

Autoreferat

Imię i Nazwisko: Wojciech Froelich

Instytut Informatyki, Uniwersytet Śląski,
ul. Będzińska 39, 41-200 Sosnowiec
Email: wojciech.froelich@us.edu.pl
www: <https://sites.google.com/site/wfroelichhomepage/home>

2. Posiadane dyplomy

Tytuł magistra inżyniera informatyki, specjalność: budowa i oprogramowanie maszyn matematycznych; dyplom ukończenia wyższych studiów magisterskich: Politechnika Śląska, Wydział Informatyki, Elektroniki i Automatyki, Gliwice, kierunek Informatyka, rok 1987; praca magisterska pt. „Symulator mikroprocesorów Motorola 6800” napisana pod kierunkiem dr inż. Krzysztofa Nałęckiego.

Stopień doktora nauk technicznych w zakresie informatyki; Akademia Górniczo-Hutnicza w Krakowie; rok 2004; rozprawa doktorska pt. „Agentowo-ewolucyjny model pozyskiwania wiedzy” napisana pod kierunkiem Prof. dr hab. inż. Edwarda Nawareckiego.

3. Informacje o dotychczasowym zatrudnieniu w jednostkach naukowych

1987 – 1989 Instytut Informatyki Czasu Rzeczywistego, Politechnika Śląska w Gliwicach (w tym roczna służba wojskowa).

1994 do teraz, Instytut Informatyki, Wydział Informatyki i Nauki o Materiałach, Uniwersytet Śląski, Sosnowiec.

4. Wskazanie osiągnięcia (zgodnie z art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. nr 65, poz. 595 ze zm.):

Jako podstawowe osiągnięcie naukowe przedstawiam cykl dwunastu publikacji monotematycznych. Wszystkie wskazane artykuły zostały zrealizowane po uzyskaniu przeze mnie tytułu doktora.

a) tytuł osiągnięcia naukowego

Rozmyte sieci kognitywne – rozwój teorii i zastosowań.

b) (autor/autorzy, tytuł/tytuły publikacji, rok wydania, nazwa wydawnictwa)

Publikacje w czasopismach posiadających współczynnik Impact Factor:

[P1] E. I. Papageorgiou, W. Froelich, Application of Evolutionary Fuzzy Cognitive Maps for Prediction of Pulmonary Infections. IEEE Transactions on Information Technology in Biomedicine 16(1), pp. 143-149, 2012

[P2] E. I. Papageorgiou, W. Froelich, Multi-step prediction of pulmonary infection with the use of evolutionary fuzzy cognitive maps. Neurocomputing, 92, pp.28-35, 2012

[P3] W. Froelich, E. I. Papageorgiou, M. Samarinas, K. Skriapas, Application of evolutionary fuzzy cognitive maps to the long-term prediction of prostate cancer, Applied Soft Computing, 12(12), pp. 3810–3817, 2012

[P4] W. Froelich, J.L. Salmeron, Evolutionary Learning of Fuzzy Grey Cognitive Maps for the Forecasting of Multivariate, Interval-Valued Time Series, International Journal of Approximate Reasoning 55 (6), pp. 1319-1335, 2014

[P5] J.L. Salmeron, W. Froelich, Dynamic Optimization of Fuzzy Cognitive Maps for Time Series Forecasting, Knowledge-based systems 105, pp. 29-37, 2016

[P6] W.Froelich, W.Pedrycz, Fuzzy Cognitive Maps in the Modeling of Granular Time Series, Knowledge-based systems, Volume 115, pp. 110-122, 2017

[P7] W.Froelich, Towards Improving the Efficiency of the Fuzzy Cognitive Map Classifier, Neurocomputing, 2016, DOI: 10.1016/j.neucom.2016.11.059

[P8] P. Juszczuk, W. Froelich, Learning Fuzzy Cognitive Map Using a Differential Evolution Algorithm, Polish Journal of Environmental Studies, Vol. 18, No. 3B, pp. 108-112, 2009

Publikacje pozostałe:

[P9] W. Froelich, P. Juszczuk, Predictive Capabilities of Adaptive and Evolutionary Fuzzy Cognitive Maps - A Comparative Study, w książce: "Intelligent Systems for Knowledge Management", series: Studies in Computational Intelligence, Vol. 252, pp. 153-174, 2009

[P10] W. Froelich, E.I. Papageorgiou, Extended Evolutionary Learning of Fuzzy Cognitive Maps for the Prediction of Multivariate Time-Series, w książce: "Fuzzy Cognitive Maps for Applied Sciences and Engineering", series: Intelligent Systems Reference Library, pp. 121-131, 2014

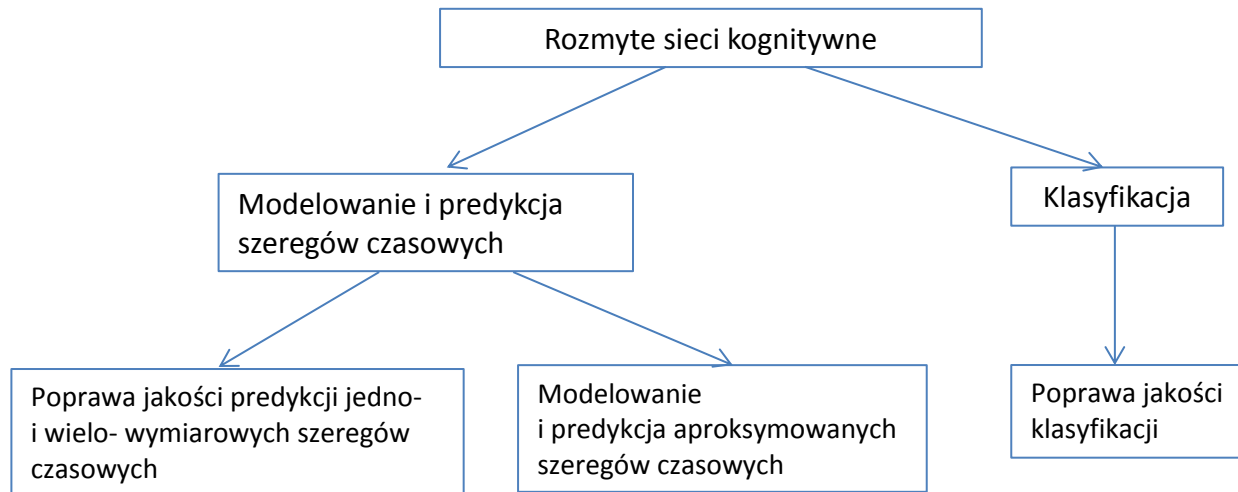
[P11] W. Froelich, A. Wakulicz-Deja, Application of Fuzzy Cognitive Maps for Stock Market Modeling and Forecasting, Lecture Notes in Artificial Intelligence, Vol. 5027, 2008, pp. 72—81

[P12] W. Froelich, A. Wakulicz-Deja, Mining Temporal Medical Data Using Adaptive Fuzzy Cognitive Maps, Proceedings of the HSI'09, 2nd International Conference on Human System Interaction, Catania, Italy, 21-23 May, 2009, pp. 16-23

c) omówienie celu naukowego wymienionych prac i osiągniętych wyników wraz z omówieniem ich ewentualnego wykorzystania.

4.1 Cel naukowy

Celem naukowym wymienionych prac był rozwój teorii i zastosowań modelu rozmytych sieci kognitywnych (ang. Fuzzy Cognitive Maps - FCMs). Prowadzone prace podejmowały problemy modelowania i predykcji szeregów czasowych oraz klasyfikacji. Poniżej przedstawiam schemat prowadzonych prac.



Szczegółowe cele naukowe wymieniam poniżej:

- poprawa efektywności sieci FCM podczas jednokrokowej predykcji szeregów czasowych (wyniki badań w tym zakresie opisano w rozdziale 4.3.1 oraz w artykułach [P1,P5,P9,P10]),
- poprawa efektywności sieci FCM podczas wielokrokowej predykcji szeregów czasowych (wyniki badań w tym zakresie opisano w rozdziale 4.3.2 oraz w artykułach [P2,P3]),
- adaptacja rozmytych szarych sieci kognitywnych FGCM w celu umożliwienia predykcji przedziałowych, wielowymiarowych szeregów czasowych (wyniki badań w tym zakresie opisano w rozdziale 4.3.3 oraz w artykule [P4]),
- konstrukcja granularnego modelu szeregu czasowego oraz jego predykcja za pomocą sieci FCM (wyniki badań w tym zakresie opisano w rozdziale 4.3.4 oraz w artykule [P6]),
- poprawa efektywności działania klasyfikatora opartego na sieci FCM (wyniki badań w tym zakresie opisano w rozdziale 4.3.5 oraz w artykule [P7]),
- Niezależnie od powyżej przedstawionych celów teoretycznych, w większości prac postawiono i rozwiązano konkretne problemy praktyczne. Prowadzono też prace, których celem były wyłącznie nowe zastosowania (opisane w artykułach [P8,P11,P12]). Wyniki badań w zakresie zastosowań opisano w rozdziale 4.3.6.

Poniżej przedstawiam wprowadzenie teoretyczne dotyczące rozmytych sieci kognitywnych oraz prezentuje poszczególne etapy przeprowadzonych badań. Na każdym etapie badań opisuję podjęty problem wraz z jego motywacjami, przedstawiam proponowane rozwiązanie (nowość teoretyczną) oraz wybrane efekty praktyczne wprowadzonych nowości.

W niniejszym autoreferacie stosuję ujednolicony sposób oznaczeń (nieznacznie zmodyfikowany w stosunku do tego używanego we wskazanych artykułach).

4.2 Wprowadzenie

Oznaczmy przez $t = 0, 1, \dots, t_k$ dyskretną skalę czasu, gdzie k jest parametrem ograniczającym jej zakres. Niech $v_1(t), v_2(t), \dots, v_n(t), n \in \mathbb{N}$ oznaczają zmienne obserwowane w czasie. Wartości tych zmiennych tworzą wektor $V(t)$. Sekwencja takich wektorów rozumiana jest jako n -wymiarowy szereg czasowy $\{V(t)\}$.

Niech C będzie rodziną zbiorów rozmytych. Dla każdej obserwacji $v_i(t)$ obliczany jest stopień jej przynależności do odpowiadającego zbioru rozmytego $c_i \in C$. W chwili czasu t mamy: $c_i(t) = \mu_i(v_i(t))$. Wartość $c_i(t)$ nazywana jest stopniem aktywacji lub stanem pojęcia c_i , w chwili t . W przypadku ogólnym, kształt funkcji przynależności może być definiowany przez eksperta lub uczony na podstawie danych. W zastosowaniach praktycznych funkcja przynależności przyjmowana jest często jako normalizacja typu min-max [12], tzn.: $c_i(t) = (v_i(t) - \min(v_i)) / (\max(v_i) - \min(v_i))$, gdzie: $\min(v_i)$, $\max(v_i)$ to minimum i maksimum osiągnięte przez zmienną v_i . W zależności od zastosowania, wartości $\min(v_i)$, $\max(v_i)$ podawane są przez eksperta lub obliczane są na podstawie danych. W efekcie otrzymujemy wektor stanu sieci FCM, tzn. $C(t)$.

Sieć FCM zdefiniowana jest jako para $\langle C, W \rangle$, gdzie C to rodzina zbiorów rozmytych (pojęć wchodzących w skład FCM); W to kwadratowa macierz wag przypisanych poszczególnym parom pojęć [2]. Stan FCM w chwili t opisany jest przez wektor $C(t)$. Elementy macierzy W to wagi $w_{ij} \in [-1, 1]$ przypisane parom pojęć. Waga $w_{ij} = 1$ reprezentuje w pełni pozytywny wpływ, a waga $w_{ij} = -1$ w pełni negatywny wpływ i -tego pojęcia na pojęcie j -te. Wartości większe niż -1 i mniejsze niż 1 reprezentują pośrednie wartości oddziaływań.

Rozmyta sieć kognitywna interpretowana jest jako graf ważony [1, 2]. Węzły grafu (pojęcia) to zbiory rozmyte opisujące modelowany problem, krawędzie grafu opisane wagami reprezentują zależności określane często jako przyczynowo skutkowe. Podczas wnioskowania opartego na modelu FCM wpływ pojęć przyczynowych na pojęcia skutkowe jest kumulowany i przekształcany przy zastosowaniu nieliniowej funkcji transformacji f . Z tego powodu FCM jest modelem nieliniowym. Wnioskowanie w sieci FCM realizowane jest jako predykcja stanów pojęć $c'_j(t+1)$, w tym celu najczęściej stosowany jest wzór (1):

$$c'_j(t+1) = f\left(\sum_{i=1, i \neq j}^n w_{ij} c_i(t)\right), \quad (1)$$

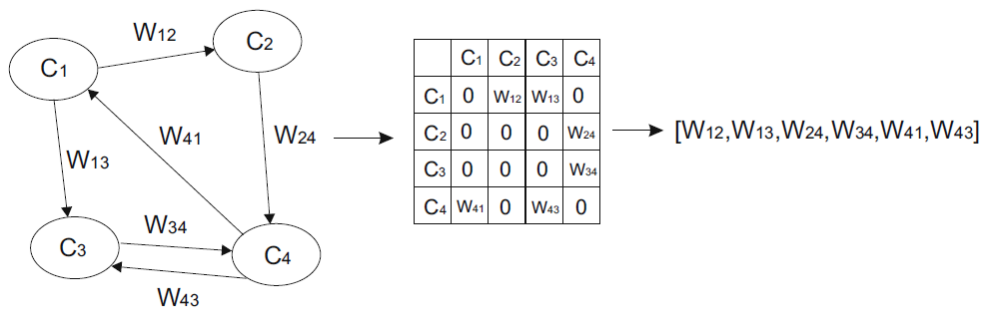
przy czym $c_i(t)$ oznacza aktualny stan i -tego pojęcia, w_{ij} - jest wagą przypisaną do skierowanej krawędzi łączącej i -te pojęcie z j -tym, $n = \text{card}(C)$ oznacza licznosc zbioru pojęć C , $f(x)$ jest funkcją transformacji. Funkcja transformacji wprowadza nieliniowość do modelu FCM oraz ogranicza ważoną sumę stanów pojęć do zakresu $[0, 1]$. Różnego typu funkcje transformacji używane są w konkretnych zastosowaniach, np. dwuwartościowa, trójwartościowa lub najczęściej stosowana logistyczna:

$f(x) = \frac{1}{1 + e^{-gx}}$, przy czym $g > 0$ jest parametrem określającym wzmocnienie uzyskiwane podczas transformacji.

Formuła (1) nazywana jest często równaniem wnioskowania sieci FCM. W przypadku konieczności przeprowadzenia predykcji wielokrokowej (długoterminowej), równanie wnioskowania (1) wywoływane jest iteracyjnie dla kolejnych kroków czasu. Po obliczeniu za pomocą równania (1) przewidywanych stanów pojęć, przeprowadzane jest wyostrzanie (defuzyfikacja) lub denormalizacja stanów pojęć.

