

205/2009

Raport Badawczy

RB/58/2009

Research Report

**Redukcja wymiarowości
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 2009

Redukcja wymiarowości szeregów czasowych

Maciej Krawczak, Grażyna Szkatuła

Spis treści:

Wprowadzenie	2
1. Koncepcja obwiedni szeregów czasowych	5
1.1. Obwiednie typu 1	5
1.2. Obwiednie typu 2	7
2. Agregacja obwiedni szeregów czasowych	9
2.1. Agregacja obwiedni typu 1	9
2.2. Agregacja obwiedni typu 2	11
3. Kompresja szeregów czasowych	13
4. Przykład obliczeniowy – zadanie klasyfikacji	14
4.1. Opis danych	14
4.2. Generowanie reguł decyzyjnych	16
4.3. Klasyfikacja do dwóch klas: E i F	18
4.3.1. Wyniki obliczeń dla obwiedni typu 1	19
4.3.2. Wyniki obliczeń dla obwiedni typu 2	24
4.4. Klasyfikacja do trzech klas: E, F i A	36
4.5. Klasyfikacja do trzech klas: B, F i A	45
4.6. Klasyfikacja do trzech klas: B, F i A (z normalizacją)	50
Literatura	57

Wprowadzenie

W ciągu ostatnich kilku lat można zaobserwować wzrastające zainteresowanie analizą danych w postaci strumieni danych (np. Xi i in., 2006). Zainteresowanie to jest wynikiem dostępności coraz większej ilości różnych urządzeń zbierających dane, które są następnie przechowywane w bazach danych. Dane te są bardzo często w postaci szeregów czasowych lub szeregów pseudo-czasowych. Typowymi przykładami tego rodzaju danych są wyniki rynków finansowych, dane medyczne, muzyczne, dane meteorologiczne czy też dane związane z genomami organizmów (np. Rodriguez, Alonso, 2004; Nanopoulos, Alcock, Manolopoulos, 2001; Wu, Chang, 2004).

Większość badań eksploracji danych w długich szeregach czasowych dotyczy następujących zagadnień (Roddick i in., 2001):

- *indeksowania*, tzn. dla określonego wzorcowego szeregu czasowego Q oraz określonej miary podobieństwa szeregów czasowych $D(Q, C)$ należy znaleźć najbardziej podobne szeregi C (do wzorcowego) w rozpatrywanej bazie danych BD ,
- *klasteringu*, tzn. znalezienie naturalnych grup szeregów czasowych w bazie danych BD zgodnie z przyjętą miarą podobieństwa szeregów $D(Q, C)$,
- *klasyfikacji*, tzn. przyporządkowanie szeregu czasowego Q do jednej z dwóch lub więcej określonych klas,
- *agregacji (summarization)* tzn. dużego zmniejszenia wymiarowości szeregu przy zachowaniu zasadniczych cech tego szeregu,
- *wykrywanie anomalii*, czyli wykrywanie sekcji szeregów czasowych zawierających anomalie lub z pewnych względów interesujące zmiany w szeregu.

Znane są podstawowe sposoby reprezentacji szeregów czasowych, m.in.:

- dyskretna transformata Fouriera,
- dyskretna transformata falkowa,
- aproksymacja odcinkami liniowa,
- aproksymacja odcinkami stała (schodkowa).

Każda z tych reprezentacji składa się z liniowych kombinacji odpowiednich funkcji bazowych, co wyklucza wprowadzenie funkcji odległości między szeregami, wymaganej przy porównywaniu szeregów czasowych.

Z drugiej strony, metody stosowane do analizy ww. zagadnień analizy szeregów czasowych oparte są właśnie na określaniu i porównywaniu:

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

przy czym odpowiednio zdefiniowane odległości oraz określona wartość progu ϵ określają podobieństwo szeregów.

W tej pracy rozważono zagadnienie klasyfikacji szeregów czasowych na podstawie zagregowanej reprezentacji szeregu czasowego. W tym celu zaproponowano:

- wprowadzenie obwiedni górnej i obwiedni dolnej – idea obwiedni powoduje utratę pewnej ilości informacji w stosunku do informacji zawartej w oryginalnym szeregu czasowym,
- dwa typy dyskretnych obwiedni szeregów czasowych, polegające na aproksymacji funkcjami odcinkami stałymi (schodkowymi),
- agregację obwiedni górnej oraz obwiedni dolnej,
- wyznaczenie wartości cech istotnych reprezentujących zagregowane obwiednie górne i dolne osobno, lub razem, przy wykorzystaniu sieci neuronowej,
- wyznaczenie reguł decyzyjnych mających postać wyrażen logicznych „JEŻELI są spełnione określone warunki TO zachodzi przynależność do określonej klasy”.

W konsekwencji, w pracy podjęto próbę dokonania klasyfikacji szeregów czasowych z zastosowaniem zbioru reguł decyzyjnych w oparciu o zagregowane cechy istotne obwiedni górnej lub/i obwiedni dolnej.

Obliczenia wykonano na danych dostępnych poprzez Internet w bazie danych Uniwersytetu Irvine w Kalifornii, które są często stosowane przy testowaniu algorytmów do eksploracji danych.

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.

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 wyrażeń 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.

1 Koncepcja obwiedni szeregów czasowych

Przyjmijmy, że dany jest zbiór N szeregów czasowych. Zakładamy, że rozpatrywany n -ty szereg lub jego fragment opisany jest wektorem składającym się z K elementów

$$\{x_k(n)\}_{k=1}^{k=K} = [x_1(n), x_2(n), \dots, x_K(n)]^T \quad (1)$$

dla $k = 1, 2, \dots, K$, $n = 1, 2, \dots, N$. Szereg czasowy (1) można też przedstawić jako szereg czasowy różnic:

$$\{x_k^1(n)\}_{k=1}^{k=K-1} = [x_1^1(n), x_2^1(n), \dots, x_{K-1}^1(n)]^T \quad (2)$$

gdzie

$$x_k^1(n) = [x_{k+1}(n) - x_k(n)], \text{ dla } k = 1, 2, \dots, K-1, \quad n = 1, 2, \dots, N.$$

Dla każdego szeregu o postaci (1) lub (2) tworzymy m -krokową obwiednię górną i obwiednię dolną. W tym opracowaniu wprowadzamy dwa typy obwiedni.

1.1. Obwiednie typu 1

Obwiednia górną, m -krokową, typu 1 $\{x_k^2(n)\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor (m-1)}$ jest zdefiniowana w następujący sposób:

$$\begin{aligned} x_1^2(n) &= \max \{(x_1^1(n), x_2^1(n), \dots, x_m^1(n))\} \\ x_2^2(n) &= \max \{(x_1^1(n), x_2^1(n), \dots, x_m^1(n))\} \\ &\dots \\ x_{m-1}^2(n) &= \max \{(x_1^1(n), x_2^1(n), \dots, x_m^1(n))\} \\ \\ x_m^2(n) &= \max \{(x_w^1(n), x_{m+1}^1(n), \dots, x_{2m-1}^1(n))\} \\ x_{m+1}^2(n) &= \max \{(x_m^1(n), x_{m+1}^1(n), \dots, x_{2m-1}^1(n))\} \\ &\dots \\ x_{2m-2}^2(n) &= \max \{(x_m^1(n), x_{m+1}^1(n), \dots, x_{2m-1}^1(n))\} \\ \\ x_{2m-1}^2(n) &= \max \{(x_{2m-1}^1(n), x_{2m}^1(n), \dots, x_{3m-2}^1(n))\} \\ x_{2m}^2(n) &= \max \{(x_{2m-1}^1(n), x_{2m}^1(n), \dots, x_{3m-2}^1(n))\} \\ &\dots \\ x_{3m-1}^2(n) &= \max \{(x_{2m-1}^1(n), x_{2m}^1(n), \dots, x_{3m-2}^1(n))\} \end{aligned} \quad (3)$$

$$x^2_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+2}(n) = \max \{ (x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+2}(n), x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n), \dots, x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)+1}(n)) \}$$

$$x^2_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n) = \max \{ (x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n), x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+4}(n), \dots, x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)+1}(n)) \}$$

$$\dots$$

$$x^2_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)}(n) = \max \{ (x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+2}(n), x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n), \dots, x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)+1}(n)) \}$$

W analogiczny sposób tworzymy m -krokową obwiednię dolną typu 1, ozn.

$\{x^3_k(n)\}_{k=1}^{k=\lfloor \frac{k-1}{m-1} \rfloor (m-1)}$, w sposób przedstawiony poniżej:

$$x^3_1(n) = \min \{ (x^1_1(n), x^1_2(n), \dots, x^1_m(n)) \}$$

$$x^3_2(n) = \min \{ (x^1_1(n), x^1_2(n), \dots, x^1_m(n)) \}$$

...

$$x^3_{m-1}(n) = \min \{ (x^1_1(n), x^1_2(n), \dots, x^1_m(n)) \}$$

$$x^3_m(n) = \min \{ (x^1_m(n), x^1_{m+1}(n), \dots, x^1_{2m-2}(n)) \}$$

$$x^3_{m+1}(n) = \min \{ (x^1_m(n), x^1_{m+1}(n), \dots, x^1_{2m-1}(n)) \}$$

...

$$x^3_{2m-1}(n) = \min \{ (x^1_m(n), x^1_{m+1}(n), \dots, x^1_{2m-1}(n)) \}$$

$$x^3_{2m-1}(n) = \min \{ (x^1_{2m-1}(n), x^1_{2m}(n), \dots, x^1_{3m-2}(n)) \}$$

$$x^3_{2m}(n) = \min \{ (x^1_{2m-1}(n), x^1_{2m}(n), \dots, x^1_{3m-2}(n)) \}$$

...

$$x^3_{3m-3}(n) = \min \{ (x^1_{2m-1}(n), x^1_{2m}(n), \dots, x^1_{3m-2}(n)) \}$$

...

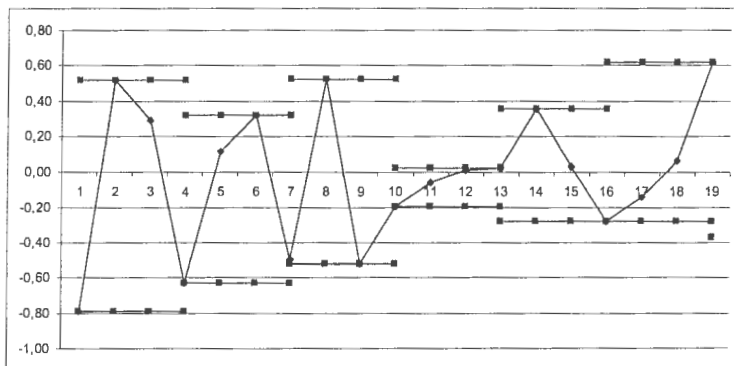
$$x^3_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+2}(n) = \min \{ (x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+2}(n), x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n), \dots, x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)+1}(n)) \}$$

$$x^3_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n) = \min \{ (x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n), x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+4}(n), \dots, x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)+1}(n)) \}$$

...

$$x^3_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)}(n) = \min \{ (x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+2}(n), x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)-m+3}(n), \dots, x^1_{\lfloor \frac{k-1}{m-1} \rfloor (m-1)+1}(n)) \}$$

Sposób tworzenia m -krokowych obwiedni typu 1 dla przykładowego szeregu (2) został zilustrowany graficznie na rysunku 1.



Rysunek 1. Obwiednia 4-krokowa g6rna typu 1 oraz 4-krokowa dolna typu 1 dla pierwszych 19 wartoŝci szeregu

1.2. Obwiednie typu 2

Dla kaŝdego szeregu czasowego (1) lub (2) tworzona jest m -krokowa obwiednia g6rna

typu 2, ozn. $\{x_k^2(n)\}_{k=1}^{\lfloor \frac{k}{m} \rfloor}$, w nast6puj1cy spos6b:

$$x_1^2(n) = \max \{x_1(n), x_2(n), \dots, x_m(n)\}$$

$$x_2^2(n) = \max \{x_1(n), x_2(n), \dots, x_m(n)\}$$

...

$$x_m^2(n) = \max \{x_1(n), x_2(n), \dots, x_m(n)\}$$

$$x_{m+1}^2(n) = \max \{x_{m+1}(n), x_{m+2}(n), \dots, x_{2m}(n)\}$$

$$x_{m+2}^2(n) = \max \{x_{m+1}(n), x_{m+2}(n), \dots, x_{2m}(n)\}$$

...

$$x_{2m}^2(n) = \max \{x_{m+1}(n), x_{m+2}(n), \dots, x_{2m}(n)\}$$

...

$$x_{\lfloor \frac{k}{m} \rfloor^{m-1}}^2(n) = \max \{x_{\lfloor \frac{k}{m} \rfloor^{m-1}}(n), x_{\lfloor \frac{k}{m} \rfloor^{m-2}}(n), \dots, x_{\lfloor \frac{k}{m} \rfloor^m}(n)\}$$

$$x_{\lfloor \frac{k}{m} \rfloor^{m-m}}^2(n) = \max \{x_{\lfloor \frac{k}{m} \rfloor^{m-1}}(n), x_{\lfloor \frac{k}{m} \rfloor^{m-2}}(n), \dots, x_{\lfloor \frac{k}{m} \rfloor^m}(n)\}$$

...

$$x_{\lfloor \frac{k}{m} \rfloor^m}^2(n) = \max \{x_{\lfloor \frac{k}{m} \rfloor^{m-1}}(n), x_{\lfloor \frac{k}{m} \rfloor^{m-2}}(n), \dots, x_{\lfloor \frac{k}{m} \rfloor^m}(n)\}$$

(5)

W analogiczny sposób tworzymy m -krokową obwiednię dolną typu 2, ozn.

$\left\{x_k^3(n)\right\}_{k=1}^{\left\lfloor \frac{k}{m} \right\rfloor^m}$, w sposób przedstawiony poniżej:

$$x_1^3(n) = \min\{x_1(n), x_2(n), \dots, x_m(n)\}$$

$$x_2^3(n) = \min\{x_1(n), x_2(n), \dots, x_m(n)\}$$

$$\dots$$

$$x_m^3(n) = \min\{x_1(n), x_2(n), \dots, x_m(n)\}$$

$$x_{m+1}^3(n) = \min\{x_{m+1}(n), x_{m+2}(n), \dots, x_{2m}(n)\}$$

$$x_{m+2}^3(n) = \min\{x_{m+1}(n), x_{m+2}(n), \dots, x_{2m}(n)\}$$

$$\dots$$

$$x_{2m}^3(n) = \min\{x_{m+1}(n), x_{m+2}(n), \dots, x_{2m}(n)\}$$

$$\dots$$

$$x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m+1}}^3(n) = \min\{x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m+1}}(n), x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m+2}}(n), \dots, x_{\left\lfloor \frac{k}{m} \right\rfloor^m}(n)\}$$

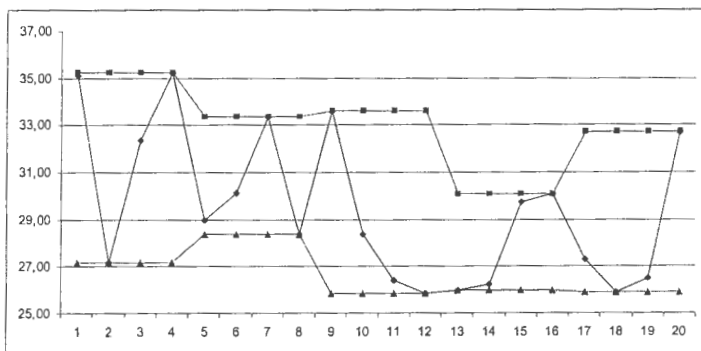
$$x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m}}^3(n) = \min\{x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m+1}}(n), x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m+2}}(n), \dots, x_{\left\lfloor \frac{k}{m} \right\rfloor^m}(n)\}$$

$$\dots$$

$$x_{\left\lfloor \frac{k}{m} \right\rfloor^m}^3(n) = \min\{x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m+1}}(n), x_{\left\lfloor \frac{k}{m} \right\rfloor^{m-m+2}}(n), \dots, x_{\left\lfloor \frac{k}{m} \right\rfloor^m}(n)\}$$

(6)

Sposób tworzenia m -krokowej obwiedni górnej typu 2 i dolnej typu 2 został zilustrowany graficznie dla przykładu obliczeniowego na rysunku 2.



