

246/2008

**Raport Badawczy**  
**Research Report**

**RB/48/2008**

**Próba zastosowania reguł  
decyzyjnych do klasyfikacji  
szeregów czasowych**

**M.Krawczak, G.Szkatuła**

**Instytut Badań Systemowych**  
**Polska Akademia Nauk**

**Systems Research Institute**  
**Polish Academy of Sciences**



# **POLSKA AKADEMIA NAUK**

## **Instytut Badań Systemowych**

ul. Newelska 6

01-447 Warszawa

tel.: (+48) (22) 3810100

fax: (+48) (22) 3810105

Kierownik Pracowni zgłaszający pracę:  
Prof. dr hab. inż. Janusz Kacprzyk

Warszawa 2008

# **Próba zastosowania reguł decyzyjnych do klasyfikacji szeregów czasowych**

Maciej Krawczak, Grażyna Szkatuła

## Spis treści:

- 1 Wprowadzenie
- 2 Ogólny opis zaproponowanego podejścia
- 3 Przykład obliczeniowy

## 1 WPROWADZENIE

Rozwój systemów informatycznych ułatwił gromadzenie dużych zbiorów danych dynamicznych dotyczących różnych zjawisk. Ważne stało się nie tylko efektywne przechowywanie takich danych, ale także ich analiza, zdolność interpretacji i wyciągania potencjalnie użytecznych i zrozumiałych dla człowieka wniosków. Wykorzystanie pełnej informacji zawartej w tych zbiorach oraz sformułowanie ukrytej w nich wiedzy zwykle przekracza możliwości percepcji człowieka. W związku z tym powstało duże zapotrzebowanie na nowe metody i narzędzia informatyczne, które mogą pomóc człowiekowi w odkrywaniu wiedzy z takich danych.

Problem klasyfikacji szeregów czasowych znajduje bardzo wiele zastosowań w różnorodnych dziedzinach, np. w medycynie, finansach, przemyśle. Istnieje wiele algorytmów specjalnie zaproponowanych dla tego typu zagadnień, np. stosujące drzewa decyzyjne (Rodriguez, Alonso, 2004), sieci neuronowe (Nanopoulos, Alcock, Manolopoulos, 2001), klasyfikatory Bayesowskie (Wu, Chang, 2004) i inne.

Wydaje się, że metody uczenia maszynowego na podstawie przykładów tworzące opisy klas w postaci zbioru reguł decyzyjnych, mających postać wyrażeń logicznych „JEŻELI są spełnione określone warunki TO zachodzi przynależność do określonej klasy”, mogą być z powodzeniem zastosowane do klasyfikacji szeregów czasowych, jednak po uprzedniej, wstępnej obróbce danych wejściowych.

W pracy podjęto próbę dokonania klasyfikacji szeregów czasowych z zastosowaniem zbioru reguł decyzyjnych. Przedstawiono ogólny opis zaproponowanego podejścia (rozdział 2), sposób wstępnego przygotowania danych, redukcji wymiaru szeregów czasowych poprzez określenie cech istotnych z zastosowaniem sieci heteroasocjacyjnej. Obliczenia wykonano na danych dostępnych poprzez Internet w bazie danych Uniwersytetu Irvine w Kalifornii, które są często stosowane przy testowaniu algorytmów do eksploracji danych (rozdział 3).

## 2. OGÓLNY OPIS ZAPROPONOWANEGO PODEJŚCIA

Można wyróżnić pięć głównych zagadnień związanych z zastosowaniem metod eksploatacji danych w analizie szeregów czasowych, bądź szeregów pseudo-czasowych (Rodrick i in., 2001), są to:

- indeksowanie, czyli dla danego szeregu czasowego  $Q$  oraz określonej miary podobieństwa szeregów czasowych  $D(Q, C)$  znalezienie podobnych szeregów  $C$  w rozpatrywanej bazie danych  $BD$ ,
- klastering, czyli znalezienie naturalnych grup podobnych szeregów czasowych w  $BD$  przy założonej mierze podobieństwa  $D(Q, C)$ ,
- klasyfikacja, czyli zakwalifikowanie niesklasyfikowanego szeregu czasowego  $Q$  do jednej z dwóch lub więcej określonych klas,
- podsumowanie, czyli dla bardzo długiego szeregu czasowego  $Q$  wygenerowanie (najlepiej w postaci graficznej) stosunkowo krótkiego przybliżenia szeregu  $Q$  eksponującego zasadnicze cechy szeregu,
- wykrywanie anomalii, czyli dla danego szeregu  $Q$  oraz określonego modelu „normalnego” przebiegu szeregu znalezienie wszystkich fragmentów szeregu  $Q$  zawierających anomalie.

W zasadzie, metody stosowane do analizy ww. zagadnień oparte są na określaniu i porównywaniu:

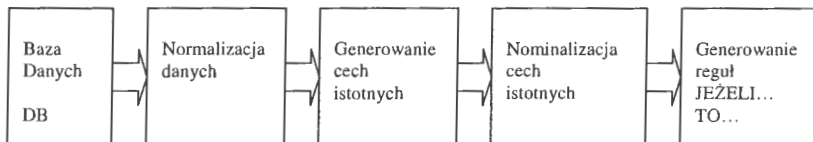
- odległości między szeregami czasowymi, albo
- odległości między cechami istotnymi szeregów czasowych,

przy czym odpowiednio zdefiniowane odległości oraz określona wartość progu  $\varepsilon$  określają podobieństwo szeregów.

Autorzy postanowili rozważyć szereg czasowy lub wybrany fragment tego szeregu czasowego jako obraz dwuwymiarowy. Zaproponowali koncepcję hybrydowego systemu do analizy szeregów czasowych. U podstaw koncepcji leży założenie, że cały szereg czasowy można traktować jako obraz lub też odpowiednie fragmenty szeregu czasowego mogą stanowić obrazy. Celem było opracowanie koncepcji systemu komputerowego, który po odpowiednim nauczaniu będzie rozwiązywał jedno z ww. zagadnień analizy szeregów czasowych.

Proponowany system składa się z czterech następujących modułów:

- modułu normalizacji danych, w tym wstępnej obróbki danych,
- modułu generowania cech istotnych,
- modułu nominalizacji cech istotnych,
- modułu generowania reguł.



### Moduł normalizacji danych

Przed analizą rozpatrywanych szeregów czasowych wartości tych szeregów muszą być poddane normalizacji, tzn. każdy szereg musi być tak przekształcony, aby jego wartość średnia była równa zero oraz odchylenie standardowe wynosiło jeden.

Dodatkowo, często zamiast oryginalnego szeregu czasowego rozpatruje się przyrosty sąsiadujących wartości. Przykładowo, rozważany szereg czasowy o postaci

$$\{x_i\}_{i=1}^{i=N} = [x_1, x_2, \dots, x_N]^T,$$

gdzie  $x_i \in R^1$ , może zostać zastąpiony szeregiem przyrostów, tzn.

$$\{x_{i+1} - x_i\}_{i=1}^{i=N-1} = [x_2 - x_1, x_3 - x_2, \dots, x_N - x_{N-1}]^T.$$

### Moduł generowania cech istotnych

Moduł generowania cech istotnych jest ściśle związany z problemem kompresji danych lub redukcją wymiarowości szeregów czasowych. Zadaniem kompresji danych jest takie zmniejszenie informacji o szeregu czasowym jako obrazu, aby można było odtworzyć ten szereg. Jest to dekompresja danych, przy możliwie małych stratach informacji w stosunku do informacji oryginalnej.