Tradycyjne, znane z literatury metody uczenia sieci FCM optymalizują wagi sieci przechowywane w macierzy W w celu uzyskania najlepszej możliwej dokładności predykcji stanu pojęć. Zbiór pojęć C jest wtedy zdefiniowany przez eksperta, jedynie macierz W podlega uczeniu na podstawie danych historycznych. Znane są dwie podstawowe metody uczenia FCM, adaptacyjne i populacyjne. Algorytmy adaptacyjne opierają się na pomysłe uczenia Hebba zaadaptowanego z teorii sztucznych sieci neuronowych. Adaptacyjne metody uczenia FCM obejmują algorytmy: dynamiczny algorytm Hebba - DHL [13], zbalansowany dynamiczny algorytm adaptacyjny – (ang. balanced dynamic algorithm BDA) [14], aktywny algorytm uczenia Hebba (active Hebbian learning – AHL) [15] oraz nieliniowy algorytm uczenia Hebba (ang. nonlinear hebbian learning - NHL [16]. Najważniejsze populacyjne algorytmy uczenia to: algorytm genetyczny (ang. real-coded genetic algorithm RCGA) [17], algorytm mrówiskowy (ang. particle swarm optimization - PSO) [18], algorytm oparty na symulowanym wyżarzaniu (ang. simulated annealing – SA) [19], algorytm oparty na ewolucji różnicowej (ang. differential evolution DE) [10].

Stosując algorytm genetyczny RCGA kolejne wiersze macierzy W odwzorowane są jeden po drugim w chromosomie (wektorze liczb rzeczywistych) [17]. Elementy na głównej przekątnej macierzy W są zerowe ponieważ nie biorą udziału podczas predykcji realizowanej za pomocą równania (1). Przykładowe odwzorowanie między siecią FCM i wektorem chromosomu ilustruje rysunek 1.



Rysunek 1. Konstrukcja chromosomu na podstawie rozmytej sieci kognitywnej

Kolejne populacje chromosomów oceniane są przez algorytm genetyczny wykorzystując funkcję przystosowania. Przykładowo, funkcja ta może być sformułowana jako: $fitness(FCM) = 1/(1 + e)$, gdzie e oznacza zakumulowany średni błąd predykcji wszystkich pojęć dla określonego okresu czasu.

Pomimo istnienia różnych ugruntowanych metod obliczania błędów predykcji szeregów czasowych, w przypadku FCM najczęściej dotychczas stosowaną w tym celu formułą jest (2)[27]:

$$e = \frac{1}{(k-1) \cdot n} \cdot \sum_{s=1}^{k-1} \sum_{i=1}^n \varepsilon_i(t_{s+1}), \quad (2)$$

przy czym: $t \in \langle 0, 1, 2, \dots, t_k \rangle$, $n = \text{card}(C)$ jest licznością zbioru pojęć. Symbol ε_j oznacza jednostkowy błąd predykcji obliczany za pomocą formuły (3):

$$\varepsilon_j(t+1) = |c'_j(t+1) - c_j(t+1)|, \quad (3)$$

gdzie $c'_j(t+1)$ jest przewidywanym przez FCM stanem j-tego pojęcia w czasie $t+1$, $c_j(t+1)$ oznacza znany, zapisany w danych historycznych stan j-tego pojęcia w czasie $t+1$.

Efektom uczenia jest chromosom, najlepiej oceniony przez funkcję przystosowania w ostatniej populacji. Chromosom ten jest dekodowany do postaci sieci FCM, która z kolei jest testowana w wyznaczonym okresie czasu. Podczas testowania, błędy predykcji oceniane są za pomocą wzoru (2), przy czym k oznacza wtedy liczbę kroków czasu w okresie testowym.

Rozmyte sieci kognitywne były do tej pory używane w różnych dziedzinach zastosowań, w szczególności w systemach wspomaganie decyzji [3, 4], predykcji szeregów czasowych [5, 6, 7, 8], sterowaniu [9], uczeniu maszynowym [10] oraz innych. Przegląd badań dotyczących FCM został przedstawiony w pracy [11].

Można zauważyć, że sieci FCM są metodą reprezentacji wiedzy podobną do sztucznych sieci neuronowych (ang. Artificial Neural Network ANN). W przeciwieństwie do ANN, sieci FCM nie wymagają wyróżniania węzłów (pojęć) wejściowych oraz wyjściowych. Wszystkie węzły FCM mogą pełnić obie role, a ich stany powiązane są bezpośrednio z danymi źródłowymi. Ponieważ sieci FCM nie zawierają węzłów ukrytych, eliminowana jest uciążliwa procedura doboru liczby ukrytych warstw i węzłów.

4.3 Podjęte problemy naukowe, zaproponowane rozwiązania oraz uzyskane wyniki.

Pierwsza grupa problemów, które podjąłem w moich badaniach dotyczy poprawy jakości predykcji szeregów czasowych uzyskiwanych za pomocą standardowych sieci FCM. W pierwszej kolejności zrealizowałem porównanie dotyczące efektywności dotychczas znanych, adaptacyjnych i populacyjnych algorytmów uczenia FCM. W pracy [P9] przedstawiono wyniki tych porównań. Również w artykule [P9] przedstawiono wyniki analiz oraz rezultaty eksperymentów dowodzące przewagę algorytmów populacyjnych RCGA i DE nad algorytmami adaptacyjnymi DHL oraz BDA. Z tego powodu, w większości moich kolejnych prac koncentrowałem się na problemie poprawy efektywności uczenia FCM z wykorzystaniem metod populacyjnych.

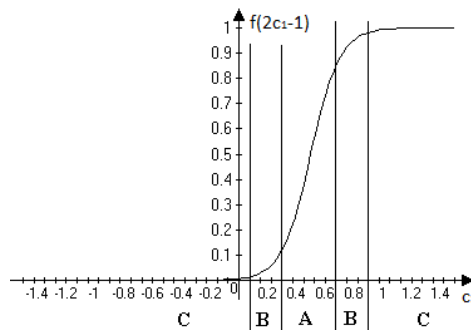
4.3.1 Poprawa efektywności jednokrokowej predykcji szeregów czasowych uzyskiwanej za pomocą modelu FCM [P1, P5, P8]

Problem

W celu wyjaśnienia pierwszego podejmowanego problemu przeanalizujemy jednokierunkowy wpływ pojęcia c_1 na c_2 [P1]. Rozważamy w tym przypadku trywialną sieć FCM złożoną tylko z tych dwóch pojęć. Załóżmy istnienie tylko jednej krawędzi między pojęciami, niech przypisana jej waga wynosi $w_{12} = 1$. Zamiast stosowania wzoru (1), stosujemy przeskalowane równanie (4) zaproponowane w pracy [14], stosowane także w artykule [P1]:

$$c_j(t+1) = f\left(\sum_{i=1, i \neq j}^n (2c_i(t) - 1)w_{ij}\right) \quad (4)$$

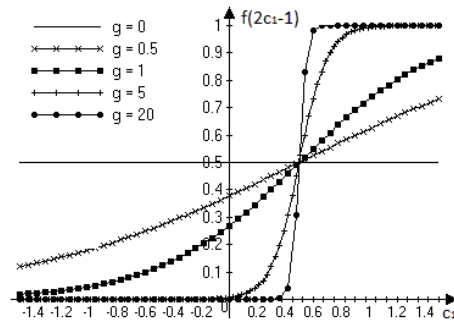
Stosując wzór (4) dla predykcji stanu pojęcia c_2 , otrzymujemy: $c_2(t+1) = f(2 \cdot c_1(t) - 1)$. Rysunek 2 ilustruje wykres tej funkcji, przy założeniu logistycznej funkcji transformacji o wzmacnieniu $g = 5$. Jak można zauważyć na Rysunku 2, wykres funkcji może zostać podzielony pod względem charakterystyki kształtu na trzy obszary. Pierwszy z tych obszarów jest zbliżony do funkcji liniowej i został zaznaczony jako A. Ten fragment kształtu funkcji znajduje się w sąsiedztwie punktu $\langle 0.5, 0.5 \rangle$. W miarę jak odległość od tego punktu rośnie, wykres funkcji staje się coraz bardziej nieliniowy. Nieliniowe części wykresu oznaczono jako obszary B. Przy dalszym oddalaniu się od punktu $\langle 0.5, 0.5 \rangle$ wykres funkcji staje się znowu zbliżony do liniowego w obszarach oznaczonych jako C. W konsekwencji, wpływ pojęcia przyczynowego c_1 na pojęcie skutkowe c_2 uzależniony jest od obszaru funkcji transformacji, który aktualnie jest wykorzystywany podczas wnioskowania. W obszarach C, tam gdzie pochodna funkcji jest bliska zeru $f'(x) \approx 0$, niezależnie od stanu pojęcia c_1 , po transformacji uzyskujemy $f(x) \approx 0$ lub $f(x) \approx 1$. Oznacza to, że w obszarach C zależność między pojęciami jest niezależna lub w znikomy sposób zależna od wagi w_{12} .



Rysunek 2. Wykres przeskalowanej funkcji transformacji dla parametru $g = 5$

Wadą logistycznej funkcji transformacji jest konieczność ustalania przez eksperta wartości parametru wzmacnienia g , często niezależnie od rozpatrywanych danych. W ten sposób powstaje możliwość zbyt małego lub zbyt dużego wzmacnienia realizowanego przez sieć. W szczególności jest to istotne w obszarze A, w którym pochodna $f'(x)$ jest prawie stała, uzależniona od wartości g . Z drugiej strony, wartości wag $w_{ij} \in [-1, 1]$ są także ograniczone w zdefiniowanym zakresie. W efekcie, skumulowany wpływ pojęć przyczynowych na skutkowe jest ograniczony i dla wybranych danych może okazać się zbyt mały w celu realizacji predykcji stanu pojęcia docelowego. Ogranicza to efektywność działania całej sieci FCM.

Przeanalizujemy jeszcze wpływ parametru g na kształt funkcji $f(x)$. Na Rysunku 3 przedstawiono kilka przykładowych wykresów funkcji transformacji dla różnych wartości parametru g . Można zauważyć, że zwiększając g , dla $g > 5$, obszar liniowości A zmniejsza się przy równoczesnym wzroście w tym obszarze wartości pochodnej $f'(x)$.



Rysunek 3. Porównanie funkcji transformacji dla różnych wartości wzmocnienia g

Obszary B prawie zanikają, przy czym obszary C powiększają się. Z drugiej strony, dla $g = 0$ funkcja transformacji staje się stałą, tzn. $f(x) = 0.5$, co prowadzi do całkowitej utraty informacji przenoszonej między pojęciami. Powyższa analiza pokazuje w jak istotny sposób kształt funkcji transformacji uzależniony jest od wartości parametru g oraz jak bardzo wpływa to na właściwości sieci FCM.

Rozwiązanie 1 [P1]

W celu zwiększenia efektywności predykcji uzyskiwanej za pomocą modelu FCM zaproponowano optymalizację parametru g i dostosowanie go do charakterystyki danych. Ze względu na wzajemną zależność, parametr g optymalizowany jest razem z macierzą W w sieci FCM. W pracy [P1] proponowane jest uwzględnienie parametru g jako elementu wektora chromosomu (genu) w procesie optymalizacji realizowanej za pomocą algorytmu genetycznego.

Problem

W celu wyjaśnienia kolejnego problemu dotyczącego działania sieci FCM, przeanalizujemy wpływ procesu normalizacji typu min-max na wartości stanów poszczególnych pojęć FCM.

W przypadku gdy $v_i = \max(v_i)$, po przeprowadzeniu normalizacji otrzymujemy wartość pojęcia $c_i = 1$. Dla $v_i = \min(v_i)$ otrzymujemy $c_i = 0$. Rozważmy przypadek wystąpienia wartości odstającej w analizowanym szeregu czasowym. Po przeprowadzeniu normalizacji wszystkich obserwacji $v_i(t)$ większość stanów odpowiadającego pojęcia c_i zostaje przesunięta w kierunku wartości 0 lub 1 (w zależności od typu wartości odstającej występującej w szeregu czasowym). Jak pokazano na przykładzie zawartym w pracy [P1] funkcja transformacji wykorzystywana jest wtedy głównie w zakresie obszarów C , w których jak opisano wcześniej, następuje utrata części informacji przekazywanej między pojęciami.

Rozwiązanie 2 [P1]

W celu rozwiązania zaistniałego problemu zaproponowano w pracy [P1] kontrolowane przesunięcie wartości pojęć FCM w kierunku centralnego obszaru A funkcji transformacji. Proponuje się zastosowanie normalizacji uzależnionej od średniej oraz odchylenia standardowego obliczonego dla danych historycznych analizowanego szeregu czasowego. Proponowana funkcja (5) sparametryzowana jest za pomocą λ_i , gdzie i jest indeksem pojęcia.

$$c_i(t) = \frac{v_i(t) - l(v_i)}{u(v_i) - l(v_i)}, \quad (5)$$

gdzie: $l(v_i) = \text{mean}(v_i) - \lambda_i \cdot \text{stdDev}(v_i)$, $u(v_i) = \text{mean}(v_i) + \lambda_i \cdot \text{stdDev}(v_i)$.

Po podstawieniu za $l(v_i)$ i $u(v_i)$ przedstawionych powyżej wzorów, funkcja (5) przyjmuje postać (6):

$$c_i(t) = \frac{v_i(t) - \text{mean}(v_i) + \lambda_i \cdot \text{stdDev}(v_i)}{2 \cdot \lambda_i \cdot \text{stdDev}(v_i)}. \quad (6)$$

Parametr λ_i determinuje długość przedziału $[l(v_i), u(v_i)]$. Zwykle $l(v_i) \neq \min(v_i)$ and $u(v_i) \neq \max(v_i)$, z tego powodu dla $v_i < l(v_i)$ lub $x_i > u(x_i)$ uzyskana wartość $c_i(t)$ może wypaść z przedziału $[0,1]$. W takim przypadku, jeżeli $c_i(t) > 1$, podstawiamy $c_i(t) = 1$ oraz jeżeli $c_i(t) < 0$, podstawiamy $c_i(t) = 0$.

