Modified Hopfield Neural Network for Identifying Faults in Symmetric Comparison Models

Mourad Elhadef
College of Engineering and Computer Science
Abu Dhabi University, Abu Dhabi, UAE
mourad.elhadef@adu.ac.ae

***Abstract*—This paper presents a modified Hopfield neural network for solving the system-level fault diagnosis problem under the symmetric comparison model. The comparison-based self-diagnosis approach assigns tasks to the nodes, and the outcomes from each pair of units performing the same task are compared. The objective is to identify the set faulty of nodes based on the matching and mismatching among the system's nodes. We consider t-comparison-based diagnosable systems in which at most t nodes can fail permanently at the same time. Results from a thorough simulation study demonstrate the effectiveness of the Hopfield-network-based self-diagnosis algorithm for randomly generated diagnosable systems of different sizes and under various fault scenarios, making it a viable addition or alternative to existing diagnosis algorithms.**

***Index Terms*—Fault tolerance, System-level diagnosis, Multiprocessor systems, Symmetric comparison models, Hopfield neural networks.**

I. Introduction

The system-level fault diagnosis problem aims mainly at answering the very simple question "Who's faulty and who's fault-free?", in systems known to be diagnosable. In recent decades, the need for dependable computing systems for critical applications has motivated researchers to investigate this problem by assuming that nodes are able to test and to be tested by other nodes of the system. From the results of the tests, nodes need to be diagnosed as faulty or fault-free. This problem, also known as the *self-diagnosis problem*, has been extensively studied in the last three decades (the reader is referred to the following surveys for more details [3], [15]). Three types of diagnosis models have been studied: testing models [19], comparison models [16], [12], [20], and probabilistic models [15]. Testing models, such as the classical PMC model [19] and its variations, assume that each node is assigned a subset of the other nodes to test and the diagnosis is based on the collection of test outcomes. While, comparison models, such as the generalized comparison model (GCM) [20], assume that a set of jobs is assigned to pairs of distinct nodes, and the results are compared. The outcomes of these comparisons, i.e., the matching and mismatching results, are used as a basis in order to identify the set of faulty nodes. In invalidation and comparison models, a worst-case behavior is always assumed. That is, only t-diagnosable systems where the maximum number of faults is bounded by t are considered in order to guarantee a certain level of diagnosis. Finally, probabilistic models [15] do not assume any bound, but instead, only fault sets that have a non-negligible probability of occurrence are considered.

In this paper, we consider the comparison-based diagnosis approach since it is considered to be more practical. The comparison approach has been introduced independently by Malek [16] and by Hakimi and Chwa [12] giving rise to two models. The Malek's model is known as the *asymmetric comparison* model and that of Hakimi and Chwa is called the *symmetric comparison* model. In both models it is assumed that two fault-free nodes give matching results while a faulty node and a fault-free node give mismatching outcomes. The two models differ in the assumption on comparison tests involving a pair of faulty nodes. In the symmetric model, both test outcomes are possible in this case (0 or 1), while in the asymmetric model two faulty nodes always give mismatching outputs. Fig. 1 summarizes all possible comparison outcomes for both models.

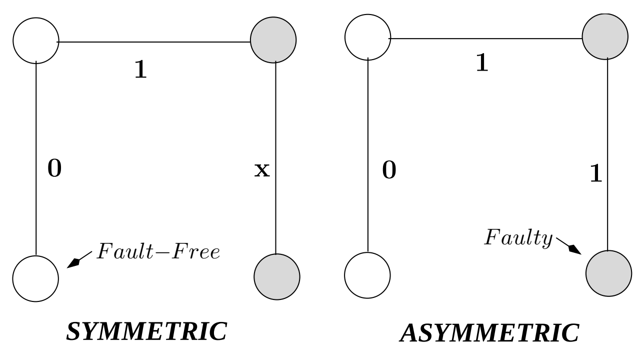

Fig. 1. Possible Comparison Outcomes.

Identifying the complete and correct set of faulty nodes using a comparison model has been shown to be NP-hard [5], but if the system is t-diagnosable, the problem is solvable in polynomial time. This problem has been extensively studied leading to elegant and efficient solutions [3], [15]. In this paper, we present a new diagnosis approach based on a modified Hopfield Neural Network (ModofiedHNN) for solving the system-level diagnosis problem under the symmetric comparison model. Hopfield neural networks (HNNs) have been shown to be able efficient in solving optimization problems [13]. They have been widely applied to various problems such as image restoration [17], channel allocation [14], and tumor boundary detection [24], to name a few. We believe that this new type of diagnosis approach will be useful in the design of future generation of dependable systems.

The remainder of this paper is organized as follow. We first provide general view of the fault and the symmetric comparison model in Section II. The modified Hopfield neural-network-based diagnosis approach is detailed in Section III. Simulation results are provided in Section IV. Section V discusses about related work. Finally, Section VI concludes the discussion and motivates future investigations on the system-level fault diagnosis problem.