Rysunek 2. Obwiednia 4-krokowa górna typu 2 oraz 4-krokowa dolna typu 2 dla pierwszych 20 wartości szeregu

2. Agregacja obwiedni szeregów czasowych

2.1. Agregacja obwiedni typu 1

Dla utworzonych m -krokowych obwiedni górnych typu 1, zapisanych w postaci:

$$\left\{ x_k^2(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor (m-1)} = \left[x_1^2(n), x_2^2(n), \dots, x_{\lfloor \frac{K-1}{m-1} \rfloor (m-1)}^2(n) \right]^T, \text{ dla } n=1, 2, \dots, N$$

dokonyjemy agregacji, otrzymując szeregi $\left\lfloor \frac{K-1}{m-1} \right\rfloor$ wymiarowe postaci:

$$\left\{ x_k^G(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor} = \left[x_1^G(n), x_2^G(n), \dots, x_{\lfloor \frac{K-1}{m-1} \rfloor}^G(n) \right]^T \quad (7)$$

Agregacja polega na tym, że w miejsce kolejnych $m-1$ powtarzających się wartości m -krokowych obwiedni górnych typu 1, ozn. $\left\{ x_k^2(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor (m-1)}$, tworzymy jedną wartość szeregu zagregowanego, ozn. $\left\{ x_k^G(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor}$, w sposób przedstawiony poniżej:

$$\begin{array}{l} \hline x_1^2(n) \\ x_2^2(n) \quad \Rightarrow \quad x_1^G(n) = \max\{x_1^1(n), x_2^1(n), \dots, x_m^1(n)\} \\ \dots \\ x_{m-1}^2(n) \\ \hline x_m^2(n) \\ x_{m+1}^2(n) \quad \Rightarrow \quad x_2^G(n) = \max\{x_m^1(n), x_{m+1}^1(n), \dots, x_{2m-1}^1(n)\} \\ \dots \\ x_{2m-2}^2(n) \\ \hline \dots \\ \dots \\ \hline x_{\lfloor \frac{K-1}{m-1} \rfloor (m-1) - m + 2}^2(n) \\ x_{\lfloor \frac{K-1}{m-1} \rfloor (m-1) - m + 1}^2(n) \quad \Rightarrow \quad x_{\lfloor \frac{K-1}{m-1} \rfloor}^G(n) = \max\{x_{\lfloor \frac{K-1}{m-1} \rfloor (m-1) - m + 2}^1(n), \dots, x_{\lfloor \frac{K-1}{m-1} \rfloor (m-1) - m + 1}^1(n)\} \\ \dots \\ x_{\lfloor \frac{K-1}{m-1} \rfloor (m-1)}^2(n) \\ \hline \end{array}$$

Następnie, dla utworzonych m -krokowych obwiedni dolnych typu 1, zapisanych w postaci:

$$\left\{ x_k^3(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor}^{(m-1)} = \left[x_1^3(n), x_2^3(n), \dots, x_{\lfloor \frac{K-1}{m-1} \rfloor}^3(n) \right]^T, \text{ dla } n = 1, 2, \dots, N$$

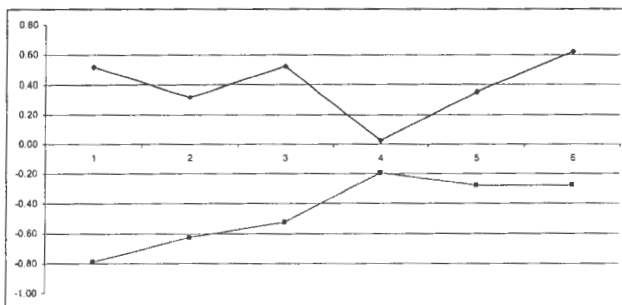
dokonyjemy agregacji, otrzymując szeregi $\left\lfloor \frac{K-1}{m-1} \right\rfloor$ wymiarowe postaci:

$$\left\{ x_k^D(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor} = \left[x_1^D(n), x_2^D(n), \dots, x_{\lfloor \frac{K-1}{m-1} \rfloor}^D(n) \right]^T \quad (8)$$

Agregacja polega na tym, że w miejsce kolejnych $m-1$ powtarzających się wartości m -krokowych obwiedni dolnych typu 1, ozn. $\left\{ x_k^3(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor}^{(m-1)}$, tworzymy jedną wartość szeregu zagregowanego, ozn. $\left\{ x_k^D(n) \right\}_{k=1}^{\lfloor \frac{K-1}{m-1} \rfloor}$, w sposób przedstawiony poniżej:

$$\begin{array}{l} \hline x_1^3(n) \\ x_2^3(n) \Rightarrow x_1^D(n) = \min \{ (x_1^1(n), x_2^1(n), \dots, x_m^1(n)) \} \\ \dots \\ x_{m-1}^3(n) \\ \hline x_m^3(n) \\ x_{m+1}^3(n) \Rightarrow x_2^D(n) = \min \{ (x_m^1(n), x_{m+1}^1(n), \dots, x_{2m-1}^1(n)) \} \\ \dots \\ x_{2m-2}^3(n) \\ \hline \dots \\ \dots \\ \hline x_{\lfloor \frac{K-1}{m-1} \rfloor}^3(n) \\ x_{\lfloor \frac{K-1}{m-1} \rfloor}^3(n) \Rightarrow x_{\lfloor \frac{K-1}{m-1} \rfloor}^D(n) = \min \{ (x_{\lfloor \frac{K-1}{m-1} \rfloor}^1(n), \dots, x_{\lfloor \frac{K-1}{m-1} \rfloor}^1(n)) \} \\ \dots \\ x_{\lfloor \frac{K-1}{m-1} \rfloor}^3(n) \\ \hline \end{array}$$

Sposób tworzenia zagregowanych m -krokowych obwiedni typu 1 (z rysunku 1) został zilustrowany graficznie dla przykładu obliczeniowego na rysunku 3.



Rysunek 3. Zagregowana obwiednia 4-krokowa górna typu 1 i 4-krokowa dolna typu 1

2.2. Agregacja obwiedni typu 2

Dla utworzonych m -krokowych obwiedni górnych typu 2, zapisanych w postaci:

$$\{x_k^2(n)\}_{k=1}^{\lfloor \frac{K}{m} \rfloor m} = \left[x_1^2(n), x_2^2(n), \dots, x_{\lfloor \frac{K}{m} \rfloor m}^2(n) \right]^T, \text{ dla } n = 1, 2, \dots, N$$

dokonywana jest agregacja wartości kolejno powtarzających się, w wyniku której otrzymywane są szeregi $\left\lfloor \frac{K}{m} \right\rfloor$ wymiarowe postaci:

$$\{x_k^G(n)\}_{k=1}^{\lfloor \frac{K}{m} \rfloor} = \left[x_1^G(n), x_2^G(n), \dots, x_{\lfloor \frac{K}{m} \rfloor}^G(n) \right]^T \quad (9)$$

Agregacja polega na tym, że w miejsce kolejnych m powtarzających się wartości m -krokowych obwiedni górnych typu 2 $\{x_k^2(n)\}_{k=1}^{\lfloor \frac{K}{m} \rfloor m}$, tworzymy jedną wartość szeregu zagregowanego $\{x_k^G(n)\}_{k=1}^{\lfloor \frac{K}{m} \rfloor}$, w sposób następujący:

$$\begin{matrix} x_1^2(n) \\ x_2^2(n) \\ \dots \\ x_m^2(n) \end{matrix} \Rightarrow x_1^G(n) = \max \{ (x_1(n), x_2(n), \dots, x_m(n)) \}$$

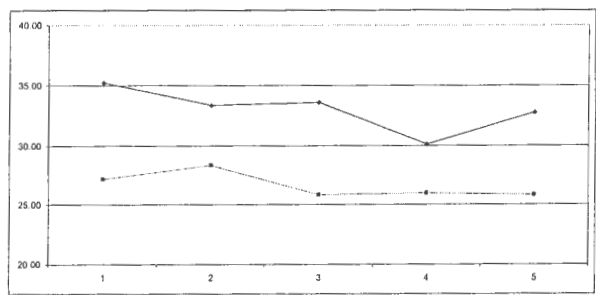
$$\begin{array}{l}
 x_{m+1}^2(n) \\
 x_{m+1}^2(n) \\
 \dots \\
 x_{2m}^2(n)
 \end{array}
 \Rightarrow x_1^G(n) = \max\{x_{m+1}(n), x_{m+2}(n), \dots, x_{2m}(n)\}$$

$$\begin{array}{l}
 x_{\lfloor \frac{K}{m} \rfloor}^2(n) \\
 x_{\lfloor \frac{K}{m} \rfloor}^2(n) \\
 \dots \\
 x_{\lfloor \frac{K}{m} \rfloor}^2(n)
 \end{array}
 \Rightarrow x_{\lfloor \frac{K}{m} \rfloor}^G(n) = \max\{x_{\lfloor \frac{K}{m} \rfloor}^2(n), \dots, x_{\lfloor \frac{K}{m} \rfloor}^2(n)\}$$

W analogiczny sposób tworzona jest zagregowana m -krokowa obwiednia dolna szeregu typu 2, w postaci szeregów $\lfloor \frac{K}{m} \rfloor$ wymiarowych

$$\{x_k^D(n)\}_{k=1}^{\lfloor \frac{K}{m} \rfloor} = \left[x_1^D(n), x_2^D(n), \dots, x_{\lfloor \frac{K}{m} \rfloor}^D(n) \right]^T \tag{10}$$

Sposób tworzenia zagregowanych m -krokowych obwiedni typu 2 (z rysunku 2) został zilustrowany graficznie dla przykładu obliczeniowego na rysunku 4.



Rysunek 4 Zagregowana obwiednia 4-krokowa gorna typu 2 i 4-krokowa dolna typu 2

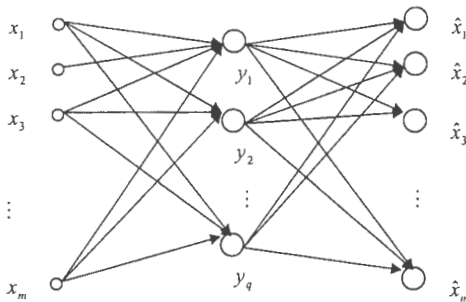
3. Kompresja szeregów czasowych

Generowanie cech istotnych jest ściśle związane z problemem kompresji danych lub redukcją wymiarowości szeregów czasowych. Zadaniem kompresji danych jest takie zmniejszenie informacji o szeregu czasowym, 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 N elementów, $\{\bar{x}_i\}_{i=1}^{mN} = [x_1, x_2, \dots, x_N]^T$, to po kompresji jest on reprezentowany przez q cech istotnych, tzn. wektor $\{y_i\}_{i=1}^{mq} = [y_1, y_2, \dots, y_q]^T$, gdzie $q \ll N$. Zakłada się, że na podstawie wektora y można odtworzyć wektor \bar{x} z pewną dokładnością jako $\{\hat{x}_i\}_{i=1}^{mN} = [\hat{x}_1, \hat{x}_2, \dots, \hat{x}_N]^T$.

Szereg czasowy 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, którą stanowi q neuronów.



Rysunek 5. Wyjścia neuronów warstwy ukrytej tworzą wektor cech istotnych.

4 Przykład obliczeniowy dla zadania klasyfikacji

4.1 Opis danych

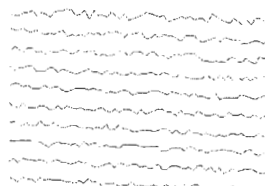
Do obliczeń wybrano zbiór danych dostępny poprzez Internet w bazie danych Uniwersytetu Irvine w Kalifornii, który jest często stosowany przy testowaniu algorytmów do eksploracji danych (Alcock i Manolopoulos, 1999):

http://kdd.ics.uci.edu/databases/synthetic_control/synthetic_control.data.html

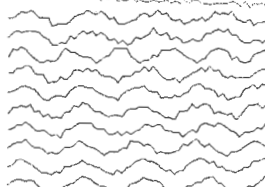
Zawiera on 600 szeregów czasowych, sztucznie wygenerowanych, z których każdy złożony jest z 60 liczb, określona jest też przynależność szeregu do jednej z sześciu klas: A, B, C, D, E lub F. Szeregi od 1-100 należą do klasy A, 101-200 należą do klasy B, 201-300 należą do klasy C, 301-400 należą do klasy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

Poniżej przedstawiono po 10 szeregów dla każdej klasy:

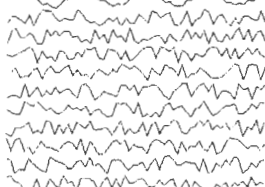
Klasa A:
normalny (normal)



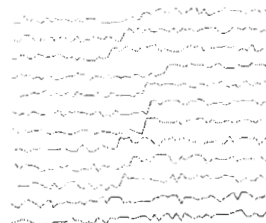
Klasa B:
cykliczny (cyclic)



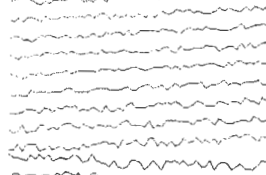
Klasa C:
trend wzrostowy
(increasing trend)



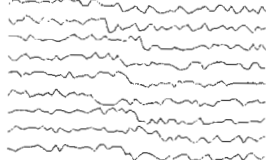
Klasa D:
trend opadający
(decreasing trend)



Klasa E:
przesunięcie w górę
(upward shift)



Klasa F:
przesunięcie w dół
(downward shift)



Poniżej zamieszczono 7 pierwszych wartości z 10 pierwszych szeregów z klasy E:

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 ...
...						

Celem naszym było między innymi sprawdzenie, na ile zaproponowana metoda agregacji szeregów czasowych sprawdza się w praktyce. Dla wybranej klasyfikacji, dokonywano agregacji szeregów ze zbioru uczącego, za pomocą cech istotnych. Następnie, na podstawie takich zagregowanych szeregów generowano zbiór reguł decyzyjnych dla rozpatrywanych klas. Dokładność klasyfikacji z zastosowaniem wygenerowanych reguł sprawdzano dla szeregów ze zbioru uczącego oraz ze zbioru testowego, który zawierał szeregi, niebiorące udziału w procesie tworzenia reguł.

4.2 Generowanie reguł decyzyjnych

Załóżmy, że dany jest zbiór przykładów $U = \{e^n\}$, $n=1,2,\dots,N$, będących szeregami czasowymi. Przykłady te opisujemy za pomocą warunków związanych ze zbiorem cech istotnych $A = \{a_1, \dots, a_q\}$ o skończonym zbiorze wartości, odpowiednio $V_{a_j} = \{x_{j,1}, x_{j,2}, \dots, x_{j,t_j}\}$ dla $j=1, \dots, q$, $a_j \in A$. Można zdefiniować funkcję $f: U \times A \rightarrow V$ taką, że $\forall e^n \in U, \forall a_j \in A, f(e^n, a_j) \in V_{a_j}$. Każdy przykład $e^n \in U$ można opisać za pomocą koniunkcji q warunków elementarnych

$$e^n = \bigwedge_{j=1}^q (a_j = f(e^n, a_j)) \quad (11)$$

gdzie $f(e^n, a_j) = x_{j,t(j,n)}$ oraz $x_{j,t(j,n)} \in V_{a_j}$. Funkcja $f(e^n, a_j)$ określa, że cecha a_j przyjmuje wartość $x_{j,t(j,n)}$ dla przykładu e^n . Indeks $t(j,n)$ dla $j \in \{1, 2, \dots, q\}$ i $n \in \{1, 2, \dots, N\}$ określa, którą wartość przyjmuje j -ta cecha w n -tym przykładzie.

Stosując wymienione powyżej cechy możemy zgodnie ze wzorem (11) opisać każdy przykład e^n (tzn. każdy szereg czasowy) w postaci koniunkcji warunków związanych z tymi cechami, oznaczanymi $s_j = (a_j = x_{j,t(j,n)})$. Koniunkcję l warunków elementarnych, $l \leq q$, postaci $\bigwedge_{j \in I} s_j = C^l$, gdzie $I \subseteq \{1, \dots, q\}$, $card(I) = l$, nazywamy kompleksem. Mówimy, że kompleks C^l opisuje przykład e^n jeżeli $\forall j \in I$ wszystkie warunki w kompleksie są opisane przez odpowiednie warunki w przykładzie.