Denormalizacja przeprowadzana jest za pomocą formuły (7):

$$v_i(t) = c_i(t) \cdot 2 \cdot \lambda_i \cdot \text{stdDev}(v_i) + \text{mean}(v_i) - \lambda_i \cdot \text{stdDev}(v_i). \quad (7)$$

Rezultaty zastosowania rozwiązań 1 i 2 [P1]

W celu przeprowadzenia oceny proponowanych rozwiązań 1 i 2 wykorzystano rzeczywiste dane medyczne. Były to wielowymiarowe szeregi czasowe zebrane dla dwudziestu pacjentów chorujących na zapalenie płuc. Dane zebrano w sposób anonimowy w greckim szpitalu (General Hospital of Lamia). Specyficzny wybór danych był zdeterminowany wymaganiami projektu europejskiego DEBUIIT (siódmy program ramowy Unii Europejskiej). Walidacja proponowanego rozwiązania była przeprowadzana w warunkach rzeczywistych z udziałem lekarzy. W celu oceny efektywności rozwiązań 1 i 2, przeprowadzono szereg eksperymentów. W celach porównawczych, stosując wzór (2) obliczono błędy predykcji generowane przez następujące sieci FCM:

FCM-I – standardowa normalizacja min-max oraz stałe (dobre przez eksperta) $g = 5$,

FCM-II – proponowana normalizacja (6), ustalony parametr λ_i , oraz stała wartość $g = 5$,

FCM-III – standardowa normalizacja oraz optymalizacja g za pomocą algorytmu genetycznego,

FCM-IV – proponowana normalizacja (6), ustalony przez eksperta parametr λ_i oraz optymalizacja g .

Walidację przeprowadzono ze względu na poszczególne szeregi czasowe przypisane pacjentom. Zastosowano procedurę walidacji krzyżowej typu LOOCV (ang. leave-one-out crossvalidation), która zwykle stosowana jest w przypadku dysponowania małą liczbą danych [27]. Szereg opisujący historię choroby jednego pacjenta pełnił rolę testową, 19 pozostałych szeregów było używanych w celu uczenia modelu FCM. Uczenie i testowanie było powtarzane 20 razy wybierając sekwencje kolejnych pacjentów w celu testowania. Po obliczeniu błędu predykcji dla każdego szeregu, obliczono średnią błędów dla wszystkich dwudziestu pacjentów. Statystyka uzyskanych błędów przedstawiona jest w Tabeli 1. Symbolem StdDev oznaczono odchylenie standardowe.

Tabela 1: Błędy predykcji

| FCM | Średnia | StdDev | Min | Max |
|---------|---------|--------|--------|--------|
| FCM-I | 0.1763 | 0.0181 | 0.1532 | 0.1934 |
| FCM-II | 0.1702 | 0.0174 | 0.1513 | 0.1942 |
| FCM-III | 0.1218 | 0.0083 | 0.1114 | 0.1377 |
| FCM-IV | 0.1047 | 0.0065 | 0.0917 | 0.1143 |

Średni błąd uzyskany dla sieć FCM-IV był najmniejszy, co oznacza, że obydwie wprowadzone nowości teoretyczne: Rozwiązanie 1 i Rozwiązanie 2 okazały się korzystne ze względu na poprawę efektywności predykcji uzyskiwanej przez model FCM.

Rozwiązanie 3 [P10]

W pracy [P10] zaproponowano usprawnienie powyżej opisanej metody. Parametr g określający wzmocnienie funkcji transformacji optymalizowany był indywidualnie dla każdego pojęcia. Oznacza to, że dla każdego pojęcia c_i chromosom zawierał odpowiadający parametr g_i . Podobnie parametr λ_i sterujący normalizacją został objęty indywidualną dla każdego pojęcia optymalizacją za pomocą algorytmu genetycznego. Ze względu na to, że dla każdego i -tego pojęcia uzyskano inną, zoptymalizowaną wartość λ_i ocena błędów tradycyjnie realizowana dla FCM przed realizacją procesu denormalizacji, musiała zostać zmodyfikowana. Zastosowano w tym celu wzór (8):

$$e = \frac{1}{(k-1) \cdot n} \cdot \sum_{t=1}^{k-1} \sum_{i=1}^n \frac{|v_i(t) - v'_i(t)|}{|\max(v_i) - \min(v_i)|}, \quad (8)$$

gdzie: k to długość szeregu, n to liczba pojęć FCM, $\max(v_i), \min(v_i)$ były podane przez ekspertów (lekarzy), $v_i(t)$ oznacza wartość znaną ze zbioru danych, $v'_i(t) = \mu^{-1}(c'(t))$ jest zdemoralizowaną wartością predykcji wygenerowaną przez FCM, $c'(t)$ to przewidywany stan pojęcia FCM.

Rezultaty zastosowania rozwiązania 3 [P10]

Również w przypadku oceny efektów rozwiązania 3 wykorzystano poprzednio opisane dane medyczne. W celach porównawczych trzy różne sieci FCM były uczone i testowane:

FCM-A – sieć referencyjna, taka sama jak poprzednio opisana FCM-I

FCM-B – sieć taka sama jak FCM-IV z pracy [P1], ustalony parametr λ_i oraz optymalizacja g wspólnego dla wszystkich [P1],

FCM-C – indywidualna optymalizacja λ_i oraz g_i dla każdego pojęcia (zaproponowana w [P10]).

Table 2: Statystyka błędów predykcji

| | Średnia | StdDev |
|-------|---------|--------|
| FCM-A | 0.1672 | 0.0224 |
| FCM-B | 0.1182 | 0.0105 |
| FCM-C | 0.1061 | 0.0107 |

Wszystkie sieci FCM były uczone i testowane stosując wzór (8) w celu obliczania błędów predykcji. Z tego względu wyniki liczbowe pokazane w Tabeli 2 nie są porównywalne z tymi pokazanymi poprzednio w Tabeli 1. Jak można zauważyć w Tabeli 2, błędy predykcji uzyskiwane przez sieć FCM-C są znacznie mniejsze niż dla referencyjnej FCM-A oraz trochę mniejsze niż dla sieci FCM-B proponowanej poprzednio w pracy [P1]. Dowodzi to, że dla wybranego, medycznego zbioru danych, nowości proponowane w pracy [P10] przyczyniły się do dalszego wzrostu efektywności modelu FCM.

Problemy

1) Na podstawie wyników licznych eksperymentów postawiono hipotezę, że uczenie modelu FCM wykorzystujące wszystkie dostępne dane uczące może nie być najlepszym rozwiązaniem. Teoretycznie, im dłuższy szereg czasowy, na podstawie którego model jest uczony, tym bardziej ogólny powinien być uzyskany model. Taki model mógłby reprezentować długoterminowe cechy charakterystyczne szeregu, np. sezonowość powtarzającą się przez dłuższy okres czasu lub długoterminowy trend. W praktyce okazuje się często, że zarówno sezonowość jak i trendy występują w szeregach czasowych lokalnie i zmieniają się w czasie. Uwzględniając zbyt dużo danych podczas uczenia FCM otrzymujemy model zbyt ogólny, nie uwzględniający lokalnej zmienności określonych cech charakterystycznych szeregu. Z drugiej strony, uwzględniając podczas uczenia zbyt mało danych istnieje niebezpieczeństwo otrzymania modelu zbyt szczególnego, taki model nie jest w stanie reprezentować długoterminowych regularności występujących w szeregu. Z powyższych powodów problemem staje się odpowiednie ustalenie okresu uczącego w celu umożliwienia jak najbardziej efektywnej predykcji w okresie testowym.

Zgodnie z wiedzą literaturową, wszystkie dotychczasowe metody uczenia FCM realizowane były w następujący sposób. Niezależnie od charakterystyki danych w okresie uczącym oraz długości tego okresu, uczona sieć FCM zawierała zawsze wybraną przez eksperta stałą liczbę pojęć oraz ustaloną funkcję transformacji. Wagi takiej sieci były uczone wykorzystując wszystkie dostępne dane uczące. Wszystkie znane populacyjne algorytmy uczenia FCM usiłowały dostosować wagi sieci do charakterystyki całego szeregu uczącego. Oznacza to, że wpływ pierwszej wartości w szeregu uczącym jak i tej ostatniej na wartości wag, był identyczny. Co więcej, takie rozwiązanie było stosowane niezależnie od długości szeregu.

2) We wszystkich dotychczasowych zastosowaniach FCM, struktura sieci (zbiór pojęć oraz krawędzi) była określana przez eksperta. W takim przypadku, po zakończeniu uczenia, uzyskana sieć zawierała wszystkie pojęcia zdefiniowane przed jej uczeniem oraz wszystkie krawędzie, nawet te, dla których nauczone wagi były bliskie zeru. W ten sposób, uzyskana sieć stawała się trudna w interpretacji. W celu rozwiązania tego problemu przeprowadzano selekcję krawędzi po zakończeniu uczenia wag sieci. Kasowano krawędzie, którym podczas uczenia przypisano wagi o wartości bezwzględnej poniżej pewnego progu. W konsekwencji poprawa czytelności sieci prowadziła to do mniejszego lub większego wzrostu błędów predykcji.

3) Jak wykazano w pracach [P1] i [P10] funkcja transformacji posiada decydujący wpływ na jakość predykcji uzyskiwanej przez sieć FCM. Pomimo tego, we wszystkich dotychczasowych pracach funkcja transformacji, nawet jeżeli jej ostateczny kształt był kontrolowany poprzez parametry, była ustalana w sposób arbitralny np. jako funkcja logistyczna [P1,P10]. Nigdy poprzednio wybór tej funkcji nie był realizowany automatycznie, nie był on też zmieniany w trakcie uczenia sieci i dostosowywany do określonego okresu uczącego.

Rozwiązanie 4 [P5]

W artykule [P5] zaproponowano następujące nowości pozwalające na rozwiązanie postawionych powyżej problemów:

- uczenie sieci FCM w sposób dynamiczny, adaptując sieć do lokalnej charakterystyki danych uczących,
- dynamiczny wybór pojęć sieci FCM. W ten sposób optymalizowany był rząd modelu predykcyjnego,
- selekcja funkcji transformacji oraz optymalizacja parametrów wybranej funkcji algorytmem populacyjnym.

W celu uczenia modelu FCM z uwzględnieniem wszystkich powyższych elementów oceniano efektywność różnych algorytmów populacyjnych.

Rezultaty rozwiązania 4 [P5]

Jak wiadomo na podstawie literatury, żaden z istniejących modeli predykcyjnych nie jest na tyle uniwersalnym, żeby można go efektywnie stosować dla każdego typu danych. W artykule [P5], po wprowadzeniu rozwiązania 4, przeprowadzono eksperymenty porównawcze FCM oraz wybranych innych, najbardziej znanych modeli predykcyjnych. Testy przeprowadzono na znacznej liczbie, ogólnie dostępnych, jednowymiarowych szeregów czasowych. Po przeprowadzeniu licznych testów statystycznych i eksperymentów zidentyfikowano kryterium wyboru szeregów czasowych, dla których rekomendowane jest zastosowanie proponowanego rozwiązania. Zastosowanie FCM rekomendowane jest dla szeregów liniowych i stacjonarnych. Wyniki eksperymentów podano w pracy [P5]. Najważniejsze z nich podajemy w Tabeli 3. Rezultaty uzyskano dla ośmiu, publicznie dostępnych szeregów czasowych:

1. Annual barley yields per acre in England - Wales (1884-1939)
2. Annual precipitation in inches, entire Great Lakes (1900-1986)
3. Annual precipitation (inches), Lake Huron (1900-1986)
4. Annual precipitation (inches), Lake Superior (1900-1986)
5. Winter negative temperature sum (° C) (1781-1988)
6. Annual rainfall (inches) Nottingham castle (1867-1939)
7. Mean annual temperature (° C) (1781-1988)
8. Mean summer temperature (° C) (1781-1988)

W przypadku realizacji eksperymentów zrezygnowano z oceny błędów za pomocą tradycyjnie stosowanego dla FCM wzoru (3). Zarówno podczas obliczenia funkcji przystosowania w czasie uczenia, jak i podczas testów sieci, zastosowany został średni bezwzględny procentowy błąd predykcji (ang. Mean Absolute Percentage Error (MAPE)) (9).

$$MAPE = \frac{1}{n} \cdot \sum_{t=1}^n \left| \frac{\varepsilon(t)}{v(t)} \right| \cdot 100\%, \text{ dla } v(t) \neq 0, \quad (9)$$

gdzie ε , podobnie jak w poprzednich artykułach oznacza jednostkowy błąd predykcji $\varepsilon_j(t) = |v'(t) - v(t)|$.

Wymagało to przeprowadzania denormalizacji stanów pojęć realizowanej w funkcji przystosowania, co zwiększało czas wymagany do uczenia sieci. Z drugiej strony, tego typu działanie ułatwiło późniejsze porównanie uzyskanych wyników predykcji, z tymi generowanymi przez inne modele.

Tabela 3: Porównanie różnych modeli predykcyjnych (błąd MAPE)

| | Naive | ARIMA | ES | HW | ANFIS | DFCM | Algorytm uczenia sieci DFCM |
|---|---------|---------|---------------|----------|---------|----------------|-----------------------------|
| 1 | 6.5211 | 6.4735 | 6.4839 | 6.4857 | 8.7278 | 6.2479 | (ABC) |
| 2 | 10.3409 | 8.4279 | 7.1678 | 7.5775 | 12.4073 | 6.8986 | (PSO) |
| 3 | 11.4775 | 9.5460 | 8.5139 | 8.6698 | 12.4627 | 8.1838 | (PSO) |
| 4 | 12.4225 | 10.4834 | 9.4372 | 9.9394 | 15.2779 | 9.1267 | (PSO) |
| 5 | 63.8697 | 54.3929 | 53.2736 | 53.4834 | 53.7581 | 50.1879 | (RCGA) |
| 6 | 18.6184 | 12.3115 | 12.2330 | 12.9659 | 34.0544 | 11.5202 | (ABC) |
| 7 | 9.9364 | 7.8881 | 7.7373 | 7.9082 | 12.6097 | 7.7338 | (PSO) |
| 8 | 5.2553 | 4.2521 | 4.0434 | 4.110208 | 7.6996 | 4.1423 | (RCGA) |

Uzyskano bardzo dobre rezultaty eksperymentalne. Jak pokazano w Tabeli 3, w wyniku proponowanych innowacji uzyskano bardzo konkurencyjny model predykcyjny nazwany DFCM (ang. Dynamic Fuzzy Cognitive Map), który w przypadku szeregów liniowych i stacjonarnych okazał się najlepszy dla siedmiu z ośmiu wybranych szeregów czasowych.

4.3.2 Poprawa efektywności modelu FCM w zakresie wielokrokowej predykcji szeregów czasowych [P2,P3]

W tym rozdziale opisuję moje dalsze badania dotyczące poprawy efektywności modelu FCM w zakresie wielokrokowej predykcji, wielowymiarowych szeregów czasowych.

Problem

W celu przeanalizowania propagacji błędów w kolejnych krokach predykcji wielowymiarowego szeregu czasowego oznaczymy przez $D(t)$ znany na podstawie danych historycznych wektor szeregu. Przez $C(t)$ oznaczymy wektor przewidywany przez sieć FCM. W ten sposób sekwencja: $D(t_1), D(t_2), \dots, D(t_k)$, oznacza wielowymiarowy szereg, przy czym k jest liczbą uwzględnionych chwil czasu.