II. Preliminaries

The system we consider is composed of N nodes that are interconnected with each other via a wired or wireless communication network. In comparison models, it is assumed that pairs of nodes are assigned the same task to be performed. The agreements (0) and disagreements (1) among the nodes are the basis for identifying the set of faulty nodes. The comparison diagnosis model can be described by two graphs, a *communication graph* and a *comparison* (or test) *graph*. The undirected communication graph $G = (V, E)$ represents the interconnection topology of the system (see example in Fig. 2 (A)). An undirected edge $e = (u, v)$ represents a communication link between the two nodes u and v. Whereas, the comparison graph shows the comparison tests that are performed in order to identify the set of faulty nodes once a faulty situation is detected, i.e., when the system deviates from its expected behavior due to faults in the nodes. An example of a comparison graph is provided in Fig. 2 (B). The set of all comparison outcomes is called the *syndrome*, and it is denoted by Ω. The set of all faulty nodes in the system is called the *fault set*. The actual fault set causing a faulty situation at a given point of time will be denoted by $\mathbb{F}$. We will refer to any comparison syndrome that can be generated under the fault set $\mathbb{F}$ by $\Omega_{\mathbb{F}}$. The objective of the fault identification algorithm is to identify $\mathbb{F}$ given $\Omega_{\mathbb{F}}$.

A. Fault Model

Faults can be classified based on their duration, their underlying cause, or on how a failed component behaves once it has failed [2]. Based on how a failed node behaves once it has failed, we could simply classify faults either as *hard* or *soft* [7]. A hard-faulted node is unable to communicate with the rest of the system, whereas a soft-faulted node can continue to operate and communicate with the other nodes in the system with altered behavior. Based on duration, faults can be classified either as *permanent*, *intermittent*, or *transient*. A transient fault will eventually disappear without any apparent intervention, whereas a permanent one will remain unless it is repaired and/or removed by some external administrator. A particularly problematic type of transient fault is the intermittent fault that recurs, often unpredictably. While it may seem that permanent faults are more severe, from an engineering perspective, they are much easier to diagnose and handle.

Once the system deviates from its normal behavior, a diagnosis algorithm is executed in order to determine which system's components caused this abnormal behavior. If faults are allowed to occur during the execution of the diagnosis algorithm, then the faults are assumed to be *dynamic*. Whereas, *static* faults are not assumed to occur during the diagnosis phase. Note that dynamic faults are hard to diagnose since a node may fail after it has been diagnosed as fault-free by other nodes.

In this work, we consider only the static permanent faults, i.e., software or hardware faults that always produce errors when they are fully exercised. However, we consider both hard and soft faults.

Definition 1: A system is *t-diagnosable* if each node can be correctly identified as fault-free or faulty based on a valid collection of comparison results, assuming that the number of faulty nodes does not exceed a given bound t.

The fault diagnosis process is based on the comparison syndrome output by the system's nodes. We consider only the deterministic diagnosis approach in which the input is a comparison syndrome and the output is the set of nodes diagnosed as faulty. In this paper, we consider the symmetric comparison model developed by Hakimi and Chwa in [12].

B. Symmetric Comparison Model

In the symmetric comparison model developed by Hakimi and Chwa [12] it is assumed that a central observer (comparator) is responsible of performing the comparisons between pairs of nodes by assigning them some tasks from the set of tasks $T = \{T_1, T_2, \ldots\}$. Each pair of nodes v_i and v_j is assigned a task $T_l \in T$. Once the task T_l is completed by both nodes, their results are compared. The comparison graph in this case, is an undirected graph $G = (V, C)$, where V denotes the set of nodes and $C = \{(v_i, v_j) : (v_i, v_j)$ is a pair of nodes performing the same task $T_l \in T\}$. From now on, we will denote a node pair (v_i, v_j) or (v_j, v_i) by c_{ij}. The result of the comparison test between the nodes v_i and v_j, a binary value, is associated with c_{ij}. This comparison result is 0 if the results generated by both nodes are identical; and it is 1, otherwise, i.e., if their results mismatch.

The outcome of a comparison test involving a pair of faulty nodes is unreliable (0 or 1). Both test outcomes are possible in this model, while in the asymmetric comparison model (Malek's model [16]) two faulty nodes always give mismatching outputs (see Fig. 1). Γ_i denotes the set of nodes with which a node $v_i \in V$, is compared, and is given by:

$$\Gamma_i = \{v_j : c_{ij} \in C\}$$

Ω^{ij} refers to the comparison outcome of the node pair c_{ij}.

Definition 2: A fault set $F \subset V$ is said to be *consistent* with a symmetric comparison syndrome Ω if for any $\Omega^{ij} \in \Omega$, such that v_i is fault-free, i.e., $v_i \in V - F$, $\Omega^{ij} = 1$ iff $v_j \in F$.

Definition 3: A comparison assignment graph $G(V, C)$ under the symmetric comparison model is a $D_\alpha(|V|)$ design iff for all $v_i \in V, |\Gamma_i| \geq \alpha$, i.e. each node is at least compared with α other nodes.

Systems belonging to this special design $D_\alpha(|V|)$ have been shown to be t-diagnosable in [18] and they can be easily generated.