Zakłada się, że jeżeli rozpatrywany szereg lub jego fragment opisany jest wektorem składającym się z  $m$  elementów

$$\{\bar{x}_i\}_{i=1}^{i=m} = [x_1, x_2, \dots, x_m]^T$$

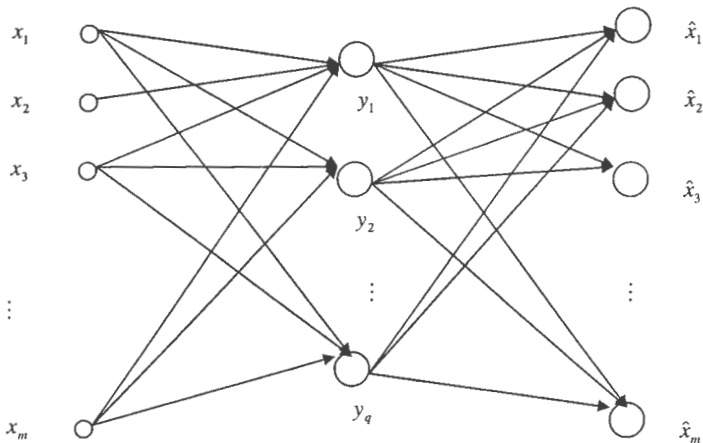
to po kompresji jest on reprezentowany przez  $q$  cech istotnych, tzn. wektor

$$\{y_i\}_{i=1}^{i=q} = [y_1, y_2, \dots, y_q]^T, \text{ gdzie } q \ll m.$$

Zakłada się, że na podstawie wektora  $y$  można odtworzyć wektor  $\bar{x}$  z pewną dokładnością jako  $\{\hat{x}_i\}_{i=1}^{i=m} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_m]^T$ .

Szereg czasowy jako obraz jest teraz reprezentowany przez wektor  $y$ , którego elementy tworzą cechy istotne, inaczej składniki główne (principal components) (Oja, 1992).

Jednym ze sposobów generowania składników głównych (w naszym przypadku cech istotnych) jest zastosowanie pamięci asocjacyjnej, realizowanej np. przez sieci neuronowe. Realizacja pamięci asocjacyjnej jest możliwa jako liniowa pamięć heteroasocjacyjna, czyli w postaci dwuwarstwowej jednokierunkowej sieci neuronowej z jedną warstwą ukrytą.



Wyjścia neuronów warstwy ukrytej tworzą właśnie wektor cech istotnych.

### **Moduł nominalizacji cech istotnych**

Otrzymane rzeczywiste wartości cech istotnych zamieniane są na wartości symboliczne. Przedziały zmienności cech istotnych dzielone są na określoną liczbę odcinków, których długość jest tak dobierana, aby liczność przykładów przypadająca na każdy odcinek była podobna. Poszczególnym odcinkom przyporządkowywane są symbole, np. w postaci kolejnych liczb całkowitych.

### **Moduł generowania reguł**

Zbiory tak określonych przykładów uczących (z ich podziałem na klasy) są punktem wyjścia w procesie uczenia maszynowego, w wyniku którego uzyskuje się opisy rozpatrywanych klas w postaci *reguł* mających postać wyrażen logicznych postaci "JEŻELI spełnione są określone warunki TO zachodzi przynależność do danej klasy".

Reguły muszą spełniać pewne dodatkowe wymagania. Powinny np. poprawnie opisywać "wszystkie" lub „prawie wszystkie” przykłady należące do rozpatrywanej klasy i nie opisywać "wszystkich" lub „prawie wszystkich” przykładów do tej klasy nienależących, mieć minimalną długość (np. w sensie liczby warunków tworzących regułę) itp.

Utworzone reguły mogą być użyte do klasyfikowania nowych szeregów, dla których nie jest znana przynależność do określonej klasy.



### 3. PRZYKŁAD OBLICZENIOWY

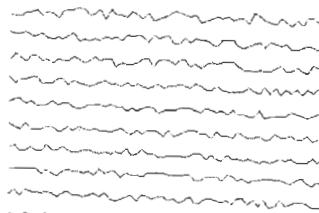
Do obliczeń wybrano zbiór danych dostępny poprzez Internet w bazie danych Uniwersytetu Irvine w Kalifornii, który jest często stosowany przy testowaniu algorytmów do eksploracji danych (Alcock i Manolopoulos, 1999):

[http://kdd.ics.uci.edu/databases/synthetic\\_control/synthetic\\_control.data.html](http://kdd.ics.uci.edu/databases/synthetic_control/synthetic_control.data.html)

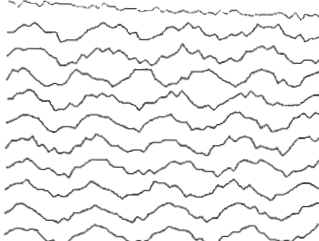
Zawiera on 600 szeregów czasowych, sztucznie wygenerowanych, z których każdy złożony jest z 60 liczb, określona jest też przynależność szeregu do jednej z sześciu klas: A, B, C, D, E lub F. Szeregi od 1-100 należą do klasy A, 101-200 należą do klasy B, 201-300 należą do klasy C, 301-400 należą do klasy D, 401-500 należą do klasy E, 501-600 należą do klasy F.

Poniżej przedstawiono po 10 przykładowych szeregów dla każdej klasy:

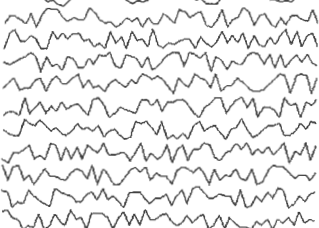
Klasa A:  
normalny (normal)



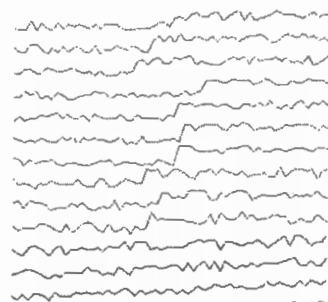
Klasa B:  
cykliczny (cyclical)



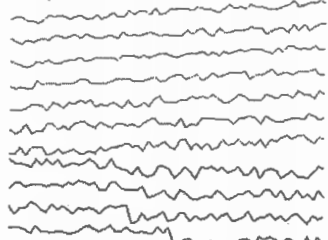
Klasa C:  
trend wzrostowy  
(increasing trend)



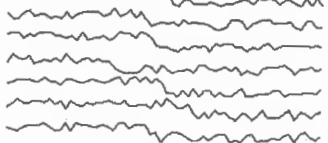
Klasa D:  
trend opadający  
(decreasing trend)



Klasa E:  
przesunięcie w górę  
(upward shift)



Klasa F:  
przesunięcie w dół  
(downward shift)

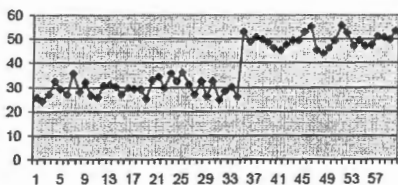


Do obliczeń wybrano dwie klasy:

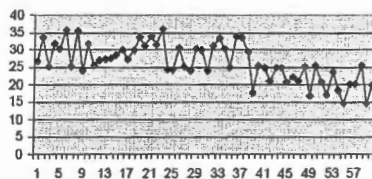
- klasa 1: szeregi typu E (25 szeregów uczących)
- klasa 2: szeregi typu F (25 szeregów uczących).

Celem naszym było utworzenie zbioru minimalnych reguł decyzyjnych dla obu rozpatrywanych klas, które pozwalałyby klasyfikować nowe przykłady szeregów czasowych, takie, które nie brały udziału w procesie tworzenia reguł.