Ze względu na wartości, które przyjmuje dodatkowa cecha a_d dokonujemy podziału zbioru przykładów na rozłączne i w sumie tworzące cały zbiór podzbiory, które nazywamy klasami dla klasyfikacji. Elementy zbioru A nazywamy cechami warunkowymi, a cechę a_d nazywamy cechą decyzyjną. Zakładamy, że liczba i rodzaj cech wystarczają do poprawnego rozdzielenia przykładów należących do różnych klas.

Bardziej formalnie możemy zapisać, że podziałem zbioru przykładów U ze względu na cechę decyzyjną a_d mającą dziedzinę $V_{a_d} = \{x_{d,1}, x_{d,2}, \dots, x_{d,t_d}\}$, nazywamy niepuste

podzbiory przykładów $\{X_{x_{d,l}} : l = 1, \dots, L_d\}$, $X_{x_{d,l}} = \{e \in U : f(e, a_d) = x_{d,l}\}$, $\forall x_{d,l} \in V_{a_d}$
i $X_{x_{d,1}} \cup \dots \cup X_{x_{d,L_d}} = U$, $X_{x_{d,i}} \cap X_{x_{d,j}} = \emptyset$ dla $i \neq j$.

Zbiory tak określonych przykładów uczących (ze znaną przynależnością do klasy) są punktem wyjścia w procesie uczenia maszynowego, w wyniku którego uzyskuje się opisy rozpatrywanych klas w postaci reguł elementarnych. Przyjęto, że tworzone reguły powinny poprawnie opisywać "większość" przykładów należących do rozpatrywanej klasy i nie opisywać "prawie wszystkich" przykładów do tej klasy nie należących. Reguły elementarne mają postać wyrażen logicznych „JEŻELI *spełnione są określone warunki* TO *zachodzi przynależność do danej klasy*”; w których poprzednik reguły zawiera koniunkcję warunków związanych z podzbiorem cech wybranych do opisu przykładów, a następnik reguły będzie określał przynależność do klasy. Implikację

$$R_r : C^{I_r} \Rightarrow (a_d = v_{d,l}) \quad (12)$$

$l \in \{1, \dots, L_d\}$ nazywamy r -tą regułą elementarną dla klasy $Y_{x_{d,l}}$, gdzie $C^{I_r} = \bigwedge_{j \in I_r} (a_j = x_{j,(j,r)})$, $I_r \subseteq \{1, \dots, q\}$. Indeks $t(j,r)$ określa, jaka wartość j -tej cechy występuje w r -tej regule.

Tworzone reguły elementarne powinny spełniać warunek spójności, tzn. rozróżniać przykłady należące od nie należących do danej klasy oraz być minimalne, tzn. usunięcie dowolnego warunku w części przesłankowej, spowodowałoby nie spełnienie warunku spójności.

Reguły, o których była mowa wyżej, można tworzyć stosując różne algorytmy uczenia maszynowego. Reguły mogą być użyte do uzyskania nieznanych wcześniej w sposób jawny informacji o istniejących w zbiorze danych zależnościach. W tym przypadku każda reguła jest rozpatrywana indywidualnie jako możliwa reprezentacja interesującego nas wzorca czy też jakiejś istniejącej zależności pomiędzy cechami.

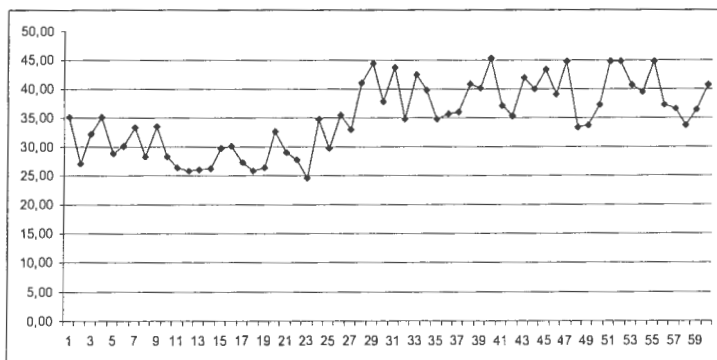
Korzystając z własnych doświadczeń, do obliczeń wybrano m.in. metodę IP wykorzystującą modyfikację zadania pokrycia zbioru, opisaną przez Szkatułę (1995, 2002), a następnie rozwiniętą przez Kacprzyka i Szkatułę (1999, 2002).

4.3 Wyniki obliczeń - klasyfikacja do dwóch klas E i F

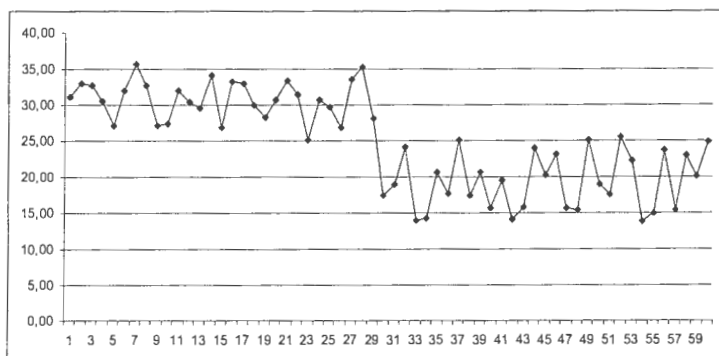
Do obliczeń wybrano dwie klasy:

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

Rozpatrywane szeregi czasowe zapisano w postaci $[x_1(n), x_2(n), \dots, x_{60}(n)]^T$ dla $n = 1, 2, \dots, 50$. Znana była również przynależność każdego szeregu do klasy 1 lub do klasy 2. Na rysunku 6 i 7 przedstawiono przykłady pojedynczych szeregów z obu rozpatrywanych klas.



Rysunek 6. Przykład szeregu czasowego z klasy 1



Rysunek 7. Przykład szeregu czasowego z klasy 2

4.3.1. Wyniki obliczeń dla obwiedni typu 1

Każdy n -ty szereg zawierający 60 wartości liczbowych był przetwarzany do szeregu różnic $\{x_1^1(n), x_2^1(n), \dots, x_{59}^1(n)\}^T$ (wzór (2)) i tworzona była 4-krokowa obwiednia górna typu 1 tego szeregu $\{x_k^2(n)\}_{k=1}^{k=59}$ (wzór (3)) oraz 4-krokowa obwiednia dolna typu 1 $\{x_k^3(n)\}_{k=1}^{k=59}$ (wzór (4)).

Obwiednie górne 4-krokowe były następnie agregowane do szeregów $\{x_1^G(n), x_2^G(n), \dots, x_{19}^G(n)\}^T$, $n = 1, 2, \dots, 50$ (wzór (7)). Analogicznie tworzono zagregowane 4-krokowe obwiednie dolne $\{x_1^D(n), x_2^D(n), \dots, x_{19}^D(n)\}^T$, $n = 1, 2, \dots, 50$ (wzór (8)).

Następnie generowano 5 cech istotnych, stosując w tym celu pamięć asocjacyjną, realizowaną poprzez dwuwarstwową jednokierunkową sieć neuronową z jedną warstwą ukrytą, którą stanowiło 5 neuronów. Do tego celu został użyty program Java Neural Networks Simulator (JavaNNS). Generowanie cech istotnych wykonano dla trzech zadań, przyjmując kolejno do obliczeń:

- I) sieć neuronową o 19 wejściach i 19 wyjściach (dane uczące stanowiły zagregowane 4-krokowe obwiednie górne typu 1)
- II) sieć neuronową o 19 wejściach i 19 wyjściach (dane uczące stanowiły zagregowane 4-krokowe obwiednie dolne typu 1).
- III) sieć neuronową o 38 wejściach i 38 wyjściach (dane uczące stanowiły zagregowane 4-krokowe obwiednie górne typu 1 oraz zagregowane 4-krokowe obwiednie dolne typu 1)

Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis szeregu. Tak więc dla każdego zadania, z zastosowaniem wartości cech istotnych zapisano każdy rozpatrywany szereg czasowy $\{x_k(n)\}_{k=1}^{k=60} = [x_1(n), x_2(n), \dots, x_{60}(n)]^T$, dla $n = 1, 2, \dots, 50$, w postaci zagregowanej $[x_1(n), x_2(n), \dots, x_5(n)]^T$, dla $n = 1, 2, \dots, 50$.

Uzyskano w ten sposób 50 szeregów, zawierających po 5 wartości liczbowych (cech istotnych). Cecha decyzyjna a_d z wartościami $\{1$ -klasa 1; 2-klasa 2 $\}$ określała przyna-

leżność szeregu do klasy. Poniżej zamieszczono postać dziesięciu skompresowanych szeregów dla zadania III.

	x1	x2	x3	x4	x5	klasa
	52	784	405	6	978	1
	174	968	170	10	944	1
	240	888	14	2	990	1
	94	821	41	9	993	1
	331	943	51	8	996	1
	61	926	30	1	993	1
	389	708	53	4	979	2
	257	842	47	36	988	2
	214	868	73	7	986	2
	365	771	703	13	921	2

Rekord: 1 z 50

Następnie wartości cech istotnych były nominalizowane.

Dla każdego rozpatrywanego zadania, na podstawie zbioru uczącego zawierającego 50 zagregowanych szeregów, tworzono minimalne zbiory reguł decyzyjnych do rozpatrywanych klas oraz zbiory wszystkich reguł decyzyjnych.

Poniżej zamieszczono wyniki obliczeń.

Zadanie I

Do wygenerowania 5 cech istotnych zastosowano pamięć asocjacyjną, realizowaną poprzez dwuwarstwową jednokierunkową sieć neuronową z jedną warstwą ukrytą, o 19 wejściach i 19 wyjściach. Do nauki posłużyły zagregowane 4-krokowe obwiednie górne typu 1, wyznaczone dla 25 szeregów typu E (klasa 1) oraz 25 szeregów typu F (klasa 2).

Z zastosowaniem wartości cech istotnych utworzono 50 szeregów, zawierających po 5 wartości liczbowych, ze znaną przynależnością do klasy 1 lub klasy 2, które stanowiły podstawę do dalszych obliczeń.

Wygenerowano zbiór 28 reguł decyzyjnych dla rozpatrywanej klasyfikacji. Uzyskano w 90% poprawną klasyfikację przykładów uczących, przykłady: 7, 15, 24, 30 i 44 nie zostały jednoznacznie zaklasyfikowane.

Zadanie II

Do wygenerowania 5 cech istotnych zastosowano pamięć asocjacyjną, realizowaną przez dwuwarstwową jednokierunkową sieć neuronową z jedną warstwą ukrytą, o 19 wejściach i 19 wyjściach. Do nauki posłużyły zagregowane 4-krokowe obwiednie dolne typu 1, wyznaczone dla 25 szeregów typu E (klasa 1) i dla 25 szeregów typu F (klasa 2).

Z zastosowaniem wartości cech istotnych utworzono 50 szeregów, zawierających po 5 wartości liczbowych, ze znaną przynależnością do klasy 1 lub klasy 2, które stanowiły podstawę do dalszych obliczeń.

Wartości cech istotnych były następnie nominalizowane, w sposób podany w Tabeli 1.

Tabela 1. Sposób nominalizacji wartości cech istotnych

Wartość nominalna	Cecha a1	Cecha a2	Cecha a3
1	> 0 and <= 975	> 0 and <= 100	> 0 and <= 500
2	> 975 and <= 980	> 100 and <= 250	> 500 and <= 600
3	> 980 and <= 985	> 250 and <= 400	> 600 and <= 700
4	> 985 and <= 990	> 400 and <= 550	> 700 and <= 800
5	> 990 and <= 995	> 550 and <= 700	> 800 and <= 900
6	> 995 and <= 1000	> 700 and <= 1000	> 900 and <= 1000

Wartość nominalna	Cecha a4	Cecha a5
1	> 0 and <= 25	> 0 and <= 10
2	> 25 and <= 50	> 10 and <= 20
3	> 50 and <= 75	> 20 and <= 30
4	> 75 and <= 100	> 30 and <= 40
5	> 100 and <= 200	> 40 and <= 50
6	> 200 and <= 1000	> 50 and <= 1000

Wygenerowano minimalny zbiór 21 reguł decyzyjnych dla rozpatrywanych klas, podany poniżej, w nawiasach zawarto liczbę przykładów opisywanych przez daną regułę).

$a_1=6 \wedge a_3=4 \Rightarrow a_6=1$ (1 przykład)

$a_1=5 \wedge a_3=5 \Rightarrow a_6=1$ (6 przykładów)

$a_1=4 \wedge a_2=2 \Rightarrow a_6=1$ (2 przykłady)

$a_1=5 \wedge a_5=1 \Rightarrow a_6=1$ (2 przykłady)

$a_5=5 \Rightarrow a_6=1$ (3 przykłady)

$a_4=6 \wedge a_5=3 \Rightarrow a_6=1$ (2 przykłady)

$a_3=4 \wedge a_5=6 \Rightarrow a_6=1$ (1 przykład)

$a_4=4 \wedge a_5=2 \Rightarrow a_6=1$ (1 przykład)

$a_3=3 \wedge a_4=6 \Rightarrow a_6=1$ (2 przykłady)

$a_4=3 \Rightarrow a_6=1$ (7 przykładów)

$a_1=4 \wedge a_5=1 \Rightarrow a_6=1$ (1 przykład)

$a_2=1 \wedge a_5=2 \Rightarrow a_6=1$ (1 przykład)

$a_1=6 \wedge a_4=1 \Rightarrow a_6=2$ (9 przykładów)

$a_1=6 \wedge a_3=5 \Rightarrow a_6=2$ (5 przykładów)

$a_4=5 \wedge a_5=2 \Rightarrow a_6=2$ (1 przykład)

$a_4=2 \wedge a_5=3 \Rightarrow a_6=2$ (5 przykładów)

$a_2=3 \wedge a_5=6 \Rightarrow a_6=2$ (3 przykłady)

$a_3=6 \wedge a_5=4 \Rightarrow a_6=2$ (1 przykład)

$a_1=3 \wedge a_4=5 \Rightarrow a_6=2$ (1 przykład)

$a_2=4 \Rightarrow a_6=2$ (1 przykład)

$a_2=5 \Rightarrow a_6=2$ (1 przykład)

Dla wygenerowanych reguł uzyskano w 100% poprawną klasyfikację wszystkich przykładów uczących.

Dokładność klasyfikacji z zastosowaniem tak utworzonych reguł sprawdzana była dla zbioru testowego, zawierającego 50 nowych szeregów czasowych, które nie były stosowane w procesie uczenia sieci heteroasocjacyjnej (25 szeregów z klasy 1 i 25 szeregów z klasy 2). 9 przykładów testowych z klasy 1 i 14 z klasy 2 nie zostało jednoznacznie zaklasyfikowanych z zastosowaniem wygenerowanych reguł.

Zadanie III

Do wygenerowania 5 cech istotnych zastosowano pamięć asocjacyjną, realizowaną poprzez dwuwarstwową jednokierunkową sieć neuronową z jedną warstwą ukrytą, o 38 wejściach i 38 wyjściach. Do nauki posłużyły zagregowane 4-krokowe obwiednie górne typu 1 oraz obwiednie dolne typu 1 (wyznaczone dla 25 szeregów typu E (klasa 1) oraz 25 szeregów typu F (klasa 2)).

Z zastosowaniem wartości cech istotnych utworzono 50 szeregów, zawierających po 5 wartości liczbowych, ze znaną przynależnością do klasy 1 lub klasy 2. Wartości cech istotnych były następnie nominalizowane, w sposób podany w Tabeli 2.

Tabela 2. Sposób nominalizacji wartości cech istotnych

Wartość nominalna	Cechy a1, a2, a3	Cecha a4	Cecha a5
1	> 0 and <= 100	> 0 and <= 15	> 0 and <= 730
2	> 100 and <= 200	> 15 and <= 30	> 730 and <= 760
3	> 200 and <= 300	> 30 and <= 45	> 760 and <= 790
4	> 300 and <= 400	> 45 and <= 60	> 790 and <= 820
5	> 400 and <= 500	> 60 and <= 75	> 820 and <= 850
6	> 500 and <= 600	> 75 and <= 90	> 850 and <= 880
7	> 600 and <= 700	> 90 and <= 105	> 880 and <= 910
8	> 700 and <= 800	> 105 and <= 120	> 910 and <= 940
9	> 800 and <= 900	> 120 and <= 135	> 940 and <= 970
10	> 900 and <= 1000	> 135 and <= 1000	> 970 and <= 1000

Wygenerowano minimalny zbiór 26 reguł decyzyjnych dla rozpatrywanej klasyfikacji, zamieszczony poniżej. W nawiasach zawarto liczbę przykładów opisywanych przez daną regułę.