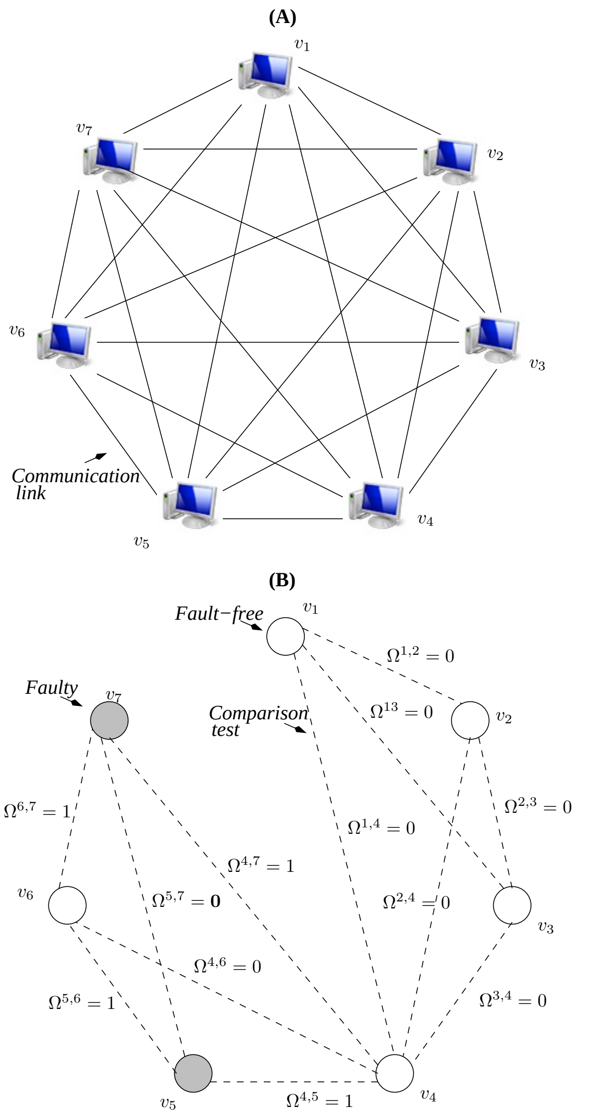

Fig. 2. A 2-Diagnosable Comparison-Based System: (A) Communication Graph. (B) A Comparison Assignment and a Symmetric Comparison Syndrome.

A small system connecting seven nodes is shown in Figure 2 (A). A typical comparison assignment is provided in Figure 2 (B), and a symmetric comparison syndrome corresponding to the actual fault set $\mathbb{F} = \{v_5, v_7\}$ is also given. Note that $\Omega^{5,7} = 0$ according to the symmetric invalidation rules. This example is a 2-diagnosable system [18].

III. Modified Hopfield Network for Fault Identification

The Hopfield neural network (HNN) assumes that all neurons are fully interconnected. The ith neuron is described by its state, which is denoted by V_i. The value of each state is determined by the total input from other neurons followed by a thresholding rule. The ith neuron's input is derived from: i) the outputs of other neurons scaled by the connection weights and ii) an appropriate external input. The total input to neuron i is denoted by S_i, and is given by

$$S_i = \sum_j w_{ij} V_j + I_i$$

where w_{ij} refers to the connection weight from neuron j to neuron i and I_i is the external input. Neurons' states are updated using either a discrete activation function with threshold θ_i as given by

$$V_i = \begin{cases} 1 & \text{if } S_i \geq \theta_i \\ 0 & \text{otherwise.} \end{cases}$$

or a continuous activation function as defined by

$$V_i = f(U_i) = \frac{1}{2}\left(1 + tanh\left(\frac{U_i}{\epsilon}\right)\right) \quad (1)$$

where U_i is the input signal and ϵ is a constant. The updating process stops when the states are unchanged or the energy has reached a minimum value. An energy function is defined for this network. For small values of ϵ, the energy function E is defined as in [13];

$$E = \frac{1}{2}\sum_i \sum_j w_{ij} V_i V_j - \sum_i V_i I_i$$

This energy function is minimized by the HNN's updating rule. Applying successively the updating rule will force the network to converge such that the energy of the network becomes smaller during the updating rule. Upon reaching a stable state, we can deduce that it has fallen into minimum energy state where this could be a local or global minimum. To adapt the HNN to any new application, w_{ij} and the I_i should be set appropriately so that E represents the function that needs to be minimized to solve the given optimization problem. The energy function should represent all the constraints of the problem.

A. *Applying Modified Hopfield Network to Diagnosis Problem*

In our algorithm, a continuous Hopfield network is developed, which is updated until a stopping criteria is met or a predefined maximum number of iterations has been reached. The modified Hopfield network for the comparison-based system level diagnosis problem is built of N neurons, where each neuron corresponds to a node in the system. The aim of the diagnosis problem is to label each node as *Faulty* (1) or *Fault-Free* (0), yielding hence a potential fault set F, while minimizing the discrepancy between the input syndrome $\Omega_{\mathbb{F}}$ and the syndrome Ω_F. An energy function is derived to represent this while at the same time taking into account any constraint a fault set must satisfy. Our energy function is given by: $E = E_1 + E_2 + E_3$, where

$$E_1 = A_1 \left(\sum_{i=1}^{N} V_i - t \right)^2 \tag{2}$$

$$E_2 = A_2 \sum_{i=1}^{N} \left(\sum_{j \in \Gamma_i} V_j - t \right)^2 \tag{3}$$

$$E_3 = A_3 \sum_{i=1}^{N} \sum_{j \in \Gamma_i} \left(1 - \alpha(V_i, V_j, \Omega_{\mathbb{F}}^{ij}) \right) \tag{4}$$

where E_1, E_2 and E_3 are the energy function components that correspond to specific constraints of t-diagnosable systems, and the function $\alpha(V_i, V_j, \Omega_{\mathbb{F}}^{ij})$ is defined as follows:

$$\alpha(V_i, V_j, \Omega_{\mathbb{F}}^{ij}) = \begin{cases} 0 & \text{if } V_i + V_j + \Omega_{\mathbb{F}}^{ij} = 1 \\ 1 & \text{otherwise.} \end{cases}$$

