ZESZYTY NAUKOWE POLITECHNIKI ŚLĄSKIEJ 1998
Seria: INFORMATYKA z. 34 Nr kol. 1381

Tomasz DZIUBICH, Henryk KRAWCZYK
Politechnika Gdańska, Katedra Architektur Systemów Komputerowych

NEURODIA - STRATEGIA DIAGNOZY ROZPROSZONEJ WYKORZYSTUJĄCA SIECI NEURONOWE TYPU NEULONET

Streszczenie. W pracy przedstawiono nową strategię diagnozy rozproszonej dla sieci komputerowej. Decyzje diagnostyczne są podejmowane w oparciu o sieci neuronowe (zwane neulonetami) umieszczone w każdym węźle sieci. Dla pewnych błędów globalna diagnoza sieci może być poprawna, ale niekompletna. Dzięki właściwej metodzie uczenia neulonetu otrzymuje się jednak możliwie maksymalną diagnozowalność.

NEURODIA - DISTRIBUTED DIAGNOSIS STRATEGY USED NEULONET NEURAL NETWORK

Summary. The paper presents a new distributed diagnosis strategy of computer network. Diagnostic decisions are taken on the base of neuronal networks (called Neulonets) located in each node of the network. For same network failures, the global diagnosis can be correct, but incomplete. However, due to the proper learning method of the neulonets the highest diagnostic resolution is obtained.

1. Wprowadzenie

We współczesnych sieciach komputerowych istnieją aplikacje, w których konieczne jest ciągłe nadzorowanie sprzętu i oprogramowania pod względem poprawności ich funkcjonowania [3]. Służą do tego programy diagnostyczne, które pozwalają stwierdzić, czy system działa dobrze, czy też pewna jego część (lub całość) funkcjonuje wadliwie. Na podstawie wyników uzyskanych przez programy diagnostyczne odpowiednie algorytmy podejmują próbę oceny stanu systemu, tzn. które jednostki systemu funkcjonują poprawnie, a które błędnie. Istotną cechą takiego algorytmu jest to, aby podjęta decyzja - diagnoza była poprawna, tzn. żadna poprawna jednostka nie została rozpoznana jako błędna. Decyzja może być jednak nie-

kompletna, tzn. w jednym przebiegu algorytmu nie muszą zostać wykryte wszystkie błędne jednostki [5]. Dlatego też zaproponowano wykorzystanie sieci neuronowych do diagnozy błędnie funkcjonujących jednostek (węzłów) w sieci komputerowej.

Opisany poniżej algorytm oparty jest na sieci neuronowej Neulonet [2]. Jego celem jest oszacowanie stopnia zaufania (wiarygodności) do danej jednostki w sieci komputerowej. W oparciu o progi zadane przez administratora (użytkownika) i ciągle modyfikowane podczas pracy systemu można eliminować błędne jednostki z sieci lub alarmować o ich niepoprawnym działaniu.

