in describing the proposed technique.

A Alarm Data

The historical alarm data of a telecommunication company is used to apply our proposed fault prediction technique. A typical alarm is shown in fig. 1.

Alarm ID	Alarm Type	Severity	Probable Case	Managed Object
42193	Link Down	Major	ALM_IMA_LINK_LCD	EMS T2000

Managed Element	Rack	Shelf	Slot	Reason	Date
PTP	1	1	7	Loss of Signal	2/09/2013:4:27

Figure 1: A typical Telecommunication Alarm

Alarms in telecommunication are messages describing some sort of abnormality or malfunctioning in the network. This malfunctioning might not be visible to the end user. The Network Operational Center (NOC) of the wireless network daily receives thousands of alarms generated by BTSs and BSCs. The receiving system stores these alarms in a database and displays these alarms to the operator. Later these alarms can be analyzed and converted into useful data by a correlation system. The generation time and generation date field together make up the alarm time. The information about the location of the network element, which has sent this alarm, is scattered in several fields. These are slot number, BTS number, Link number etc.

The process of alarm correlation is difficult because of the complexity of network elements. These network elements produce different types of alarms and usually there are large variations in the order of alarms in a given time frame. A model based analysis of the data can be carried out with the assistance of a network management expert, who knows what patterns to expect. However the statistical analysis, as proposed here, gives us more insight into network behavior and we can detect patterns that enable us to predict future faults. Another important reason for statistically analyzing the data is that different telecommunication management experts may not agree on same set of rules. The historical alarm data is in chronicle order. The basic idea behind a correlation system is to discover recurrent patterns from alarm database. It usually consists of discovering alarm rules within a time window in the alarm database.

1) *Analysis of the Real Time Alarm Data*

Here we present some statistical analysis of the historical alarm data [9]. This analysis is essential for the formulation of the input matrix for the ANN application. The alarm data is analyzed to categorize alarms according to alarm predicates like severity, type and sub rack and slot numbers. Our goal in fault prediction is to predict one of the critical alarms as these alarms are actually faults.

Table 1: Percentages Of Alarms With Different Severities And Types

	Trunk	Running (define)	Comm.	SW	Signal
Warn.	0.0000	1.9107	0.0000	0.0014	0.0000
Minor	2.9367	0.0000	0.0014	0.0070	0.0000
Major	54.5566	4.5594	0.0000	2.7186	0.3131
Critical	0.0000	0.0000	25.7128	7.2822	0.0000

From the table 1 we can see the types and severities of faults that actually occur. Furthermore 11 categories from a total of 20 appear with non-zero percentage and some of these, three in this case, have less than 1% chance of appearing.

B *Input Matrix Formulation for ANN*

One major contribution of this paper is the formulation of the input matrix that is applied to the ANN for fault prediction. This input matrix contains a particular state of the historical alarms over a given time period. There are two types of information contained in the matrix: frequency of occurrence of an alarm property (alarm predicates) and frequency of occurrence of sequence of alarms (event sequences). The alarm predicates are determined by analysis of the alarm data. One predicate of an alarm means presence or absence of certain property of that particular alarm. The alarms being used here have five types and four levels of severities or stages. This makes a total of 5x4 possibilities. These 20 alarm categories can appear on different sub racks and slots. There are sixteen sub racks and each of these have 16 Slots. Apart from these there are some other important alarms that usually occur on BTS – BSC interface (called PCF define). These alarms can also have the 20 types and severity. Therefore this results in a total of (20 x 16 x 16) + 20= 5140 predicates. We have analyzed the given historical alarm data and eliminated those predicates that do not have significant (less than 1 %) contribution. We identify 21 predicates that are present in the given alarm data. Table 3 shows these 21 predicates. These predicates are allotted predicate numbers for ease of use in our ANN system.