The term E1 ensures that the proposed solution will not have more than t nodes labeled as faulty. In fact, we are looking for the smallest fault set that is consistent with the input syndrome. From (2), we can easily deduce that a potential fault set with more than t faulty nodes will end up with a positive value for $E1$. While, fault sets with cardinalities smaller than t will result in a negative value of $E1$ decreasing hence the energy value. Smaller fault sets will be given more priority.

The second term $E2$ aims at avoiding fault sets where all the neighbors of any given node are faulty. In fact, if all neighbors of any node are faulty, then the system is not diagnosable. $E2$ is a more specialized version of $E1$ as it is applied only at the node's view, i.e., its neighbors.

Finally, the last term $E3$ which is the most important is related to the consistency of the potential fault set with the input syndrome $\Omega_{\mathbb{F}}$. To explain how we ended up with such a term, we need first to comment the key idea behind it. Table I shows all possible outcomes of the function $\alpha(V_i, V_j, \Omega_{\mathbb{F}}^{ij})$. For example, the first row of this table indicates that if $V_i = 0$, i.e., v_i is fault-free, and $V_j = 0$, i.e., v_j is fault-free, then the comparison outcome between node v_i and v_j will match with the input comparison outcome which is $\Omega_{\mathbb{F}}^{ij} = 0$. As a result, notice that $1 - \alpha(V_i, V_j, \Omega_{\mathbb{F}}^{ij}) = 0$ in this case. That is, it will not affect the energy term $E3$. However, consider now the second row, and follow the same reasoning. Since $V_i = 0$, i.e., fault-free, and $V_j = 0$, then the comparison outcome between node v_i and v_j should be 0, while this time the comparison outcome is $\Omega_{\mathbb{F}}^{ij} = 1$, and hence, there is a mismatch that needs to be avoided. Thus, this case and similar ones need to be avoided by having them affecting negatively the energy factor $E3$. All the three cases in rows indicated by ◄ should also be avoided as they contradict with the input syndrome. Intuitively, this means that if the comparison outcome $\Omega_{\mathbb{F}}^{ij} = 1$, then for sure at least one of the two nodes is faulty, and if the comparison outcome $\Omega_{\mathbb{F}}^{ij} = 0$, then for sure both compared nodes should be either fault-free, or faulty.

TABLE I
POSSIBLE OUTCOMES $\alpha(V_i, V_j, \Omega_{\mathbb{F}}^{ij})$.

V_i	V_j	$\Omega_{\mathbb{F}}^{ij}$	$\alpha(V_i, V_j, \Omega_{\mathbb{F}}^{ij})$
0	0	0	1
0	0	1	0 ◄
0	1	0	0 ◄
0	1	1	1
1	0	0	0 ◄
1	0	1	1
1	1	0	1
1	1	1	1

B. Modified Hopfield Diagnosis Algorithm

In this section, the implementation of the modified hopfield neural network-based (ModifiedHNN) diagnosis algorithm is discussed. Step 1) below explains the proposed initialization method, which uses specific characteristics of the problem. Step 2) presents the updating procedure after the initialization. There are N neurons in the network arranged in one-dimensional array. The network is fully connected as shown in Fig. 3. The most important task is finding an appropriate connection weight matrix. It is constructed taking into account the structure of the comparison graph. A consideration in this regard is, for example, no neuron should fire on another one if it is not its neighbor. An element in the weight matrix for a connection between two neurons i and j is computed as follows:

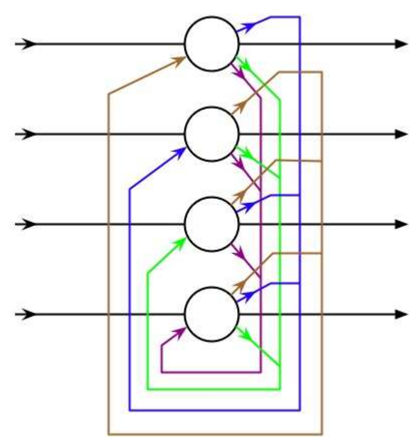

Fig. 3. Layout of the Hopfield Network for the Diagnosis Problem.

$$w_{ij} = -A_1(1 - \delta_{ij}) - A_2(1 - \beta_{ij}) - A_3 \tag{5}$$

δ_{ij} is the Kronecker delta function and defined with β_{ij} as follows:

$$\delta_{ij} = \begin{cases} 1 & \text{if } i = j \\ 0 & \text{otherwise.} \end{cases}$$

$$\beta_{ij} = \begin{cases} 0 & \text{if } j \in \Gamma_i \\ 1 & \text{otherwise.} \end{cases}$$

The overall diagnosis algorithm is summarized in the following steps.

i) The initial state of neurons is set to one or zero according to the chosen initialization method, and the weights are initialized using (5).
ii) Compute all outputs using (1).
iii) Repeat until a stopping criteria is met or after running a certain number of iterations
  a) Calculate the activations of all neurons in asynchronous way using the updating procedure as described below.
  b) Recompute all outputs using (1).
iv) Determine the fault set using the network's outputs as detailed below.