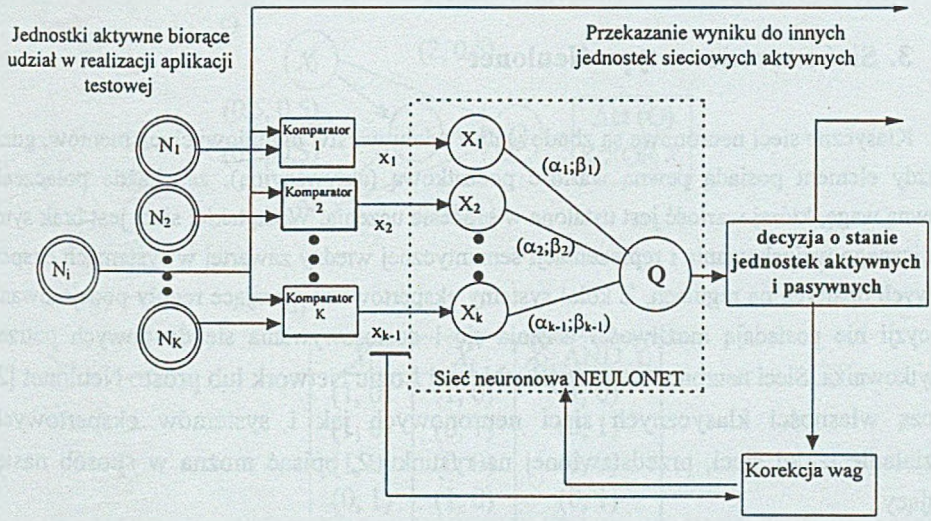
Przez wiarygodność jednostki należy rozumieć wartość liczbową (rzeczywistą R), odzwierciedlającą zaufanie do informacji otrzymanych z danej jednostki systemu rozproszonego. Im wartość ta jest większa, tym większy stopień zaufania do poprawnego funkcjonowania. Przy bardzo niskim stopniu zaufania jednostki informacje przesyłane z takiej jednostki do innych jednostek mogą być przez nie ignorowane. Najprostszą definicją **stopnia zaufania** może być stosunek wyników poprawnych do wszystkich otrzymanych wyników w zadanym przedziale czasu przez daną jednostkę. W pracy zaprezentowano zarówno budowę sieci Neulonet, jej odpowiednią modyfikację i algorytm korekcji wag określających aktualny stopień zaufania do informacji zawartej w danej jednostce sieci. Zaproponowano również algorytm diagnostyczny umożliwiający podejmowanie globalnych decyzji diagnostycznych z maksymalną wiarygodnością.

2. Model diagnostyczny węzła sieci komputerowej

Dowolna sieć komputerowa składa się z dwóch rodzajów urządzeń. Pierwsze to urządzenia przetwarzające (aktywne), mogące generować własne komunikaty, np. stacja robocza. Drugie - urządzenia pasywne, przekazujące tylko odebrane komunikaty, np. koncentrator pasywny.

Oznaczmy jednostki aktywne przez *N*, natomiast jednostki pasywne przez *P*. Przez węzeł sieci komputerowej należy rozumieć jednostkę aktywną *N*, która jest zdolna do wykonywania przygotowanych procedur testowych, zaimplementowanych w postaci oprogramowania lub umieszczonych w pamięci EPROM. Magistrale bądź koncentratory są przykładami jednostek pasywnych, których uszkodzenie może wpłynąć na stan funkcjonowania całej sieci. Jednak tylko jednostka aktywna jest w stanie podjąć decyzję zarówno o własnej poprawności funkcjonowania, jak i innych jednostek. Na rysunku 1 przedstawiono schemat funkcjonalny jednostki aktywnej (sieciowej), zamodelowany poprzez sieć typu Neulonet, która realizuje obliczenia dla danej aplikacji testowej. Do podstawowych elementów jednostki decyzyjnej zalicza się:

- blok testowania - wykonujący zadanie testowe i przekazujący jego wynik do wejść własnego zespołu komparatorów oraz do innych jednostek decyzyjnych; na rys. 1 mieści się on wewnątrz jednostki N_i,
- zespół komparatorów - porównujący wyniki otrzymane z innych jednostek decyzyjnych z własnym wynikiem otrzymanym z bloku testowania. Rezultat porównania wyników jest przekazywany na wejście sieci neuronowej,
- sieć neuronowa typu Neulonet - na podstawie wejść neuronów i ich wag podejmuje decyzję o prawidłowym wyniku diagnostycznym. Decyzja ta jest przekazywana do bloku korekcji wag,
- blok korekcji wag - w oparciu o prawidłowy rezultat następuje korekcja wag α i β opisujących gałąź neuronu.



Rys. 1. Model diagnostyczny i-tej jednostki aktywnej
Fig. 1. Diagnostic model of i-th active unit

Pracę jednostki decyzyjnej opisują funkcję podstawowych bloków: blok testowania i zespół komparatorów. Każda z K jednostek decyzyjnych zawiera blok testowania. Jest on odpowiedzialny za wyznaczanie identycznego zadania dla wszystkich jednostek oraz generowanie sygnatury wyniku, podobnie jak w metodzie porównywania zadań testowych [1], z tą różnicą że nie istnieje jeden wspólny dla wszystkich menedżer zadań, lecz każda z jednostek posiada taki własny blok funkcjonalny. Po wyznaczeniu sygnatury wyników testowania zostaje ona przesłana do wszystkich jednostek w sieci realizujących zadaną aplikację testową.