Table 2: The 21 Predicates Present In The Alarm Data

Equipment	Param. 1 (what do you mean by parameter, it is not clear)	Param.2	Should be changed	Alarm type	Pred. No.
Subrack	Subrack=2	PCF alarm	Communication	Critical	1
Subrack	Subrack=3	Slot=0	Trunk	Major	2
Subrack	Subrack=4	Slot=0	Trunk	Minor	3
Subrack	Subrack=4	Slot=0	Trunk	Major	4
Subrack	Subrack=5	Slot=0	Trunk	Minor	5
Subrack	Subrack=5	Slot=0	Trunk	Major	6
Subrack	Subrack=5	Slot=12	Software	Critical	7
Subrack	Subrack=6	Slot=0	Trunk	Major	8
Subrack	Subrack=6	Slot=13	Software	Critical	9
Subrack	Subrack=7	Slot=0	Trunk	Major	10
Subrack	Subrack=8	Slot=0	Trunk	Minor	11
Subrack	Subrack=8	Slot=0	Trunk	Major	12
Subrack	Subrack=9	Slot=0	Trunk	Major	13
Subrack	Subrck=10	Slot=0	Trunk	Major	14
Subrack	Subrck=10	Slot=12	Software	Critical	15
Subrack	Subrck=11	Slot=0	Trunk	Major	16
Subrack	Subrck=12	Slot=0	Trunk	Minor	17
Subrack	Subrck=12	Slot=0	Trunk	Major	18
BTS	BTS	PCF	Software	Major	19
BTS	BTS	X	Running	Warning	20
BTS	BTS	X	Running	Major	21

Please note that the predicates in table 3 are not generic and are identified for the given 6 months of alarm data. For any other alarm data, an analysis (table 1, 2& 3) must be done to identify the predicates to be used in the input matrix. The 21 predicates form one side of the state of the alarm data and correlates the alarms in a spatial sense, the other side is the

frequency of sequence of events that correlates the alarms in a temporal sense. For a given time period, all occurrences of all the episodes (of the selected 21 predicates) are calculated. An episode is any sequence of 2 alarms occurring during a given time period one after the other. Hence there can be 21x21=441 possible episodes of alarms and we can find the frequency with which each of these episodes occur in a given time. Therefore frequency of an episode can be used to calculate the probability (relative frequency) of one alarm occurring after another alarm so that following type of statement can be made.
"If an alarm with predicate 1 occurs then another alarm with predicate 6 will follow within 20 conds with probability (relative frequency) 0.54"

An alarm A at time t1 can have one of the 21 predicate values. Suppose alarm A is followed by alarm B. Then alarm B can also have one of the 21 predicate values. Therefore (1) is the formula that is used to calculate the probability of the episode A→B (B follows A or A leads to B)

$$P(A \to B) = \frac{frequency(A \to B)}{\sum_{i=1}^{21} frequency(A \to X_i)} \quad \forall\, A, B \in [X_1 \ldots . X_{21}] \qquad (1)$$

The formula states that the probability that alarm A leads to B is equal to the frequency with which alarm A leads to alarm B divided by the sum of frequencies with which alarm A leads to any alarm having each of the 21 predicates values. Here it is important to note the difference between an alarm and a predicate. Each entry in our alarm data base is called an alarm and these alarms are then (after statistical analysis of data) categorized into 21 predicates. Each predicate is a set of properties. For example if an alarm has predicate value 1 then it is a major communication alarm on sub track 6 slot no. 3 of a BTS. (2) shows that the sum of frequencies with which A leads to alarms with each of the 21 predicate values is equal to 1.

$$P(A \to B) = 1 \qquad \forall\, A \in [X_1 \ldots . X_{21}] \qquad (2)$$

The probability (relative frequency) values for all episodes contain the temporal information needed for the input matrix to the ANN system. The input matrix contains spatial and temporal information about the current state of the telecommunication system as contained in the alarm data. The input matrix is basically a 21x21 matrix of probabilities. These are the probabilities of occurrence of all the episodes (21x21=441) in the current state. This input matrix is one of the main contributions of this paper.

C Proposed Fault Prediction Methodology

We propose an ANN system (fig. 2) for the fault prediction from a given set of historical alarm data.

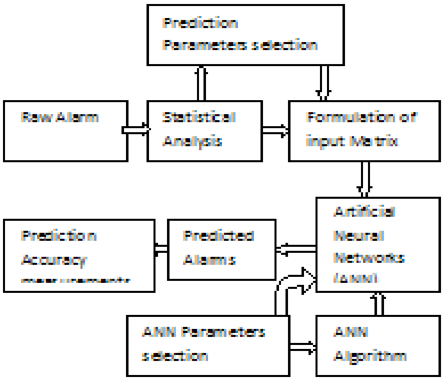


Figure 2: The Proposed ANN System for Fault Prediction

The raw alarm data is fed to a Statistical Analysis module to help select the prediction parameters. The statistical data is used to formulate the input matrix for the ANN module. The ANN module uses a particular algorithm and selected ANN parameters to output the predicted alarms. Finally prediction accuracy measurements are done by comparing the predicted alarms with the actually happened alarms. A number of experiments are run to find the optimal prediction and ANN parameters.