Step 1–Initialization: In general, a random initialization method is used as the initial states for the neurons in the Hopfield network. In our algorithm, all inputs have been initialized to 1. Other heuristics could be used such as initializing the states based on the probability of failure of the corresponding nodes. Another heuristic could be by assuming that two nodes are fault-free and then generating the states of the remaining ones by using the input syndrome. A problem may arise that the network get stuck at a local minimum. To avoid such an occurrence, random noise is added. Weights are initialized following (5) which mainly gives more weight to connections that involve nodes compared together.

Step 2–Updating Procedure: In an asynchronous Hopfield network, the neurons are selected randomly or sequentially by a certain order for the updating. In our algorithm, a sequential selection technique is used along with the following updating rule. We denote the activation of the ith neuron by a_i, and the output is denoted by o_i. The change in the activation is given by a_i^{t+1}, where

$$\begin{aligned} a_i^{t+1} &= a_i^t - \beta \left(\frac{a_i^t}{A} + Term_1 + Term_2 + Term_3 \right) \\ Term_1 &= A_1 \left(\sum_{i=1}^{N} o_i^t - t \right)^2 \\ Term_2 &= A_2 \sum_{i=1}^{N} \left(\sum_{j \in \Gamma_i} o_j^t - t \right)^2 \\ Term_3 &= A_3 \sum_{i=1}^{N} \sum_{j \in \Gamma_i} \left(1 - \alpha(o_i^t, o_j^t, \Omega_{\mathbb{F}}^{ij}) \right) \end{aligned}$$

The output of the ith neuron is calculated using (1) with $\epsilon = 1000$.

Step 3–Converting Network's Outputs to a Fault Set: To clearly explain how we extract the set of faulty nodes consider the neurons' outputs, sorted in an ascending order, shown in Table II for the comparison graph $D_5(10)$. Faulty nodes are pointed by ◀. First note that the Hopfield neural network was able to separate between the two classes: the faulty nodes and the fault-free ones. However, from the extensive simulations we have conducted we noticed that the boundary between the two classes is not all the time well defined. Hence, we adopted the following heuristic to be able to extract the set of faulty nodes by using the comparison graph and the input syndrome $\Omega_{\mathbb{F}}$.

TABLE II
NEURONS' OUTPUTS FOR A $D_5(10)$ SYMMETRIC COMPARISON GRAPH.

i	V_i
4	0.0996961 ◀
7	0.0996958 ◀
5	0.0997321 ◀
1	0.119535
3	0.115852
8	0.119291
9	0.115835
2	0.116041
6	0.119491
0	0.116068

The heuristic proceeds in the following steps.

i) Set position variable pos to end of the array.
ii) Label the node in position pos as fault-free and add it to the set $PendingFF$. In the provided example, the state of node v_9 will be fault-free, and the set $PendingFF = \{v_0\}$. Decrease value of pos by one.
iii) Repeat the following steps until all nodes are labeled either as fault-free or as faulty, or $pendingFF$ is empty.
  a) Consider $v_i \in PendingFF$ if not empty. For each v_i's neighbor, i.e. $v_j \in \Gamma_i$, if $\Omega_{\mathbb{F}}^{ij} = 0$ then label v_j as fault-free and added to $pendingFF$, else ($\Omega_{\mathbb{F}}^{ij} = 1$) and hence we need to label v_j as faulty.
  b) If $pendingFF =$, then goto step ii).

The described heuristic has been implemented and excessively tested as it will be shown in Section IV. In all scenarios this heuristic was successful in converting correctly the neurons' outputs to faulty or faut-free states.

IV. SIMULATION RESULTS

We have implemented the modified hopfield neural network-based (ModifiedHNN) diagnosis algorithm using C++, and we have performed extensive simulations using a PC equipped with an Intel Core 2 QUAD Q8300 CPU 2.5GHz and 4GB of RAM. All diagnosable comparison graphs the we have used have been generated randomly. In addition, faulty situations have been generated randomly and all possible fault sets that may occur in a t-diagnosable system have been simulated by varying the number of faults from 1 to t. We relied mainly on diagnosable comparison graphs from the special design $D_t(|V|)$, with $t \leq |V| - 2$, introduced in Definition 3 as it has been proven to be t-diagnosable in [18], and it can be easily generated even for large systems.

We will first start by showing results from a thorough simulation study to demonstrate the effectiveness of the new diagnosis approach. Then, a comparison with similar diagnosis approaches is provided.

A. Effectiveness of the ModefiedHNN-Based Fault Identification Algorithm

Extensive simulations have been conducted to check the efficiency of the new diagnosis algorithm. Various types of experiments have been run and the fault identification algorithm has been tested under all possible faulty situations. In the following, we summarize the outcomes of such experiments showing only the results for the symmetric comparison graph $D_9(20)$. We have also tested the new approach with various other comparison graphs ranging from small systems composed from few nodes to large systems composed of hundred of nodes. All simulations' results were similar to the ones provided below. Note that the maximum number of possible fault sets in a t-diagnosable system is bounded by $\sum_{i=1}^{t} C_i^n$, where $C_i^n = \frac{n!}{(n-i)!i!}$. For the $D_9(20)$ comparison graph, there is exist 1,046,528 possible faulty situations.