$a4=4 \Rightarrow a6=1$ (1 przykład)
 $a5=2 \Rightarrow a6=1$ (1 przykład)
 $a2=6 \Rightarrow a6=1$ (1 przykład)
 $a1=1 \wedge a4=1 \Rightarrow a6=1$ (8 przykładów)

$a_1=4 \wedge a_3=1 \Rightarrow a_6=1$ (2 przykłady)
 $a_1=1 \wedge a_4=3 \Rightarrow a_6=1$ (1 przykład)
 $a_2=8 \wedge a_4=2 \Rightarrow a_6=1$ (2 przykłady)
 $a_1=3 \wedge a_2=9 \wedge a_4=1 \Rightarrow a_6=1$ (3 przykłady)
 $a_2=10 \wedge a_4=1 \wedge a_5=10 \Rightarrow a_6=1$ (4 przykłady)
 $a_3=5 \Rightarrow a_6=1$ (3 przykłady)
 $a_2=7 \wedge a_4=1 \Rightarrow a_6=1$ (1 przykład)
 $a_1=4 \wedge a_4=2 \Rightarrow a_6=1$ (1 przykład)
 $a_2=10 \wedge a_3=2 \wedge a_5=9 \Rightarrow a_6=1$ (1 przykład)

$a_1=4 \wedge a_2=8 \Rightarrow a_6=2$ (3 przykłady)
 $a_1=3 \wedge a_3=1 \wedge a_5=10 \Rightarrow a_6=2$ (2 przykłady)
 $a_1=2 \wedge a_2=9 \Rightarrow a_6=2$ (4 przykłady)
 $a_5=4 \Rightarrow a_6=2$ (2 przykłady)
 $a_3=1 \wedge a_5=9 \Rightarrow a_6=2$ (4 przykłady)
 $a_1=6 \Rightarrow a_6=2$ (2 przykłady)
 $a_1=5 \wedge a_5=9 \Rightarrow a_6=2$ (1 przykład)
 $a_1=5 \wedge a_5=10 \Rightarrow a_6=2$ (2 przykłady)
 $a_4=8 \Rightarrow a_6=2$ (1 przykład)
 $a_2=5 \Rightarrow a_6=2$ (2 przykłady)
 $a_3=4 \Rightarrow a_6=2$ (1 przykład)
 $a_3=2 \wedge a_4=3 \Rightarrow a_6=2$ (1 przykład)
 $a_2=8 \wedge a_5=8 \Rightarrow a_6=2$ (2 przykłady)

Dwa przykłady ze zbioru uczącego należące do klasy 1: przykład 6 i 10 oraz dwa należące do klasy 2: przykład 26 i 28 nie zostały jednoznacznie zaklasyfikowane z zastosowaniem wygenerowanych reguł. Uzyskano 92% poprawnych klasyfikacji przykładów uczących.

4.3.2. Wyniki obliczeń dla obwiedni typu 2

Dla każdego n -tego szeregu zawierającego 60 wartości liczbowych tworzona była 4-krokowa obwiednia górna typu 2 $\{x_k^2(n)\}_{k=1}^{k=60}$ (wzór (5)), która następnie była agregowana do szeregu postaci $[x_1^G(n), x_2^G(n), \dots, x_{15}^G(n)]^T$, $n = 1, \dots, 50$ (wzór (9)) oraz 4-krokowa

obwiednia dolna typu 2 $\{x_k^D(n)\}_{k=1}^{k=60}$ (wzór (6)), która następnie była agregowana do szeregu postaci $[x_1^D(n), x_2^D(n), \dots, x_{15}^D(n)]^T$, $n = 1, \dots, 50$ (wzór (10)).

Do wygenerowania 5 cech istotnych zastosowano pamięć asocjacyjną, realizowaną przez dwuwarstwową jednokierunkową sieć neuronową z jedną warstwą ukrytą, o 15 wejściach i 15 wyjściach. Generowanie cech istotnych wykonano dla dwóch zadań:

IV) danymi uczącymi były zagregowane 4-krokowe obwiednie górne typu 2

V) danymi uczącymi były zagregowane 4-krokowe obwiednie dolne typu 2.

Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis szeregu dla każdego zadania. Uzyskano w ten sposób 50 szeregów, zawierających po 5 wartości liczbowych (cech istotnych) w postaci zagregowanej $[x_1(n), x_2(n), \dots, x_5(n)]^T$, $n = 1, 2, \dots, 50$. Cecha decyzyjna a_d z wartościami {1-klasa 1; 2-klasa 2} określała przynależność do klasy.

Następnie wartości cech istotnych były nominalizowane, w sposób podany w Tabeli 3.

Tabela 3. Sposób nominalizacji wartości cech istotnych

Wartości cech istotnych z przedziału:	Wartość nominalna
> 0 and <= 100	1
> 100 and <= 200	2
> 200 and <= 300	3
> 300 and <= 400	4
> 400 and <= 500	5
> 500 and <= 600	6
> 600 and <= 700	7
> 700 and <= 800	8
> 800 and <= 900	9
> 900 and <= 1000	10

Dla każdego zadania, na podstawie zbioru uczącego zawierającego zagregowane szeregi ze znaną przynależnością do klasy, generowano zbiory reguł decyzyjnych do rozpatrywanych klas. Tworzono zarówno minimalne zbiory reguł decyzyjnych poprawnie opisujących przykłady ze zbioru uczącego, jak również zbiory wszystkich reguł dla rozpatrywanej klasyfikacji.

Sprawdzana była dokładność klasyfikacji wygenerowanych reguł dla przykładów uczących i dla przykładów testowych, które nie brały udziału w procesie uczenia sieci heteroasocjacyjnej. Poniżej zamieszczono wyniki obliczeń.

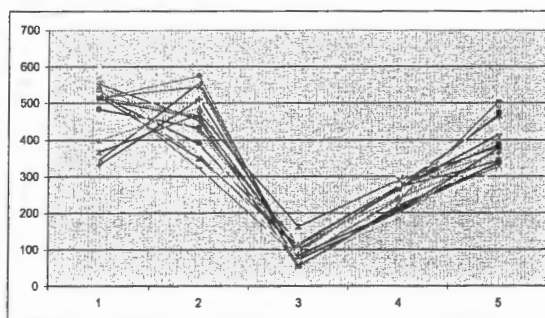
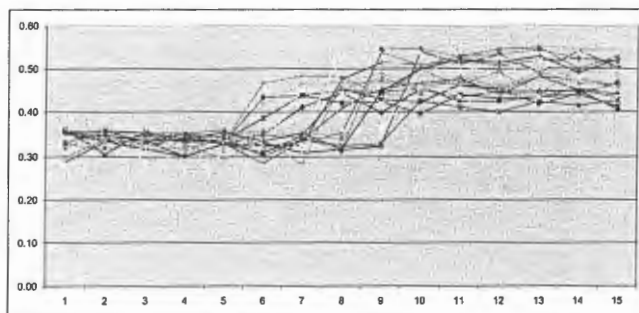
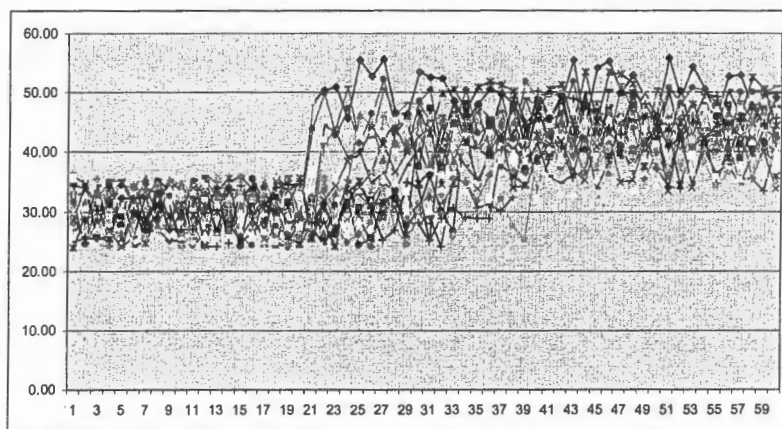
Zadanie IV

Do wygenerowania 5 cech istotnych zastosowano pamięć asocjacyjną, realizowaną poprzez dwuwarstwową jednokierunkową sieć neuronową z jedną warstwą ukrytą, o 15 wejściach i 15 wyjściach. Do nauki sieci neuronowej posłużyły zagregowane 4-krokowe obwiednie górne typu 2 $\{x_k^G(n)\}_{k=1}^{k=15}$ dla $n = 1, 2, \dots, N$, wyznaczone dla 25 szeregów typu E (klasa 1) oraz 25 szeregów typu F (klasa 2). (plik GG_MAX, błąd uczenia = 0.04, wartość współczynnika uczenia $\eta = 0.1$, maksymalna dopuszczalna różnica między wzorcem a wyjściem sieci $d_{\max} = 0.05$, 10000 cykli uczenia).

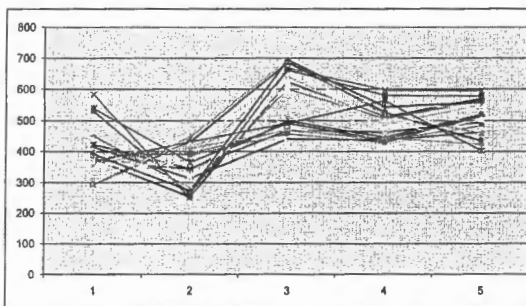
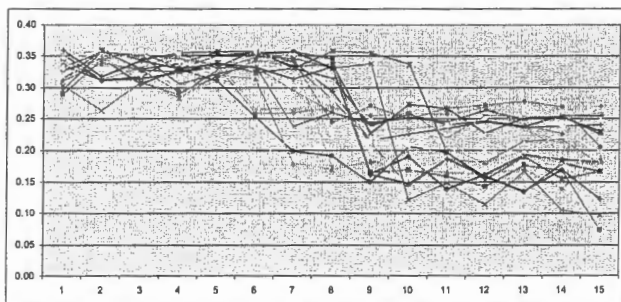
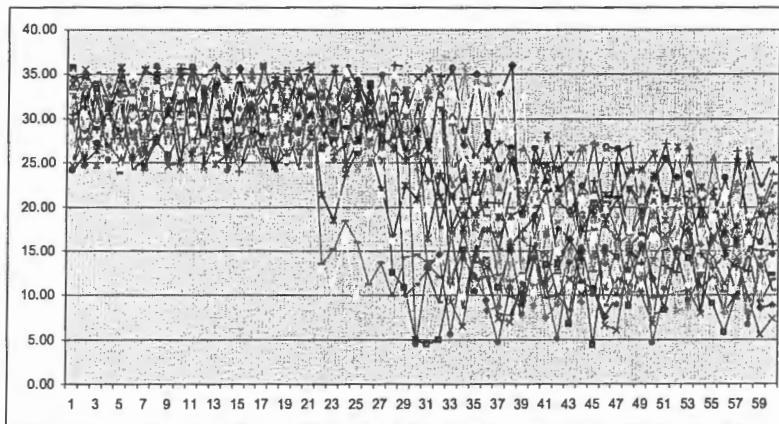
Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis tego szeregu. Poniżej przedstawiono przykładowych pięć takich szeregów dla danych uczących, dla których szósta cecha decyzyjna a_6 z wartościami {1- klasa 1; 2- klasa 2} określała przynależność szeregu do klasy.

a1	a2	a3	a4	a5	Dec
368	471	163	289	380	1
512	575	53	209	340	1
333	509	84	204	378	1
378	255	624	512	568	2
531	267	663	580	576	2

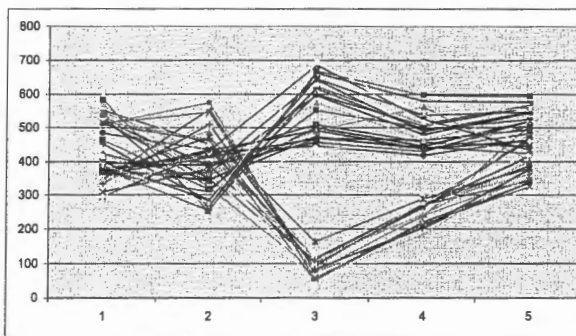
Poniżej na rysunku 8 i 9 zamieszczono ilustrację graficzną wyżej wymienionych szeregów.



Rysunek 8. (górn) wszystkie szeregi czasowe z klasy 1 ze zbioru uczącego (środek) ich zagregowana 4-krokowa obwiednia górna typu 2 (dół) zapis szeregów za pomocą cech istotnych



Rysunek 9. (górn) wszystkie szeregi czasowe z klasy 2 ze zbioru uczącego (środek) ich zagregowana 4-krokowa obwiednia górna typu 2 (dół) zapis szeregow za pomocą cech istotnych



Rysunek 10. Zapis szeregów za pomocą cech istotnych (dla obwiedni górnych typu 2) dla obu rozpatrywanych klas

Następnie wartości cech istotnych były nominalizowane. Sposób nominalizacji podano w Tabeli 3.

Poniżej zamieszczono wygenerowany minimalny zbiór reguł decyzyjnych do rozpatrywanych klas, w nawiasach podano liczbę przykładów, które opisują:

$a_4=3 \Rightarrow a_6=1$ (23 przykłady)

$a_4=2 \Rightarrow a_6=1$ (2 przykłady)

$a_4=6 \Rightarrow a_6=2$ (11 przykłady)

$a_4=5 \Rightarrow a_6=2$ (14 przykłady)

Utworzone reguły w 100% poprawnie klasyfikują wszystkie przykłady uczące.

Poniżej zamieszczono wszystkie wygenerowane reguły decyzyjne dla rozpatrywanych klas oraz w nawiasach liczbę przykładów, które opisują.

$a_3=1 \Rightarrow a_6=1$ (20 przykładów)

$a_4=3 \Rightarrow a_6=1$ (23 przykłady)

$a_1=6 \wedge a_2=5 \Rightarrow a_6=1$ (6 przykładów)

$a_5=4 \Rightarrow a_6=1$ (16 przykładów)

$a_2=6 \Rightarrow a_6=1$ (7 przykładów)

$a_4=2 \Rightarrow a_6=1$ (2 przykłady)

$a_3=2 \Rightarrow a_6=1$ (5 przykładów)

$a_1=5 \wedge a_2=5 \Rightarrow a_6=1$ (2 przykłady)

$a_1=6 \wedge a_2=4 \wedge a_5=6 \Rightarrow a_6=1$ (1 przykład)

$a_2=3 \Rightarrow a_6=2$	(6 przykładów)
$a_3=7 \Rightarrow a_6=2$	(11 przykładów)
$a_4=6 \Rightarrow a_6=2$	(11 przykładów)
$a_1=4 \wedge a_5=6 \Rightarrow a_6=2$	(10 przykładów)
$a_3=5 \Rightarrow a_6=2$	(9 przykładów)
$a_4=5 \Rightarrow a_6=2$	(14 przykładów)
$a_1=4 \wedge a_5=5 \Rightarrow a_6=2$	(6 przykładów)
$a_3=6 \Rightarrow a_6=2$	(5 przykładów)
$a_1=3 \Rightarrow a_6=2$	(2 przykłady)
$a_1=4 \wedge a_2=4 \Rightarrow a_6=2$	(8 przykładów)
$a_1=5 \wedge a_5=6 \Rightarrow a_6=2$	(3 przykłady)

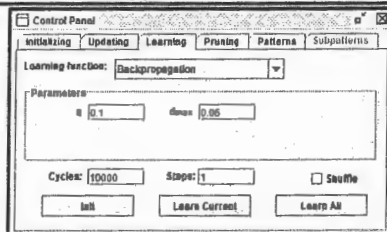
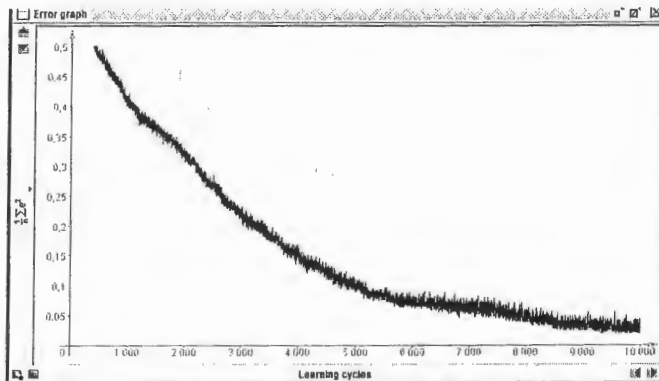
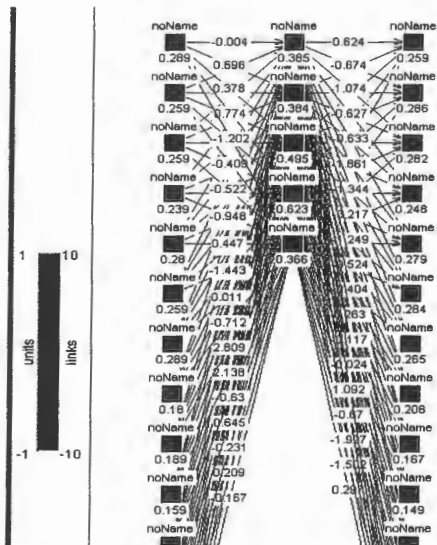
Dokładność klasyfikacji z zastosowaniem tak utworzonych reguł sprawdzana była dla zbioru testowego, zawierającego 50 nowych szeregów czasowych, które nie były stosowane w procesie uczenia sieci. Zbiór testowy zawierał 25 szeregów z klasy 1 i 25 szeregów z klasy 2. Uzyskano w 100% poprawną klasyfikację wszystkich przykładów testowych. Poniżej, dla każdej reguły z wygenerowanego minimalnego zbioru reguł, podano liczbę przykładów testowych, które ona opisuje.