Poniżej przedstawiono przykładowe szeregi z obu rozpatrywanych klas.



Przykład szeregu typu E



Przykład szeregu typu F

Poniżej przedstawiono 7 pierwszych wartości z 10 pierwszych szeregów z klasy 1:

28.7812	34.4632	31.3381	31.2834	28.9207	33.7596	25.3969 ...
24.8923	25.7410	27.5532	32.8217	27.8789	31.5926	31.4861 ...
31.3987	30.6316	26.3983	24.2905	27.8613	28.5491	24.9717 ...
25.7740	30.5262	35.4209	25.6033	27.9700	25.2702	28.1320 ...
27.1798	29.2498	33.6928	25.6264	24.6555	28.9446	35.7980 ...
25.5067	29.7929	28.0765	34.4812	33.8000	27.6671	30.6122 ...
28.6989	29.2101	30.9291	34.6229	31.4138	28.4636	35.9115 ...
30.9493	34.317	35.5674	34.8829	30.6691	35.2667	35.8950 ...
35.2538	34.6402	35.7584	28.5510	25.6518	29.6442	31.9400 ...
29.1734	31.5089	33.1944	35.6177	31.5890	35.1223	28.6222 ...
...						

Zgodnie z powyższymi ustaleniami, rozpatrywano szeregi czasowe  $O_n$ ,  $n = 1, 2, \dots, 50$ . Pierwszych 25 szeregów przynależało do klasy 1, pozostałe do klasy 2. Każdy  $n$ -ty szereg zawierający 60 wartości liczbowych zapisano w postaci:

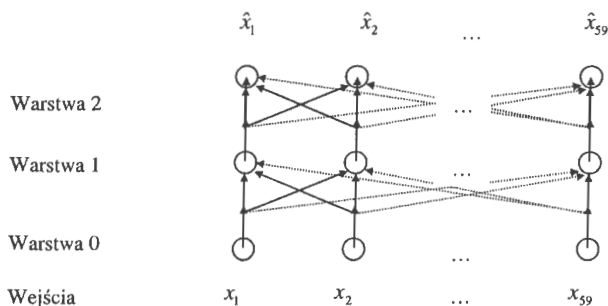
$$x_k(n), k = 1, 2, \dots, 60 \text{ dla } n = 1, 2, \dots, 50.$$

Następnie wartości każdego  $n$ -tego szeregu czasowego były wstępnie przetwarzane w następujący sposób:

$$y_k(n) = [x_{k+1}(n) - x_k(n)] / 10, k = 1, 2, \dots, 59.$$

Tak więc każdy przykład nowego, przetworzonego szeregu czasowego  $R_n$ ,  $n=1, \dots, 50$  (szeregu różnic) był reprezentowany przez 59 wartości liczbowych.

Do kompresji przetworzonych szeregów czasowych zastosowano dwuwarstwową sieć neuronową o strukturze przedstawionej na poniższym rysunku. Jest to sieć neuronowa, w której jest 59 wejść, 59 wyjść oraz w warstwie ukrytej tylko 5 neuronów. Neurony w warstwie ukrytej oraz wyjściowej mają liniowe funkcje aktywacji.



Neurony w warstwie ukrytej odpowiadają cechom istotnym, które opisują rozpatrywane szeregi czasowe różnic.

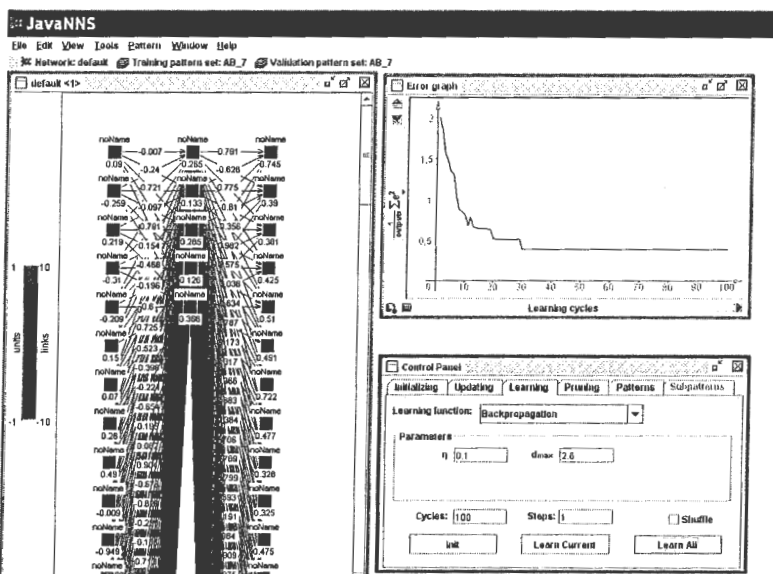
Do wyznaczenia wartości cech istotnych opisujących szeregi czasowe różnic został użyty program Java Neural Networks Simulator (JavaNNS). W celu uruchomienia programu komputerowego przygotowano następujące pliki:

AB\_7.net – przechowujący opis struktury i wartości wag sieci neuronowej

AB\_7.pat – przechowujący dane do uczenia sieci neuronowej

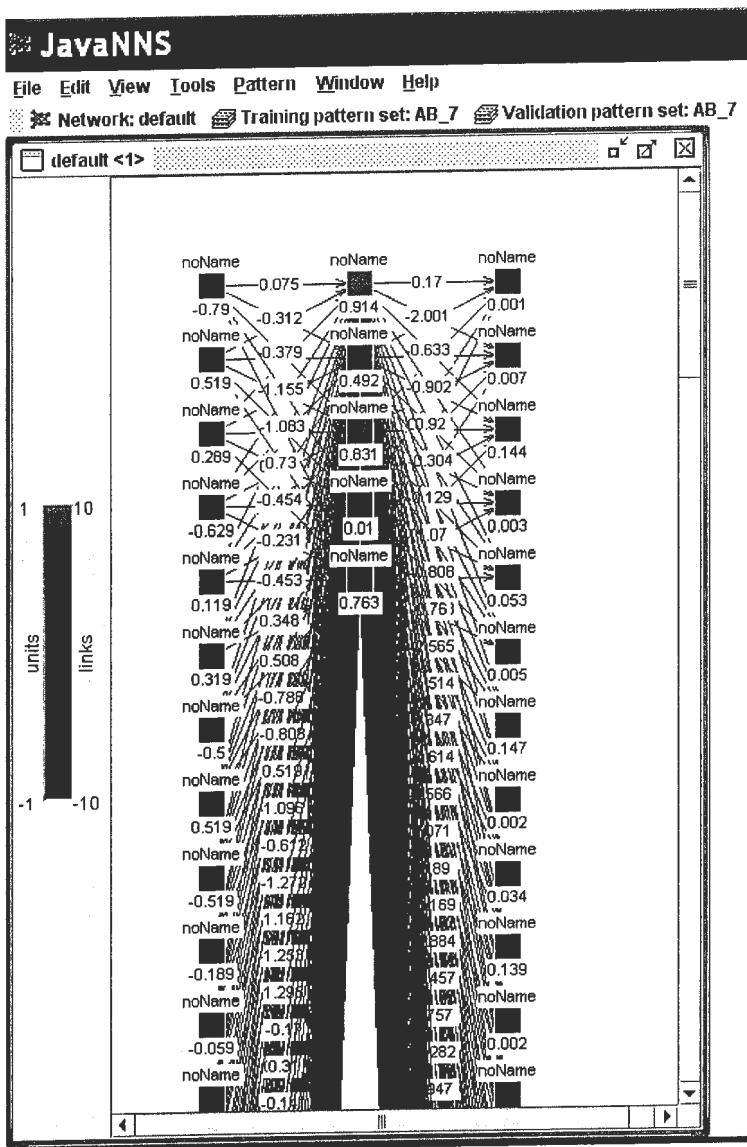
AB\_7\_test.pat – przechowujący dane do testowania sieci.