The first set of experiments aimed at checking if the ModifiedHNN-based diagnosis algorithm was correct, i.e., was able to identify the faulty nodes of any given faulty situation. To do so, we have created a hopfield neural network, as described in Section III, and we have tested it as follow. We have generated 100,000 random fault sets and their corresponding consistent syndromes, and we have input them to the ModifiedHNN. The ModifiedHNN-based diagnosis algorithm was able to identify almost all faulty nodes, yielding hence around 100% correctness.

The second set of experiments, quite similar to the the first one, had the objective of checking the effectiveness of the new diagnosis approach under various syndromes. We have hence randomly generated 1000 faulty situations, and for each one we have generated randomly 10,000 symmetric syndromes. In all these tested faulty situations the ModifiedHNN-based diagnosis algorithm was able to identify the almost all corresponding faulty nodes. That is, around 100% correctness.

Our third set of experiments involved this time all faulty situations that may occur in a t-diagnosable system. Note that we were able to check this only for small systems where we were able to generate all these possible fault sets. For larger systems we have adopted a different approach that will be described below. For the considered $D_9(20)$ diagnosable system we have tested it using all possible fault sets of cardinality ranging from 1 to 9. For each cardinality c, $1000c$ randomly generated symmetric syndromes have been tested. In almost all these tested faulty situations the ModifiedHNN-based fault identification algorithm was able to identify the corresponding faulty nodes. That is, almost 100% correctness.

The last set of experiments that we have conducted involved generating different comparison graphs ranging from small systems composed of tens of nodes to large systems composed of hundreds of nodes. The number of nodes, n, was varied from 10 to 1000 with different paces as follows. If $n \leq 100$ the pace was set to 10. But, for $100 < n \leq 1000$ the considered pace was 100. Each time a Hopfield neural network is created and tested using various comparison graphs. For each value of n, we have randomly generated $100n$ comparison graphs and tested them for $1000n$ times using randomly generated fault sets, and where the maximum number of faults was also random. Over almost all these extensive simulations, the diagnosis algorithm was able to determine the exact fault sets, providing us hence with around 100% correctness.

As matter of fact, we can conclude that the ModifiedHNN-based diagnosis algorithm is correct, i.e., it identifies all faulty nodes, and that Hopfield neural networks can be used to solve the system-level fault diagnosis problem adding hence an alternative to the existing diagnosis algorithms.

B. Performance Comparison With Similar Approaches

Artificial neural networks have been introduced to the system-level diagnosis problem in [8] where it has been shown that a simple perceptron neural network was able to identify faulty nodes under the asymmetric comparison model. It has been also shown in [8] that the perceptron neural network-based diagnosis algorithm has failed, i.e., correctness less then 100%, when tested with symmetric syndromes (see Fig. 4). The main reason was that the diagnosis problem under asymmetric comparison models is a separable problem as a fault set can only be identified by a unique syndrome. In fact, in asymmetric comparison models each fault set produces one consistent syndrome. While, under symmetric comparison models a fault set may result in many consistent syndromes since the comparison outcomes between two faulty nodes is unreliable and can be 0 or 1. Thus, the diagnosis problem under the symmetric model is a nonseparable one. In a subsequent work in [11] Elhadef et al. tried to improve the performance of the neural network approach by considering a multilayered network as it is known to be efficient for nonseparable problems. They developed a backpropagation neural network (BPNN) to diagnose faulty situations under the symmetric model. The BPNN-based diagnosis has been shown to efficiently diagnose faulty situations. The only criticism is that it has failed when the number of faulty nodes approaches the bound t as shown in Fig. 4. This has motivated the present work as Hopfield neural networks have been extensively used recently to solve nonlinear problems.

The ModifiedHNN-based diagnosis algorithm succeeded in diagnosing more faulty situations, hence, outperforming the BPNN-based diagnosis. The second advantage of the ModifiedHNN-based diagnosis algorithm is that it does not require a learning phase like the BPNN-based diagnosis. In addition, BPNN-based diagnosis relied on a postprocessing phase in order to escape local optima, while the ModifiedHNN-based diagnosis algorithm does not need such correcting process. However, it terms of diagnosis latency, i.e. time required to diagnose a faulty situation, the BPNN-based diagnosis was faster as it exploited the off-line learning phase to speedup the fault identification phase. Fig. 5 shows the average time taken to diagnose randomly generated fault sets. As one can easily deduce that both Perceptron-based and BPNN-based diagnosis are taking less time to diagnose a faulty situation thanks to the off-line learning phase. On the other hand, the

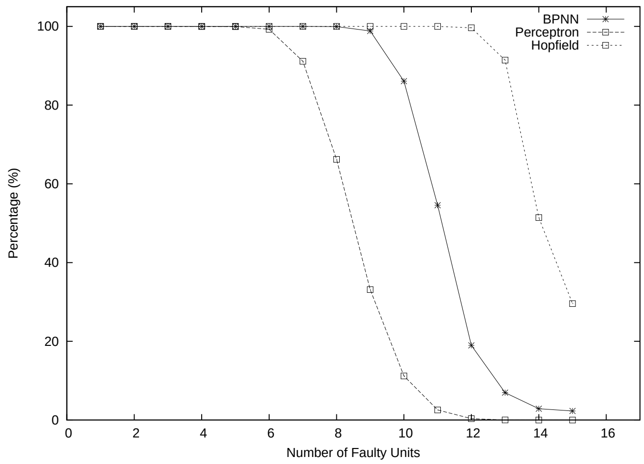

Fig. 4. Perceptron vs. Backpropagation vs. Hopfield Neural-Network-Based Diagnosis Approaches.