Podobnie jak poprzednio, problem analizujemy dla trywialnej sieci FCM złożonej tylko z dwóch pojęć. Założymy stosowanie trywialnej funkcji transformacji $f(x) = x$. Przy takich założeniach, równanie wnioskowania sieci FCM przyjmuje postać: $c_j(t_{s+1}) = w_{ij}c_i(t_s)$, gdzie $j \neq i$. Obliczmy krok po kroku poziom aktywacji obydwu pojęć oraz błędy predykcji stosując standardowy wzór (3). Przykładowo, w czasie t_2 , poziom aktywacji c_2 wynosi: $c_2(t_2) = w_{12}(t_1)d_1(t_1)$, błąd predykcji to: $\varepsilon_2(t_2) = |d_2(t_2) - c_2(t_2)|$. W celu realizacji kolejnego kroku predykcji dla chwili t_3 , wzór wnioskowania sieci musi zostać zastosowany kolejny raz. Przewidywany stan pierwszego pojęcia $c_1(t_3) = w_{21}(t_2)c_2(t_2)$ uzależniony jest od poprzednio przewidywanego stanu $c_2(t_2)$, a tym samym od poprzednio wygenerowanego błędu $\varepsilon_2(t_2)$. Zauważmy, że dla tej samej chwili czasu t_3 , błąd predykcji obliczony za pomocą wzoru (3) wynosi $\varepsilon_1(t_3) = |d_1(t_3) - c_1(t_3)|$ i nie zależy od poprzednio wynikłego błędu $\varepsilon_2(t_2)$. Podobna sytuacja występuje dla kolejnej chwili czasu t_4 , gdzie $c_2(t_4)$ oraz $\varepsilon_2(t_4)$ zależą od poprzednio przewidywanego stanu $c_1(t_3)$.

oraz błędu $\varepsilon_1(t_3)$. Oznacza to, że za pomocą dotychczas stosowanego wzoru (2) obliczony zostaje błąd $\varepsilon_2(t_4) = |d_2(t_4) - c_2(t_4)|$ niezależny od $\varepsilon_1(t_3)$.

Jak powyżej wyjaśniono, wzór (2) i w konsekwencji wzór (3) nie są odpowiednie do obliczania wielokrotnych błędów predykcji dla danych wielowymiarowych. Wzór (2) nie bierze pod uwagę propagacji błędów między pojęciami FCM. W konsekwencji, algorytmy stosujące wzór (3) podczas uczenia, np. algorytm genetyczny, optymalizują wagi sieci FCM osobno dla każdego pojęcia. Podczas optymalizacji macierzy wag W , każdy jej wiersz optymalizowany jest bez uwzględnienia jego zależności od pozostałych wierszy. Problem ten kumuluje się przy wzroście liczby pojęć oraz wydłużeniu horyzontu predykcji. W efekcie następuje utrata efektywności działania nauczonej sieci FCM.

Rozwiązanie 5 [P2,P3]

W pierwszej kolejności, w celu prawy efektywności obliczania błędów predykcji i tym samym zwiększenia efektywności uczenia FCM dla wielokrotowej predykcji wielowymiarowych szeregów czasowych, zamiast używania wzoru (2) zaproponowano stosowanie dwóch wzorów. Pierwszy z nich (10) zaproponowany w pracy [P2] jest prostą modyfikacją wzoru (3).

$$e = \frac{1}{(k-h) \cdot n} \cdot \sum_{s=1}^{k-h} \sum_{i=1}^n \varepsilon_i(t_{s+h}) \quad (10)$$

Za pomocą wzoru (10) obliczamy tylko błędy uzyskane w h -tym kroku predykcji pomijając obliczanie błędów dla kroków poprzednich, tzn. dla chwil $t_{s+1}, t_{s+2}, \dots, t_{s+h-1}$. Tylko takie błędy są uśredniane dla całej sekwencji uczącej. Parametr h określa wymagany horyzont predykcji.

Mimo, że proponowana zmiana jest niewielka, zaletą używania wzoru (10) zamiast wzoru (3) podczas uczenia FCM jest uzyskanie sieci dedykowanej dla określonego horyzontu czasowego (uwzględniającej propagację błędów predykcji między pojęciami) zamiast sieci zoptymalizowanej dla horyzontu $h = 1$ i jej wykorzystanie, gdy predykcja jest wymagana dla $h > 1$.

W dalszej kolejności, w pracy [P3] zaproponowano alternatywny wzór (11) obliczania błędów dla wielokrotowej predykcji szeregów wielowymiarowych z zastosowaniem FCM.

$$e = \frac{1}{(m-h) \cdot h \cdot n} \cdot \sum_{s=1}^{m-h} \sum_{k=s+1}^{s+h} \sum_{i=1}^n \varepsilon_i(t_k). \quad (11)$$

Za pomocą wzoru (11) obliczane są uśredniane błędy predykcji wygenerowane przez FCM w rozpatrywanym horyzoncie predykcji, czyli w okresie $[t_{s+1}, t_{s+h}]$.

W zależności od wymagań aplikacji, wzory (10) lub (11) powinny być stosowane przy obliczaniu funkcji przystosowania podczas ewolucyjnego uczenia sieci FCM. W przypadku zaistnienia potrzeby uzyskania jak najmniejszego błędu dokładnie w h -tym kroku predykcji, stosujemy wzór (10). W przypadku gdy istotna jest minimalizacja błędów osiąganych w całym okresie predykcji proponujemy stosowanie wzoru (11).

Rezultaty rozwiązania 5

W pracy [P2] przeprowadzono ewaluację proponowanego wzoru (10) stosując wielowymiarowe, rzeczywiste dane pacjentów hospitalizowanych w General Hospital of Lamia (Grecja) chorych na zapalenie płuc (projekt DEBUGIT). Ponieważ praca [P2] była realizowana wcześniej niż [P1], do dyspozycji były dane 19 pacjentów (nie 20 tak jak we później realizowanej pracy [P1]). Również z tego względu w pracy [P2] nie była stosowana optymalizacja procesu normalizacji oraz współczynnika wzmocnienia funkcji transformacji. Referencyjna sieć FCM-A była uczona z zastosowaniem funkcji przystosowania przy obliczaniu błędów za pomocą wzoru (10). Każda z siedmiu sieci, dedykowanych dla określonego horyzontu predykcji $h \in [1,7]$) była uczona stosując wzór (10) w funkcji przystosowania. Sieci te oznaczano jako FCM-B(1), FCM-B(2), ..., FCM-B(7). W celu wyboru danych uczących i testowych zastosowano metodę LOOCV ze względu na poszczególnych pacjentów. Rezultaty eksperymentów pokazane w Tabeli 4 wykazują zmniejszenie błędów predykcji wielokrokowej, po zastosowaniu wprowadzonej innowacji.

Tabela 4: Błędy predykcji wielokrokowej, metoda LOOCV (zapalenie płuc [P2])

| h | FCM-A | | FCM-B(h) | |
|---|--------|----------|----------|----------|
| | (Mean) | (StdDev) | (Mean) | (StdDev) |
| 1 | 0.1691 | 0.1022 | 0.1691 | 0.1022 |
| 2 | 0.2568 | 0.1181 | 0.1762 | 0.1163 |
| 3 | 0.3225 | 0.1141 | 0.1742 | 0.1162 |
| 4 | 0.3643 | 0.1273 | 0.1883 | 0.1121 |
| 5 | 0.4117 | 0.1261 | 0.2112 | 0.1242 |
| 6 | 0.4202 | 0.1235 | 0.2232 | 0.1212 |
| 7 | 0.4386 | 0.1282 | 0.2673 | 0.1220 |

Alternatywne równanie (11) było oceniane z wykorzystaniem danych zebranych dla 40 pacjentów hospitalizowanych ze względu na nowotwór prostaty [P3]. Uzyskane wyniki były w tym wypadku interpretowane pod względem medycznym przez dwóch lekarzy, pomocniczych współautorów pracy [P3]. W celach porównawczych nauczono dwie sieci FCM. FCM-I nauczono wykorzystując znany wzór (3), FCM-II wykorzystując wzór (11) uwzględniony w funkcji przystosowania algorytmu genetycznego. Uzyskane wyniki pokazano w Tabeli 5. Proponowana Sieć FCM-II okazała się lepsza niż referencyjna FCM-I dla wszystkich $h > 2$.

Table 5: Błędy predykcji wielokrokowej, metoda LOOCV (nowotwór prostaty [P3])

| h | FCM-I | | FCM-II | |
|---|--------|----------|--------|----------|
| | (Mean) | (StdDev) | (Mean) | (StdDev) |
| 1 | 0.0985 | 0.0505 | 0.1012 | 0.0728 |
| 2 | 0.1091 | 0.0511 | 0.1026 | 0.0690 |
| 3 | 0.1245 | 0.0613 | 0.1020 | 0.0678 |
| 4 | 0.1380 | 0.0688 | 0.1022 | 0.0703 |
| 5 | 0.1518 | 0.0829 | 0.1006 | 0.0722 |
| 6 | 0.1670 | 0.0937 | 0.1003 | 0.0732 |
| 7 | 0.1855 | 0.1079 | 0.1011 | 0.0727 |

4.3.3 Adaptacja rozmytych szarych sieci kognitywnych w celu predykcji przedziałowych, wielowymiarowych szeregów czasowych. [P4]

Problem

Przedziałowe szeregi czasowe (ang. interval-valued time series – ITS) mogą być stosowane wtedy, gdy dane obserwowane są w sposób ciągły i dokładny, jednakże z punktu widzenia zastosowania istnieje jedynie potrzeba monitorowania i predykcji zakresu ich zmian w poszczególnych przedziałach czasowych. Przykładowe dane tego typu mogą dotyczyć minimalnych i maksymalnych temperatur dziennych lub minimalnych i maksymalnych cen akcji osiągniętych np. w ciągu dnia, tygodnia lub miesiąca na giełdzie papierów wartościowych. W pracy [P4] podejmowany jest problem predykcji szeregów przedziałowych ITS.

W literaturze zaproponowano model Rozmytych Szarych Sieci Kognitywnych (ang. Fuzzy Grey Cognitive Maps - FGCMs) [25]. FGCMs reprezentują przybliżoną wiedzę dotyczącą problemu i opierają się na liczbach szarych (przedziałowych).

Liczba szara oznaczona jest jako $\otimes g \in [\underline{g}, \bar{g}]$, gdzie $\underline{g} \leq \bar{g}$ [28]. Arytmetyka liczb szarych jest identyczna z arytmetyką przedziałową [P4].

FGCM zdefiniowano jako czwórkę[25]:

$$FGCM = \langle \otimes C, \otimes W, f(\cdot), \eta \rangle, \quad (11)$$

gdzie: $\otimes C$ jest zbiorem szarych pojęć, $\otimes W$ jest macierzą szarych liczb, $f(\cdot)$ to funkcja transformacji, η oznacza zakres (dziedzinę) pojęć szarych, w pracy [P4] założono, że η jest przedziałem $[0,1]$.

W każdym kroku czasu stan i-tego pojęcia szarego jest szarą liczbą (przedziałem) oznaczoną jako $\otimes c_i(t)$. $\otimes W$ jest macierzą kwadratową zawierającą szare wagi $\otimes w_{ij} \in [\underline{w}_{ij}, \bar{w}_{ij}]$, przy czym $(\underline{w}_{ij} \leq \bar{w}_{ij})$ oraz $\underline{w}_{ij}, \bar{w}_{ij} \in [-1,+1]$. Wnioskowanie realizowane jest za pomocą wzoru (12).

$$\begin{aligned} \otimes c_j(t+1) &= f\left(\sum_{i=1, i \neq j}^n \otimes w_{ij} \cdot \otimes c_i(t)\right) \\ &= f([\underline{c}_j(t+1), \bar{c}_j(t+1)]) \\ &= [f(\underline{c}_j(t+1)), f(\bar{c}_j(t+1))] \\ &= [\underline{c}_j(t+1), \bar{c}_j(t+1)] \end{aligned} \quad (12)$$

Podobnie jak dla FCM zakłada się, że przekątna macierzy $\otimes W$ nie jest używana. Logistyczna funkcja transformacji $f(\cdot)$ przyjmuje postać (13).

$$\otimes c_j(t+1) = \left[\left(1 + e^{-\lambda \cdot \sum_{i=1}^n \overline{\otimes w_{ij} \cdot \otimes c_i(t)}}\right)^{-1}, \left(1 + e^{-\lambda \cdot \sum_{i=1}^n \underline{\otimes w_{ij} \cdot \otimes c_i(t)}}\right)^{-1} \right], \quad (13)$$

gdzie:

$$\underline{\otimes w_{ij} \cdot \otimes c_i(t)} = \min(\underline{w_{ij}} \cdot \underline{c_i(t)}, \underline{w_{ij}} \cdot \overline{c_i(t)}, \overline{w_{ij}} \cdot \underline{c_i(t)}, \overline{w_{ij}} \cdot \overline{c_i(t)}), \quad (14)$$

$$\overline{\otimes w_{ij} \cdot \otimes c_i(t)} = \max(\underline{w_{ij}} \cdot \underline{c_i(t)}, \underline{w_{ij}} \cdot \overline{c_i(t)}, \overline{w_{ij}} \cdot \underline{c_i(t)}, \overline{w_{ij}} \cdot \overline{c_i(t)}). \quad (15)$$

Rozwiązanie 6 [P4]

W literaturze dotyczącej szeregów ITS można wyróżnić trzy podejścia do ich predykcji[P4]. Pierwsze z nich zakłada niezależną predykcję górnych i dolnych zakresów przedziałów. Należy zwrócić uwagę, że przy takim podejściu, zaniedbywana jest informacji dotycząca szerokości przedziałów. Inne podejście polega na powiązaniu ze sobą zakresów przedziałów i zastosowaniu tradycyjnych metod predykcji zwykłych szeregów czasowych. Tego typu podejście realizuje predykcję zależności między zakresami przedziałów bez wykorzystania arytmetyki przedziałowej. Trzecim podejściem jest zastosowanie metod opartych na arytmetyce przedziałowej.

Pierwszą nowością proponowaną w pracy [P4] jest podjęcie po raz pierwszy problemu predykcji wielowymiarowych szeregów przedziałowych. Tabela 6 umiejscawia podejmowany problem badawczy w relacji do dotychczas rozpatrywanych problemów.

Tabela 6: Metody predykcji szeregów przedziałowych ITS

| Typ szeregu | Zakresy przewidywane są niezależnie od siebie | Model Predykcja zależności między zakresami | Arytmetyka przedziałowa |
|----------------|---|---|-------------------------|
| jednowymiarowy | [20, 21] | [20, 21, 22, 24] | [21, 23] |
| wielowymiarowy | - | - | Nowość[P4] |

Drugą nowością proponowaną w pracy [P4] jest adaptacja modelu sieci FGCM do predykcji wielowymiarowych przedziałowych szeregów czasowych. W pracy [P4] zaproponowano również adaptację algorytmu genetycznego opartego na arytmetyce przedziałowej, algorytm ten wykorzystywany jest do uczenia sieci FGCM.