Poniższy rysunek przedstawia zrzut ekranu, na którym jest widoczna sieć neuronowa, wykres błędu uczenia oraz panel sterowania do zarządzania procesem uczenia i testowania sieci neuronowej.



Poniżej przedstawiono zrzut ekranu, na którym widoczna jest nauczona sieć. Do nauki posłużyły szeregi różnic  $R_n$ ,  $n=1, \dots, 50$  (25 szeregów typu E oraz 25 szeregów typu F). Przyjęto następującą wartość współczynnika uczenia  $\eta = 0.1$ , natomiast jako maksy-  
10

małą dopuszczalną różnicę między wzorcem a wyjściem sieci określono jako  $d_{max} = 2.6$ , dodatkowo przyjęto liczbę cykli uczenia jako 100.



Po procesie uczenia sieci, otrzymane dla każdego szeregu różnic wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis tego szeregu. Użytkano w ten sposób 50 szeregów, zawierających po 5 wartości liczbowych (cech istotnych). Poniżej zamieszczono przykładowo postać 11 takich skompresowanych szeregów (F określa przynależność do klasy).

	A	B	C	D	E	F
17	97	968	525	564	721	1
18	387	561	579	390	50	1
19	819	4	178	721	532	1
20	99	871	368	270	239	1
21	109	126	197	109	725	1
22	996	6	26	796	888	1
23	988	508	256	571	808	1
24	257	149	686	965	973	1
25	470	3	953	180	250	1
26	783	913	964	969	147	2
27	951	997	120	623	15	2

Następnie dokonano nominalizacji wartości cech istotnych. Określono 4 przedziały wartości w taki sposób, aby uzyskać w miarę możliwości równoliczną liczbę szeregów w każdym z nich. Zastosowany sposób nominalizacji wartości cech istotnych przedstawiono poniżej.

Wartości cech istotnych z przedziału:	Wartość nominalna
$\geq 0$ and $< 250$	1: mała
$\geq 250$ and $< 600$	2: średnia
$\geq 600$ and $< 850$	3: duża
$\geq 850$ and $\leq 1000$	4: bardzo duża

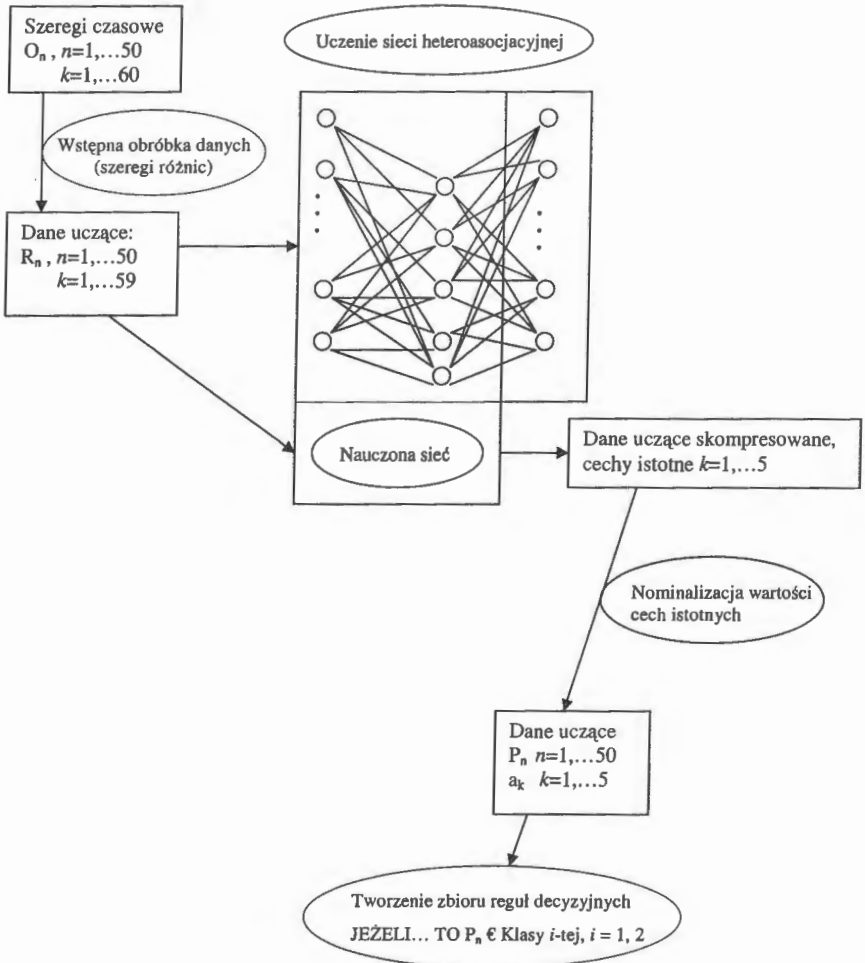
Poniżej przedstawiono szeregi uzyskane po nominalizacji,  $P_n$  dla  $n=1, \dots, 50$ .

lp.	a1	a2	a3	a4	a5	klasa
1	2	4	2	1	4	1
2	1	1	1	1	3	1
3	2	3	4	3	4	1
4	4	1	4	1	1	1
5	4	3	1	3	1	1
6	1	3	2	3	3	1
7	2	3	4	2	3	1

8	2	2	1	4	4	1
9	3	1	3	4	1	1
10	4	2	4	4	2	1
11	1	1	2	4	2	1
12	2	1	3	1	1	1
13	4	3	3	4	3	1
14	1	1	3	1	4	1
15	2	3	4	2	2	1
16	3	2	2	2	3	1
17	1	4	2	2	3	1
18	2	2	2	2	1	1
19	3	1	1	3	2	1
20	1	4	2	2	1	1
21	1	1	1	1	3	1
22	4	1	1	3	4	1
23	4	2	2	2	3	1
24	2	1	3	4	4	1
25	2	1	4	1	2	1
26	3	4	4	4	1	2
27	4	4	1	3	1	2
28	2	3	4	4	4	2
29	1	2	2	2	4	2
30	3	3	3	3	3	2
31	3	2	3	1	1	2
32	3	2	1	1	2	2
33	3	1	2	2	1	2
34	1	4	2	1	1	2
35	4	3	4	2	1	2
36	1	4	1	3	4	2
37	4	2	1	2	1	2
38	2	4	2	4	2	2
39	4	3	1	4	3	2
40	1	4	1	3	4	2
41	4	2	4	2	2	2
42	1	1	2	3	4	2
43	2	4	4	1	4	2
44	4	3	1	1	4	2
45	4	2	1	1	4	2
46	3	4	2	4	2	2
47	2	3	2	1	4	2
48	2	3	2	3	4	2
49	1	1	3	3	1	2
50	2	4	2	3	4	2

Powyższe dane mogą stanowić zbiór danych uczących dla różnych metod uczenia maszynowego na podstawie przykładów, tworzących reguły decyzyjne dla dwóch rozpatrywanych klas.