ModifiedHNN-based diagnosis is taking a little bit much more time to diagnose a faulty situation compared to the BPNN-based diagnosis, but overall its diagnosis latency is very low, i.e., around 16ms.

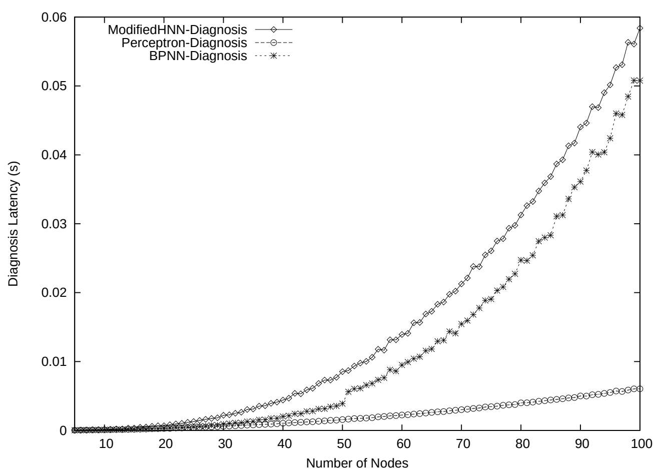

Fig. 5. Comparison of Diagnosis Latency.

We believe that both approaches will be useful. For systems known to be more reliable and where extreme faulty situations, involving many nodes, are rare and where faster diagnosis is required the BPNN-based diagnosis would be a perfect choice. While, for unreliable systems that can suffer from many simultaneous faults and where a learning phase cannot be conducted, the ModifiedHNN-based diagnosis would the best choice.

V. Related Work and Discussion

Identifying the correct set of all faulty nodes using the comparison approach has been shown to be NP-Hard [5], but if the system is t-diagnosable, the problem is solvable in polynomial time. This problem has been extensively studied leading to elegant and efficient solutions. In the following, t, V, and C denote, respectively, the maximum number of faults allowed in a system, the set of nodes, and the set of comparison tests. For their symmetric comparison model, Hakimi and Chwa developed an $O(|C|)$ diagnosis algorithm [12]. While, for the asymmetric comparison model, various fault identification algorithms have been proposed. In [1], Ammann and Cin proposed a $O(|V|^2)$ sequential diagnosis algorithm for a subset of t-diagnosable systems. Sengupta and Dahbura introduced next in [20] an $O(|V|^5)$ polynomial diagnosis algorithm for all t-diagnosable systems. Recently, Yang and Tang [23] developed a more efficient diagnosis algorithm which requires only $O(n\Delta^3\delta)$, where Δ and δ denote the maximum and minimum degrees of a node, respectively. In [5], Blough and Pelc studied the complexity of fault diagnosis under comparison models and they provided efficient algorithms for diagnosing systems for which the comparison assignment is a bipartite graph. A diagnosis algorithm has also been proposed, by Blough and Brown, for their broadcast comparison model which requires $O(|C| + t^2|V|)$ steps under asymmetric assumptions.

Other evolutionary approaches have been also used to solve the comparison-based fault diagnosis problem such as genetic algorithms [9].

Recently, in [7], Chessa and Santi presented a new comparison-based diagnostic model based on one-to-many communication paradigm which takes advantage of the shared nature of ad-hoc networks. They introduced a diagnosis protocol and two implementations of their model considering whether the network topology can change during diagnosis or not. Their work has been improved more recently in [10] using a more adaptable approach.

In this paper, we have solved the symmetric comparison-based diagnosis using the a modified Hopfield neural network. The new algorithm does not require any prior learning or knowledge about the system or about faulty situations, hence, providing better generalization performance. It can be considered as a viable addition to the other existing diagnosis algorithms [12], [1], [20], [5], [23].

VI. Conclusion

The modified Hopfield neural network-based (ModifiedHNN) diagnosis algorithm presented in this paper aims at solving the well known system-level diagnosis problem using the symmetric comparison model. The proposed approach adapted a Hopfield network to the diagnosis problem by profiting from the availability of the input syndrome to direct the neurons to the optimal solution. The results from an extensive simulation study have shown the efficiency of this novel approach in detecting all faulty situations, even under rare circumstances. That is, when extremely rare faulty situations, e.g., those where for example almost half of the system nodes fail at the same time, are simulated. We believe that the Hopfield-networks-based diagnosis approach is a viable addition to the existing diagnosis algorithms. In addition, we have shown that the novel approach scales very well for large diagnosable systems. Further experimental analysis and comparisons with existing solutions would be helpful in understanding the pros and cons of using artificial neural network systems in designing solutions to the system-level diagnosis problem.

As future investigations, we plan to apply the ModifiedHNN-based diagnosis to other diagnosis models, such as the PMC model [19], the generalized comparison model [4], and the probabilistic models [15]. It would be also interesting to experiment and analyze the use of alternative mechanisms, such as Support vector machines [21], for solving the system-level fault diagnosis problem. In addition, we are adapting the proposed solution to other types of faults such as dynamic faults [22] and intermittent faults [6].