Zgodnie z notacją stosowaną wcześniej dla tradycyjnych sieci FCM, oznaczmy przez V zbiór zmiennych jednowartościowych, obserwowanych w czasie. Załóżmy, że zmienne te obserwowane są w sposób ciągły. Wektor obserwacji $V(\tau)$ jest rejestrowany w czasie $\tau \in \mathfrak{R}, \tau \in [0, \tau_e]$, gdzie $\tau_e \in \mathfrak{R}$ jest ustalonym parametrem. Sekwencja takich wektorów to tradycyjny (jednowartościowy) wielowymiarowy szereg

czasowy oznaczony jako $\{V(\tau)\}$. Przedział $[0, \tau_e]$ dzielony jest na $t_e \in \mathbb{N}, \tau_e \leq t_e$ przedział indeksowane przez $t \in \mathbb{N}$. W ten sposób otrzymujemy sekwencję przedziałów $[0,1], (1,2), \dots, (t_e-1, t_e]$. Indeksy przedziałów tworzą nową dyskretną skalę czasu $t \in [1, t_e]$ gdzie t_e jest parametrem.

Założmy, że obserwacje $v(\tau)$ każdej zmiennej $v \in V$ zmieniają się znacznie w miarę upływu czasu τ lecz rozważamy jedynie minimum v_{min} i maksimum v_{max} w każdym z przedziałów $\tau \in (t-1, t]$. Skala czasu zdefiniowana przez zmienną t , jest znacznie mniej szczegółowa niż τ tzn. w każdym z przedziałów określonych przez t obserwujemy znaczną liczbę wektorów $V(t)$. Niech dla wybranej zmiennej v $min(v(\tau)), \tau \in (t-1, t]$ będzie dolnym zakresem przedziału liczby szarej $\otimes g(t)$ obserwowanej w czasie t , tzn. $\underline{g}(t) = min(v(\tau))$. Podobnie, niech $\bar{g}(t) = max(v(\tau)), \tau \in (t-1, t]$.

Oznaczmy jako $\otimes G$ zbiór wszystkich rozważanych zmiennych szarych, odpowiednik zbioru V rozważanego poprzednio podczas analizy tradycyjnych sieci FCM. Wartością zmiennej $\otimes g_i \in \otimes G$ w czasie t jest: $\otimes g_i(t) = [\underline{g}_i(t), \bar{g}_i(t)]$. Przez jednowymiarowy przedziałowy szereg czasowy będziemy rozumieli $\{\otimes g_i(t)\}$, czyli sekwencję liczb szarych $\otimes g_i(t), i \in [1, card(\otimes G)]$ obserwowanych w chwilach czasu $t = \{1, 2, \dots, t_e\}$. Uwzględniając wszystkie szare zmienne ze zbioru $\otimes G$, otrzymujemy wektor $\otimes G(t)$ obserwowany w chwili t . Wielowymiarowy szereg przedziałowy przyjmuje formę sekwencji: $\{\otimes G(t)\} = \{\otimes G(1), \otimes G(2), \dots, \otimes G(n)\}$.

W celu zastosowania sieci FGCM do predykcji $ITS = \{\otimes G(t)\}$, szereg ten musi ulec normalizacji: $\otimes c_i(t) = \mu(\otimes g_i(t))$. Realizowane jest to osobno dla dolnego, $\underline{c}_i(t) = \mu(\underline{g}_i(t))$, i górnego, $\bar{c}_i(t) = \mu(\bar{g}_i(t))$ zakresu pojęcia. Dla obydwu zakresów normalizację przeprowadzamy w przedziale $[\underline{g}_{min}, \bar{g}_{max}]$

otrzymując $\underline{c}_i(t) = \frac{\underline{g}(t) - \underline{g}_{min}}{\underline{g}_{max} - \underline{g}_{min}}, \bar{c}_i(t) = \frac{\bar{g}(t) - \underline{g}_{min}}{\underline{g}_{max} - \underline{g}_{min}}$, przy czym $\underline{g}_{min} = min(\underline{g}(t)), \bar{g}_{max} = max(\bar{g}(t))$

dla $t \in [1, t_e]$. Stałe $\underline{g}_{min}, \bar{g}_{max}$ są parametrami podawanymi przez eksperta lub obliczonymi na podstawie danych. W efekcie otrzymujemy wektor liczb szarych $\otimes C(t) = \mu(\otimes G(t))$ oraz dla kolejnych kroków czasu sekwencję stanów szarych pojęć $\{\otimes C(t)\} = \{\otimes C(1), \otimes C(2), \dots, \otimes C(t_e)\}$.

Celem uczenia sieci jest optymalizacja macierzy $\otimes W$ oraz parametrów wzmocnienia logistycznej funkcji transformacji, dla każdego pojęcia osobno. W celu uniknięcia konfliktu nazw z oznaczeniami genów, parametry transformacji, indywidualne dla każdego pojęcia oznaczono jako λ_i . W pracy [P4] zaproponowano adaptację algorytmu genetycznego uwzględniającą przeprowadzenie wybranych operacji z zastosowaniem arytmetyki przedziałowej.

Każdy chromosom rozpatrywanej populacji składa się z dwóch części:

1) Część szara, czyli przedziałowa to wektor liczb szarych utworzony na podstawie macierzy $\otimes W$. Kolejne wiersze $\otimes W$ umieszczane są jeden za drugim w wektorze chromosomu, z pominięciem elementów na przekątnej $\otimes W$ nie biorących udziału we. Długość tej części chromosomu wynosi $n^2 - n$, gdzie $n = card(\otimes C)$.

2) Część tradycyjna (jednowartościowa) to wektor współczynników funkcji transformacji λ_i , wspólnych dla górnego i dolnego zakresu liczb szarych. Długość tej części chromosomu wynosi $n = card(\otimes C)$.

Obliczenie funkcji przystosowania $fitness(FCM) = 1/(1 + e)$ zostało odpowiednio dostosowane. Błąd predykcji obliczany jest za pomocą wzoru (17).

$$e = \frac{1}{n} \cdot \sum_{i=1}^n ERR(\otimes c_i(t), \otimes c_i'(t)), \quad (16)$$

gdzie: $n = card(\otimes C)$, $\otimes c_i(t)$ jest stanem pojęcia szarego znanym z danych historycznych, $\otimes c_i'(t)$ jest stanem pojęcia szarego przewidywanym przez FGCM, symbol ERR oznacza jeden ze wzorów definiującą odległość między liczbami przedziałowymi (Ichino-Yaguchi, Hausdorff lub Kernel-based) [P4]. Pozostałe szczegóły zaproponowanego algorytmu opisano w pracy [P4].

Rezultaty rozwiązania 6 [P4]

W celu oceny proponowanego rozwiązania wykorzystano rzeczywiste dane meteorologiczne [26]. Pojęcia sieci FGCM odpowiadały wybranym obserwacjom meteorologicznym: c_1 – temperatura typu „dry bulb” [Celsius], c_2 - temperatura typu „dew point” [Celsius], c_3 - względna wilgotność, c_4 - szybkość wiatru, c_5 - ciśnienie.

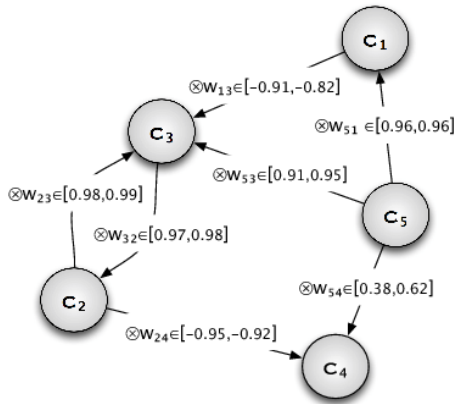
W celach porównawczych przeprowadzono predykcję dolnego i górnego przedziału szeregu ITS za pomocą metod standardowych: metody naiwnej zakładając $\forall i. \otimes c_j(t+1) = \otimes c_j(t)$, ARIMA (ang. autoregressive integrated moving average), VAR (ang. vector auto regression), oraz ES (ang. exponential smoothing).

W Tabeli 7 zaprezentowano rezultaty testów dla długości okresu uczącego (okna przesuwne) $length(W_L) = 30$ oraz dwukrokowej predykcji, $length(W_T) = 2$. Błędy obliczono stosując uśrednianie typu MDE (ang. Mean Distance Error) oraz wzór (16) dla różnych odległości Ichino-Yaguchi (MDE_{IY}), Hausdorff (MDE_H) oraz jądrowej (ang. kernel-based distance - MDE_K). Jak można zauważyć w Tabeli 7, sieć FGCM okazała się najlepsza, niezależnie od zastosowanego sposobu obliczania błędów.

Table 7: Błędy predykcji dla $length(W_L) = 30$ oraz $length(W_T) = 2$

| Model | MDE_{IY} | MDE_H | MDE_K |
|-------|---------------|---------------|---------------|
| naive | 0.1606 | 0.2290 | 0.1798 |
| ARIMA | 0.1582 | 0.2156 | 0.1705 |
| VAR | 0.1423 | 0.2023 | 0.1603 |
| ES | 0.1407 | 0.1965 | 0.1573 |
| FGCM | 0.1373 | 0.1922 | 0.1528 |

W porównaniu z innymi modelami predykcyjnymi typu czarna skrzynka (ang. black box) sieci FCM i FGCM mogą być interpretowane przez eksperta. Rysunek 4 przedstawia przykładową sieć FGCM wygenerowaną na podstawie danych meteorologicznych. Przedstawiono tylko te krawędzie, dla których wybielone (patrz [P4]) wartości wag $|\dot{w}_{ij}| \geq 0.4$.



Rysunek 4: Przykładowa rozmyta szara sieć kognitywna

W pracy [P4] potwierdzono znaczną zgodność uzyskanej sieci FGCM z istniejącą wiedzą meteorologiczną.

4.3.4 Konstrukcja granularnego modelu szeregu czasowego oraz jego predykcja za pomocą rozmytych sieci kognitywnych [P6]

Problem

Zauważmy, że w wielu zastosowaniach praktycznych informacja dotycząca dokładnych, numerycznych wartości szeregu czasowego jest mniej istotna. W takich przypadkach szereg może być aproksymowany, rozważa się problem znalezienia kompromisu między ogólnością i szczegółowością informacji zawartej w szeregu. Z jednej strony mamy do czynienia z dokładnymi wartościami numerycznymi szeregu oryginalnego, z drugiej strony przybliżamy ten szereg za pomocą sekwencji pewnych jednostek informacji, np. granul. Poprzez takie przybliżenie uzyskujemy mniej dokładną reprezentację szeregu, w zamian oczekujemy zwiększenia dokładności predykcji, która w takim przypadku może odbywać się na wyższym poziomie abstrakcji, tzn. na poziomie granularnym.

Podczas tworzenia granularnej reprezentacji szeregu czasowego następuje odfiltrowanie (do pewnego stopnia) komponentu losowego zawartego w szeregu. Podczas granularyzacji możemy stosować określone kryterium pozwalające na znalezienie kompromisu między szczegółowością i ogólnością uzyskiwanej aproksymacji. W konsekwencji uzyskujemy zoptymalizowaną sekwencję granul określaną jako granularny szereg czasowy (ang. Granular Time Series – GTS). Efektywność predykcji takiej sekwencji mierzona jest poprzez obliczenie zgodności (ang. overlapping) między przewidywanymi i rzeczywistymi granulami. Większa zgodność oznacza lepszą jakość predykcji.

W moich badaniach podjąłem następujące dwa problemy:

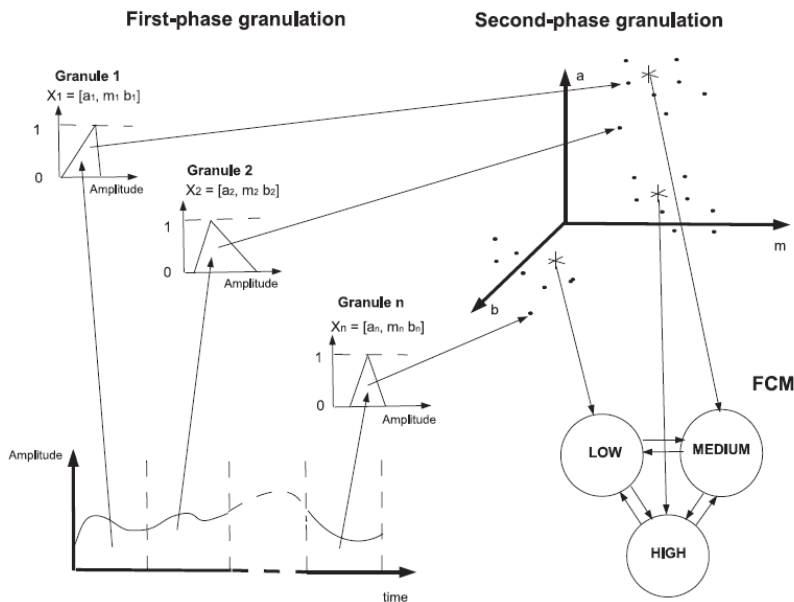
1. problem aproksymacji tradycyjnego szeregu czasowego w formie szeregu granularnego GTS,
2. problem konstrukcji modelu predykcyjnego GTS opartego na rozmytych sieciach kognitywnych, model powinien zapewniać zadowalającą jakość predykcji mierzoną na poziomie granularnym.

Rozwiązanie 7 [P6]

Rozwiązanie proponowane w pracy [P6] zostało zilustrowane na rysunku 5.

W pierwszej kolejności, w celu aproksymacji szeregu czasowego, wyznaczamy stałej długości przedziały czasu, w których będą konstruowane granule. W szczególnym przypadku granule te rozumiane są jako trójkątne liczby rozmyte, których funkcje przynależności są optymalizowane stosując zasadę uzasadnionej granularyzacji (ang. Principle of Justifiable Granularity - PJG) [P6]. Proces tworzenia granul określony jest jako pierwsza faza granularyzacji (ang. first-phase granulation). Uzyskana sekwencja granul nazywana jest granularnym szeregiem czasowym.

W celu konstrukcji modelu predykcyjnego szeregu granularnego GTS, wszystkie granule szeregu podlegają grupowaniu. Grupowanie metodą Fuzzy C-Means przeprowadzone jest w trójwymiarowej przestrzeni, zgodnie z reprezentacją trójkątnych liczb rozmytych w formie ich zakresów (a,b) oraz wartości modalnych (m) . Proces grupowania granul nazywamy w naszym przypadku drugą fazą granularyzacji szeregu czasowego (ang. second-phase granulation). Reprezentanci klastrów będący także granulami przyjmowani są jako pojęcia rozmytej sieci kognitywnej. Pojęcia FCM są granulami drugiego poziomu (ang. second-phase granules). W celu interpretacji szeregu GTS, pojęcia FCM porządkowane są ze względu na ich wartości modalne. W dalszej kolejności przypisujemy im adekwatne oznaczenia symboliczne, np.: 'LOW', 'MEDIUM', 'HIGH' (jak pokazano na rysunku 5).