$a_4=3 \Rightarrow a_6=1$	(opisuje 47 przykładów testowych)
$a_4=2 \Rightarrow a_6=1$	(opisuje 3 przykłady testowe)
$a_4=6 \Rightarrow a_6=2$	(opisuje 24 przykłady testowe)
$a_4=5 \Rightarrow a_6=2$	(opisuje 26 przykładów testowych)

Zadanie V

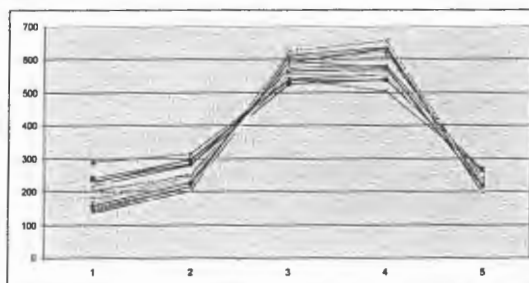
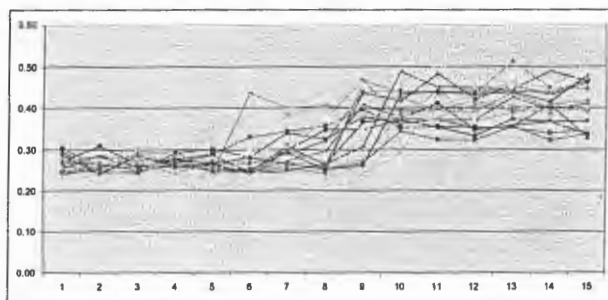
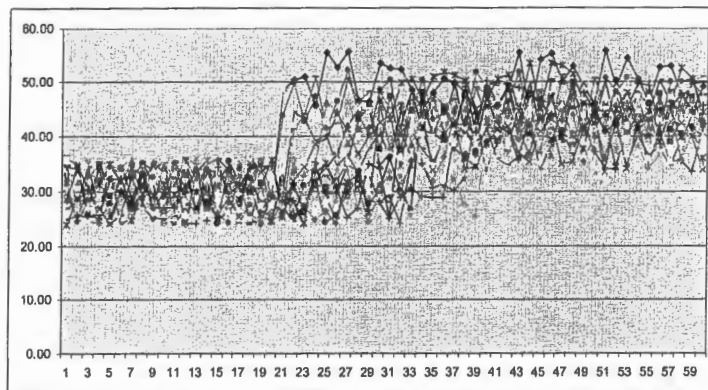
Do wygenerowania 5 cech istotnych zastosowano pamięć asocjacyjną, realizowaną poprzez dwuwarstwową jednokierunkową sieć neuronową z jedną warstwą ukrytą, o 15 wejściach i 15 wyjściach. Do nauki posłużyły zagregowane 4-krokowe obwiednie dolne typu 2 $\{x_k^D(n)\}_{k=1}^{k=15}$ dla $n=1,2,\dots,N$, wyznaczone dla 25 szeregów typu E (klasa 1) oraz 25 szeregów typu F (klasa 2).

Na rysunku 11 zamieszczono zrzut ekranu, na którym widoczna jest nauczona sieć, przyjęte parametry i wykres błędu. (GG_MIN, błąd uczenia =0.02, wartość współczynnika uczenia $\eta=0.1$, maksymalna dopuszczalna różnica między wzorcem a wyjściem sieci $d_{\max}=0.1$, 10000 cykli uczenia).

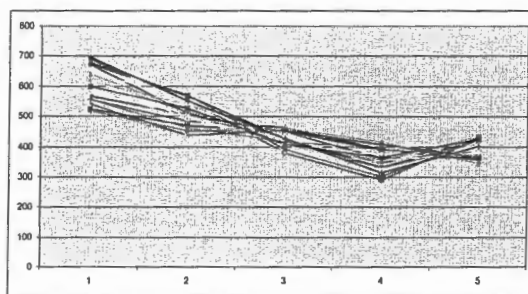
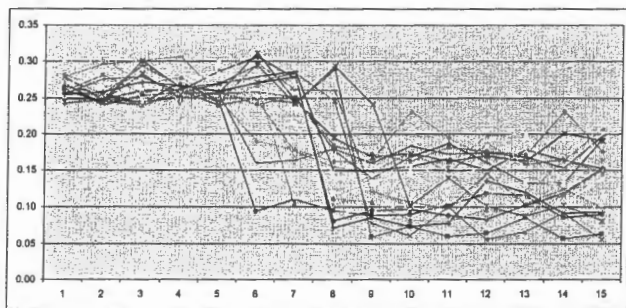
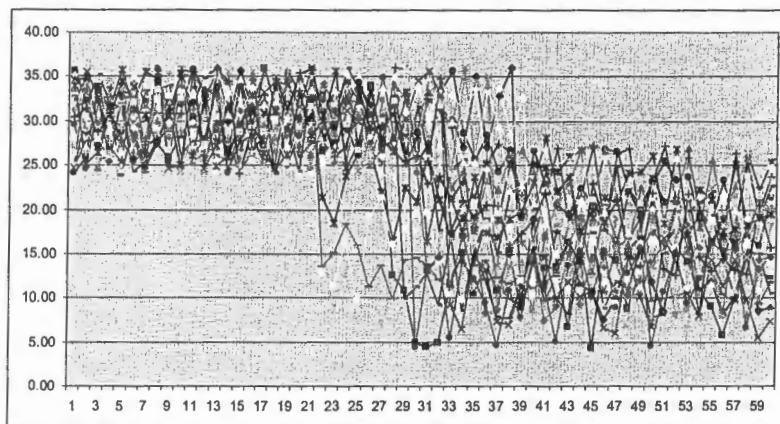


Rysunek 11. Nauczona sieć oraz wykres błęd i parametry uczenia sieci

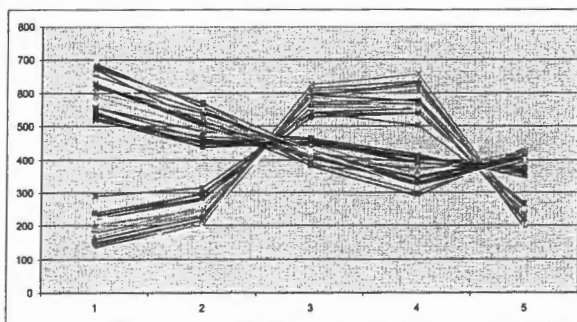
Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis szeregu (50 szeregów po 5 wartości liczbowych). Cecha a_j z wartościami $\{1, 2\}$ określała przynależność szeregu do klasy. Poniżej zamieszczono ilustrację graficzną dla wyżej wymienionych szeregów.



Rysunek 12. (górn) wszystkie szeregi czasowe z klasy 1 ze zbioru uczącego (środek) zagregowana 4-krokowa obwiednia dolna typu 2 dla wszystkich szeregów czasowych z klasy 1 ze zbioru uczącego, oraz (dół) zapis szeregów za pomocą cech istotnych



Rysunek 13. (górną) wszystkie szeregi czasowe z klasy 2 ze zbioru uczącego (środek) zagregowana 4-krokowa obwiednia dolna typu 2 dla wszystkich szeregów czasowych z klasy 2 ze zbioru uczącego, oraz (dół) zapis szeregów za pomocą cech istotnych.



Rysunek 14. Zapis szeregów z klasy 1 oraz klasy 2 za pomocą cech istotnych (dla obwiedni dolnych typu 2)

Sposób nimalizacji cech istotnych podano w Tabeli 3.

Poniżej zamieszczono wygenerowany minimalny zbiór reguł decyzyjnych dla rozpatrywanych klas, w nawiasach podano liczbę przykładów uczących, które opisują:

$a5=3 \Rightarrow a6=1$ (23 przykłady)

$a5=2 \Rightarrow a6=1$ (2 przykłady)

$a4=4 \Rightarrow a6=2$ (19 przykładów)

$a4=3 \Rightarrow a6=2$ (1 przykład)

$a5=4 \Rightarrow a6=2$ (13 przykładów)

Utworzone reguły w 100% poprawnie klasyfikują wszystkie przykłady uczące.

Poniżej zamieszczono wszystkie wygenerowane reguły decyzyjne dla rozpatrywanej klasyfikacji:

$a1=3 \Rightarrow a6=1$ (8 przykładów)

$a2=3 \Rightarrow a6=1$ (22 przykładów)

$a3=6 \Rightarrow a6=1$ (19 przykładów)

$a4=6 \Rightarrow a6=1$ (15 przykładów)

$a5=3 \Rightarrow a6=1$ (23 przykładów)

$a1=2 \Rightarrow a6=1$ (17 przykładów)

$a2=2 \Rightarrow a6=1$ (2 przykładów)

(a4=7) => (a6=1) (10 przykładów)
(a3=7) => (a6=1) (6 przykładów)
(a5=2) => (a6=1) (2 przykładów)
(a2=4) => (a6=1) (1 przykład)

(a1=7) => (a6=2) (13 przykładów)
(a2=6) => (a6=2) (13 przykładów)
(a3=5) => (a6=2) (22 przykłady)
(a4=4) => (a6=2) (19 przykładów)
(a5=5) => (a6=2) (12 przykładów)
(a3=4) => (a6=2) (3 przykłady)
(a4=3) => (a6=2) (1 przykład)
(a1=6) => (a6=2) (12 przykładów)
(a2=5) => (a6=2) (12 przykładów)
(a4=5) => (a6=2) (5 przykładów)
(a5=4) => (a6=2) (13 przykładów)

Dokładność klasyfikacji z zastosowaniem tak utworzonych reguł sprawdzana była dla zbioru testowego, zawierającego 50 nowych szeregów czasowych, które nie były stosowane w procesie uczenia sieci heteroasocjacyjnej. Zbiór testowy zawierał 25 szeregów z klasy 1 i 25 szeregów z klasy 2.

Uzyskano w 100% poprawną klasyfikację wszystkich przykładów testowych. Poniżej, dla każdej reguły podano liczbę przykładów ze zbioru testowego, które opisuje dana reguła.

(a5=3) => (a6=1) (opisuje 25 przykładów testowych)
(a5=2) => (a6=1) (nie opisuje żadnego przykładu testowego)
(a4=4) => (a6=2) (opisuje 19 przykładów testowych)
(a4=3) => (a6=2) (opisuje 2 przykłady testowe)
(a5=4) => (a6=2) (opisuje 12 przykładów testowych).

4.4 Wyniki obliczeń - klasyfikacja do trzech klas E, F i A

Do obliczeń wybrano trzy klasy:

- klasa 1: szeregi typu E (25 szeregów uczących + 25 szeregów testowych)
- klasa 2: szeregi typu F (25 szeregów uczących + 25 szeregów testowych)
- klasa 3: szeregi typu A (25 szeregów uczących + 25 szeregów testowych)

Rozpatrywane szeregi czasowe zapisano w postaci $[x_1(n), x_2(n), \dots, x_{60}(n)]^T$ $n = 1, 2, \dots, 75$. Znana była również przynależność każdego szeregu do klasy 1, klasy 2 lub 3.

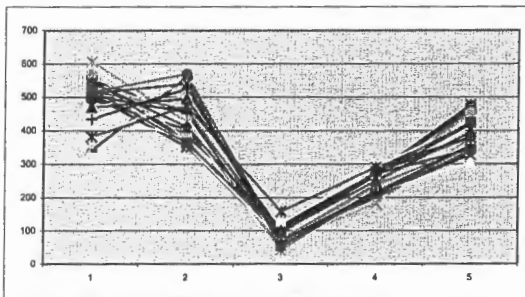
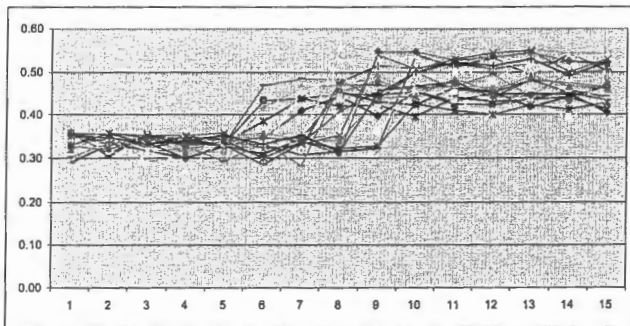
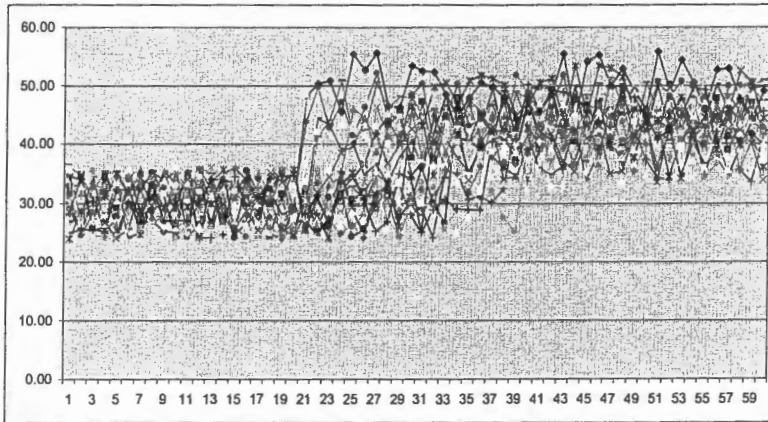
Dla każdego n -ty szeregu zawierającego 60 wartości liczbowych tworzona była 4-krokowa obwiednia górna typu 2 (wzór (5)). Obwiednie górne 4-krokowe były następnie agregowane do szeregów $[x_1^\sigma(n), x_2^\sigma(n), \dots, x_{15}^\sigma(n)]^T$, $n = 1, 2, \dots, 75$ (wzór (9)).

Następnie wygenerowano 5 cech istotnych, stosując w tym celu pamięć asocjacyjną, realizowaną przez dwuwarstwową jednokierunkową sieć neuronową o 15 wejściach i 15 wyjściach, z jedną warstwą ukrytą, którą stanowiło 5 neuronów. Do tego celu został użyty program Java Neural Networks Simulator (JavaNNS). Dane uczące stanowiły zagregowane 4-krokowe obwiednie górne typu 2.

