

IDENTIFICARE DE PROCES PRIN TEHNICI NEURALE

prof.dr.ing. Dumitru Popescu

as. ing. Monica Alexandru

Universitatea Politehnică București

ing. E. Naum

Universitatea Tehnică "Gh. Asachi", Iași

Rezumat: Lucrarea își propune să abordeze problema identificării proceselor industriale prin tehnica modernă a rețelelor neurale. Articolul prezintă, în prima parte, metodologia de identificare a unui proces integrat în mediul industrial cu toate precauțiile ce trebuie luate pentru a nu perturba regimul nominal de funcționare a sistemului. În cea de a doua parte, este descris algoritmul de identificare prin rețele neurale și sunt analizate rezultatele obținute. În final, se anticipează performanțele ce pot rezulta prin aplicarea metodelor neconvenționale în domeniul controlului industrial.

Cuvinte cheie: rețele neurale, identificare de proces, model neliniar, control predictiv, algoritmul "backpropagation".

1. Introducere

De cele mai multe ori, pentru a explica comportarea intrare-ieșire a variabilelor unui proces industrial este necesară cunoașterea unui model calitativ al procesului sau o determinare precisă a acestuia. Un model de proces este o legătură funcțională între variabile, care explică relațiile cauză-efect între intrări și ieșiri. Identificarea de proces constă în determinarea acestui model și este prioritar cerută în cazul unui control avansat în care modelul rezultat este folosit direct în faza de proiectare a regulatorului.

Modelul procesului poate fi obținut pe baza legilor fundamentale ale fizicii, cum ar fi spre exemplu legea conservării masei, energiei, momentului etc. Un astfel de model este capabil să explice mecanismul fizic al procesului și este numit model fenomenologic. Un dezavantaj al acestei abordări de identificare este că, adeseori, ea conduce la obținerea de modele neliniare complexe pentru a căror folosire în etapa de proiectare a regulatorului se impun simplificări structurale și parametrice de model care afectează dezavantajos, mai mult sau mai puțin, performanțele în buclă închisă, impuse de tehnolog. Din acest punct de vedere, în multe situații practice industriale, o interpretare suficientă a fenomenelor fizice existente nu poate fi formulată și, prin urmare, modelul fenomenologic sau nu poate fi obținut sau este o aproximare imprecisă a problemei.

O altă metodă în identificarea practică de proces este abordarea "black-box", în care modelul este obținut exclusiv din date experimentale. Modelele "black-box", spre deosebire de cele fenomenologice, nu furnizează informații detaliate despre fizica procesului, ci descriu comportamentul dinamic intrare-ieșire. Pentru o reglare

satisfăcătoare, este suficientă informația dinamică sub forma unui model "black-box".

Există însă aplicații industriale pentru care cele două abordări amintite nu soluționează problema identificării. Este cazul proceselor industriale, care manifestă un comportament neliniar semnificativ. Dacă controlul acestor procese se realizează prin implementarea regulatorilor adaptive, care au în structura lor un model liniar al procesului, cel mult variant în parametri, rezultatele nu vor fi cel predictate datorită limitărilor inerente de modelare. Altfel spus, modelul liniar utilizat este o aproximare mai mult sau mai puțin riguroasă a procesului neliniar. Procesul tehnologic neliniar "condus" de un regulator bazat pe un model "aproximație liniară", va reacționa în limitele dorite din punctul de vedere al stabilității și al performanțelor cerute numai dacă în faza de proiectare a regulatorului au fost prevăzute filtre de robustețe.

Revenind la procesul de identificare a sistemelor neliniare, o alternativă la modelarea matematică ar fi modelarea neurală. Noua tehnică de modelare prin rețele neuronale (neural networks modelling) și-a demonstrat capacitatea de a reprezenta eficient relații neliniare, existente între variabilele vizate ale unui proces.

2. Rețele neurale

O rețea neurală (RN) este o structură computațională, capabilă să modeleze o relație funcțională neliniară între variabilele de intrare și variabilele de ieșire. Există un număr disponibil de configurații de rețele și un număr de algoritmi de antrenare pentru fiecare configurație. Rețeaua constă dintr-un număr de elemente de calcul numite neuroni, care sunt dispuși în mai multe straturi într-o arhitectură relațională. Primul strat este cel de intrare, care primește semnalele variabilelor de intrare. Ultimul strat este cel de ieșire, care furnizează variabilele de ieșire predictate. Straturile intermediare se numesc straturi ascunse, iar numărul lor variază de la o aplicație la alta.