Po skompletowaniu K-1 sygnatur i wyznaczeniu własnej sygnatury przez i-tą jednostkę zostaje uruchomiony zespół komparatorów. Składa się on z K-1 pojedynczych komparatorów, z których każdy ma dwa wejścia i jedno wyjście. Na jedno z wejść podana jest sygnatura otrzymana z j-tej jednostki decyzyjnej, na drugie zaś zawsze z i-tej jednostki, czyli własnego

bloku testowania. Jeśli sygnatury są identyczne na wyjściu komparatora, otrzymujemy wartość 1, gdy zaś sygnatury są różne - wartość 0. Otrzymane rezultaty przekazywane są na wejścia sieci neuronowej i utrzymują się do momentu zadziałania zespołu korekcji wag. Zapamiętanie wyników porównania do momentu korygowania wag jest konieczne ze względu na późniejszą modyfikację wag gałęzi sieci neuronowej. Jeśli zostanie podjęta decyzja o poprawności wyniku „1", wagi α gałęzi neuronów o takim wejściu zostaną odpowiednio zwiększone (wzrost współczynnika zaufania), zaś wagi β gałęzi neuronów o wejściu przeciwnym („0") zostaną zmniejszone (brak zaufania).

Istotną rolę w tej diagnozie odgrywają sieć neuronowa i blok korekcji wag. Szczegółowe działanie sieci neuronowej typu Neulonet oraz bloku korekcji wag zostanie przedstawione w następnych rozdziałach.

3. Sieć neuronowa typu Neulonet

Klasyczne sieci neuronowe są zbudowane z wielu warstw nieliniowych elementów, gdzie każdy element posiada pewną wartość początkową (aktywacyjną), zaś każde połączenie pewną wagę, której wartość jest ustalona w procesie uczenia. Wadą takiej sieci jest brak symbolicznego wnioskowania i reprezentacji semantycznej wiedzy zawartej w systemach ekspertowych opartych na regułach. Z kolei systemy ekspertowe zawierające reguły podejmowania decyzji nie posiadają możliwości uczenia się i dostosowywania się do nowych potrzeb użytkownika. Sieci neuronowe zwane jako Neural Logic Network lub prosto Neulonet [2] łączą własności klasycznych sieci neuronowych jak i systemów ekspertowych. Działanie takiej sieci, przedstawionej na rysunku 2, opisać można w sposób następujący.

Niech Q będzie węzłem wyjściowym, a $X_1, X_2, \ldots, X_n$ węzłami wejściowymi. Wartości przyporządkowane węzłowi X_i oznaczmy przez $x_i = (a_i, b_i)$, a wagi połączeń występujących między węzłami X_i oraz Q przez (α_i, β_i). Każda para przyporządkowana do węzła może przyjąć jedną z trzech wartości (1,0) dla prawdy, (0,1) dla fałszu lub (0,0) dla „nie wiem"; (1,1) jest niezdefiniowane.

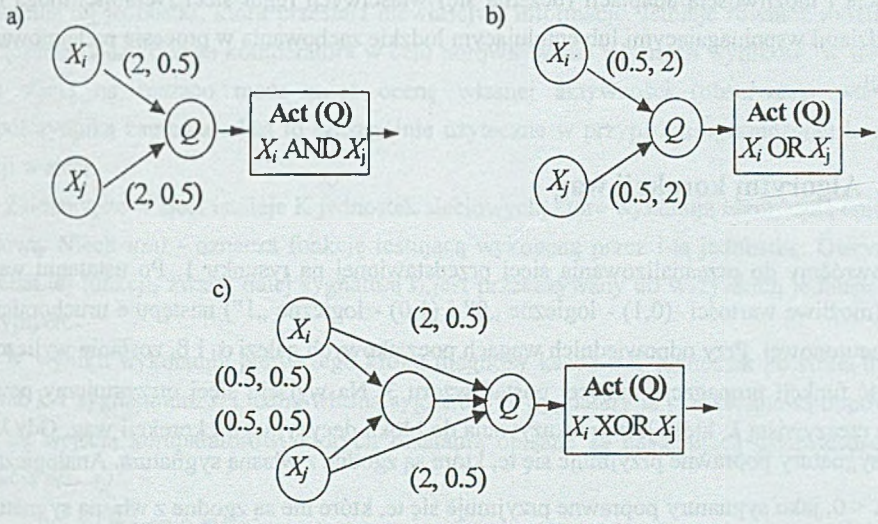
Wówczas funkcję propagacji określa się następująco:

$$\text{net}(Q) = \sum_{i=1}^{N} \left(a_i \alpha_i - b_i \beta_i\right) \tag{1}$$

Zaś wyjście z węzła Q jest opisane następującą funkcją aktywacji:

$$Act(Q) = \begin{cases} (1,\ 0) \text{ jeśli net(Q)} \geq \lambda \\ (0,\ 1) \text{ jeśli net(Q)} \leq -\lambda \\ (0,\ 0) \text{ w innym wypadku} \end{cases} \tag{2}$$

gdzie λ jest progiem zazwyczaj przyjmującym wartość 1.



d)

X_i	X_j	X_i AND X_j
(1, 0)	(1, 0)	(1, 0)
(1, 0)	(0, 1)	(0, 1)
(1, 0)	(0, 0)	(0, 0)
(0, 1)	(1, 0)	(0, 1)
(0, 1)	(0, 1)	(0, 1)
(0, 1)	(0, 0)	(0, 1)
(0, 0)	(1, 0)	(0, 0)
(0, 0)	(0, 1)	(0, 1)
(0, 0)	(0, 0)	(0, 0)

Rys. 2. Modele sieci realizujące operatory AND (a), OR (b) i XOR (c) oraz definicja operatora AND (d)
Fig 2. Nelonet network models realizing the following functions: AND (a), OR (b) and (c) XOR and AND operator definition (d)

Stosując powyższe funkcje możemy zbudować tablicę prawdy, ukazującą, iż zachowania sieci są podobne do regułowego systemu ekspertowego z logicznym operatorem AND (rys. 2a). Zmieniając ustawienia wag możemy skonstruować regułę OR lub regułę XOR przez wprowadzenie dodatkowego węzła. Sieci OR i XOR pokazano na rysunku 2b i 2c.

Każda z sieci AND, OR i XOR reprezentuje podstawową akcję podejmowania decyzji, zwaną regułą Neulonet. Reguły te mogą być zmieniane podobnie jak te w konwencjonalnych systemach ekspertowych opartych na regułach. Zdolności takich reguł sieciowych nie są ograniczone do konwencjonalnych operacji logicznych. Dzięki różnym ustawieniom wag reguły sieci mogą reprezentować różne typy podejmowanych decyzji. W połączeniu z równoległością i możliwością adaptacji (uczenia się) właściwych reguł sieci Neulonet mogą być narzędziami wspomagającymi lub emulującym ludzkie zachowania w procesie podejmowania decyzji.

4. Algorytm korekcji wag

Powróćmy do przeanalizowania sieci przedstawionej na rysunku 1. Po ustaleniu wejść sieci (możliwe wartości (0,1) - logiczne „0" i (1,0) - logiczne „1") następuje uruchomienie sieci neuronowej. Przy odpowiednich wagach początkowych gałęzi α_i i β_i zostanie wyliczona wartość funkcji propagacji, mającej postać wzoru 3. Na wyjściu sieci otrzymujemy pewną liczbę rzeczywistą λ, która jest przekazywana do bloku decyzyjnego i korekcji wag. Gdy $\lambda>0$, jako sygnatury poprawne przyjmuje się te, które są zgodne z własną sygnaturą. Analogicznie, gdy $\lambda < 0$, jako sygnatury poprawne przyjmuje się te, które nie są zgodne z własną sygnaturą. W przypadku gdy $\lambda = 0$, nie jest przeprowadzana korekcja wag.