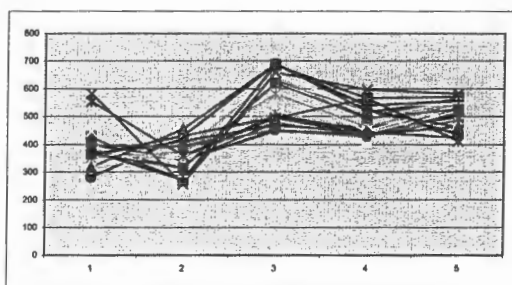
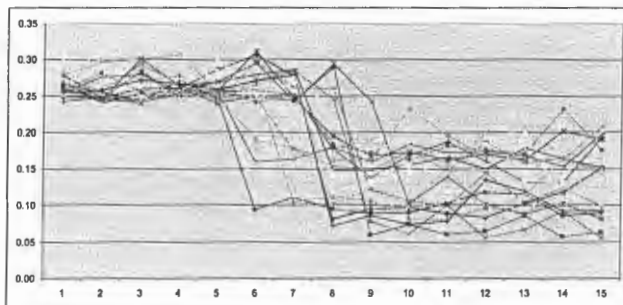
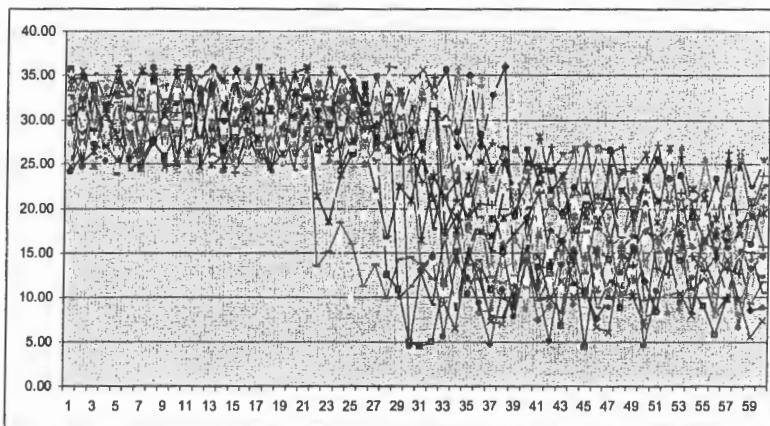
Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis szeregu. Tak więc, dla każdego zadania, z zastosowaniem wartości cech istotnych zapisano każdy rozpatrywany szereg czasowy $\{x_i(n)\}_{i=1}^{k=60} = [x_1(n), x_2(n), \dots, x_{60}(n)]^T$, dla $n=1, 2, \dots, 75$, w postaci zagregowanej $[x_1(n), x_2(n), \dots, x_5(n)]^T$, dla $n=1, 2, \dots, 75$.

Uzyskano w ten sposób 75 szeregów, zawierających po 5 wartości liczbowych (cech istotnych). Cecha decyzyjna a_d z wartościami {1-klasa 1; 2-klasa 2, 3-klasa 3} określała przynależność szeregu do klasy. (plik HH_MAX_3_klasy, błąd uczenia =0.02, wartość współczynnika uczenia $\eta=0.3$, maksymalna dopuszczalna różnica między wzorcem a wyjściem sieci 0.1, 10000 cykli uczenia).

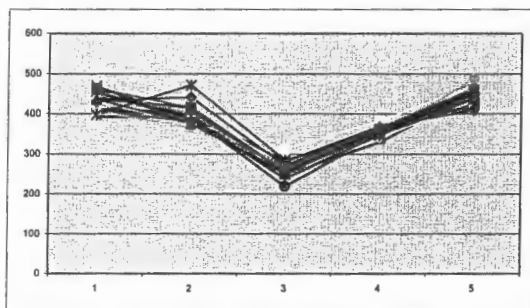
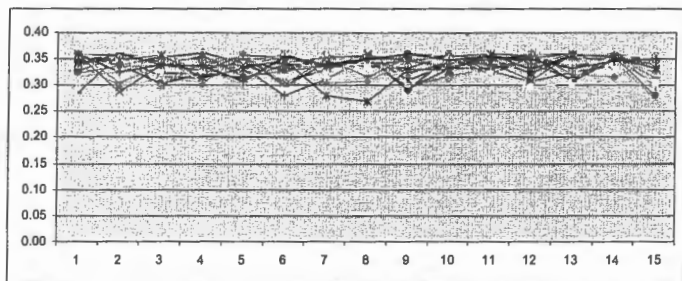
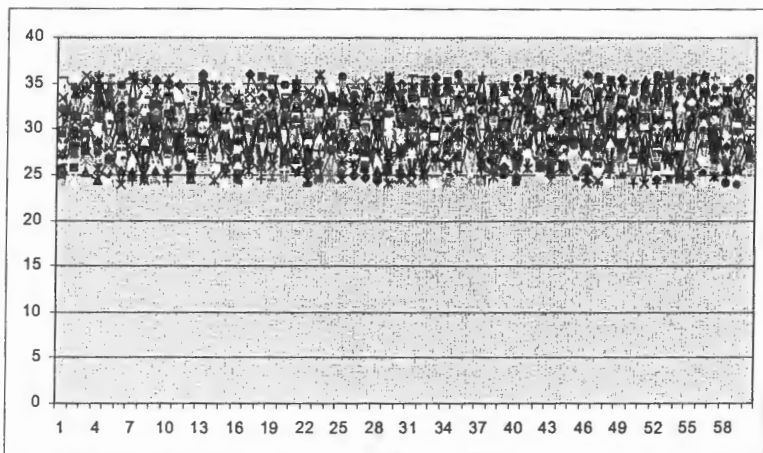
Poniżej zamieszczono ilustrację graficzną dla wyżej wymienionych szeregów.



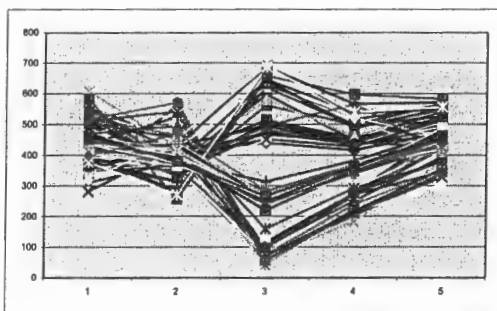
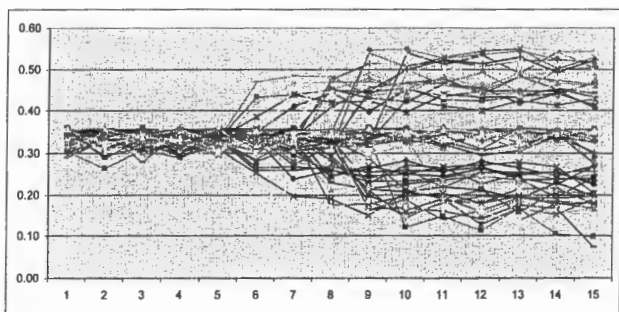
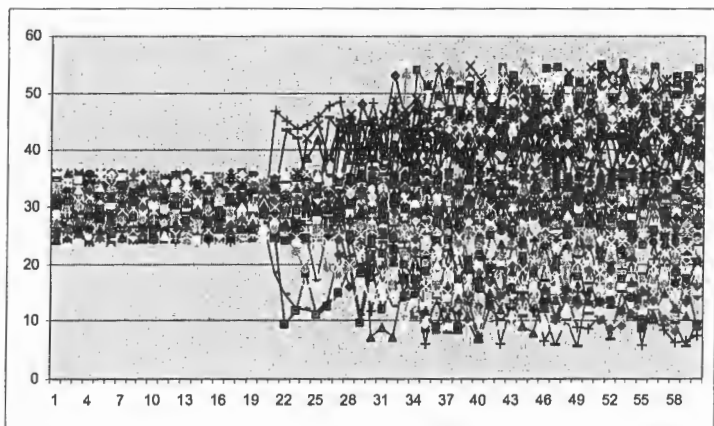
Rysunek 15. (góra) wszystkie szeregi czasowe z klasy 1 ze zbioru uczącego (środek) ich 4-krokowe zagregowane obwiednie górne typu 2 oraz (dół) zapis szeregów z zastosowaniem cech istotnych.



Rysunek 16. (góra) wszystkie szeregi czasowe z klasy 2 ze zbioru uczącego (środek) ich 4-krokowe zagregowane obwiednie dolne typu 2 oraz (dół) zapis tych szeregów z zastosowaniem cech istotnych.



Rysunek 17. (górn) wszystkie szeregi czasowe z klasy 3 ze zbioru uczącego (środek) ich 4-krokowe zagregowane obwiednie górne typu 2 oraz (dół) zapis tych szeregów z zastosowaniem cech istotnych.



Rysunek 18. (góra) wszystkie szeregi czasowe z klasy 1, 2 i 3 ze zbioru uczącego (środek) ich 4-krokowe zagregowane obwiednie górne typu 2 (dół) zapis tych szeregów z zastosowaniem cech istotnych

Wartości cech istotnych były następnie nominalizowane, w sposób podany w Tabeli 3.

Na podstawie zbioru uczącego, zawierającego 75 szeregów opisanych za pomocą 5 cech istotnych, po nominalizacji, wygenerowano minimalny zbiór pięciu pewnych reguł elementarnych przedstawiony poniżej (w nawiasach podano liczbę przykładów, które opisuje dana reguła):

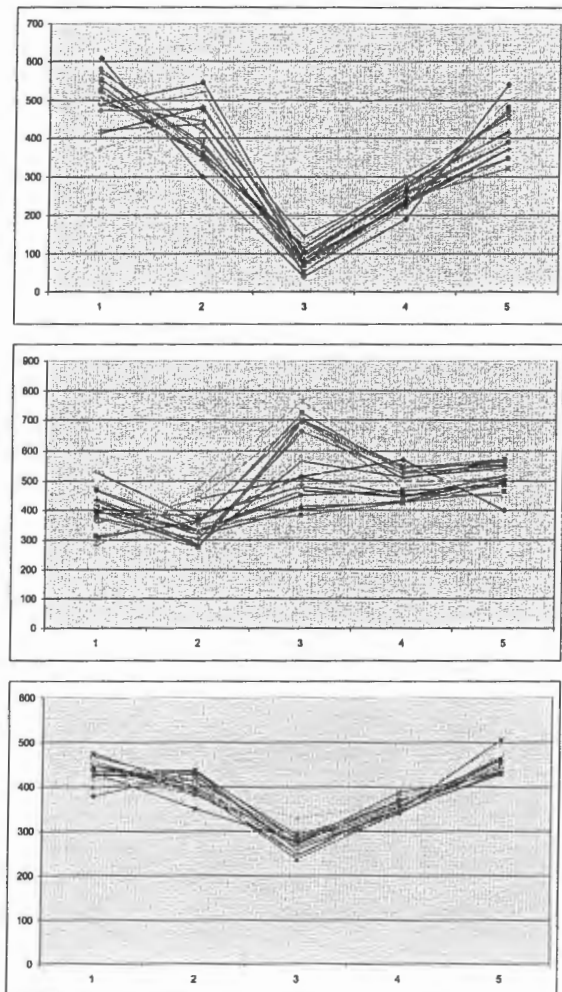
$(a_4=3) \Rightarrow (a_6=1)$	(24 przykłady)
$(a_4=2) \Rightarrow (a_6=1)$	(1 przykład)
$(a_4=6) \Rightarrow (a_6=2)$	(9 przykładów)
$(a_4=5) \Rightarrow (a_6=2)$	(16 przykładów)
$(a_4=4) \Rightarrow (a_6=3)$	(25 przykładów)

Wygenerowane reguły w 100% poprawnie opisują wszystkie przykłady ze zbioru uczącego (25 przykładów z klasy 1, 25 przykładów z klasy 2 i 25 przykładów z klasy 3).

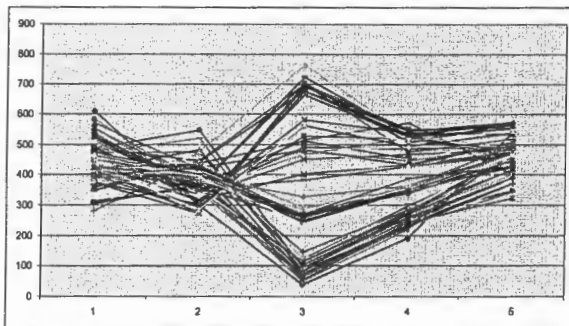
Zbiór wszystkich wygenerowanych reguł elementarnych zamieszczono poniżej:

$(a_3=1) \Rightarrow (a_6=1)$	(20 przykładów)
$(a_4=3) \Rightarrow (a_6=1)$	(24 przykłady)
$(a_1=7) \Rightarrow (a_6=1)$	(2 przykłady)
$(a_5=4) \Rightarrow (a_6=1)$	(17 przykładów)
$(a_2=6) \Rightarrow (a_6=1)$	(6 przykładów)
$(a_4=2) \Rightarrow (a_6=1)$	(1 przykład)
$(a_3=2) \Rightarrow (a_6=1)$	(5 przykładów)
$(a_1=6) \wedge (a_2=5) \Rightarrow (a_6=1)$	(5 przykładów)
$(a_2=3) \Rightarrow (a_6=2)$	(5 przykładów)
$(a_3=7) \Rightarrow (a_6=2)$	(11 przykładów)
$(a_4=6) \Rightarrow (a_6=2)$	(9 przykładów)
$(a_5=6) \Rightarrow (a_6=2)$	(15 przykładów)
$(a_3=5) \Rightarrow (a_6=2)$	(9 przykładów)
$(a_4=5) \Rightarrow (a_6=2)$	(16 przykładów)
$(a_3=6) \Rightarrow (a_6=2)$	(5 przykładów)
$(a_1=3) \Rightarrow (a_6=2)$	(1 przykład)
$(a_3=3) \Rightarrow (a_6=3)$	(24 przykładów)
$(a_4=4) \Rightarrow (a_6=3)$	(25 przykładów)
$(a_1=5) \wedge (a_2=5) \wedge (a_5=5) \Rightarrow (a_6=3)$	(11 przykładów)
$(a_3=4) \Rightarrow (a_6=3)$	(1 przykład)

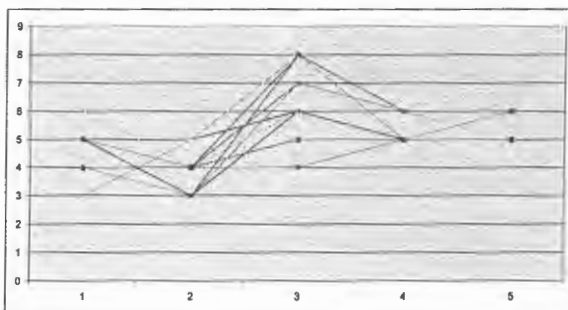
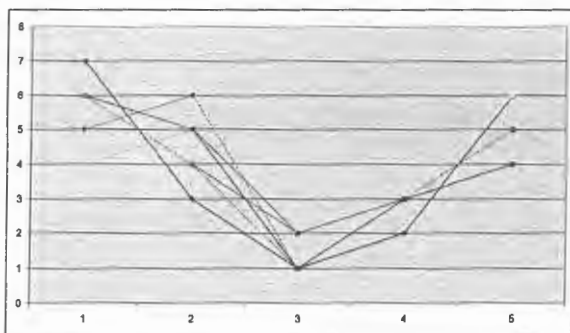
Dokładność klasyfikacji z zastosowaniem tak utworzonych reguł sprawdzana była dla zbioru testowego, zawierającego 75 nowych szeregów czasowych, które nie były stosowane w procesie uczenia sieci heteroasocjacyjnej. Zbiór testowy zawierał 25 szeregów z klasy 1, 25 szeregów z klasy 2 i 25 szeregów z klasy 3.