D Prediction Parameters

There a number of prediction parameters that can be varied to find an optimal mix.

Alarm Period is the total time duration for which we have the historical alarm data e.g. 6 months.
Training Period is the portion of the Alarm Period used for training the ANN.
Prediction Interval (PI) is the portion of the period of time that will be used for fault predictions.

Time Window (TW) is an important parameter that requires a bit of explanation. The overall training period is equally divided into time slots called time windows. First of all, statistical analysis is done for the first Time Window (TW) and that is the first sample of statistical data. Next the TW is shifted by an amount called Time shift (TS) and a second sample of statistical data is collected. Similarly the time window is successively shifted and statistical data is collected until the end of the training period is reached. The characteristics of these collected samples are used to formulate the input matrix.

The Time window and Time shift are 2 important parameters and have significant impact on the prediction accuracy. A large Time window will has more data in one sample but less number of samples. A wide window shift will has less overlapping data and less number of samples. A narrow window shift will has more overlapping data and more number of samples. The optimal window and window shift can only be obtained by experiments.

Fig. 3 shows the time windows. Also, it is to be noted that:

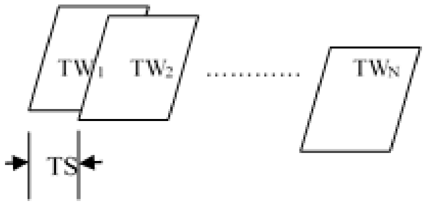


Figure 3: Windowing Technique for Correlation

$$time\ shift(seconds) \leq window(seconds)$$

In fig. 3 TW is time window and TS is time shift.Now for each window taken, the first task is to find all the 441 episodes. We have to calculate three values for each episode:

1) Frequency of an Episode: is the frequency of occurrence of an episode
2) Probability of an Episode: is the probability (relative frequency) that an episode will occur
3) Time bound (delta) of an Episode: is the maximum time period between 2 alarms of an episode.

In each window, frequency of episodes is a 21x21 matrix that stores the frequencies of each of the 441 episodes. Similarly the Probabilities and Time Bounds are also 21x21 matrices in each window. Each row of the 441 matrix represents a predicate which appears first in an episode and each column represents a predicate which follows the first one in the episode. Hence these predicate matrices store episode characteristics for all 441 episodes.

E ANN Input and outputs

Let us suppose that for a particular Window Size, there are total m windows in the Training Period. The 441 values for the episodes (frequencies, probabilities and delta) of the first window is used as an input to the neural network (window 2 in fig. 3). The target of the ANN for this input (three 21x21 matrices) will be three 21x21 matrices in the second window. This means that we are predicting future window (see window 3 in fig. 3) state from the current window state. The first window is only used once as an input. As there are m windows, the number of training examples we have are m-1.

F Evaluation of results

The input to the proposed ANN is three 21x21 binary matrices for frequency, probability and delta of episodes of the current window. The output of ANN is the predicted three 21x21 matrices for frequency, probability and delta of episodes of the next window. Our next task is to obtain a list of predicted alarms for the next window. Let us show an example to understand how this prediction is done.

Suppose Window Length = 2700seconds (45min) and Time Shift=900seconds (15min)

Note that the length of the two windows on both sides of the overlap area is equal. Because of the overlap area, we already know the initial alarms of the next window. These alarms (in overlap area) are used to predict the remaining alarms in the next window. Hence the remaining length of the next window will be: Window length - overlap area= time shift. So our prediction interval (during which we are doing the prediction) is equal to the time shift.

For example, if an alarm with predicate 18 occurs in overlap area, then we look at the 18th row of each of the three predicted matrices (21x21). If predicted frequency, probability and delta matrices all show a value greater than 0.5 for ith column of 18th row, then i is added in the list of predicted alarms. This procedure is done for all the alarms in the overlap area and a list of predicted alarms is obtained. Some of these predicted alarms actually occur(true positives), and the rest are wrongly predicted (false positives). Also some alarms that actually occur are not predicted at all (false negatives).