Działanie sieci neuronowej różni się w przyjętym modelu od opracowania źródłowego [3] w dwóch aspektach. Po pierwsze zrezygnowano z możliwości podania na wejście sieci neuronowej sygnału „nie wiem", o symbolu (0,0). Jest to związane z zastosowaniem komparatora dającego wynik dyskretne ze zbioru dwuelementowego {0,1}. Stąd wynik komparatora o wartości „1" jest przekazany do sieci jako (1,0), zaś wynik o wartości „0" jako para (0,1). Drugą różnicą jest rezygnacja z funkcji progowej *Act(Q)* (wzór 2), której wartości należą do zbioru skończonego. Do oceny diagnozowalności bardziej odpowiednie są wartości rzeczywiste λ niż wartości zbioru skończonego {0,1}.

Wynik otrzymany na wyjściu sieci neuronowej oraz wyniki otrzymane na wyjściu komparatorów są przekazane do bloku korekcji wag. Blok ten stanowi sprzężenie zwrotne sieci neuronowej. Rezultat otrzymany w wyniku działania sieci neuronowej odzwierciedla stopień zaufania do sygnatur otrzymanych od innych jednostek decyzyjnych w odniesieniu do własnej sygnatury. Jeżeli wynik końcowy jest dodatni, oznacza to, że prawdopodobnie rezultatem poprawnym jest „1". Stąd wszystkie gałęzie, na których wejściu pojawiła się para (1,0), muszą mieć skorygowaną wagę - współczynnik zaufania α, natomiast dla gałęzi z wejściem (0,1) należy zmodyfikować wagę - współczynnik braku zaufania β. Analogiczna sytuacja ma miejsce, gdy wystąpi wynik ujemny. Wówczas rezultatem poprawnym jest „0". I tak wszystkie

gałęzie, na których wejściu pojawiła się (1,0), muszą mieć skorygowaną wagę β, natomiast dla gałęzi z wejściem (0,1) należy zmodyfikować wagę α. W przypadku otrzymania zera nie przeprowadza się korekcji wag.

Jeżeli po przeprowadzonej korekcji jeden ze współczynników α lub β przekroczy dopuszczalną wartość graniczną, to jednostka decyzyjna powinna podjąć działania w celu wykluczenia tej jednostki, która przesłała niewłaściwą informację. Istnieje również możliwość dołączenia dodatkowego komparatora w celu porównywania własnych wyników. W ten sposób stacja na bieżąco może mieć ocenę własnej aktywności (obserwacja własnego współczynnika zaufania). Jest to szczególnie użyteczne w przypadku ograniczonej komunikacji w sieci.

Załóżmy, że w sieci istnieje K jednostek sieciowych, które wykonują identyczną aplikację testową. Niech *t(m)* - oznacza funkcję testującą wykonaną przez i-tą jednostkę. Otrzymany rezultat tej funkcji, zwany dalej sygnaturą s_i, jest przekazywany do wszystkich jednostek decyzyjnych.

W wyniku wykonania pierwszego kroku diagnozy każda z K jednostek powinna dysponować k-1 sygnaturami s_j i jedną własną sygnaturą s_i. Sygnatury te przekazane są odpowiednio na wejścia komparatorów, których działania opisane są następującą dwuwartościową funkcją $p(s_i, s_j)$:

$$p(s_i, s_j) = 1 \Leftrightarrow s_i = s_j$$

$$p(s_i, s_j) = 0 \Leftrightarrow s_i \neq s_j$$

Oznaczmy jako x_i wynik otrzymany na wyjściu i-tego komparatora.

$$x_i = (1,0) \Leftrightarrow p(s_i, s_j) = 1$$

$$x_i = (0,1) \Leftrightarrow p(s_i, s_j) = 0$$

X_i jest wartością wejściową dla i-tego neuronu w sieci neuronowej. Odpowiednie wagi gałęzi każdego neuronu wynoszą α_i i β_i. Działanie sieci neuronowej, a zarazem jej wyjście jest opisane następującą funkcją propagacji (porównaj z wzorem 1),

$$f(x_1, x_2, .. x_{k-1}) = \sum_{i=1}^{K} (\alpha a_i + \beta b_i) \tag{3}$$

gdzie a_i i b_i są składowymi danej wejściowej $x_i = (a_i, b_i)$.