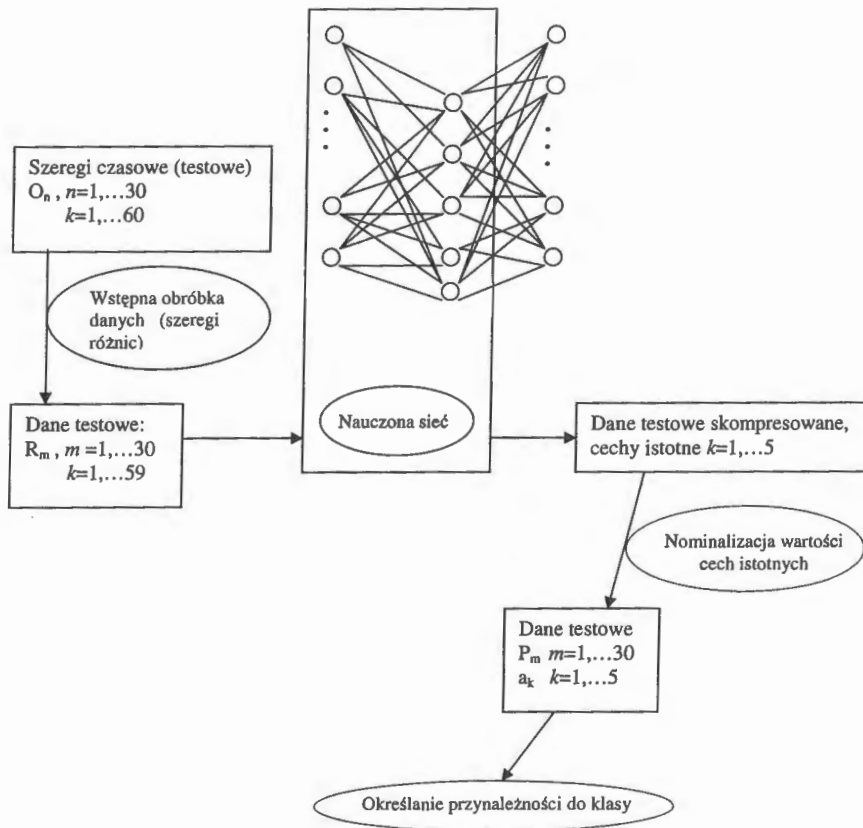
Poniżej przedstawiono uproszczony schemat zaproponowanego podejścia, omówionego powyżej, pozwalającego tworzyć reguły decyzyjne na podstawie zbioru 50 szeregów czasowych  $O_n$ , złożonych z 60 liczb, ze znaną przynależnością do klasy.





Dokładność klasyfikacji z zastosowaniem tak utworzonych reguł sprawdzana była dla zbioru testowego zawierającego 30 innych szeregów czasowych, nie stosowanych w procesie uczenia sieci heteroasocjacyjnej. Zbiór testowy zawierał 15 szeregów z klasy 1 i 15 szeregów z klasy 2.

Uproszczony schemat postępowania, w tym określanie cech istotnych dla 30 szeregów ze zbioru testowego przedstawiono poniżej.



Proces tworzenia opisu klasy na podstawie zbioru przykładów posiadających pewne wspólne właściwości, które odróżniają ją od innych klas, charakteryzowany jest poprzez przyjęty język reprezentacji danych oraz zastosowany algorytm uczenia maszynowego. Do reprezentowania zbioru przykładów uczących w pamięci komputera przyjęto tabelę, w której każdemu przykładowi i każdej cesze została jednoznacznie przyporządkowana wartość, jaką dana cecha przyjmuje w danym przykładzie. Korzystając z własnych doświadczeń, do obliczeń wybrano dwie modyfikacje metody IP wykorzystującej modyfikację zadania pokrycia zbioru, opisaną przez Szkatułę (1995), a następnie rozwiniętej przez Kacprzyka i Szkatułę (1999, 2002) oraz podejście oparte na zaproponowanej przez Pawlaka teorii zbiorów przybliżonych (1982).

Przyjęto, że reguła dla rozpatrywanej klasy jest zbiorem reguł elementarnych mających postać wyrażeń logicznych typu „JEŻELI... TO...”. Zbiór taki nazywamy kompletnym, jeżeli jest niepusty i skończony oraz dla każdego przykładu ze zbioru uczącego istnieje co najmniej jedna reguła, która go opisuje. Minimalnym zbiorem reguł nazywamy taki zbiór, który jest kompletny i zawiera minimalną liczbę reguł.

Dla rozpatrywanego zadania tworzono zarówno minimalne zbiory reguł decyzyjnych jak również zbiory wszystkich minimalnych reguł elementarnych dla rozpatrywanych klas oraz zbiory reduktów. Tworzone reguły elementarne charakteryzowane były przez współczynnik siły reguły, zależny od liczby przykładów opisywanych przez część warunkową każdej reguły.

Poniżej przedstawiono wyniki obliczeń, tzn. zbiory reguł elementarnych dla dwóch rozpatrywanych klas utworzone z zastosowaniem dwóch modyfikacji metody IP oraz wyniki klasyfikacji szeregów testowych, z zastosowaniem tych reguł. Klasyfikacja była dokonywana poprzez sprawdzanie spełnienia warunków w częściach przesłanowych reguł, a w przypadkach niejednoznacznych uwzględniany był największy stopień dopasowania do klasy (Szkatuła, 1995). Zamieszczono także zbiory wszystkich wygenerowanych minimalnych reguł decyzyjnych dla rozpatrywanych klas, oraz zbiory reduktów.

- Wyniki uzyskane z zastosowaniem metody IP (2)

Na podstawie zbioru uczącego zawierającego 25 przykładów należących do klasy 1 (du\_k1.ddd) oraz 25 przykładów należących do klasy 2 (du\_k2.ddd) utworzono zbiory reguł. Poniżej przedstawiono zrzut ekranu, na którym jest widoczny utworzony dla klasy 1 minimalny zbiór reguł elementarnych K1.fff.

```

TWORZENIE REGUŁ KLASYFIKACJI i KLASYFIKACJA PRZYKŁADÓW TESTOWYCH
2 q=0.040 [a2=1][a5=3]
3 q=0.060 [a1=2][a2=1]
4 q=0.090 [a4=2][a5=3]
3 q=0.060 [a2=1][a4=4]
2 q=0.040 [a1=4][a2=1]
1 q=0.020 [a2=3][a5=2]
1 q=0.020 [a1=2][a3=4][a4=3]
3 q=0.060 [a3=3][a4=4]
1 q=0.020 [a1=1][a3=3][a4=1]
2 q=0.040 [a1=2][a2=2]
1 q=0.020 [a4=3][a5=2]
1 q=0.020 [a1=1][a2=3]
1 q=0.020 [a2=3][a3=1][a5=1]
2 q=0.040 [a2=4][a3=2][a4=2]
1 q=0.020 [a3=4][a4=4][a5=2]
1 q=0.020 [a1=2][a2=4][a3=2][a4=1]

średnia q = 0.036

```

Zbiór K1.fff zawiera 16 reguł elementarnych dla klasy 1, zawierających w sumie 39 warunków. Poniżej zamieszczono dla każdej cechy  $a_k$ ,  $k=1, \dots, 5$ , tylko te wartości, które występują w warunkach tworzących część przestankową utworzonych reguł, wraz z sumą stopni przynależności reguł w których występują.

```

TWORZENIE REGUŁ KLASYFIKACJI i KLASYFIKACJA PRZYKŁADÓW TESTOWYCH
0.220 a1 <1, 0.040> <2, 0.140> <4, 0.040>
0.360 a2 <1, 0.200> <2, 0.040> <3, 0.060> <4, 0.060>
0.200 a3 <1, 0.020> <2, 0.060> <3, 0.080> <4, 0.040>
0.340 a4 <1, 0.040> <2, 0.120> <3, 0.040> <4, 0.140>
0.200 a5 <1, 0.020> <2, 0.060> <3, 0.120>