Next task is to define a measure for calculating the accuracy of this prediction. We calculate the accuracy in terms of two factors:

1) CPA(Percentage of correctly predicted alarms with respect to actual alarms)

For an input window, we find the CPA from actual and predicted alarms vectors as follows:

$$CPA = \frac{length(actual\ alarms\ \cap predicted\ alarms)}{length(actual\ Alarm)}\ x\ 100 \quad (4)$$

This factor tells us what percentage of actual alarms is correctly predicted and is the so called true positives in literature. If this process is done for all the windows we have in our data, and then a mean is calculated then it is called mean CPA.

Mean CPA= (∑_(i=1)^p▒〖CPA〗_i)/p (5)

Where p is the total number of windows.

2) CPP(Percentage of correctly predicted alarms with respect to predicted alarms)

For an input window, we find CPP from actual and predicted alarm vectors as follows:

CPP= (length(actual alarms ∩predicted alarms))/(length(predicted Alarm)) x 100 (6)

This factor tells us what percentage of predicted alarms actually occurred. If this is low, then it may be that we end up predicting a lot of wrong alarms even if our CPA is quite high. Mean CPP is given as:

Mean CPP= (∑_(i=1)^p▒〖CPP〗_i)/p (7)

Where p is the total number of windows. The optimal parameters will maximize both CPA and CPP.

4 Results

A number of experiments had been done on a six months of alarm data from a telecommunication company. The purpose of these experiments is to find optimal ANN and prediction parameters that provide the best Mean CPP and CPA values. These experiments are specific to the alarm data at hand and will have to be done for any other data.

A Optimal ANN parameters

There are a number of parameters available for the ANN. The following parameters are not dependent on the dynamic data and can be decided based on static knowledge about the data and prediction goals.

Number of inputs are 441 (matrix of 21x21 predicates).
Numbers of outputs are 441 (matrix of 21x21 predicates)
Number of neurons in first layer is equal to the number of inputs in the first layer and is equal to the number of outputs in the last layer.
Transfer functions are TANSIG (-1 to 1) and LOGSIG (0 to 1)
Training algorithm are Back propagation Gradient Descent (TRAINGDX) and Back propagation Conjugate Gradient Training algorithms (TRAINCGB).We have used these two training algorithms and also compared them for better results.
Data sets for training, validation and test data are data sets needed to control ANN. The training data set is used to update weights and biases, and to calculate error. The validation dataset is used to stop the training before the net starts over fitting. The test data just tells us at run time about how the network will perform in case of unseen data. We have used 20% training data as validation data, 10% as test data and 70% as training data
Maximum number of epochs is the maximum number of times all the inputs are iterated through the network. If the number of epochs reaches this number during training, the training stops. We have used a value of 1000 for maximum number of epochs.

The parameters, Learning Rate and (LR) and Number of Hidden Layer Neurons (NHLN), are dependent on the data and are decided through experiments using a window size of 2 days and a time shift of 1 day. In these experiments we calculate and analyze Mean Square Error (MSE) of prediction, the convergence rate, CPA and CPP values.

Table 3: Effect Of Learning Rate On Mean Square Error (MSE)

Learning Rates	Prob. net MSE	Freq. net MSE	Delta net MSE	Average MSE
0.1	0.0374	0.0348	0.0286	0.0336
0.3	0.0258	0.0336	0.0289	0.0294
0.5	0.0327	0.0311	0.0336	0.0325
0.7	0.0341	0.0456	0.0453	0.0417
0.9	0.0259	0.0297	0.0262	0.0273

It can be seen from table II that the best learning rate (minimum average of all three nets MSE) is 0.9. Although a learning rate of 0.9 will be very unstable (does not converge). Therefore we recommend using a learnig rate of 0.3 because it is close to the performance at 0.9 learning rate and convergence is better.

Table 4: Learning Rate And Prediction Accuracy

Learning Rate Analysis TRAINGDX on test data

Learning Rate.	CPA	WPA	CPP	WPP
0.1	83.70	16.30	77.91	22.09
0.3	84.43	15.57	78.24	21.76
0.5	84.34	15.66	78.16	21.84
0.7	84.62	15.38	77.75	22.25
0.9	83.32	16.68	78.46	21.54

Table IV shows the accuracy parameters for different Learning Rates. Here CPA and CPP are as defined earlier. WPA is the percentage of actually occurring alarms that are not predicted by ANN and is equal to 100 – CPA. WPP is percentage of predicted alarms that actually never occurred and is equal to 100 - CPP.