În majoritatea aplicațiilor, s-a constatat că este satisfăcătoare o configurație de rețea cu un singur strat ascuns. Fiecare neuron din stratul de intrare este conectat la toți neuronii din stratul ascuns și, în mod similar, toți neuronii din stratul ascuns sunt conectați la toți neuronii din stratul de ieșire.

Fiecare dintre aceste conexiuni este caracterizată de un parametru numit "pondere de conexiune" sau, mai simplu, pondere (weight). În afara neuronilor conectați total (în sensul că prezintă legături cu stratul anterior și cu cel următor) mai există și neuroni "bias", aflați în stratul de intrare și în cele intermediare, câte unul pe fiecare nivel. Acești neuroni prezintă conexiuni numai cu stratul următor, iar semnalul de activare este întotdeauna unitar.

Primul pas în modelarea cu rețele neuronale este antrenarea rețelei printr-un set de date (eșantioane). Rețeaua "învață", adaptându-și ponderile pe baza setului de date de antrenament. Învațarea sau antrenarea rețelei se realizează prin algoritmi dedicați, cum ar fi, spre exemplu, algoritmul de propagare înapoi a erorii de predicție (backpropagation algorithm, BPA), algoritmi de gradienti conjugați etc. Principiul algoritmilor de antrenare este de a modifica ponderile rețelei în

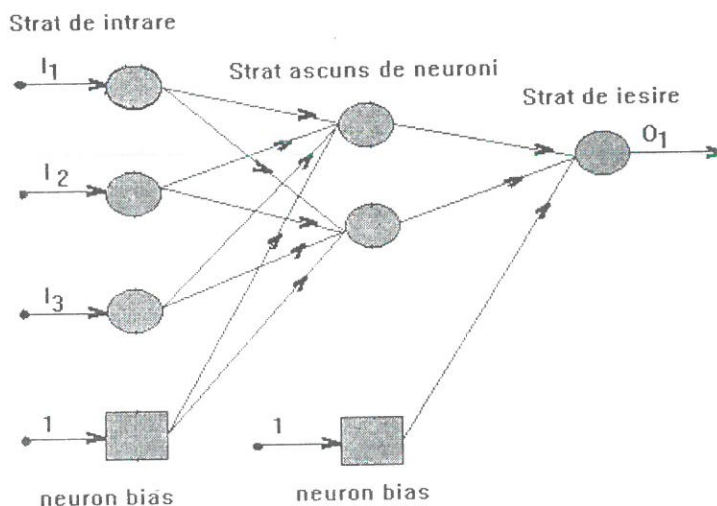


Figura 1 Structura generală de rețea feedforward cu trei straturi

Neuronilor prezenți în stratul ascuns și în cel de ieșire, le sunt asociați funcții de calcul continue și diferențiale precum funcția sigmoidă sau funcția tangent hiperbolică, cunoscute ca funcții de activare. Neuronii stratului de intrare au numai rolul de distribuire a semnalelor aplicate la intrarea rețelei către nodurile computaționale. Tuturor nodurilor din straturile ascunse și din cel de ieșire, le este asociat același tip de funcție de activare. Dacă I_i este intrarea în al i -lea neuron din stratul anterior, w_i este ponderea asociată acestei conexiuni și β este ponderea asociată conexiunii cu neuronul de bias, atunci ieșirea, O_j a neuronului din stratul curent, este dată de:

$$S = \sum (w_i I_i) + \beta, \quad (1)$$

$$O_j = \frac{1}{1 + \exp(-S)} \quad (2)$$

Capacitatea rețelei de a simula un comportament funcțional complex, se datorează unor astfel de funcții de activare și nu mai puțin unei arhitecturi de interconectare a semnalelor de activare a cărei complexitate diferă de la o aplicație la alta.

vederea minimizării unui criteriu pătratic de forma:

$$\min J = \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^n (\tilde{y}(t) - y(t))^2 \quad (3)$$

unde l este dimensiunea setului de antrenare și n reprezintă în numărul neuronilor ultimului strat.