Rysunek 5: Proponowana idea granularyzacji i predykcji szeregu czasowego

W celu konstrukcji modelu i realizacji predykcji GTS odkrywamy i wykorzystujemy zależności czasowe między pojęciami FCM. Ponieważ pojęcia FCM są uporządkowane ze względu na ich wartości modalne i odpowiadające wartości symboliczne, krawędzie grafu FCM odpowiadają zależnościom temporalnym między wartościami lingwistycznymi obserwowanymi w szeregu granularnym. Wagi przypisane krawędziom reprezentują siłę zależności. W celu odkrycia tych zależności stosowany jest algorytm genetyczny.

Relacja każdej granuli szeregu GTS do wszystkich pojęć FCM opisana jest za pomocą wektora numerycznego. W celu obliczenia wartości elementów tego wektora, proponowana jest wyspecjalizowana funkcja oceniająca względny stopień pokrywania się granul w stosunku do ich łącznej wielkości (sumarycznej liczby elementów). Predykcja realizowana jest za pomocą sieci FCM. Efektem predykcji na poziomie pojęciowym, jest pojęcie FCM o największym przewidywanym stopniu aktywacji. Na poziomie numerycznym, efektem predykcji jest granula szeregu GTS przewidywana w kolejnym przedziale czasowym, opisana w formie parametrów liczby rozmytej. Wartości tych parametrów uzyskiwane są w wyniku zaproponowanej w artykule [P6] metody degranulacji.

W artykule [P6] zaproponowano:

- nową, dwu-fazową metodę granulacji szeregu czasowego,
- nową funkcję oceniającą pokrywanie się granul informacji,
- nowy sposób degranulacji, umożliwiający obliczenie parametrów predykowanej granuli na podstawie przewidywanego stanu wszystkich pojęć sieci FCM.

Rezultaty rozwiązania 7 [P6].

Wyniki przeprowadzonych eksperymentów przedstawiono w pracy [P6]. Dowodzą one, że zaproponowana metoda modelowania może zostać skutecznie zastosowana w celu aproksymacji i predykcji szeregów czasowych. Zbadano pięć szeregów czasowych o różnej charakterystyce. Predykcja szeregu czasowego zawierającego stabilny cykl (sezonowość) okazała się najbardziej efektywna. Zgodnie z oczekiwaniami, proponowana metoda staje się mniej efektywna dla szeregów zawierających silniejszy komponent losowy. Rezultaty eksperymentów wykazały ponadto, że liczba pojęć sieci FCM powinna być dobierana indywidualnie dla każdego szeregu czasowego. W ten sposób użytkownik posiada możliwość ustalenia wymaganego poziomu aproksymacji szeregu i w konsekwencji uzyskiwanego błędu predykcji.

4.3.5 Poprawa efektywności działania klasyfikatora opartego na sieci FCM [P7]

Problem

Sieć FCM stosowana jest zwykle jako narzędzie wspomaganie decyzji lub jako model predykcji szeregów czasowych. Mniej znane jest zastosowanie FCM w formie klasyfikatora. W moich dalszych badaniach podjąłem problem poprawy efektywności klasyfikacji uzyskiwanej za pomocą FCM.

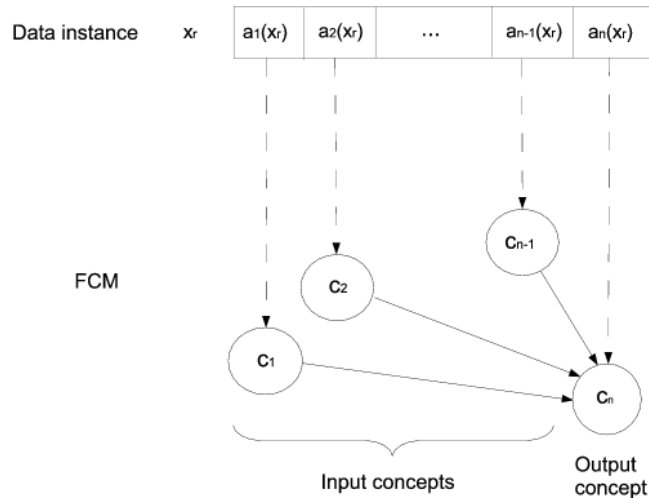
W artykule [P7] problem klasyfikacji został sformalizowany w następujący sposób.

Niech X jest zbiorem, którego elementy $x_r \in X$ opisane są za pomocą atrybutów ze zbioru A . Zakłada się, że atrybuty warunkowe $a_1, a_2, \dots, a_{n-1} \in A$ są typu numerycznego (całkowitego lub rzeczywistego) lub że są nominalne. Atrybut decyzyjny $a_n \in A$ jest zawsze nominalny, dziedziną a_n jest zbiór etykiet przyporządkowany poszczególnym klasom, tzn.: $domain(a_n) = \{class_1, class_2, \dots, class_m\}$.

Przez M oznaczamy klasyfikator. Dla każdego elementu x_r , klasyfikator przypisuje wartość atrybutu decyzyjnego na podstawie atrybutów warunkowych, argumentów klasyfikatora, tzn.: $a'_n(x_r) = M(a_1(x_r), a_2(x_r), \dots, a_{n-1}(x_r))$. Podczas testowania klasyfikatora, dla danego elementu x_r , predykowana klasa $a'_n(x_r)$ porównywana jest z wartością znaną na podstawie danych $a_n(x_r)$. Klasyfikacja jest poprawna gdy: $a'_n(x_r) = a_n(x_r)$.

W moich pracach rozważałem klasyfikator zbudowany w oparciu o rozmytą sieć kognitywną. Celem było uzyskanie klasyfikatora o jak najlepszej efektywności klasyfikacji mierzonej za pomocą wskaźników określonych w artykule [P7].

W pierwszej kolejności potwierdzono wyniki uzyskane w literaturze. Zastosowano algorytm genetyczny w celu uczenia wag sieci FCM. Przeprowadzone eksperymenty wykazały, że najlepsze wyniki klasyfikacji uzyskiwane są dla prostej struktury FCM przedstawionej na rysunku 6. W tym szczególnym przypadku, w sieci FCM wydzielono pojęcia wejściowe odpowiadające atrybutom warunkowym oraz jedno pojęcie wyjściowe odpowiadające atrybutowi decyzyjnemu.



Rysunek 6: Klasyfikator FCM

Wartości $a_i(x_r), i = 1, 2, \dots, n - 1$ atrybutów warunkowych dla elementu $x_r \in X$ podlegają normalizacji i stanowią stany pojęć $c_i, i = 1, 2, \dots, n - 1$ sieci FCM. Wartości nominalne atrybutu decyzyjnego a_n przekształcane są do odpowiadających wartości numerycznych $\alpha_k \in [0, 1]$. Odzworowanie to przeprowadzone jest zgodnie z formułą (17), zapewniającą zgodność indeksów wartości symbolicznych i numerycznych.

$$\{class_1, class_2, \dots, class_m\} \rightarrow \{\alpha_1, \alpha_2, \dots, \alpha_m\}. \quad (17)$$

W celu obliczenia α_k , dzielimy przedział $[0, 1]$ na $m - 1$ równych podprzedziałów, gdzie $m \geq 2$ jest liczbą klas w dziedzinie $domain(a_n)$. Zakresy tych przedziałów to wartości $\alpha_k, k = 1, \dots, m$ obliczone zgodnie ze wzorem (18):

$$\alpha_k = \frac{k-1}{m-1}. \quad (18)$$

W ten sposób, dla każdej klasy $class_k$ otrzymujemy odpowiadającą wartość $\alpha_k \in [0,1]$, która określa stan pojęcia wyjściowego c_n sieci FCM.

W powyżej opisany sposób, dla każdego x_r otrzymujemy wektor stanów poszczególnych pojęć FCM; tzn. $[c_1(r), c_2(r), \dots, c_{n-1}(r), c_n(r)]$.

Klasyfikacja realizowana jest z zastosowaniem wzoru wnioskowania (1). W efekcie uzyskujemy przewidywaną wartość $c'_n(r)$ pojęcia wyjściowego. W celu realizacji klasyfikacji, wartość $c'_n(r)$ zostaje przypisana do jednej z wartości α_k , odpowiednika klasy $class_k$.

Problemem podejmowanym w artykule [P7] jest znalezienie wartości progów $tr_0 < tr_2 < \dots < tr_m$ umożliwiających dyskryminację wartości $c'_n(r)$ w celu zyskania jak najlepszego przyporządkowania $a'_n = class_k$. Szczegóły działania dyskryminacji, włącznie z przykładem numerycznym przedstawiono w artykule [P7].

Rozwiązanie 8 [P7]

W celu rozwiązania postawionego problemu zaproponowano nowy algorytm przedstawiony jako Algorytm 1. W artykule [P7] przedstawiono motywacje dotyczące opracowanego algorytmu oraz porównano jego działanie z dotychczas znanymi algorytmami realizującymi takie same zadanie.

Zasadniczym pomysłem realizowanym przez Algorytm 1 jest sekwencyjne przeszukiwanie zbioru stanów pojęcia wyjściowego uzyskiwanego dla podzbioru uczącego klasyfikator. Przeszukiwanie prowadzone jest w wybranych zakresach, w taki sposób, aby uzyskać możliwie najmniejszy błąd klasyfikacji dla całego zbioru uczącego. Efektem działania algorytmu są progi dyskryminacyjne $tr_0 < tr_2 < \dots < tr_m$ wykorzystywane do testowania klasyfikatora.

Wprowadzenie proponowanego algorytmu spowodowało, że uczenie klasyfikatora FCM składa się z dwóch faz: 1) uczenia wag sieci 2) ustalenia odpowiednich progów dyskryminacyjnych.

Proponowany Algorytm 1 działa w następujący sposób.

W pierwszej kolejności wprowadzamy dwie stałe tr_0 i tr_1 stanowiące progi trywialne, nie podlegające optymalizacji (wiersze 1 i 2 w pseudokodzie proponowanego algorytmu). Zakładamy, że $tr_0 = 0$ określa dolny zakres przedziału $[0,1]$, $tr_m = 1$ jest jego górnym zakresem.

Pętla FOR (wiersz 3) wybiera aktualnie testowany próg. Przykładowo dla problemu klasyfikacji binarnej, gdy $m = 2$, zadaniem pętli FOR jest znalezienie tylko jednego nietrywialnego progu tr_1 . Dla aktualnie analizowanego progu, w wierszu 4, inicjowana jest liczba osiągniętych błędów klasyfikacji, która w tym wypadku ustalana jest na wartość maksymalną $card(X_L)$ równą liczności zbioru uczącego.

Algorithm 1

Input: - Actual $c_n(r)$ and predicted $c'_n(r)$ states of the FCM output concept,
for the learning data set X_L ,
m - Number of classes to be discriminated,
d - Minimal distance between the thresholds (accuracy of the search),
 $\alpha_1 < \alpha_2 < \dots < \alpha_m$ - Numerical representatives of the class labels.
Output: Set of thresholds $tr_0 < tr_1 < \dots < tr_m$.

```
1.  $tr_0 = 0$  // first threshold, lower bound of the searched interval
2.  $tr_m = 1$  // last threshold, upper bound of the searched interval
3. FOR k=1 TO m-1 { // searching for the threshold  $tr_k$ 
4.    $err_{min} = card(X_L)$  ; //temporary number of the classification errors
5.    $tr = tr_{k-1} + d$  ; // the first possible value for the threshold  $tr_k$ 
6.   WHILE  $tr < 1$  { // a loop related to the search of the [0,1] interval
7.      $err = 0$  // current number of the classification errors
8.     FOR r=1 TO  $card(X_L)$  { // a loop related to the data instances
9.       IF ( $tr_{k-1} \leq c'_n(r) < tr$  AND NOT  $c_n(r) == \alpha_k$ )
10.         $err = err + 1$  ; // a classification error occurred
11.      } // end FOR r
12.      IF( $err < err_{min}$ ){// calculate the lower bound of  $tr$ 
13.         $err_{min} = err$  ;
14.         $tr_l = tr$  ;
15.      } // end IF
16.      IF( $err == err_{min}$ )// calculate the upper bound of  $tr$ 
17.         $tr_u = tr$  ; // end IF
18.       $tr = tr + d$  ; // the next threshold value to check
19.    } // end WHILE
20.     $tr_k = (tr_u + tr_l)/2$  // calculate  $tr_k$ 
21.} // end FOR k
```

Przeszukiwanie przedziału [0,1] rozpoczyna się od pierwszej wartości następującej po poprzednio wyszukanej wartości proggu (wiersz 5). Parametr d określa minimalną odległość między kolejnymi wartościami progów. Wartością zmiennej tr jest tymczasowa wartość proggu, dla której obliczany jest błąd klasyfikacji dla odpowiadającej klasy. Zadaniem pętli WHILE w wierszu 6 jest przeszukiwanie przedziału $[tr_{k-1} + d, 1]$ w celu znalezienia takiej wartości tr , która najlepiej dyskryminuje klasy $class_k$ oraz $class_{k-1}$. Dla każdej wartości tr , testowany jest cały zbiór uczący (wiersz 8). Błąd klasyfikacji występuje gdy przewidywany przez FCM stan pojęcia $c'_n(t + 1)$ należy do $k - tego$ przedziału lecz rzeczywisty, ustalony na podstawie danych stan $c_n(t + 1)$ nie jest równy α_k (wiersz 9). Jeżeli wystąpi błąd klasyfikacji, wartość zmiennej err jest inkrementowana w wierszu 10.

Celem instrukcji IF w wierszu 12 oraz kolejnych instrukcji w wierszach 13 i 14 jest obliczenie minimalnej liczby błędów klasyfikacji oraz odpowiednie ustalenie wartości proggu. Ponieważ wartość spodziewana dla zbioru testowego znajduje w przedziale $tr_k \in [tr_l, tr_u]$, dla zbioru uczącego obliczane są obydwa zakresy tego przedziału, dolny zakres tr_l w linii 14, górny zakres tr_u w linii 17. Środek tego przedziału przyjmowany jest jako poszukiwana wartość proggu tr_k (wiersz 20).

Proponowany algorytm posiada liniową złożoność obliczeniową ze względu na licznosc zbioru uczącego. Ponieważ uczenie wag FCM oraz ustalanie progów za pomocą proponowanego algorytmu przeprowadzane jest tylko dla danych uczących, czas wykonania algorytmu jest akceptowalny w praktyce (0.5 – 3 minuty dla procesora Intel Core M)

Otrzymane wartości progów $tr_0 < tr_1 < \dots < tr_m$ wykorzystywane są przy klasyfikacji elementów należących do zbioru testowego.

Rezultaty rozwiązania 8 [P7]

Wyniki klasyfikacji dla 5-krotnej walidacji krzyżowej zostały uzyskane dla 15 zbiorów wybranych z repozytorium UCI Machine Learning: 1.Breast, 2.Echocardiogram, 3.Pima, 4.Hepatitis, 5.Parkinson, 6.Mammographic, 7.Fertility, 8.ILPD, 9.Blood, 10.Iris, 13.Vehicles, 12.Wine, 11.Thyroid Transfusion, 14.Glass, 15.Ecoli.