In considering all these factors (MSE, convergence speed and prediction accuracy parameters) we conclude that a learning rate of 0.3 is a good compromising value.

Table 5: Number of Nhln on Mse

For GDX (Effect of increasing Hidden neurons)

Hidden Neurons	Prob. net MSE (least)	Freq. net MSE (least)	Delta net MSE	Mean MSE
220 (inp/2)	0.0405	0.0361	0.0353	0.0373
441 (inp)	0.0327	0.0311	0.0336	0.0325
882 (inp*2)	0.0225	0.018	0.0208	0.0204
1764 (inp*4)	0.0193	0.0164	0.0201	0.0186

Table 6: Number of Nhln on Mse

For CGB (Effect of increasing Hidden neurons)

Hidden Neurons	Prob. net MSE (least)	Freq. net MSE (least)	Delta net MSE	Mean MSE
220 (inp/2)	0.027	0.0321	0.0358	0.0316
441 (inp)	0.0286	0.0274	0.0245	0.0268
882 (inp*2)	0.0403	0.0193	0.0272	0.0289
1764 (inp*4)	0.0488	0.031	0.0222	0.0340

It can be seen from tables 5 and 6 that 441 values for number of NHLN is a good number. NHLN of 441 provides good convergence as the higher values are an over kill.

It can be seen from table 7 that 441 provides good enough prediction accuracy parameters. Furthermore a NHLN of 441 has a low mean training time. Therefore after due consideration we conclude that 441 is a good compromising number for NHLNs.

Table 7: Number of Nhln and Prediction Accuracy

Hidden Layer Neurons Analysis TRAINGDX on test data				
Hid. Neurons	CPA	WPA	CPP	WPP
220	82.87	17.13	78.86	21.14
441	84.34	15.66	78.16	21.84
882	84.27	15.73	77.77	22.23
1764	83.45	16.55	78.31	21.69
Hidden Layer Neurons Analysis TRAINCGB on test data				
Hid. Neurons	CPA	WPA	CPP	WPP
220	83.93	16.07	78.63	21.37
441	83.92	16.08	78.02	21.98
882	83.91	16.09	78.22	21.78
1764	83.75	16.25	78.12	21.88

B Optimal Prediction Parameters

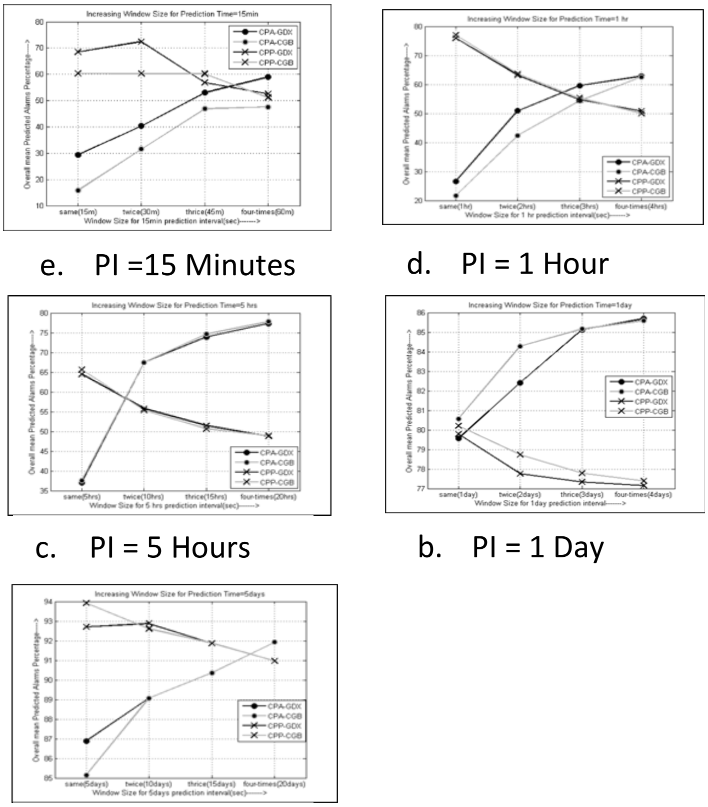
The proposed fault prediction algorithm depends on the Time Window (TW) and Time shift (TS) parameters (see Section III). Therefore search for optimal TW and TS parameters is an important contribution of this paper.