Astfel, o RN poate fi văzută ca o funcție neliniară:

$$Y = f_{RN}(X)$$

unde X și Y reprezintă vectorii de intrare, respectiv de ieșire.

3. Procedura de identificare

Identificarea unui model din date experimentale intrare-ieșire implică următoarele etape:

1. Achiziționarea datelor intrare-ieșire
2. Prelucrarea datelor achiziționate
3. Alegerea arhitecturii de rețea

4. Estimarea parametrilor
5. Validarea modelului

1. Achiziționarea datelor intrare-ieșire

În funcție de tipul testelor ce vor opera ulterior cu datele din proces, organizarea operației de achiziție este o etapă importantă. Pe durata achiziției pot surveni schimbări în tipul și dimensiunea intrărilor, domeniul de operare, ordinea de comutare a canalelor de achiziție multiplă (în cazul unui vector de intrare). Procesul de colectare a datelor se realizează în mod uzual pe perioada funcționării instalației, fapt pentru care trebuie să se țină cont ca regimul de lucru normal al procesului să nu fie deplasat în afara limitelor admise de toleranță pentru a nu se deteriora calitatea produsului. În termeni practici, aceasta s-ar traduce printr-o variație limitată a amplificării variabilelor de intrare. În același timp, pentru un experiment de identificare reușit, intrările trebuie să aibă un profil pseudoaleator, adică să fie bogate în informație dinamică (semnalul de excitație să conțină un spectru larg de frecvență). Datorită acestui conflict de obiective, etapa de colectare a datelor este un stagiul crucial în procesul de identificare. Dacă ieșirea procesului depășește limitele admise de deviație față de valoarea sa nominală, atunci operatorul trebuie să introducă schimbări compensatorii în mărimile de intrare pentru a readuce ieșirile în banda admisă. Un număr exagerat de operații compensatorii poate introduce corelații nedorite între mărimile de intrare.

2. Prelucrarea datelor achiziționate

Datele direct colectate nu pot fi folosite în această formă în studiul identificării. Ele trebuie să fie filtrate pentru a îndepărta zgomotul nemăsurabil, prezent întotdeauna în aplicațiile de timp real. Blocurile de date, colectate pe perioada când în proces au survenit defecțiuni, vor fi îndepărtate din set pentru a nu conduce la identificarea unui model inadecvat.

3. Arhitectura de rețea

În sistemele digitale de control, un proces multivariabil poate fi reprezentat prin:

$$Y_k = f(Y_{k-1}, Y_{k-2}, Y_{k-3}, \dots, Y_{k-m})$$

$$X_{k-1}, X_{k-2}, X_{k-3}, \dots, X_{k-n}$$

unde Y_k și X_k reprezintă vectorii de ieșire și, respectiv, intrare. Indicele k desemnează momentele de eșantionare sau timpul discret, iar f semnifică relația funcțională între intrări și ieșiri. Procesul de identificare prin metode clasice are ca rezultat

forma funcției de transfer f prin specificarea ordinului sistemului și prin determinarea parametrilor săi (coeficienții numitorului și ai numărătorului funcției de transfer). Procedura de identificare clasică este adecvată proceselor liniare sau celor neliniare ale căror neliniarități pot fi depășite prin adoptarea unor modele liniare astfel încât comportarea în buclă închisă să fie stabilă și performantă. În cel mai des întâlnite situații, caracterul neliniar este predominant și identificarea procesului nu mai poate fi evaluată ca o identificare de funcție de transfer, neputându-se specifica, în acest caz, o relație funcțională intrare-ieșire printr-un model matematic (transformata Laplace sau transformata Z) accesibil prelucrărilor ulterioare. În acest cadru, tehnicile de identificare cu rețele neurale și-au dovedit eficacitatea. Metoda nu necesită ca funcția de transfer să fie precizată, însă trebuie specificată topologia rețelei prin numărul de neuroni al fiecărui strat. Numărul de noduri din nivelurile de intrare și de ieșire este fixat de intrările, respectiv ieșirile procesului real. Vectorul de intrare al rețelei neurale conține valorile intrărilor și ale ieșirilor instalației la momentele trecute de eșantionare. Numărul de "valori trecute" ale intrărilor folosite în vectorul de activare (intrare) al rețelei este precizat de timpul mort al procesului (multiplu al perioadei de eșantionare), iar cel al ieșirilor este stabilit în funcție de ordinul procesului (se presupune o cunoaștere a priori aproximativă a ordinului și a timpului mort). Numărul de noduri ale stratului ascuns se determină prin încercări și experiență.