Jak można zauważyć w Tabeli 8, proponowany klasyfikator FCM okazał się najlepszy dla siedmiu spośród analizowanych zbiorów.

Tabela 8: Porównanie klasyfikatorów ze względu na dokładność (ang. classification rate)

| Set | FCM | DT | NB | k-NN | SVM | FR | NN | PNN | RF |
|-----|--------------|-------|-------|--------------|-------|--------------|--------------|--------------|-------|
| 1 | 0.957 | 0.941 | 0.949 | 0.965 | 0.906 | 0.968 | 0.953 | 0.969 | 0.965 |
| 2 | 0.690 | 0.589 | 0.689 | 0.589 | 0.673 | 0.613 | 0.636 | 0.645 | 0.664 |
| 3 | 0.653 | 0.698 | 0.725 | 0.707 | 0.685 | 0.711 | 0.767 | 0.762 | 0.724 |
| 4 | 0.918 | 0.787 | 0.912 | 0.800 | 0.838 | 0.810 | 0.775 | 0.838 | 0.838 |
| 5 | 0.848 | 0.867 | 0.805 | 0.954 | 0.790 | 0.905 | 0.887 | 0.851 | 0.779 |
| 6 | 0.776 | 0.805 | 0.812 | 0.751 | 0.783 | 0.761 | 0.829 | 0.796 | 0.802 |
| 7 | 0.875 | 0.840 | 0.860 | 0.830 | 0.870 | 0.812 | 0.820 | 0.880 | 0.860 |
| 8 | 0.760 | 0.715 | 0.724 | 0.677 | 0.746 | 0.733 | 0.758 | 0.750 | 0.744 |
| 9 | 0.764 | 0.850 | 0.840 | 0.860 | 0.870 | 0.823 | 0.800 | 0.880 | 0.800 |
| 10 | 0.985 | 0.947 | 0.953 | 0.960 | 0.953 | 0.960 | 0.953 | 0.947 | 0.947 |
| 11 | 0.987 | 0.926 | 0.963 | 0.949 | 0.916 | 0.962 | 0.958 | 0.944 | 0.958 |
| 12 | 0.925 | 0.888 | 0.966 | 0.961 | 0.404 | 0.977 | 0.972 | 0.955 | 0.955 |
| 13 | 0.732 | 0.724 | 0.531 | 0.705 | 0.639 | 0.689 | 0.759 | 0.479 | 0.739 |
| 14 | 0.786 | 0.615 | 0.502 | 0.718 | 0.667 | 0.733 | 0.653 | 0.592 | 0.723 |
| 15 | 0.837 | 0.818 | 0.841 | 0.848 | 0.815 | 0.832 | 0.844 | 0.857 | 0.830 |

Wyniki testu Wilcoxon'a przedstawione w Tabeli 9 potwierdzają przewagę klasyfikatora FCM-A dla wybranych zbiorów danych.

Tabela 9: Test Wilcoxon'a

| Model 1 | Model 2 | p-value |
|---------|---------|---------|
| FCM-A | DT | 0.9609 |
| FCM-A | NB | 0.8029 |
| FCM-A | k-NN | 0.8257 |
| FCM-A | SVM | 0.9866 |
| FCM-A | FR | 0.8399 |
| FCM-A | NN | 0.6401 |
| FCM-A | PNN | 0.7246 |
| FCM-A | RF | 0.8183 |

Na podstawie uzyskanych wyników można stwierdzić, że po zastosowaniu proponowanego algorytmu, klasyfikator FCM jest bardzo efektywny, jego zastosowanie powinno zostać rozważone jako alternatywa dla innych, bardziej znanych klasyfikatorów.

4.4 Rozwój zastosowań sieci FCM

Przedstawione wcześniej nowości teoretyczne znalazły zastosowanie w następujących dziedzinach:

- predykcja wielowymiarowych szeregów czasowych dotyczących choroby zapalenia płuc [P1,P2,P10],
- predykcja wielowymiarowych szeregów czasowych dotyczących choroby nowotworu prostaty [P3],
- predykcja jednowymiarowych szeregów czasowych dla szeregów liniowych i stacjonarnych [P5],
- predykcja przedziałowych szeregów czasowych w meteorologii [P4].

W przypadku artykułów [P1,P2,P3] dotyczących problemów medycznych uzyskane rezultaty były pozytywnie przyjęte przez lekarzy.

Zależności reprezentowane przez sieć FGCM [P4] znalazły potwierdzenie w literaturze meteorologicznej.

Dodatkowym osiągnięciem praktycznym prowadzonych prac są propozycje nowatorskich zastosowań sieci FCM w następujących dziedzinach:

- predykcja wielowymiarowych szeregów czasowych w meteorologii [P8],
- system wspomaganie decyzji dla inwestorów giełdy papierów wartościowych [P11],
- odkrywanie wzorców w danych medycznych [P12].

W artykule [P8] zastosowano sieci FCM w celu predykcji danych meteorologicznych. Analizowano uczenie sieć FCM za pomocą algorytmu ewolucji różnicowej.

W artykule [P11] zaproponowano zastosowanie modelu FCM w celu odkrywania zależności między wskaźnikami analizy technicznej na giełdzie papierów wartościowych. Uzyskane rezultaty wykazały korzyści z zastosowania modelu FCM w porównaniu ze strategią inwestowania „kup i trzymaj”.

Celem pracy [P12] było zastosowanie FCM w systemie wspomaganie decyzji dotyczący cukrzycy. Artykuł [P12] zainicjował dalsze badania oraz powstanie szeregu nowatorskich rozwiązań teoretycznych opisywanych w dalszej kolejności jako pozostałe osiągnięcie.

5. Omówienie pozostałych osiągnięć naukowo – badawczych.

Jako pozostałe osiągnięcie naukowe wskazuję publikacje w zakresie:

- systemów wspomaganie decyzji dotyczących terapii i diagnozy medycznej (rozdział 5.1),
- systemów wspomaganie decyzji dotyczących efektywnego zarządzania zasobami wodnymi (rozdział 5.2).

Wszystkie wskazane artykuły zostały napisane po uzyskaniu przeze mnie tytułu doktora.

5.1 Wspomaganie decyzji dotyczących terapii i diagnozy medycznej

Publikacje posiadające współczynnik Impact Factor:

[D1] W. Froelich, R. Deja, G. Deja, Mining Therapeutic Patterns from Clinical Data for Juvenile Diabetes, *Fundamenta Informaticae*, Volume 127, Number 1-4, pp. 513-528, 2013, IOS Press.

[D2] R. Deja, W. Froelich, G. Deja. Differential sequential patterns supporting insulin therapy of new-onset type 1 diabetes. *BioMedical Engineering OnLine*, Volume 14, pp. 1: 13. 2015, BioMed Central (Springer Nature).

[D3] R. Deja, W. Froelich, G. Deja, A. Wakulicz-Deja, Hybrid Approach to the Generation of Medical Guidelines for the Insulin Therapy of Children, *Information Sciences*, DOI: 10.1016/j.ins.2016.07.066, 2016. 2016

[D4] W.Froelich, K.Wróbel, P.Porwik, Diagnosis of Parkinson's Disease Using Speech Samples and Threshold-based Classification, *Journal of Medical Imaging and Health Informatics*, 2015, American Scientific Publishers

Problem

W tej części moich badań podejmowałem problem dotyczący wspomagania terapii medycznej. Celem było odkrycie na podstawie danych historycznych modelu reprezentującego przebieg choroby. W pierwszej kolejności badania dotyczyły pacjentów chorych na cukrzycę typu 1.

Cukrzyca jest chorobą intensywnie rozprzestrzeniającą się na całym świecie. Pomimo licznych istniejących systemów wspomagania decyzji problem zarządzania poziomem glukozy we krwi pacjentów cierpiących na cukrzycę nie został w pełni rozwiązany i wciąż wzbudza zainteresowanie w świecie nauki. Modele odzwierciedlające reakcję organizmu pacjentów na zastrzyki insulinowe w formie zmieniającego się poziomu glukozy we krwi (glikemii) są jednym z aktualnych kierunków badań.

Podstawowym celem terapii cukrzycy jest doprowadzenie do stabilizacji poziomu glukozy we krwi pacjenta, czyli tzw. normoglikemii. W celu kontroli tego poziomu przeprowadza się w ciągu dnia wielokrotne pomiary. W przypadku przekroczenia bezpiecznego zakresu glukozy pacjentowi podawane są w formie zastrzyków określone dawki insuliny. Problemem jest ustalanie wymaganych dawek insuliny.

Rozwiązanie 9 [D1]

W celu wspomagania decyzji lekarza dotyczących ustalania wymaganych dawek insuliny zaproponowano nową metodę opartą na modelowaniu czasowej zależności poziomu glukozy od podanej insuliny. Proponowana metoda opiera się na odkrywaniu wyspecjalizowanych wzorców sekwencji z danych historycznych. W pierwszej kolejności zaproponowano realizację abstrakcji funkcjonalnej dotyczącej poziomu glikemii w okresie nocnym. Kolejnym krokiem budowy modelu było zaproponowanie meta-wzorców definiowanych przez lekarzy i wykorzystywanych w celu odkrywania wzorców sekwencji. Zaproponowane meta-wzorce pozwalają na odkrywanie wzorców sekwencji ukierunkowanych na osiągnięcie celu terapeutycznego, czyli normoglikemii. W celu zastosowania proponowanej metodologii,

zostały zdefiniowane dwu- i trzelementowe meta-wzorce. W części eksperymentalnej pracy [D1] oceniono otrzymane wzorce sekwencji i wykazano użyteczność zaproponowanej wcześniej nowości teoretycznej przy wspomaganiu decyzji lekarskich.

Rozwiązanie 10 [D2]

Komplementarnym rozwiązaniem w stosunku do tego prezentowanego w pracy [D1] jest propozycja wprowadzenia pojęcia różnicowych wzorców sekwencji (ang. differential sequential patterns DSPs). Pojęcie to wprowadzono w pracy [D2] w celu umożliwienia reprezentacji zmian w poziomie glikemii oraz dawek insulinowych. W pracy [D2] zastosowano algorytm odkrywania różnicowych wzorców sekwencji. Poprzez zastosowanie takich wzorców ułatwiono lekarzowi podejmowanie decyzji w zakresie zmian w prowadzonej terapii (tzn. zmianie podawanie dawki insuliny). W pracy [D2] zaproponowano także alternatywny sposób oceny glikemii nocnej. Użyteczność proponowanej metody została zweryfikowana na podstawie rzeczywistych danych młodocianych pacjentów szpitala. Potwierdzono możliwość zastosowania różnicowych wzorców sekwencji jako narzędzia wspomagającego decyzje lekarza.

Rozwiązanie 11 [D3]

W celu wspomaganie decyzji lekarskich dotyczących terapii insulinowej, w pracy [D3] zaproponowano nowe podejście do automatycznego generowania zaleceń lekarskich (ang. medical guidelines). W pierwszej kolejności, statyczne dane dotyczące pacjentów np. wiek, waga, oraz inne zostały poddane grupowaniu. W ten sposób uzyskano grupy pacjentów o podobnej charakterystyce medycznej. W dalszej kolejności, dla każdego pacjenta określonej grupy zaproponowano model rozwoju choroby. Stosując dyskretyzację oraz wprowadzając symboliczną skalę czasu dane numeryczne pacjenta zostały przekształcone do postaci sekwencji zdarzeń. Zaproponowano pojęcie zdarzenie złożonego, reprezentującego jednostkę terapeutyczną dla terapii insulinowej, do posiłkowej. Sekwencja zdarzeń złożonych reprezentowała przebieg indywidualnej terapii pacjenta. W celu odkrywania wzorców w sekwencjach zdarzeń złożonych zaproponowano nowy algorytm oparty na idei epizodów częstych (ang. frequent episodes). Uzyskane wzorce zostały przedstawione lekarzom w formie grafu ilustrującego możliwy przebieg terapii. Zaproponowany model umożliwia reprezentację zarówno podobieństwa między różnymi typami terapii (dla grup pacjentów) jak i wzorców powtarzających się w obrębie terapii. W oparciu o taki model zrealizowano system wspomaganie decyzji medycznych. Efektywność systemu w sensie wiarygodności proponowanej terapii insulinowej [D3] została oceniona w oparciu o dane rzeczywiste zebrane na dziecięcym oddziale diabetologicznym szpitala w Katowicach.

Problem

Kolejnym etapem podjętych przeze mnie prac w zakresie medycznych systemów wspomaganie decyzji była diagnoza choroby Parkinsona na podstawie próbek głosu. Choroba Parkinsona przejawia się niesprawnością centralnego systemu nerwowego i charakteryzuje się szeregiem symptomów medycznych: nie zrównoważeniem, trudnością w mówieniu, powolnością ruchową. Jednym z symptomów charakteryzujących chorobę Parkinsona są zaburzenia mowy. Z tego powodu możliwa jest diagnostyka choroby na podstawie próbek głosu pacjenta.

Rozwiązanie 12 [D4]

W pracy [D4] zaproponowano dwustopniową metodę klasyfikacji próbek głosu w celu poprawy efektywności diagnozowania choroby Parkinsona. W pierwszym kroku, wektory zawierające wartości cech charakterystycznych głosu klasyfikowane są za pomocą klasyfikatorów standardowych, znanych z literatury. Dla każdego pacjenta możliwe jest przeprowadzenie dowolnej liczby pomiarów i uzyskanie różnej liczby wektorów charakterystycznych (różnej dla różnych pacjentów). W drugim kroku, sklasyfikowane wektory wartości cech podlegają grupowej, dodatkowej ocenie. W celu realizacji tej oceny zaproponowano metodę polegającą na stosowaniu ustalonego progu procentowego dotyczącego liczby klasyfikowanych wektorów wskazujących na chorobę. Wartość tego progu ustalana jest eksperymentalnie. Tylko w przypadku przekroczenia ustalonego progu, podejmowana jest decyzja dotycząca pozytywnego zdiagnozowania choroby. Eksperymenty zostały przeprowadzone wykorzystując publicznie dostępne dane medyczne. Ustalono, że drzewa decyzyjne są najlepszym standardowym klasyfikatorem możliwym do zastosowania w pierwszym kroku klasyfikacji. Dla kroku drugiego, ustalono że wartości progu w zakresie 20% – 40% wystarczają do uzyskania ponad 90% poprawnie zdiagnozowanych pacjentów. Wykazano przydatność zaproponowanej metody w diagnozowaniu choroby Parkinsona.

5.2 System wspomaganie efektywnym zarządzaniem zasobami wodnymi.

Jako drugie osiągnięcie, należące do grupy pozostałych osiągnięć, wskazują zbiór artykułów dotyczących systemu zarządzania zasobami wodnymi. Prace w tym zakresie prowadzone są w ramach projektu ISS-EWATUS, finansowanego przez 7 Program Ramowy Unii Europejskiej. W projekcie pełnią rolę reprezentanta naukowego konsorcjum.