In our experiments we had used five different values for TS: 15 minutes, 1 hour, 5 hours, 1 day and 5 days. The TW is varied as equal to: TS, twice that of TS, thrice that of TS and four times that of TS. Total number of experiments is 20 and 20 neural net sets were trained (20 nets each for frequency, probability and delta) for analyzing the effect of increasing TW for each TS and the effect of increasing TS for each TW.

1) Optimal Time Window (TW) Size Analysis

The proposed fault prediction algorithm depends on the Time Window (TW) and Time shift (TS) parameters (see Section III). Therefore, search for optimal TW and TS parameters is an important contribution of this paper.

For the following experiments we have analyzed the values of CPA and CPP for the 2 training algorithms: GDX and CGB. These values of CPA and CPP are calculated for the training, validation and test data. First four months of training data is used and then 2 months of actual predictions are done.



e. PI =15 Minutes

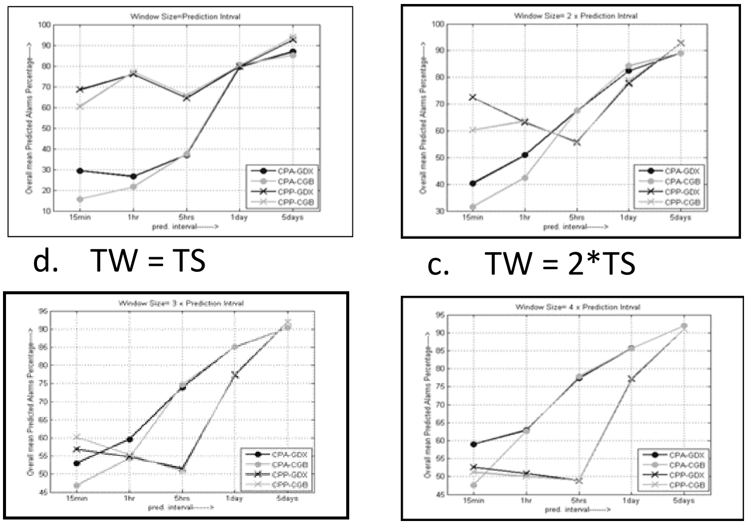
d. PI = 1 Hour

c. PI = 5 Hours

b. PI = 1 Day

a. PI = 5 Days

Figure 4: TS and TW Analysis for different PI



d. TW = TS

c. TW = 2*TS

b. TW = 3*TS

a. TW = 4*TS

Figure 5: TW and TS Analysis

Fig. 4 shows graphically the values of CPA and CPP for the 2 months of predictions on unknown test data as window size is increased. In fig.4a, fig.4b and fig.4e, the best accuracy is for a TW that is 3 times the PI. In fig.4c and fig.4d, the best accuracy is for TW of 4 times the PI. Note that fig. 4 shows the mean of the training, validation and test result values (CPA, CPP) for the two algorithms. Fig. 4 thus gives us an overall picture for choosing the best TW value for a particular TS value.

Fig. 5 shows graphically the values of CPA and CPP for the 2 months of predictions on unknown test data as window size is increased. Fig.5 show that for TW = TS, TW = 2*TS, TW = 3*TS and TW = 4*TS, the accuracy keeps getting better for increasing PI. It can be seen that a window size equal to PI is a good compromise. Based on our experiments we believe that our approach is useful for use in a self-healing network. Here is our proposed procedure for proactive fault prediction so that the faults can be predicted and corrected in self-healing network

Training Phase

1. Statistically analyze the raw historical alarm data to identify contributing Predicates
2. Use raw historical alarm data to formulate the input predicate matrix
3. Train the ANNs using the input matrix and the raw historical alarm data
4. Run experiments and select optimal ANN (learning rate and number of NHLN) and prediction parameters (TW and TS)
5. Update the input matrix for the prediction phase.

Prediction Phase

6. Use optimal parameters, resultant input matrix and ANN to predict future (within the prediction period) faults
7. Initiate corrective measures for the predicted future faults.