Wartość funkcji propagacji $f(x_1, x_2, ..., x_{K-1})$ oznaczmy przez λ. Wartość ta zostaje przekazana do bloku korekcji wag, który podejmuje decyzję opisaną za pomocą następującego wzoru:

$$\begin{aligned} &\lambda > 0 \Rightarrow \text{jeśli } x_i = (0,1)\ \beta_{i(n+1)} = g(t) * \beta_{i(n)} \quad \text{jeśli } x_i = (1,0)\ \alpha_{i(n+1)} = f(t) * \alpha_{i(n)} \\ &\lambda = 0 \Rightarrow \beta_{i(n+1)} = \beta_{i(n)} \quad \alpha_{i(n+1)} = \alpha_{i(n)} \\ &\lambda < 0 \Rightarrow \text{jeśli } x_i = (1,0)\ \beta_{i(n+1)} = f(t) * \beta_{i(n)} \quad \text{jeśli } x_i = (0,1)\ \alpha_{i(n+1)} = g(t) * \alpha_{i(n)} \end{aligned} \tag{4}$$

gdzie f(t) i g(t) są funkcjami czasowymi korygującymi wagi.

Łatwo zauważyć, że jednym z podstawowych problemów jest właściwy dobór wag gałęzi neuronu α i β. O ile wartości początkowe, intuicyjnie rzecz biorąc, powinny być liczbami przeciwnymi, równymi co do wartości bezwzględnej, tzn. α = -β, o tyle zmiana tych wag powinna być podczas funkcjonowania sieci na tyle szybka, aby zareagować jak najwcześniej na nieprawidłowości działania sieci komputerowej, a jednocześnie na tyle wolna, aby nie powodować „fałszywych alarmów". Stąd też proponuje się funkcje f(t) i g(t) spełniające następujące warunki:

$$\lim_{t\to\infty} f(t) = 1 \quad \text{i} \quad \lim_{t\to\infty} g(t) = -1$$

$$\frac{df(t)}{dt} << \frac{dg(t)}{dt} \tag{5}$$

Zapewnią one powolny wzrost zaufania (α rośnie powoli do 1) i gwałtowny spadek zaufania (β maleje szybko do -1). Przykładowe funkcje spełniające te zależności to

$$f(t) = 1 - \frac{1}{2e^t} \qquad g(t) = -\frac{1}{t} \tag{6}$$

Podjęta decyzja przy właściwych wagach α i β jest poprawna tylko w pojedynczym kroku diagnozy. Następna decyzja zależy od właściwej korekcji wag. Reakcja i ustalenie wag gałęzi w kolejnym kroku zależy od funkcji f(t) i g(t). Jeśli funkcje te będą niepoprawne (źle dobrane), otrzyma się w następnych krokach niepoprawne decyzje. Może być tutaj również pośrednio wykorzystana reguła Bayesa czy typowy regulator dwupołożeniowy.

5. Wnioski końcowe

W oparciu o przedstawiony schemat funkcjonalny sieci można przedstawić następujący globalny algorytm diagnostyczny. Sprowadza się on do przedstawienia zachowania się i-tej jednostki:

1. i-ta jednostka przeprowadza samotestowanie lub testowanie lokalne dotyczące wybranych węzłów sieci,
2. wyniki obliczeń przekazuje do wszystkich ostatnio poprawnie działających jednostek,
3. oczekuje na decyzje diagnostyczną podejmowaną przez te jednostki przez czas Δt,
4. dokonuje oceny poprawności diagnozy wybranych bądź wszystkich jednostek sieciowych,
5. realizuje korekcja wag sieci neuronowej dla następnego kroku testowania,
6. powyższe kroki są wykonywane w równych bądź ściśle określonych odstępach czasu.