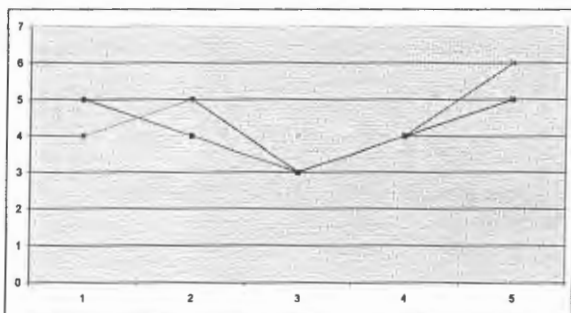


Rysunek 19. Zapis szeregów ze zbioru testowego (dla 3 klas) z zastosowaniem cech istotnych (górze) z klasy 1 (środek) z klasy 2 (dół) z klasy 3

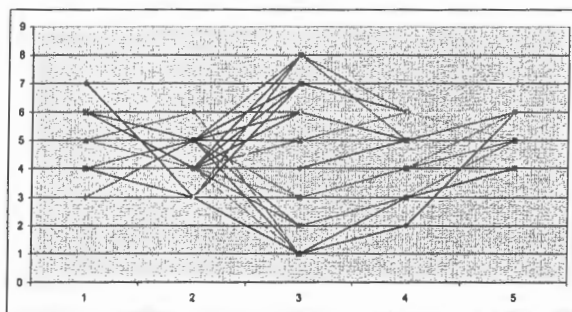


Rysunek 20. Zapis szeregów ze zbioru testowego, z 3 klas, z zastosowaniem cech istotnych





Rysunek 21. Zapis szeregów ze zbioru testowego za pomocą zdyskretyzowanych cech istotnych, odpowiednio (góła) z klasy 1 (środek) z klasy 2 (góła) z klasy 3



Rysunek 22. Zapis szeregów ze zbioru testowego za pomocą dyskretyzowanych cech istotnych, dla trzech klas

Uzyskano w 100% poprawną klasyfikację wszystkich przykładów testowych. Poniżej, dla każdej reguły (z minimalnego zbioru reguł opisujących wszystkie przykłady uczące) podano liczbę przykładów ze zbioru testowego, które dana reguła opisuje.

- $(a_4=3) \Rightarrow (a_6=1)$ (24 przykłady)
- $(a_4=2) \Rightarrow (a_6=1)$ (1 przykład)
- $(a_4=6) \Rightarrow (a_6=2)$ (13 przykładów)
- $(a_4=5) \Rightarrow (a_6=2)$ (12 przykładów)
- $(a_4=4) \Rightarrow (a_6=3)$ (25 przykładów)

4.5 Wyniki obliczeń - klasyfikacja do trzech klas B, F i A

Do obliczeń wybrano trzy klasy:

- klasa 1: szeregi typu B (25 szeregów uczących)
- klasa 2: szeregi typu F (25 szeregów uczących)
- klasa 3: szeregi typu A (25 szeregów uczących)

Rozpatrywane szeregi czasowe zapisano w postaci $[x_1(n), x_2(n), \dots, x_{60}(n)]^T$ $n = 1, 2, \dots, 75$. Znana była również przynależność każdego szeregu do klasy 1, klasy 2 lub 3.

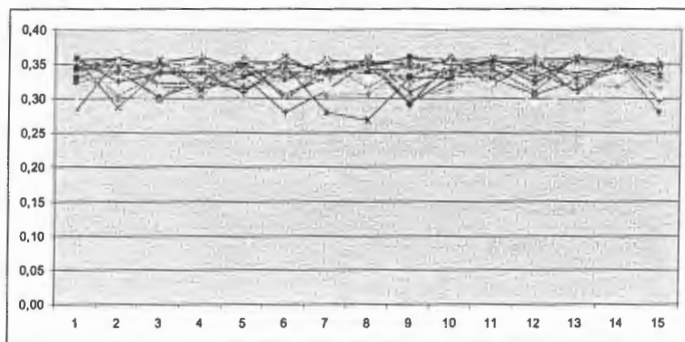
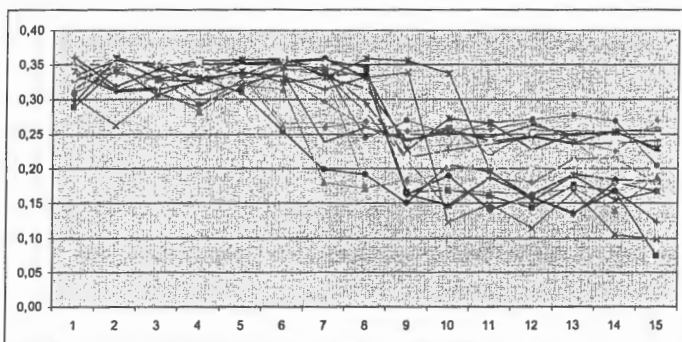
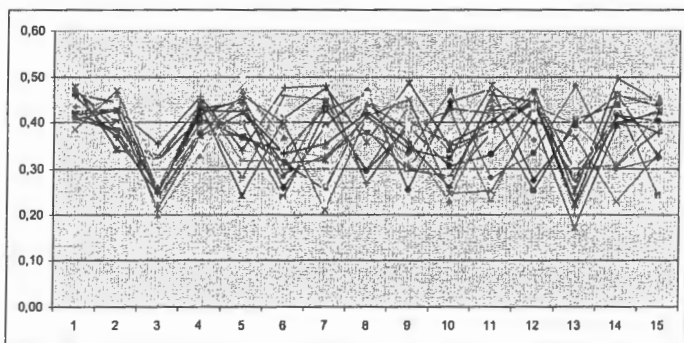
Dla każdego n -ty szeregu zawierającego 60 wartości liczbowych tworzona była 4-krokowa obwiednia górna typu 2 (wzór (5)). Obwiednie górne 4-krokowe były następnie agregowane do szeregów $[x_1^G(n), x_2^G(n), \dots, x_{15}^G(n)]^T$, $n = 1, 2, \dots, 75$ (wzór (9)).

Następnie wygenerowano 5 cech istotnych, stosując w tym celu pamięć asocjacyjną, realizowaną poprzez dwuwarstwową jednokierunkową sieć neuronową o 15 wejściach i 15 wyjściach, z jedną warstwą ukrytą, którą stanowiło 5 neuronów. Do tego celu został użyty program Java Neural Networks Simulator (JavaNNS). Dane uczące stanowiły zagregowane 4-krokowe obwiednie górne typu 2.

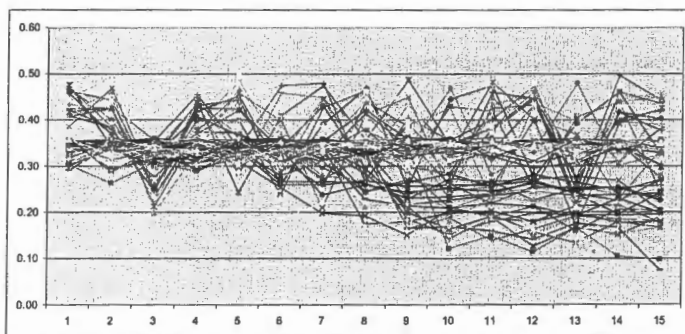
Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis szeregu. Tak więc, dla każdego zadania, z zastosowaniem wartości cech istotnych zapisano każdy rozpatrywany szereg czasowy $\{x_k(n)\}_{k=1}^{k=60} = [x_1(n), x_2(n), \dots, x_{60}(n)]^T$, dla $n = 1, 2, \dots, 75$, w postaci zagregowanej $[x_1(n), x_2(n), \dots, x_5(n)]^T$, dla $n = 1, 2, \dots, 75$.

Uzyskano w ten sposób 75 szeregów, zawierających po 5 wartości liczbowych (cech istotnych). Cecha decyzyjna a_d z wartościami {1-klasa 1; 2-klasa 2, 3-klasa 3} określała przynależność szeregu do klasy.

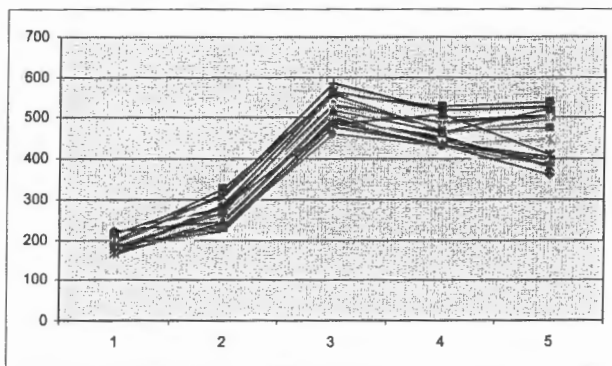
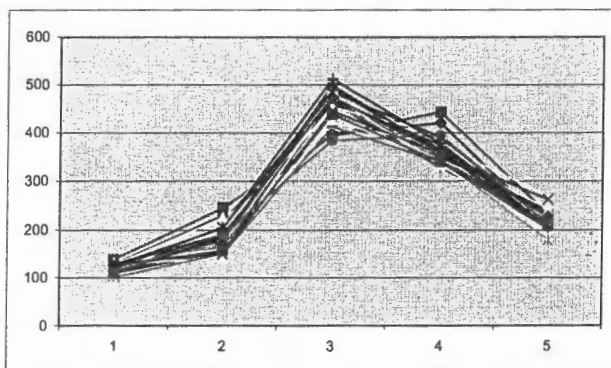
Poniżej zamieszczono ilustrację graficzną dla wyżej wymienionych szeregów.

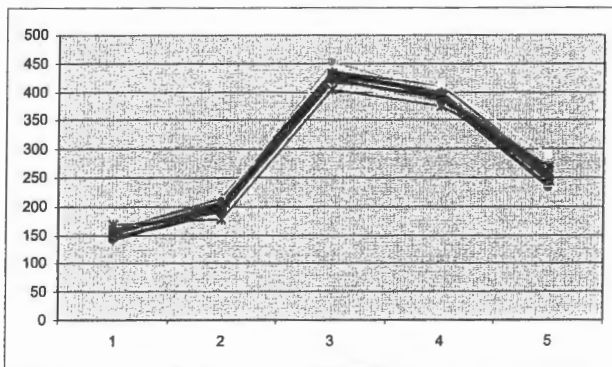


Rysunek 23. 4-krokowe zagregowane obwiednie górne typu 2 dla wszystkich szeregów czasowych ze zbioru uczącego (górze) z klasy 1 (środek) z klasy 2 (dół) z klasy 3

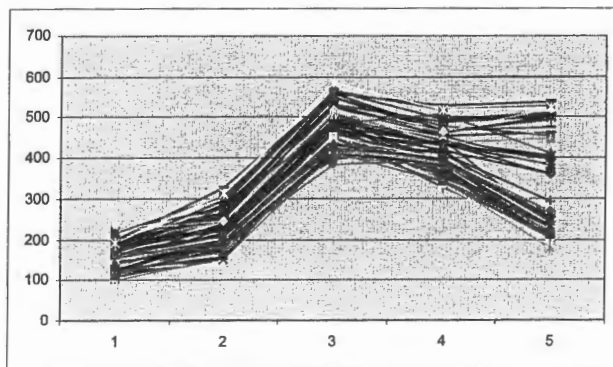


Rysunek 24. 4-krokowe zagregowane obwiednie górne typu 2 dla wszystkich szeregów czasowych z klasy 1, 2 i 3 ze zbioru uczącego





Rysunek 25. Zapis wszystkich szeregów ze zbioru uczącego z zastosowaniem cech istotnych (góra) szeregi z klasy 1 (środek) z klasy 2 (dół) z klasy 3



Rysunek 26. Zapis wszystkich szeregów ze zbioru uczącego z zastosowaniem cech istotnych, z klasy 1, 2 i 3

Uzyskane wartości cech istotnych były następnie nominalizowane, w sposób podany w Tabeli 4.

Tabela 4. Sposób nominalizacji wartości cech istotnych

Wartości cech istotnych z przedziału:	Wartość nominalna
> 0 and <= 150	1
> 150 and <= 200	2
> 200 and <= 250	3
> 250 and <= 300	4
> 300 and <= 350	5
> 350 and <= 400	6
> 400 and <= 450	7
> 450 and <= 500	8
> 500 and <= 550	9
> 550 and <= 1000	10

Na podstawie zbioru uczącego, zawierającego 75 szeregów opisanych za pomocą 5 cech istotnych po nominalizacji, wygenerowano minimalny zbiór reguł elementarnych przedstawiony poniżej (w nawiasach podano liczbę przykładów, które opisuje dana reguła):

- $(a_3=6) \Rightarrow (a_6=1)$ (3 przykłady)
 $(a_5=2) \Rightarrow (a_6=1)$ (3 przykłady)
 $a_4=5) \Rightarrow (a_6=1)$ (8 przykładów)
 $(a_3=8) \wedge (a_5=3) \Rightarrow (a_6=1)$ (12 przykładów)
 $(a_2=1) \Rightarrow (a_6=1)$ (5 przykładów)
 $(a_3=9) \wedge (a_5=3) \Rightarrow (a_6=1)$ (2 przykłady)

 $(a_5=9) \Rightarrow (a_6=2)$ (7 przykładów)
 $(a_5=6) \Rightarrow (a_6=2)$ (9 przykładów)
 $(a_5=7) \Rightarrow (a_6=2)$ (3 przykłady)
 $(a_5=8) \Rightarrow (a_6=2)$ (6 przykładów)

 $(a_2=3) \wedge (a_5=4) \Rightarrow (a_6=3)$ (4 przykłady)
 $(a_2=2) \wedge (a_3=7) \wedge (a_4=6) \Rightarrow (a_6=3)$ (17 przykładów)
 $(a_1=2) \wedge (a_5=3) \Rightarrow (a_6=3)$ (6 przykładów)

Wygenerowane reguły w 100 % poprawnie opisują wszystkie przykłady ze zbioru uczącego (25 przykładów z klasy 1, 25 przykładów z klasy 2 i 25 przykładów z klasy 3).

4.6 Klasyfikacja do trzech klas: B, F i A (z normalizacją)

Do obliczeń wybrano trzy klasy:

- klasa 1: szeregi typu B (25 szeregów uczących)
- klasa 2: szeregi typu F (25 szeregów uczących)
- klasa 3: szeregi typu A (25 szeregów uczących)

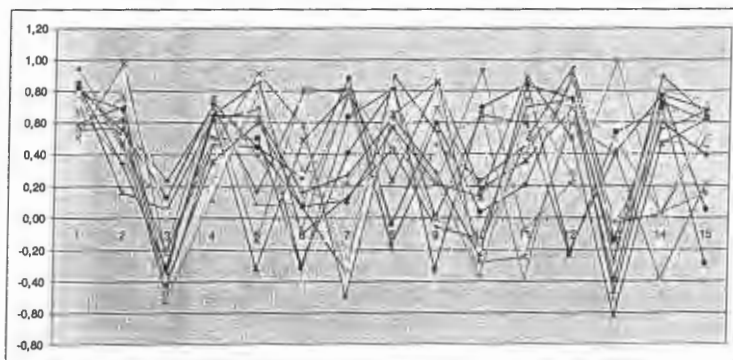
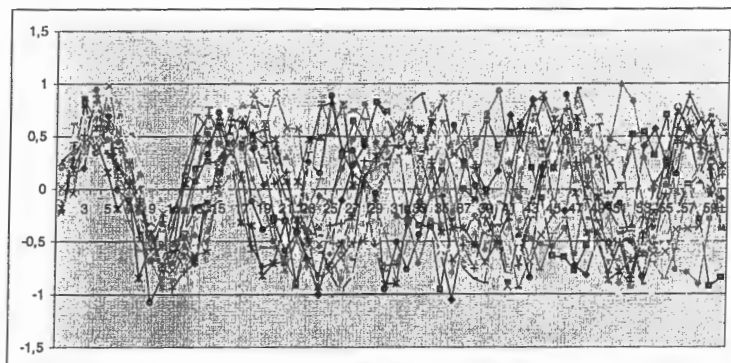
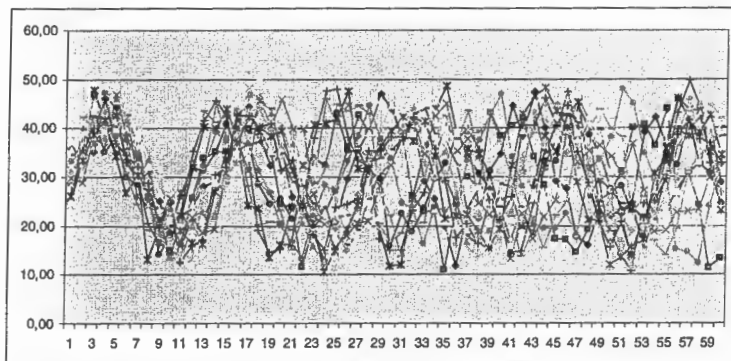
Rozpatrywane szeregi czasowe zapisano w postaci $[x_1(n), x_2(n), \dots, x_{60}(n)]^T$ $n = 1, 2, \dots, 75$. Znana była również przynależność każdego szeregu do klasy 1, klasy 2 lub 3. Szeregi były następnie normalizowane, tzn. każdy szereg ma wartość średnią równą zero zaś odchylenie standardowe wartość jeden.