Please note that steps of the training phase are done one time only. Therefore the time spent in the training phase is not important. The prediction phase is rather fast and can predict faults in seconds.

Proposed Parameters selection for Real Time Fault prediction

Following are the recommended ANN and prediction parameters based on our experiments and analysis of the results.

1. Learning Rate: 0.5
2. Hidden Layer Neurons: Equal to number of inputs (441)
3. Prediction Interval (PI): Availability of 6 month alarm data.
4. Time Window (TW): As per need of the self-healing network operational requirements
5. Time shift (TS): 2*TW

5 Conclusion

A fault prediction technique had been presented that is useful in a self-healing network. The proposed fault prediction technique used historical alarm data and trained an Artificial Neural Network (ANN) for future alarm predictions. A thorough analysis is done to find optimal ANN and prediction parameters. The prediction results indicated that the proposed fault prediction technique will be useful in prediction and correction of faults before these faults occur.

For future work, other artificial intelligence techniques will be applied and compared using the proposed technique. Furthermore the proposed fault prediction technique will be tested on an actual Telecommunication network to develop a practical application for self-healing of the network.

References

[1] Giljong Yoo, Jeongmin Park and Eunseok Lee,"Hybrid Prediction Model for improving Reliability in Self-Healing System", Proceedings of the Fourth International Conference on Software Engineering Research, Management and Applications 2006 (SERA'06)

[2] Raquel Barco, Pedro Lázaro, and Pablo Muñoz, "A Unified Framework for Self-Healing in Wireless Networks", IEEE Communications Magazine • December 2012 p 134-142

[3] Zhengguo Xu, Yindong Ji, and Donghua Zhou, "A New Real-Time Reliability Prediction Method for Dynamic Systems Based on On-Line Fault Prediction", IEEE TRANSACTIONS ON RELIABILITY, VOL. 58, NO. 3, SEPTEMBER 2009 p 523-538

[4] Arsalan Saeed, Osianoh Glenn Aliu, Muhammad Ali Imran, "Controlling Self Healing Cellular Networks using Fuzzy Logic", IEEE Wireless Communications and Networking Conference: Mobile and Wireless Networks 2012

[5] Peter Szilagyi and Szabolcs Novaczki, "An Automatic Detection and Diagnosis Framework for Mobile Communication Systems", IEEE TRANSACTIONS ON NETWORK AND SERVICE MANAGEMENT, VOL. 9, NO. 2, JUNE 2012 p 184-197

[6] Marco Canini, Dejan Novakovic, Vojin Jovanovic, and Dejan Kosti, "Fault Prediction in Distributed Systems Gone Wild", LADIS 10 Zürich, Switzerland 2010

[7] K. Hatonen, M. Klemettinen, H. Mannila, P. Ronkainen, H. Toivonen "Knowledge Discovery from Telecommunication. Network Alarm Databases".In 12th International Conference Data Engineering (ICDE 96'), pp 115-122, New Orleans Louisiana. Feb. 1996.

[8] P. Fröhlich, W. Nejdl, K. Jobmann, H. Wietgrefe "Model-based alarm correlation in cellular phone networks". InProc. of MASCOTS 1997 (1997)

[9] C. Lo, S. Chen, B. Lin. "Coding based schemes for fault identification in communication networks". In International Journal of Network Management; 10:157-164. 2000.

[10] Z.Q. Liu and F. Yan, "Fuzzy Neural Network in Case-Based Diagnostic System", IEEE Transactions on Fuzzy Systems, Vol. 5, No. 2, 1997.

[11] M. Klemettinen, H. Mannila, and H. Toivonen. "Rule discovery in telecommunication

alarm data".In Journal of Network and Systems Management, 1999.Vol 7 No.4: 395-423 (1999).

[12] M. Khalid, "Statistical techniques for fault prediction in telecom Networks", MS Thesis, University of Computer and Emerging Sciences, Pakistan, 2009

[13] O.P. Kogeda and J.I. Agbinya, "Prediction of Faults in Cellular Networks Using Bayesian Network Model". In the Proceedings of First IEEE International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2006), Sydney, Australia, March 13'16, 2006.

[14] Mamdani, E.H. and S. Assilian, "An experiment in linguistic synthesis with a fuzzy logic controller," International Journal of Man-Machine Studies, Vol. 7, No. 1, pp. 1-13, 1975.