3.4. Antrenarea rețelei și validarea modelului

În scopul modelării, datele experimentale de tip intrare-ieșire sunt divizate în două seturi: setul de antrenare și setul de validare al arhitecturii de rețea. Datele setului de antrenare alimentează rețeaua pe perioada învățării și ponderile sunt în mod continuu adaptate în scopul minimizării erorii de predicție din ecuația (3). La sfârșitul unei iterații (epoci), o iterație însemnând o parcurgere integrală a setului de date de antrenare, rețeaua este capabilă să prediceze în limitele unei marje de eroare ieșirile setului de validare, menținându-și ponderile la valorile setate în finalul antrenării. Dacă erorile de predicție sunt mai mari decât toleranța impusă, urmează o nouă epocă de antrenare pentru rețea la sfârșitul căreia testul se reia pe setul de validare. Acest proces iterativ, antrenare-testare, continuă până când, fie numărul epocilor de antrenare este atins, fie erorile de predicție ale setului de validare intră în banda de toleranță. Identificarea procesului luat în considerare se încheie la finalul procedurii iterative, modelul instalației fiind caracterizat de arhitectura rețelei și de ponderile conexiunilor.

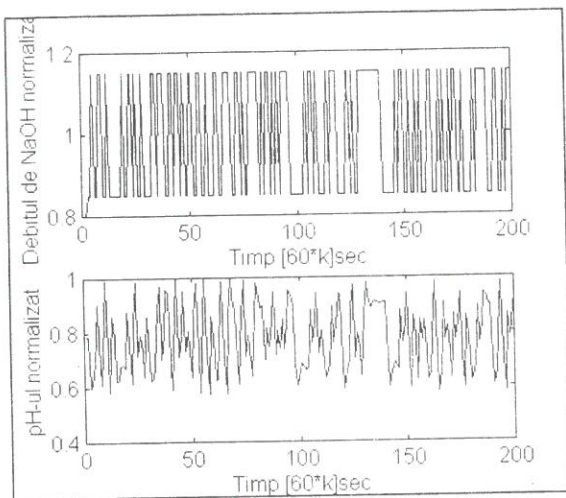


Figura 2. Baza de date de antrenare și validare

3.5. Problema industrială

În cele ce urmează, se va încerca determinarea unui model de tip rețea neurală (model RN) pe baza unor date experimentale în scopul utilizării lui ulterioare în procedura de control al nivelului de aciditate, înregistrat într-un reactor. Presupunând un debit constant de acid acetic, se variaza debitul de intrare de hidroxid de sodiu (NaOH) pentru a neutraliza acidul cu un pH impus. Tehnica utilizată este controlul predictiv, bazat pe model, care presupune dezvoltarea unui model al instalației (fie liniar, fie neliniar, în funcție de complexitatea procesului) care va constitui apoi obiectul unei rutine de optimizare. Modelul dinamic al procesului va fi generat de către o rețea cu trei straturi, având patru neuroni de intrare, șapte pe stratul ascuns și unul singur pe stratul de ieșire. La fiecare moment de timp discret, vectorul de intrare $X(t)$ are patru componente corespunzătoare lui $pH(k-2)$, $pH(k-1)$, $F(k-2)$, $F(k-1)$ unde cu $pH(k)$ s-a notat nivelul de aciditate măsurat la momentul actual în rezervor, iar cu $F(k)$ s-a notat debitul de hidroxid de sodiu, măsurat la același moment de timp. Ieșirea $y(k)$ furnizată de neuronul de ieșire al rețelei corespunde unei predicții $pH(k)$.

În scopul estimării valorilor optime ale ponderilor rețelei, s-a construit mai întâi o bază de date din valorile $X(k)$ și $pH(k)$ pentru diferite valori ale timpului discret. Pentru generarea acesteia, un semnal pseudoaleator a fost suprapus peste componenta continuă a semnalului de comandă corespunzătoare regimului staționar, iar ecuațiile de echilibru acid-bază împreună cu ecuația de electroneutralitate au condus la o serie de timp a

valorii pH-ului. Seriile de timp generate pentru $F(k)$, respectiv $pH(k)$ sunt prezentate în figura 2.