```

Poniżej przedstawiono zrzut ekranu, na którym jest widoczny utworzony dla klasy 2 minimalny zbiór reguł K2.fff.

```

TWORZENIE REGUŁ KLASYFIKACJI i KLASYFIKACJA PRZYKŁADÓW TESTOWYCH
3 q=0.060 [a2=4][a4=3][a5=4]
2 q=0.040 [a1=3][a2=4]
1 q=0.020 [a1=4][a2=3][a5=4]
3 q=0.060 [a2=2][a4=1]
1 q=0.020 [a2=3][a4=2][a5=1]
1 q=0.020 [a2=1][a3=2][a4=3]
1 q=0.020 [a2=4][a4=1][a5=1]
2 q=0.040 [a2=3][a3=2][a5=4]
1 q=0.020 [a1=4][a2=4]
1 q=0.020 [a2=2][a3=4][a4=2]
1 q=0.020 [a1=3][a4=2][a5=1]
1 q=0.020 [a2=3][a4=4][a5=4]

3 q=0.060 [a2=4][a4=4]
1 q=0.020 [a3=1][a4=2]
2 q=0.040 [a3=3][a4=3]
1 q=0.020 [a2=4][a3=4][a5=4]
1 q=0.020 [a1=1][a2=2]
1 q=0.020 [a2=3][a3=1][a4=4]

średnie q = 0.030

```

Zbiór K2.fff zawiera 18 reguł elementarnych dla klasy 2, zawierających w sumie 47 warunków. Poniżej zamieszczono dla każdej cechy  $a_k$   $k=1, \dots, 5$ , tylko te wartości, które występują w warunkach tworzących część przestankową tych reguł, wraz z sumą stopni przynależności reguł w których występują.

```

TWORZENIE REGUŁ KLASYFIKACJI i KLASYFIKACJA PRZYKŁADÓW TESTOWYCH
0.120 a1 <1, 0.020> <3, 0.060> <4, 0.040>
0.460 a2 <1, 0.020> <2, 0.100> <3, 0.120> <4, 0.220>
0.180 a3 <1, 0.040> <2, 0.060> <3, 0.040> <4, 0.040>
0.380 a4 <1, 0.080> <2, 0.080> <3, 0.120> <4, 0.100>
0.220 a5 <1, 0.060> <4, 0.160>

```

Następnie dokonano klasyfikacji przykładów ze zbioru uczącego (du\_k1.ddd oraz du\_k2.ddd) do jednej z dwóch rozpatrywanych klas, stosując wygenerowane reguły K1.fff oraz K2.fff. Uzyskano w 100% poprawną klasyfikację wszystkich 50 przykładów uczących, szczegóły przedstawiono poniżej.

Wyniki klasyfikacji do dwóch klas, dla przykładów ze zbioru uczącego

zbiory danych:	reguła: K1.fff	reguła: K2.fff
du_k1.ddd	25 przykładów	-
du_k2.ddd	-	25 przykładów

Następnie dokonano klasyfikacji 30 przykładów ze zbioru testowego (dt\_k1.ddd oraz dt\_k2.ddd). Uzyskano dokładność klasyfikacji 90%, szczegóły przedstawiono poniżej.

Wyniki klasyfikacji do dwóch klas, dla przykładów ze zbioru testowego

zbiory danych:	reguła: K1.fff	reguła: K2.fff
dt_k1.ddd	13 przykładów	2 przykłady
dt_k2.ddd	1 przykład	14 przykładów

- **Wyniki uzyskane z zastosowaniem metody IP (1)**

Na podstawie zbioru uczącego zawierającego 25 przykładów należących do klasy 1 (du\_k1.ddd) oraz 25 przykładów należących do klasy 2 (du\_k2.ddd) utworzono reguły: Kk1.fff oraz Kk2.fff do rozpatrywanych klas.

Następnie dokonano klasyfikacji przykładów ze zbioru uczącego, stosując wygenerowane reguły. Uzyskano w 100% poprawną klasyfikację wszystkich 50 przykładów uczących, szczegóły przedstawiono poniżej.

Wyniki klasyfikacji do dwóch klas, dla przykładów ze zbioru uczącego

zbiory danych:	reguła: Kk1.fff	reguła: Kk2.fff
du_k1.ddd	25 przykładów	-
du_k2.ddd	-	25 przykładów

Następnie dokonano klasyfikacji 30 przykładów ze zbioru testowego (dt\_k1.ddd oraz dt\_k2.ddd). Uzyskano dokładność klasyfikacji ok. 93%, szczegóły przedstawiono poniżej.

Wyniki klasyfikacji do dwóch klas, dla przykładów ze zbioru testowego

zbiory danych:	reguła: Kk1.fff	reguła: Kk2.fff
dt_k1.ddd	13 przykładów	2 przykłady
dt_k2.ddd	-	15 przykładów

• Tworzenie wszystkich minimalnych reguł decyzyjnych

Dla rozpatrywanego zadania określono 2 redukty, które zapewniają taką samą możliwość klasyfikacji przykładów uczących jak oryginalny zbiór cech:  $\{a_1, a_2, a_3, a_4\}$  oraz  $\{a_2, a_3, a_4, a_5\}$ . Jądro stanowią cechy:  $a_2, a_3, a_4$ , są one niezbędne w zbiorze cech, nie można ich pominąć bez utraty jakości klasyfikacji.

Poniżej przedstawiono zbiór wszystkich utworzonych 148 minimalnych reguł decyzyjnych do rozpatrywanych klas.

$a_1=3 \ a_3=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_2=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=4 \ a_3=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=4 \ a_4=4 \Rightarrow a_6=2$  ( 3 przykłady )  
 $a_3=4 \ a_4=4 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_2=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=4 \ a_3=1 \Rightarrow a_6=2$  ( 3 przykłady )  
 $a_2=4 \ a_4=3 \Rightarrow a_6=2$  ( 4 przykłady )  
 $a_1=2 \ a_2=3 \ a_4=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=3 \ a_3=4 \ a_4=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=3 \ a_4=4 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=2 \ a_3=4 \ a_4=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=4 \ a_4=4 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_2=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_3=2 \ a_5=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=2 \ a_3=2 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_4=2 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_2=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_4=3 \ a_5=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_3=3 \ a_5=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=3 \ a_4=3 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=2 \ a_3=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=2 \ a_4=1 \Rightarrow a_6=2$  ( 3 przykłady )  
 $a_1=3 \ a_2=2 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_4=1 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=3 \ a_2=2 \ a_3=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_2=2 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=1 \ a_4=1 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=2 \ a_3=1 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_2=1 \ a_3=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_3=2 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=1 \ a_3=2 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=1 \ a_4=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_4=2 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_2=4 \ a_4=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_3=2 \ a_4=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_4=1 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=4 \ a_4=1 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )

$a_3=2 \ a_4=1 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_2=3 \ a_3=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=3 \ a_3=4 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_2=3 \ a_4=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_3=4 \ a_4=2 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=4 \ a_4=2 \ a_5=1 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=3 \ a_4=2 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=4 \ a_4=2 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_2=4 \ a_5=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=1 \ a_3=1 \ a_4=3 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=1 \ a_3=1 \ a_5=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=1 \ a_4=3 \ a_5=4 \Rightarrow a_6=2$  ( 3 przykłady )  
 $a_1=4 \ a_2=2 \ a_3=1 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=4 \ a_2=2 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=2 \ a_3=1 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=1 \ a_4=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=2 \ a_3=2 \ a_4=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=2 \ a_3=2 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=2 \ a_4=4 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=4 \ a_5=2 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=4 \ a_3=1 \ a_4=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=3 \ a_3=1 \ a_4=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_3=1 \ a_5=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=3 \ a_3=1 \ a_5=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=1 \ a_4=4 \ a_5=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_4=2 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=2 \ a_3=4 \ a_4=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=2 \ a_4=2 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_2=1 \ a_4=3 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=1 \ a_3=2 \ a_4=3 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=1 \ a_3=2 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=2 \ a_4=3 \ a_5=4 \Rightarrow a_6=2$  ( 3 przykłady )  
 $a_3=4 \ a_4=1 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_2=3 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=4 \ a_4=1 \ a_5=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=3 \ a_4=1 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=4 \ a_3=1 \ a_4=1 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=3 \ a_3=1 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_3=1 \ a_4=1 \ a_5=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=4 \ a_2=2 \ a_5=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_3=2 \ a_4=4 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_3=2 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=3 \ a_4=4 \ a_5=2 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=2 \ a_2=3 \ a_3=2 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_2=3 \ a_3=2 \ a_5=4 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=2 \ a_3=2 \ a_4=3 \Rightarrow a_6=2$  ( 2 przykłady )  
 $a_1=1 \ a_2=1 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_3=3 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=1 \ a_4=3 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_2=1 \ a_4=3 \ a_5=1 \Rightarrow a_6=2$  ( 1 przykład )  
 $a_1=2 \ a_2=4 \ a_3=2 \ a_4=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=4 \ a_3=2 \ a_4=1 \ a_5=4 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_3=4 \ a_4=3 \Rightarrow a_6=1$  ( 1 przykład )

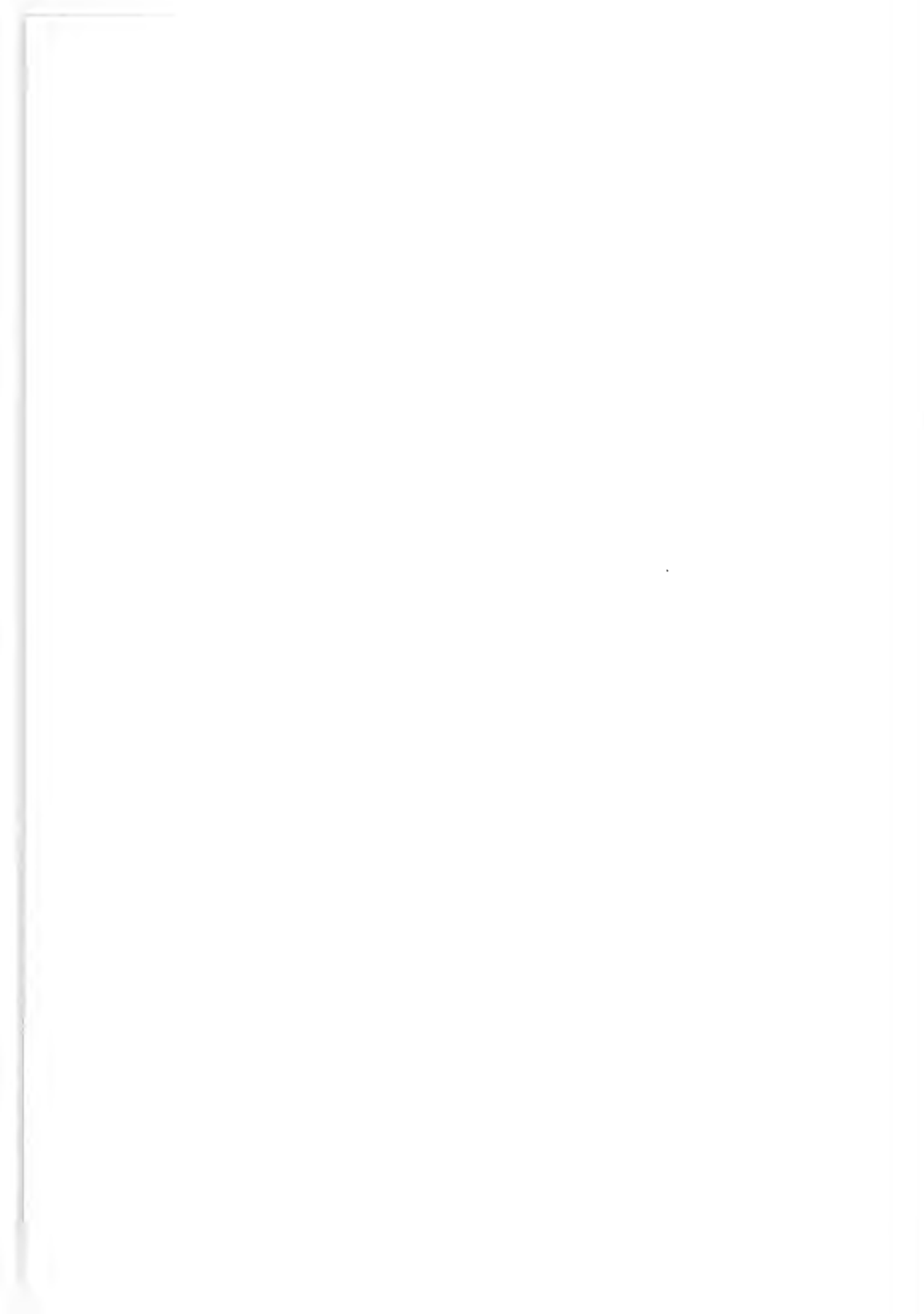
$a_1=4 \ a_2=1 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_1=4 \ a_3=4 \ a_4=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=4 \ a_4=1 \ a_5=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=1 \ a_4=1 \Rightarrow a_6=1$  ( 6 przykładów )  
 $a_2=1 \ a_3=4 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_3=4 \ a_4=1 \ a_5=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=4 \ a_2=3 \ a_4=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=3 \ a_3=1 \ a_4=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=3 \ a_3=1 \ a_5=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=3 \ a_4=3 \ a_5=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=1 \ a_2=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=1 \ a_5=3 \Rightarrow a_6=1$  ( 4 przykłady )  
 $a_3=2 \ a_5=3 \Rightarrow a_6=1$  ( 4 przykłady )  
 $a_1=2 \ a_4=2 \Rightarrow a_6=1$  ( 3 przykłady )  
 $a_4=2 \ a_5=3 \Rightarrow a_6=1$  ( 4 przykłady )  
 $a_1=2 \ a_5=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_3=4 \ a_5=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=2 \ a_2=2 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_2=2 \ a_4=4 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_1=2 \ a_3=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_3=1 \ a_4=4 \ a_5=4 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=3 \ a_2=1 \ a_3=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=1 \ a_4=4 \Rightarrow a_6=1$  ( 3 przykłady )  
 $a_3=3 \ a_4=4 \Rightarrow a_6=1$  ( 3 przykłady )  
 $a_1=4 \ a_3=4 \ a_4=4 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=4 \ a_4=4 \ a_5=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_3=4 \ a_4=4 \ a_5=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=1 \ a_4=4 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=1 \ a_5=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=1 \ a_5=2 \Rightarrow a_6=1$  ( 3 przykłady )  
 $a_1=2 \ a_3=3 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_1=2 \ a_5=1 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_1=2 \ a_2=1 \Rightarrow a_6=1$  ( 3 przykłady )  
 $a_1=4 \ a_3=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=1 \ a_3=3 \ a_4=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_3=3 \ a_5=4 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_1=1 \ a_4=1 \ a_5=4 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=3 \ a_5=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=2 \ a_3=4 \ a_5=2 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_1=3 \ a_2=2 \ a_3=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=3 \ a_2=2 \ a_4=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=2 \ a_5=3 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_2=4 \ a_4=2 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_2=4 \ a_5=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=2 \ a_3=2 \ a_5=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=3 \ a_3=1 \ a_4=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_2=1 \ a_3=1 \Rightarrow a_6=1$  ( 4 przykłady )  
 $a_1=3 \ a_2=1 \ a_4=3 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_4=3 \ a_5=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=1 \ a_4=2 \ a_5=1 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=1 \ a_3=1 \ a_4=1 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_2=1 \ a_5=3 \Rightarrow a_6=1$  ( 2 przykłady )  
 $a_4=1 \ a_5=3 \Rightarrow a_6=1$  ( 2 przykłady )