[W1] E.Magiera, W.Froelich, Integrated Support System for Efficient Water Usage and Resources Managements, Proceedings of the Conference - WDSA'2014 - Water Distribution System Analysis, 14-17 July, 2014, Bari, Italy, Procedia Engineering, Volume 89 (2014), pp. 1066-1072, Elsevier

[W2] W. Froelich, Daily Urban Water Demand Forecasting - Comparative Study, Communications in Computer and Information Science, Proceedings of the conference BDAS 2016, pp 633-647, Springer

[W3] W.Froelich, Forecasting Daily Urban Water Demand Using Dynamic Gaussian Bayesian Network, Communications in Computer and Information Science, Proceedings of the conference BDAS 2015, pp. 333-342, Springer

[W4] E.Magiera, W.Froelich, Application of Bayesian Networks to the Forecasting of Daily Water Demand, Intelligent Decision Technologies, Proceedings of the Conference KES IDT 2015, pp. 385-393, Springer

[W5] W. Froelich, E. Magiera, Forecasting Domestic Water Consumption Using Bayesian Model, Intelligent Decision Technologies, Proceedings of the Conference KES IDT 2016, Springer

[W6] W.Froelich, Dealing with Seasonality While Forecasting Urban Water Demand, Intelligent Decision Technologies, Proceedings of the Conference KES IDT 2015, pp. 171-180, Springer

Problem

W sytuacji coraz większego deficytu wody, oszczędność wody oraz efektywne zarządzanie jej zasobami okazuje się ważnym problemem badawczym. Wzrost zapotrzebowania na wodę wywołany wzrostem liczby ludności, rozwojem przemysłowym oraz niekorzystna polityka w zakresie cen wody utrudniają dostęp do zasobów wodnych, szczególnie w krajach o niskim dochodzie narodowym. W celu rozwiązania problemu zarządzania zasobami wodnymi podejmowane są kolejne badania natury interdyscyplinarnej.

Rozwiązanie 13 [W1]

Moim osiągnięciem w przedstawionej tematyce badawczej jest zaprojektowanie zintegrowanego systemu wspomagania decyzji dotyczącego zarządzania zasobami wodnymi (ang. Integrated Support System for Efficient Water Usage and Resources Management ISS-EWATUS). Wspólnie z Dr Ewą Magierą zaprojektowałem system oraz przygotowałem wniosek projektowy, który uzyskał akceptację w ramach siódmego programu ramowego Unii Europejskiej. Projekt ten jest aktualnie realizowany. Strona internetowa projektu to <http://issewatus.eu/>. W projekcie uczestniczy 10 instytucji z różnych krajów Unii Europejskiej. W projekcie pełnię rolę reprezentanta naukowego konsorcjum (ang. scientific representant of the consortium) oraz przewodniczącego rady naukowej (ang. chair of the science and quality board). Głównym celem projektu ISS-EWATUS jest wykorzystanie istniejącego potencjału w zakresie oszczędności wody na poziomie domowym i miejskim. Na poziomie domowym ISS-EWATUS obejmuje projekt tanich urządzeń i oprogramowania wspierających konsumenta w zakresie oszczędności wody. Za pomocą inteligentnych sensorów (ang. smart meters) ISS-EWATUS informuje konsumentów o zużyciu wody w czasie zbliżonym do rzeczywistego. Na podstawie danych zebranych w mieszkaniach, system ocenia możliwości w zakresie oszczędności wody i udziela porad użytkownikom, także w zakresie zmiany zachowań. System oprogramowania składa się z części serwerowej oraz interfejsu uruchamianego na urządzeniach mobilnych (tablety, telefony). W ramach tego systemu zaimplementowałem część oprogramowania serwerowego dotyczącego predykcji zapotrzebowania na wodę oraz generowania porad dla konsumentów wody.

Innym elementem projektu ISS-EWATUS jest platforma socjalna używana w celu interakcji między użytkownikami wody. Dzięki temu między użytkownikami przekazywane są informacje na temat pozytywnych zachowań w zakresie oszczędności wody. Na poziomie miejskim celem projektu ISS-EWATUS jest zmniejszenie strat wody w sieci wodociągowej (upływy w tle typu „background”) poprzez redukcję ciśnienia wody. W zakresie tego systemu zaimplementowałem szereg algorytmów predykcji zapotrzebowania na wodę. Kolejny element wchodzący w zakres projektu ISS-EWATUS dotyczy opracowania systemu wspomagani decyzji w zakresie adaptacyjnych cen wody. W Tabeli 10 przedstawiono listę modułów wchodzących w skład projektu ISS-EWATUS.

Tabela 10: Podstawowe moduły projektu ISS-EWATUS

- | | |
|---|---|
| 1 | System informacyjny dla gospodarstw domowych , uruchamiany na urządzeniach mobilnych, informuje konsumentów na temat zużycia wody. |
| 2 | System wspomagania decyzji dla gospodarstw domowych , uruchamiany na urządzeniach mobilnych. |
| 3 | Platforma sieci socjalnej promująca zachowania sprzyjające oszczędności wody |
| 4 | System wspomagani decyzji ułatwiający redukcję strat wody w miejskiej sieci wodociągowej. |
| 5 | System wspomagania decyzji symulujący i oceniający adaptacyjne strategie ustalania cen wody. |
-

Implementacja i ocena praktyczna wszystkich proponowanych modułów realizowana jest w Polsce, w Sosnowcu oraz w Grecji w miejscowości Skiathos (na wyspie o tej samej nazwie). Moduł domowy zainstalowany został w czterdziestu mieszkaniach, po dwadzieścia w każdej z powyższych lokalizacji. Moduł miejski funkcjonuje aktualnie w obydwu miastach. Pozostałe moduły udostępnione są poprzez platformy internetowe.

Rozwiązanie 14 [W2]

Podczas implementacji projektu ISS-EWATUS powstał problem wyboru najlepszej metody predykcji zapotrzebowania na wodę w sieci miejskiej. Ponieważ w sieci wodociągowej zostały zainstalowane sensory oraz zawory regulujące ciśnienie wody w czasie zbliżonym do rzeczywistego, predykcja zapotrzebowania na wodę została zrealizowana przy założeniu ciągłego napływu danych. W pracy [W2] przeprowadziłem kompleksowe porównanie piętnastu modeli predykcyjnych. Dodatkowo uwzględniłem podejście polegające na uśrednianiu predykcji generowanych przez różne modele. Przeprowadzałem testy statystyczne danych źródłowych, analizę wpływu wartości brakujących i odstających oraz regresorów zewnętrznych (dotyczących pomiarów meteorologicznych) na jakość predykcji. Eksperymenty wykazały, że uśrednianie predykcji wybranych modeli jest najlepszą z rozważanych metod.

Rozwiązanie 15 [W3,W4,W5]

W pracach [W3,W4,W5] zaproponowałem zastosowanie sieci Bayesa do predykcji zapotrzebowania na wodę. W pracy [W3] były to dynamiczne Gaussowskie sieci Bayesa (zawierające zmienne losowe o ciągłych rozkładach prawdopodobieństwa) stosowane do predykcji zapotrzebowania na wodę w sieci miejskiej. Eksperymenty wykazały, że sieci Bayesa, których struktura uczona jest na podstawie danych, nie są najbardziej efektywne. Lepszą efektywność uzyskano dla sieci, których struktura została zaprojektowana przez eksperta. Eksperymenty wykazały, że tego typu sieci Bayesa są metodą konkurencyjną w stosunku do innych, znanych modeli predykcyjnych.

W pracy [W4], w tym samym celu co w artykule [W3] zastosowałem tradycyjne dyskretne sieci Bayesa. Z tego względu zastosowałem dyskretyzację szeregów czasowych oraz zaproponowałem nową miarę do oceny jakości predykcji takich zdyskretyzowanych szeregów. W pracy [W5] zastosowałem tradycyjne sieci Bayesa do predykcji zapotrzebowania na wodę w gospodarstwach domowych. W tym przypadku zastosowałem specyficzną, powiązaną z wybranymi okresami dnia i tym samym różną charakterystyką zużycia wody, symboliczną skalą czasu. Także w tym wypadku uzyskano zadawalające wyniki eksperymentów.

Rozwiązanie 16 [W6]

Kontrola ciśnienia wody w sieciach dystrybucyjnych wymaga realizacji wielokrokowej predykcji zapotrzebowania na wodę. W pracy [W6] podjąłem problem predykcji takiego szeregu czasowego w skali godzinowej 24 kroki wprzód. Podstawowym problemem wynikłym przy takim postawieniu zadania okazuje się równoczesne uwzględnienie różnego rodzaju sezonowości: dziennej (noc – dzień), tygodniowej (dni robocze – dni wolne od pracy), rocznej/miesięcznej (pory roku). W pracy [W6] zbadano trzy znane metody uwzględniania sezonowości przy konstrukcji modeli predykcyjnych: autoregresyjną (ustalanie autoregresorów o odpowiednim opóźnieniu czasowym), metoda różnicowania szeregu (ang. differentiation), oraz zastosowanie tzw. "dummy variables", sztucznie wprowadzonych regresorów oznaczających poszczególne sezony. W pracy [W6] przeprowadzono szczegółową analizę tych trzech

metod w zastosowaniu do hydrologicznych szeregów czasowych. Eksperymenty wykazały, że najlepszym okazał się prosty model regresji liniowej w połączeniu z zastosowaniem dodatkowych zmiennych typu „dummy variables”.

Literatura

- [1] B. Kosko, Fuzzy cognitive maps, *Int. J. Man-Mach. Stud.* 24 (1) (1986) 65--75.
- [2] J. Dickerson, B. Kosko, Virtual worlds as fuzzy cognitive maps, *Presence* 3 (2) (1994) 173--189.
- [3] J. L. Salmeron, E. I. Papageorgiou, A fuzzy grey cognitive maps-based decision support system for radiotherapy treatment planning, *Knowledge-Based Systems* 30 (1) (2012) 151--160.
- [4] J. L. Salmeron, Supporting decision makers with fuzzy cognitive maps, *Research-Technology Management* 52 (3) (2009) 53--59.
- [5] W. Foelich, J. L. Salmeron, Evolutionary learning of fuzzy grey cognitive maps for the forecasting of multivariate, interval-valued time series, *International Journal of Approximate Reasoning* 55 (5) (2014) 1319--1335.
- [6] W. Homenda, A. Jastrzebska, W. Pedrycz, Modeling time series with fuzzy cognitive maps, in: *FUZZ-IEEE 2014, Beijing, China, 2014*, pp. 2055--2062.
- [7] W. Lu, J. Yang, X. Liu, W. Pedrycz, The modeling and prediction of time series based on synergy of high-order fuzzy cognitive map and fuzzy c-means clustering, *Knowledge-Based Systems* 70 (0) (2014) 242 -- 255.
- [8] W. Froelich, E. I. Papageorgiou, Extended evolutionary learning of fuzzy cognitive maps for the prediction of multivariate time-series, in: *Fuzzy Cognitive Maps for Applied Sciences and Engineering*, Springer Berlin Heidelberg, 2014, pp. 121--131.
- [9] J. L. Salmeron, E. I. Papageorgiou, Fuzzy grey cognitive maps and nonlinear hebbian learning in process control, *Applied Intelligence* 41 (1) (2014) 223--234.
- [10] P. Juszczuk, W. Froelich, Learning fuzzy cognitive maps using a differential evolution algorithm, *Polish Journal of Environmental Studies* 12 (3B) (2009) 108--112.
- [11] E. I. Papageorgiou, J. L. Salmeron, A review of fuzzy cognitive maps research during the last decade, *IEEE Transactions on Fuzzy Systems* 21 (1) (2013) 66--79.
- [12] M. Glykas (Ed.), *Fuzzy Cognitive Maps, Advances in Theory, Methodologies, Tools and Applications*, Series: Studies in Fuzziness and Soft Computing, Springer, 2010.
- [13] B. Kosko, Differential hebbian learning, *Neural Networks for Computing*, American Institute of Physics April (1986) 277--282.
- [14] A. V. Huerga, A balanced differential learning algorithm in fuzzy cognitive maps, in: *Proceedings of the 16th International Workshop on Qualitative Reasoning, 2002*, pp. 1--7.
- [15] E. Papageorgiou, C. D. Stylios, P. P. Groumpos, Active hebbian learning algorithm to train fuzzy cognitive maps, *International Journal of Approximate Reasoning* 37 (3) (2004) 219--249.
- [16] E. I. Papageorgiou, C. D. Stylios, P. P. Groumpos, Fuzzy cognitive map learning based on nonlinear hebbian rule, in: *Australian Conference on Artificial Intelligence, 2003*, pp. 256--268.
- [17] W. Stach, L. Kurgan, W. Pedrycz, M. Reformat, Genetic learning of fuzzy cognitive maps, *Fuzzy Sets and Systems* 153 (3) (2005) 371--401.
- [18] E. I. Papageorgiou, K. E. Parsopoulos, C. D. Stylios, P. P. Groumpos, M. N. Vrahatis, Fuzzy cognitive maps learning using particle swarm optimization, *Journal of Intelligent Information Systems* 25 (2005) 95--121.
- [19] S. Alizadeh, M. Ghazanfari, Learning fcm by chaotic simulated annealing, *Chaos, Solitons & Fractals* 41 (3) (2009) 1182--1190.
- [20] J. Arroyo, C. Mate, Introducing interval time series: Accuracy measures, in: *COMPSTAT 2006, proceedings in computational statistics, Heidelberg: Physica-Verlag, 2006*, pp. 1139--1146.

- [21] J. Arroyo, A. M. S. Roque, C. Mate, A. Sarabia, Exponential smoothing methods for interval time series, in: Proceedings of the 1st European Symposium on Time Series Prediction, 2007, pp. 231--240.
- [22] E. de A. Lima Neto, F. de A. T. de Carvalho, Constrained linear regression models for symbolic interval-valued variables., in: Computational Statistics and Data Analysis, 2010, pp. 333--347.
- [23] L. He, C. Hu, Impacts of interval computing on stock market variability forecasting, Computational Economics 33 (3) (2009) 263--276.
- [24] A. L. S. Maia, F. d. A. de Carvalho, Holt's exponential smoothing and neural network models for forecasting interval-valued time series, International Journal of Forecasting 27 (3) (2011) 740--759.
- [25] J. L. Salmeron, Modelling grey uncertainty with fuzzy grey cognitive maps, Expert Systems with Applications 37 (2010) 7581--7588.
- [26] N. C. D. Center, Quality controlled local climatological data, <http://cdo.ncdc.noaa.gov/qclcd/QCLCD?prior=N>
- [27] T. Dietterich, Approximate statistical tests for comparing supervised classification learning algorithms, Neural Computation 10, 1895--1923, 1998.

M. Froehic