Pentru adaptarea ponderilor, s-a utilizat în a doua etapă algoritmul "backpropagation", care minimizează eroarea de predicție pe întregul set de date.

Algoritmul implementat constă în următorii pași:

1. Calculul valorii predictate sau pasul "forward":

$$v_j^- = w_{j0}^- + \sum_{k=1}^4 w_{jk}^- X_k(t), \quad 1 \leq j \leq 7 \quad (4)$$

$$x_j^- = s(v_j^-) = 1 / (1 + e^{-v_j^-}), \quad 1 \leq j \leq 7 \quad (5)$$

$$v_1^+ = w_{10}^+ + \sum_{j=1}^7 w_{1j}^+ x_j^- \quad (6)$$

$$\tilde{y}(t) = x_1^+ = s(v_1^+) = 1 / (1 + e^{-v_1^+}) \quad (7)$$

Pentru fiecare pereche $(X(t), y(t))$ a setului de date de antrenare se repetă ecuațiile 4-7 obținându-se o serie de valori predictate pentru pH.

2. Calculul erorii de predicție pe întregul set de date:

$$E = \sum_{t=1}^{\text{lungime set date}} \frac{1}{2} (\tilde{y}(t) - y(t))^2,$$

3. Adaptarea ponderilor fiecărui strat:

$$W_{1j}^+ \text{ nou} = W_{1j}^+ \text{ vechi} - \lambda * \frac{\partial E}{\partial w_{1j}^+}, \quad 1 \leq j \leq 7$$

$$W_{jk}^- \text{ nou} = W_{jk}^- \text{ vechi} - \lambda * \frac{\partial E}{\partial w_{jk}^-},$$

$$1 \leq j \leq 7, \quad 1 \leq k \leq 4$$

unde:

s = funcția sigmoidă ce are proprietatea $s'(x) = s(x) * (1 - s(x))$

W = matricea de ponderi a conexiunilor strat intrare - strat ascuns

W^+ = matricea de ponderi a conexiunilor strat ascuns - strat ieșire

v^- = semnalele de activare ale neuronilor din stratul ascuns

v^+ = semnalele de activare ale neuronilor din stratul de ieșire

x^- = valorile citite la ieșirea neuronilor stratului ascuns

x^+ = valorile citite la ieșirea neuronilor stratului de ieșire

λ = rata de învățare

Derivatele erorii de predicție în raport cu ponderile se pot calcula astfel:

$$\frac{\partial E}{\partial W_{ij}} = (\tilde{y}(t) - y(t)) * s'(v_i) * x_j$$

și

$$\frac{\partial E}{\partial W_{jk}} = (\tilde{y}(t) - y(t)) * s'(v_i) * W_{ij} * s'(v_j) * X_k(t)$$

4. Rezultate

Rezultatele identificării sunt ilustrate în figurile de mai jos. Antrenarea s-a produs pe durata a 2756 de epoci cu un prag de eroare de 0.1. Din figura 3, se observă o scădere exponentială a erorii de predicție în prima jumătate a perioadei de antrenare, urmată în cea de-a doua jumătate de o atingere liniară a limitei impuse. În aplicația prezentă, rata de învățare a rămas constantă la valoarea 0.3.

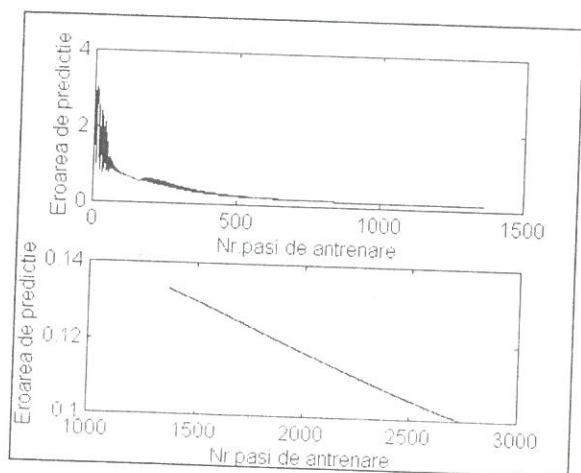


Figura 3. Evoluția erorii de predicție pe durata antrenării

Setul de date generat cuprinde 200 de eșantioane (perechi intrare-ieșire), dintre care primele 100 formează baza de antrenare, iar restul testul de validare. Așa cum se poate observa din figurile 4 și 5 predicțiile rețelei sunt satisfăcătoare atât pe datele de antrenare, cât și pe cele de validare.