W powyższym algorytmie można wyróżnić trzy fazy: testowanie lokalne (1), przekazanie i odbiór wyników pośrednich (2 i 3), ustalenie wyniku ostatecznego i korekcja wag (4 i 5). Algorytm taki adaptuje się do aktualnych warunków istniejących w sieci komputerowej. Jego

zaletą jest przeprowadzenie jedynie lokalnej diagnozy dla pewnego zbioru jednostek systemu, w którym występować mogą zarówno jednostki sprawne, jak i uszkodzone. Każda jednostka sprawna nie podejmuje jednoznacznej decyzji w jednym cyklu diagnostycznym, a stara się obserwować „jednostki podejrzane" przez określony czas, by powoli przekonać się o ich rzeczywistym stanie. Takie podejście jest istotne w systemach samorekonfigurowalnych, gdzie jednostki uszkodzone mogą być wyłączone lub wymienione na sprawne.

Sprawą istotną jest dobór funkcji f(t) i g(t). Funkcje te powinny być wnikliwie przeanalizowane, bowiem decydują o jakości realizowanej diagnostyki sieci komputerowej. W pracy przedstawiono wykorzystanie sieci Neulonet do diagnozy porównawczej. Może być ona również wykorzystana w algorytmicznej metodzie tolerowania błędów (algorithm-based fault tolerance [4]), gdzie odpowiednio buduje się dane wejściowe i wyjściowe oraz tak modyfikuje się algorytm przetwarzania, by w prosty sposób wykryć pojawiające się błędy. Sieci Neulonet mogą być wykorzystane do sprawdzenia, czy dla prawidłowo zakodowanych danych wejściowych uzyskano odpowiednio zakodowane wyniki obliczeń.

LITERATURA

[1] Dziubich T., Krawczyk H.: Implementacja diagnostyki rozproszonej w sieci komputerowej. ZN Pol. Śląskiej s. Informatyka, Gliwice 1997.

[2] Krawczyk H.: Dependable Processing. TEMPUS JEP 3470 PG, Gdańsk 1996.

[3] Tan C.L., Quah T.S.: An Artifical Neural Network that Models Human Decision Making. IEEE Computer 1996, nr 3, s. 64-70.

[4] Tannenbaum A.: Sieci komputerowe. WNT, Warszawa 1991.

[5] Yannik S.: Analysis and Randomized Design of Algorithm-Based Fault Tolerance Multiprocesor System under an Extended Model. IEEE Transaction on Parellel and Distributed System, 1997, t. 8, nr 7, s. 757-768.

Recenzent: Prof. dr hab. inż Andrzej Grzywak

Wpłynęło do Redakcji 15 grudnia 1997 r.

Abstract

The paper presents a new strategy of distributed diagnosis for computer networks. This strategy consists of the following steps:

1. A computer network node is tested by comparision its results with results obtained by other nodes executing the same task.

2. The obtained outcomes are dessiminated to all nodes.

3. The nodes take diagnostic decisions autonomously and determine the global state of the computer subnetwork.

Points 1, 2, 3 can be repeated for finding the consistent state of the whole computer network.

We propose the diagnostic model of i-th nodes (see Fig. 1) as neuronal network called Neulonet. This kind of the neuronal network combains properties of classical neuronal networks and expert systems. Such a neulonet is located in each computer network node and allows to take diagnostic decision about both the state of the node and the whole computer network. In the paper we describe how the neulonet works (Fig. 2 and formulas 1 and 2) we propose the modification of its functions and describe how to correct the weights (formulas 3, 4, 5, 6) to obtain the required diagnosability. For same network failures, the global diagnosis can be correct, but incomplete. However, due to the proper learning method of the neulonets the highest diagnostic resolution is obtained.

The advantage of this approach is hardware implementation of computer network diagnosis which improve system performance. Besides, the proposed method can be used for other kinds of testing, for instance for algorithm-based fault tolerance, where neulonets can check if for acceptable input data we obtain correct outcomes.