$a_1=4 \ a_4=3 \ a_5=4 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=4 \ a_3=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_1=2 \ a_4=1 \ a_5=2 \Rightarrow a_6=1$  ( 1 przykład )  
 $a_3=4 \ a_4=1 \ a_5=2 \Rightarrow a_6=1$  ( 1 przykład )

## Literatura

- Alcock, R. J., Manolopoulos, Y. (1999) Time-Series Similarity Queries Employing a Feature-Based Approach. 7<sup>th</sup> Hellenic Conference on Informatics, Ioannina, Greece.
- Benedikt, L., Kajic, V., Cosker, D., Marshall, D., Rosin, P. L. (2008) Facial Dynamics in Biometric Identification. In Proc. of British Machine Vision Conference, Leeds, 2008.
- Kacprzyk J., Szkatuła G. (1998) *IP1 - An Improved Inductive Learning Procedure with a Preprocessing of Data*. Proceedings of IDEAL'98 (Hong Kong), Springer-Verlag.
- Kacprzyk J., Szkatuła G. (1999) An inductive learning algorithm with a preanalysis od data. *International Journal of Knowledge - Based Intelligent Engineering Systems*, vol. 3, 135-146.
- Kacprzyk J., Szkatuła G. (2002) An integer programming approach to inductive learning using genetic and greedy algorithms. W: L.C. Jain and J.Kacprzyk (eds.) *New learning paradigms in soft computing*. Studies in Fuzziness and Soft Computing. Physica-Verlag Heidelberg, 323 - 367.
- Krawczak, M., Miklewski, A., Jakubowski, A., Konieczny, P. (2000) Investment Risk Management, (in Polish). Polish Academy of Sciences, Systems Research 25.
- Lin, J., Keogh, E., Lonardi, S., Chiu, B. (2003) A Symbolic Representation of Time Series, with Implications for Streaming Algorithms. Proceedings Data Mining and Knowledge Discovering, San Diego.
- Nanopoulos, A., Alcock, R., & Manolopoulos, Y. (2001): Feature-based Classification of Time-series Data. *International Journal of Computer Research*, 49-61.
- Oja E. (1992) Principal components, minor components and linear neural networks. *Neural Networks*, vol.5, ss 927-935.
- Pawlak Z. (1982): Rough Set. *International Journal of Computer and Information Sciences*, Vol. 11, No 5, 341-356.
- Pawlak Z. (1991): Rough Set. Theoretical Aspect of Reasoning about Data. Kluwer Academic Publishers.
- Roddick J. F., Hornsby K., Spilopoulos M. (2001): An updated bibliography of temporal, spatial and spatio-temporal data mining research. In Proceedings of the International Workshop on Temporal, Spatial and Spatio-Temporal data Mining. Berlin, Springer, Lecture Notes in Artificial Intelligence, 147-163.
- Rodríguez, J.J. & Alonso, C.J. (2004): Interval and dynamic time warping-based decision trees. In Proceedings of the 2004 ACM symposium on Applied computing (SAC), 548-552.
- Szkatuła G. (1995) Machine learning from examples under errors in data, Ph.D. thesis, SRI PAS Warsaw, Poland.
- Szkatuła G. (2002). *Zastosowanie zmodyfikowanego zadania pokrycia w uczeniu maszynowym*. W: Guttenbaum J. (eds.): *Automatyka Sterowanie Zarządzanie*. SRI PAS, Warszawa, 431-445.
- Wei, L., Keogh, E. (2006). Semi-Supervised Time Series Classification. In *Proc. of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD 2006)*, pp. 748 - 753, Philadelphia, PA, U.S.A., August 20-23, 2006.
- Wu, Y. & Chang, E.Y. (2004): Distance-function design and fusion for sequence data. *CIKM '04*, 324-333.



the 1990s, the number of people in the UK who are aged 65 and over has increased from 10.5 million to 13.5 million, and the number of people aged 75 and over has increased from 4.5 million to 6.5 million (Office for National Statistics 2000).

There is a growing awareness of the need to address the needs of older people, and the need to ensure that the health care system is able to meet the needs of older people. The Department of Health (2000) has published a strategy for older people, which sets out the government's commitment to older people and the need to ensure that the health care system is able to meet the needs of older people.

The strategy for older people is based on the following principles: (1) older people should be able to live independently in their own homes; (2) older people should be able to access the health care services that they need; (3) older people should be able to participate in the decisions that affect their lives; (4) older people should be able to live in a safe and secure environment; (5) older people should be able to access the services that they need to live well.

The strategy for older people is based on the following principles: (1) older people should be able to live independently in their own homes; (2) older people should be able to access the health care services that they need; (3) older people should be able to participate in the decisions that affect their lives; (4) older people should be able to live in a safe and secure environment; (5) older people should be able to access the services that they need to live well.

The strategy for older people is based on the following principles: (1) older people should be able to live independently in their own homes; (2) older people should be able to access the health care services that they need; (3) older people should be able to participate in the decisions that affect their lives; (4) older people should be able to live in a safe and secure environment; (5) older people should be able to access the services that they need to live well.

The strategy for older people is based on the following principles: (1) older people should be able to live independently in their own homes; (2) older people should be able to access the health care services that they need; (3) older people should be able to participate in the decisions that affect their lives; (4) older people should be able to live in a safe and secure environment; (5) older people should be able to access the services that they need to live well.

The strategy for older people is based on the following principles: (1) older people should be able to live independently in their own homes; (2) older people should be able to access the health care services that they need; (3) older people should be able to participate in the decisions that affect their lives; (4) older people should be able to live in a safe and secure environment; (5) older people should be able to access the services that they need to live well.

The strategy for older people is based on the following principles: (1) older people should be able to live independently in their own homes; (2) older people should be able to access the health care services that they need; (3) older people should be able to participate in the decisions that affect their lives; (4) older people should be able to live in a safe and secure environment; (5) older people should be able to access the services that they need to live well.

the 1990s, the number of people in the UK who are aged 65 and over has increased from 10.5 million to 13.5 million, and the number of people aged 75 and over has increased from 4.5 million to 6.5 million (Office for National Statistics 2000).

There is a growing awareness of the need to address the needs of older people, and the UK Government has set out a strategy for the 21st century (Department of Health 2000). The strategy is based on the principle of 'active ageing', which is defined as 'the process of optimising opportunities for health, participation in society and security in old age' (Department of Health 2000, p. 1).

The strategy is based on three pillars: health, participation and security. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action. The key areas for action are: health, participation, security, and the environment. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action.

The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action. The key areas for action are: health, participation, security, and the environment. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action.

The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action. The key areas for action are: health, participation, security, and the environment. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action.

The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action. The key areas for action are: health, participation, security, and the environment. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action.

The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action. The key areas for action are: health, participation, security, and the environment. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action.

The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action. The key areas for action are: health, participation, security, and the environment. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action.

The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action. The key areas for action are: health, participation, security, and the environment. The Department of Health has set out a number of objectives for each pillar, and has identified a number of key areas for action.