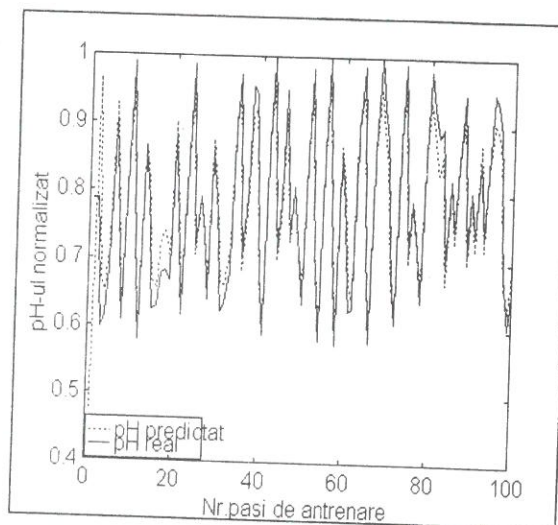


Figura 4. Predicția pe datele de antrenare

În figura 5, se pot nota regiunile în care nu există o suprapunere perfectă între pH-ul real și cel predictat cum ar fi spre exemplu regiunea 0-10 sau 30-50. Un argument posibil în justificarea acestor deviații ar putea fi existența unui spectru de frecvență insuficient de larg în setul datelor de antrenare.

Cu toate acestea, din punctul de vedere al tehnicii de control, modelul dinamic al procesului este bine direcționat (tendențele de evoluție sunt bine predictate) de către rețeaua neurală.

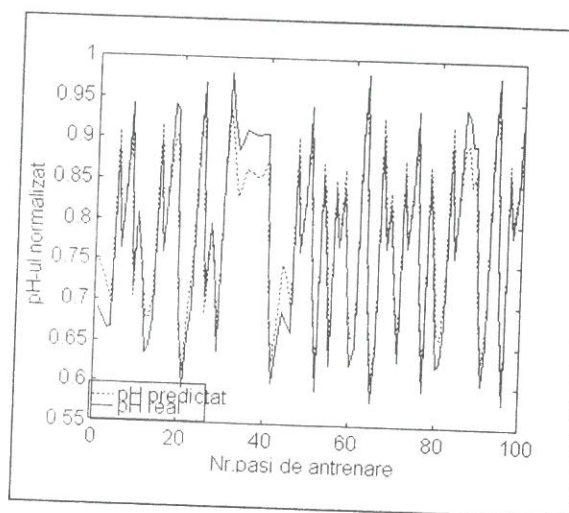


Figura 5. Predicția pe datele de validare

5. Concluzie

Lucrarea a încercat o abordare a problemei de identificare a unui sistem industrial prin metode neconvenționale, care și-au dovedit eficacitatea în situațiile dificil de rezolvat prin tehnicile convenționale. În această idee, autorii și-au propus să elaboreze o serie de articole care să trateze problemele de control, diagnosticare, decizii optime și de supervizare prin mecanisme neurale, algoritmi fuzzy sau neuro-fuzzy.

Bibliografie

1. **CHEN, S., S.A BILINGS:** Neural Networks for Nonlinear Dynamic System Modelling and Identification. În: Int. Journal Control, 56,1992.
2. **BHAT, N.V., P.A. MINDERMAN:** Modelling Chemical Process Systems via Neural Computation. În: IEEE Control System Magazine, aprilie 1990.
3. **TEODOREAN,G., M. COSTEIU:** Rețele neuronale, Editura Microinformatica, 1994.
4. **KOSKO,B.:** Neural Networks and Fuzzy Systems: a Dynamical Systems Approach, Prentice Hall, NJ, 1991.
5. **LJUNG, L.:** System Identification: Theory for User, Prentice Hall, N.J., 1987.
6. **WIDROW, B., D.H. NGUYEN:** Neural Networks for Self-learning Control System. În: IEEE Control Systems Magazine, April 1990.
7. **HOPTRUFF, R.G.:** The Principles and Practice of Time Series Forecasting and Business Modelling Using Neural Nets. În: Neural Computing & Applications. Vol. 1, No. 1, pp. 59-66.