References

[1] E. Ammann and M. D. Cin. Efficient Algorithms for Comparison-Based Self-Diagnosis. In *Self-Diagnosis and Fault-Tolerance*, pages 1–18, Werkhefte der Universität Tübingen, 4 At-tempo-Verlag, Tübingen, 1981.

[2] A. Avizienis, J.-C. Laprie, B. Randell, and C. Landwehr. Basic Concepts and Taxonomy of Dependable and Secure Computing. *IEEE Trans. Dependable Secur. Comput.*, 1(1):11–33, 2004.

[3] M. Barborak, M. Malek, and A. Dahbura. The Consensus Problem in Fault-Tolerant Computing. *ACM Computing Surveys*, 25(2):171–220, June 1993.

[4] D. Blough and H. Brown. The Broadcast Comparison Model for On-Line Fault Diagnosis in Multiprocessor Systems: Theory and Implementation. *IEEE Trans. on Computers*, 48(5):470–493, May 1999.

[5] D. Blough and A. Pelc. Complexity of Fault Diagnosis in Comparison Models. *IEEE Trans. on Computers*, 41(3):318–324, March 1992.

[6] D. Blough, G. Sullivan, and G. Masson. Intermittent Fault Diagnosis in Mutltiprocessor Systems. *IEEE Trans. on Computers*, 41:1430–1441, 1992.

[7] S. Chessa and P. Santi. Comparison-Based System-Level Fault Diagnosis in Ad Hoc Networks. In *Proc. of the 20th IEEE Symp. on Reliable Dist. Systems*, pages 257–266, 2001.

[8] M. Elhadef. A Perceptron Neural Network for Asymmetric Comparison-Based System-Level Fault Diagnosis. *CDROM of the 5th Int. Conf. on Availability, Reliability and Security (ARES 2009)*, Fukuoka, Japan, March 2009.

[9] M. Elhadef and B. Ayeb. Efficient Comparison-Based Fault Diagnosis of Multiprocessor Systems Using an Evolutionary Approach. In *proc. of the 15th Int. Parallel and Distributed Processing Symp. (IPDPS-2001)*, CA, USA, April 2001.

[10] M. Elhadef, A. Boukerche, , and H. Elkadiki. A Distributed Fault Identification Protocol for Mobile Ad-Hoc and Wireless Mesh Networks. *Journal of Parallel and Distributed Computing*, 68(3):321–335, Mar. 2008.

[11] M. Elhadef and A. Nayak. Efficient Symmetric Comparison-Based Self-Diagnosis Using Backpropagation Artificial Neural Networks. pages 264–271, *CDROM of the 28th IEEE nternational Performance Computing and Communications Conference (IPCCC 2010)*, Phoenix, Arizona, USA, Dec. 2010.

[12] S. Hakimi and K. Chwa. Schemes for Fault Tolerant Computing: A Comparison of Modularly Redundant and t-Diagnosable Systems. *Inform. Contr.*, 49:212–238, June 1981.

[13] J. Hopfield and D. Tank. "Neural" Computation of Decisions in Optimization Problems. *Biological Cybernetics*, 52:141–152, 1985.

[14] O. Lazaro and D. Girma. A Hopfield neural-network-based dynamic channel allocation with handoff channel reservation control. *IEEE Transactions on Vehicular Technology*, 49(5):1578–1587, 2000.

[15] S. Lee and K. Shin. Probabilistic Diagnosis of Multiprocessor Systems. *ACM Computing Surveys*, 26(1):121–139, March 1994.

[16] M. Malek. A Comparison Connection Assignment for Diagnosis of Multiprocessor Systems. In *Proc. 7th Int. Symp. on Comput. Architecture, New York*, pages 31–35. Association for Computing Machinery Publ., 1980.

[17] J. Paik and A. Katsaggelos. Image restoration using a modified Hopfield network. *IEEE Transactions on Image Processing*, 1(1):49–63, 1992.

[18] A. Pelc. Undirected Graph Models for System Level Fault Diagnosis. *IEEE Trans. on Electron. Comput.*, 40(11):1271–1276, Nov. 1991.

[19] F. Preparata, G. Metze, and R. Chien. On the Connection Assignment of Diagnosable Systems. *IEEE Trans. on Electron. Comput.*, 16(6), Dec. 1967.

[20] A. Sengupta and A. Dahbura. On Self-Diagnosable Multiprocessor Systems: Diagnosis by the Comparison Approach. *IEEE Trans. on Computers*, 41(11):1386–1395, Nov. 1992.

[21] J. Shawe-Taylor and N. Cristianini. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press, 2000.

[22] A. Subbiah and D. M. Blough. Distributed Diagnosis in Dynamic Fault Environments. *IEEE Trans. Parallel Distrib. Syst.*, 15(5):453–467, 2004.

[23] X. Yang and Y. Y. Tang. Efficient Fault Identification of Diagnosable Systems under the Comparison Model. *IEEE Trans. on Computers*, 56(12):1612–1618, Dec. 2007.

[24] Y. Zhu and Z. Yan. Computerized tumor boundary detection using a Hopfield neural network. *IEEE Transactions on Medical Imaging*, 16(1):55–67, 1997.