Dla każdego n -ty szeregu zawierającego 60 wartości liczbowych tworzona była 4-krokowa obwiednia górna typu 2 (wzór (5)). Obwiednie górne 4-krokowe były następnie agregowane do szeregów $[x_1^{\sigma}(n), x_2^{\sigma}(n), \dots, x_{15}^{\sigma}(n)]^T$, $n = 1, 2, \dots, 75$ (wzór (9)).

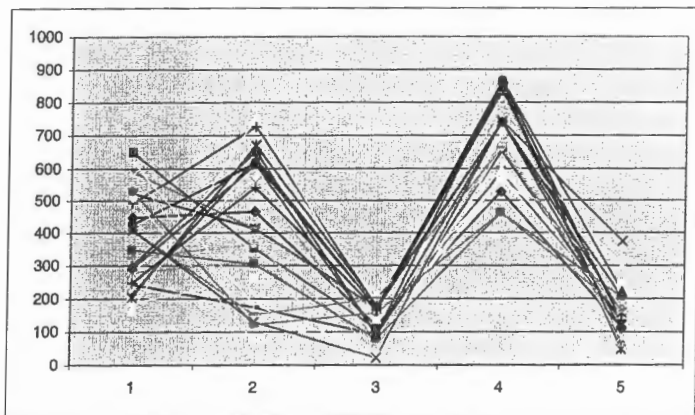
Następnie wygenerowano 5 cech istotnych, stosując w tym celu pamięć asocjacyjną, realizowaną poprzez dwuwarstwową jednokierunkową sieć neuronową o 15 wejściach i 15 wyjściach, z jedną warstwą ukrytą, którą stanowiło 5 neuronów. Do tego celu został użyty program Java Neural Networks Simulator (JavaNNS). Dane uczące stanowiły zagregowane 4-krokowe obwiednie górne typu 2.

Po procesie uczenia sieci, otrzymane dla każdego szeregu wartości w warstwie ukrytej (przemnożone przez 1000) utworzyły skompresowany opis szeregu. Tak więc, dla każdego zadania, z zastosowaniem wartości cech istotnych zapisano każdy rozpatrywany szereg czasowy $\{x_k(n)\}_{k=1}^{k=60} = [x_1(n), x_2(n), \dots, x_{60}(n)]^T$, dla $n = 1, 2, \dots, 75$, w postaci zagregowanej $[x_1(n), x_2(n), \dots, x_5(n)]^T$, dla $n = 1, 2, \dots, 75$. Uzyskano w ten sposób 75 szeregów, zawierających po 5 wartości liczbowych (cech istotnych). Wartości cech istotnych były nominalizowane, w sposób podany w Tabeli 3. Cecha decyzyjna a_d z wartościami {1-klasa 1; 2-klasa 2, 3-klasa 3} określała przynależność szeregu do klasy.

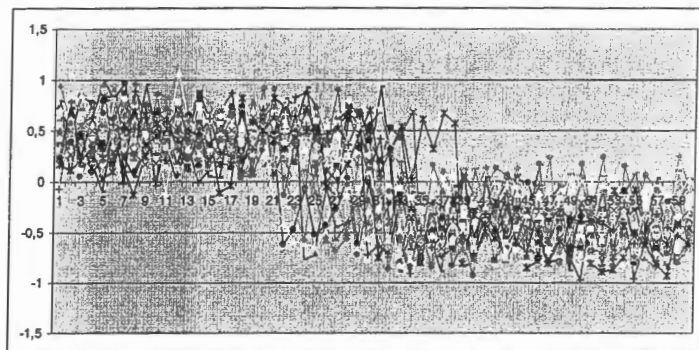
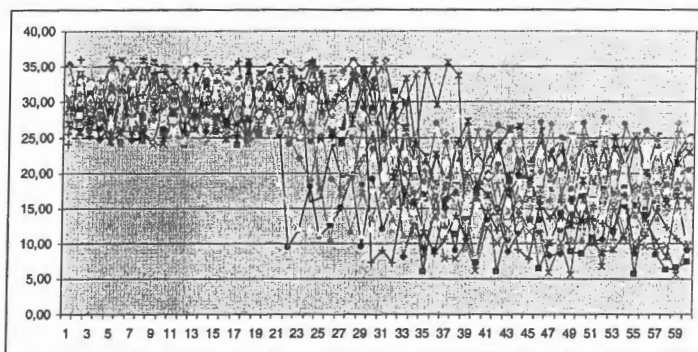
Wyżej wymienione szeregi, ich zapis po normalizacji i po agregacji ich obwiedni górnych typu 2 przedstawiono poniżej. Zamieszczono również ich zapis z zastosowaniem cech istotnych.

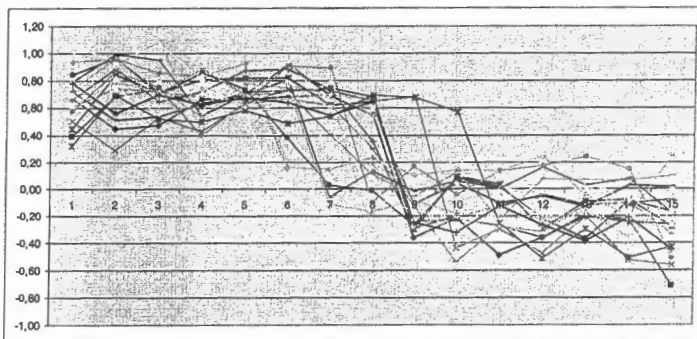


Rysunek 27. (góra) Szeregi z klasy 1 - typu B (środek) szeregi po normalizacji (dół) szeregi po agregacji obwiedni typu 2

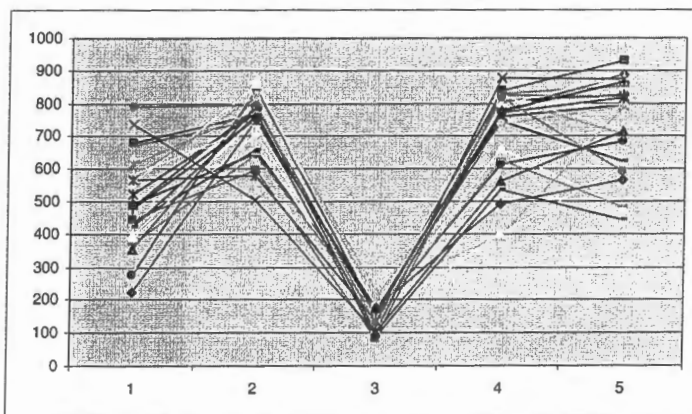


Rysunek 28. Zapis szeregów z klasy 1 z zastosowaniem cech istotnych

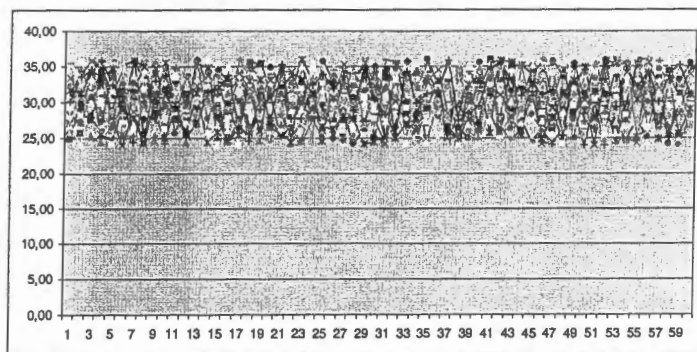


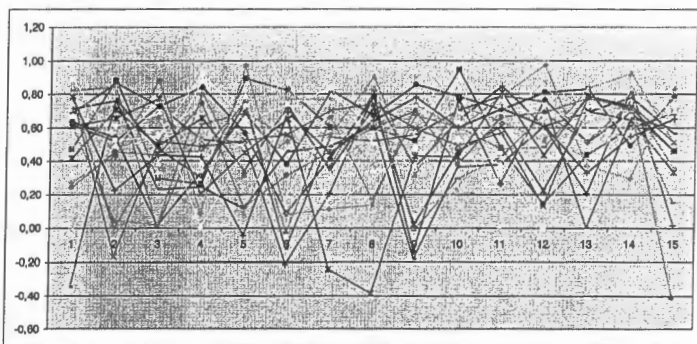
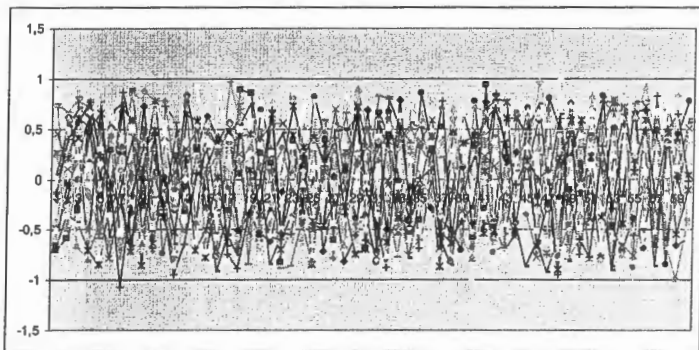


Rysunek 29. (górną) Szeregi z klasy 2 - szeregi typu F (środek) szeregi po normalizacji (dół) szeregi po agregacji obwiedni typu 2

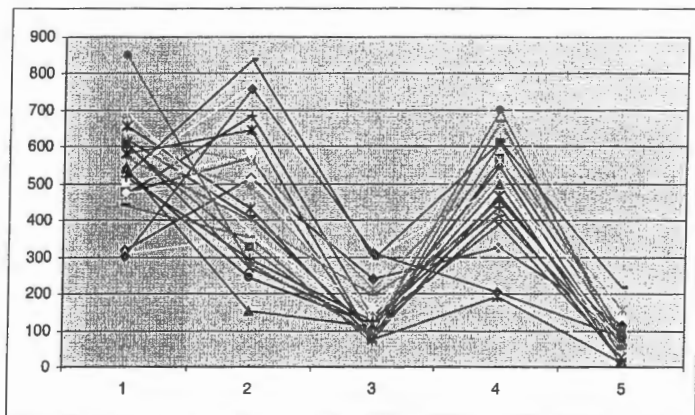


Rysunek 30. Zapis szeregów z klasy 2 z zastosowaniem cech istotnych

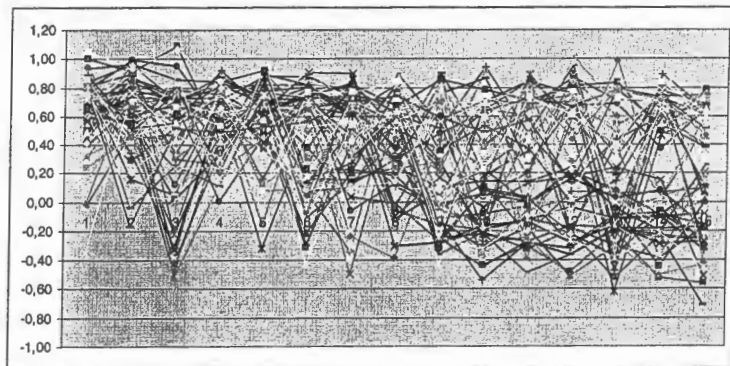
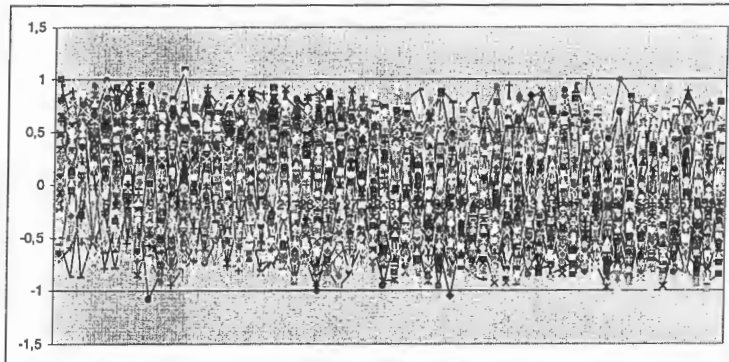
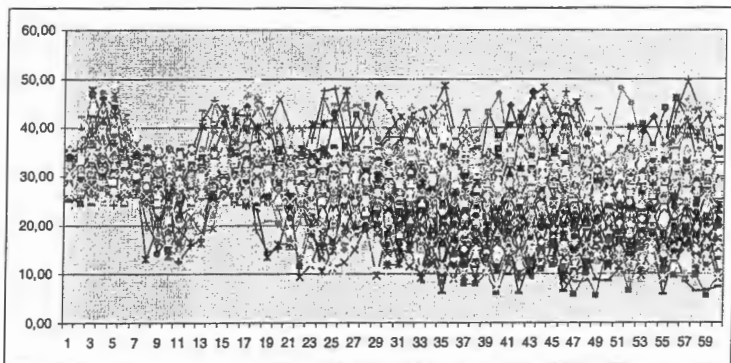




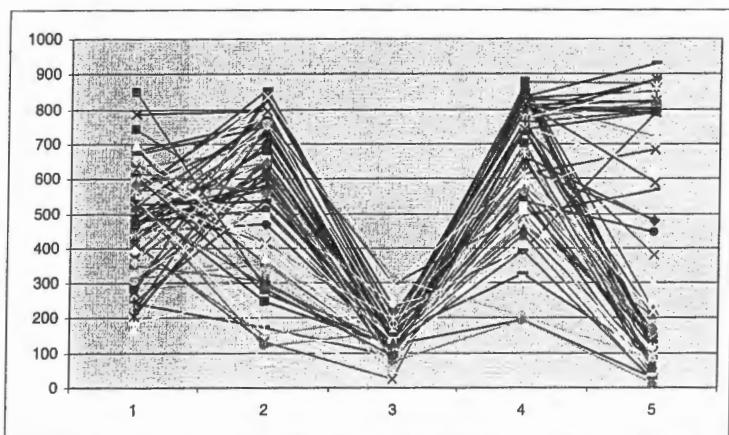
Rysunek 31. (górn) Szeregi z klasy 3 - szeregi typu A (środek) szeregi po normalizacji (dół) szeregi po agregacji obwiedni typu 2



Rysunek 32. Zapis szeregów z klasy 3 z zastosowaniem cech istotnych



Rysunek 33. (górn) Szeregi z klasy 1, klasy 2 i klasy 3 (środek) szeregi po normalizacji (dół) szeregi po agregacji obwiedni typu 2



Rysunek 34. Zapis szeregów z klasy 1, klasy 2 i klasy 3 z zastosowaniem cech istotnych

Na podstawie zbioru uczącego, zawierającego 75 szeregów opisanych za pomocą 5 cech istotnych, po nominalizacji, wygenerowano minimalny zbiór 29 reguł elementarnych, w nawiasach podano liczbę przykładów opisywanych przez daną regułę:

- | | |
|---|------------------|
| $(a1=5) \wedge (a5=2) \Rightarrow (a6=1)$ | (3 przykłady) |
| $(a2=4) \wedge (a5=2) \Rightarrow (a6=1)$ | (2 przykłady) |
| $(a1=3) \wedge (a5=2) \Rightarrow (a6=1)$ | (6 przykładów) |
| $(a4=9) \wedge (a5=1) \Rightarrow (a6=1)$ | (3 przykłady) |
| $(a2=2) \wedge (a5=2) \Rightarrow (a6=1)$ | (3 przykłady) |
| $(a1=2) \Rightarrow (a6=1)$ | (2 przykłady) |
| $(a5=4) \Rightarrow (a6=1)$ | (2 przykłady) |
| $(a1=3) \wedge (a5=1) \Rightarrow (a6=1)$ | (1 przykład) |
| $(a3=1) \wedge (a5=3) \Rightarrow (a6=1)$ | (5 przykładów) |
| $(a3=2) \wedge (a5=3) \Rightarrow (a6=1)$ | (1 przykład) |
| | |
| $(a5=7) \Rightarrow (a6=2)$ | (3 przykłady) |
| $(a5=9) \Rightarrow (a6=2)$ | (9 przykładów) |
| $(a5=8) \Rightarrow (a6=2)$ | (7 przykładów) |
| $(a5=5) \Rightarrow (a6=2)$ | (3 przykłady) |
| $(a5=6) \Rightarrow (a6=2)$ | (2 przykłady) |
| $(a5=10) \Rightarrow (a6=2)$ | (1 przykład) |
| | |
| $(a4=4) \wedge (a5=2) \Rightarrow (a6=3)$ | (1 przykład) |
| $(a4=6) \wedge (a5=1) \Rightarrow (a6=3)$ | (4 przykłady) |

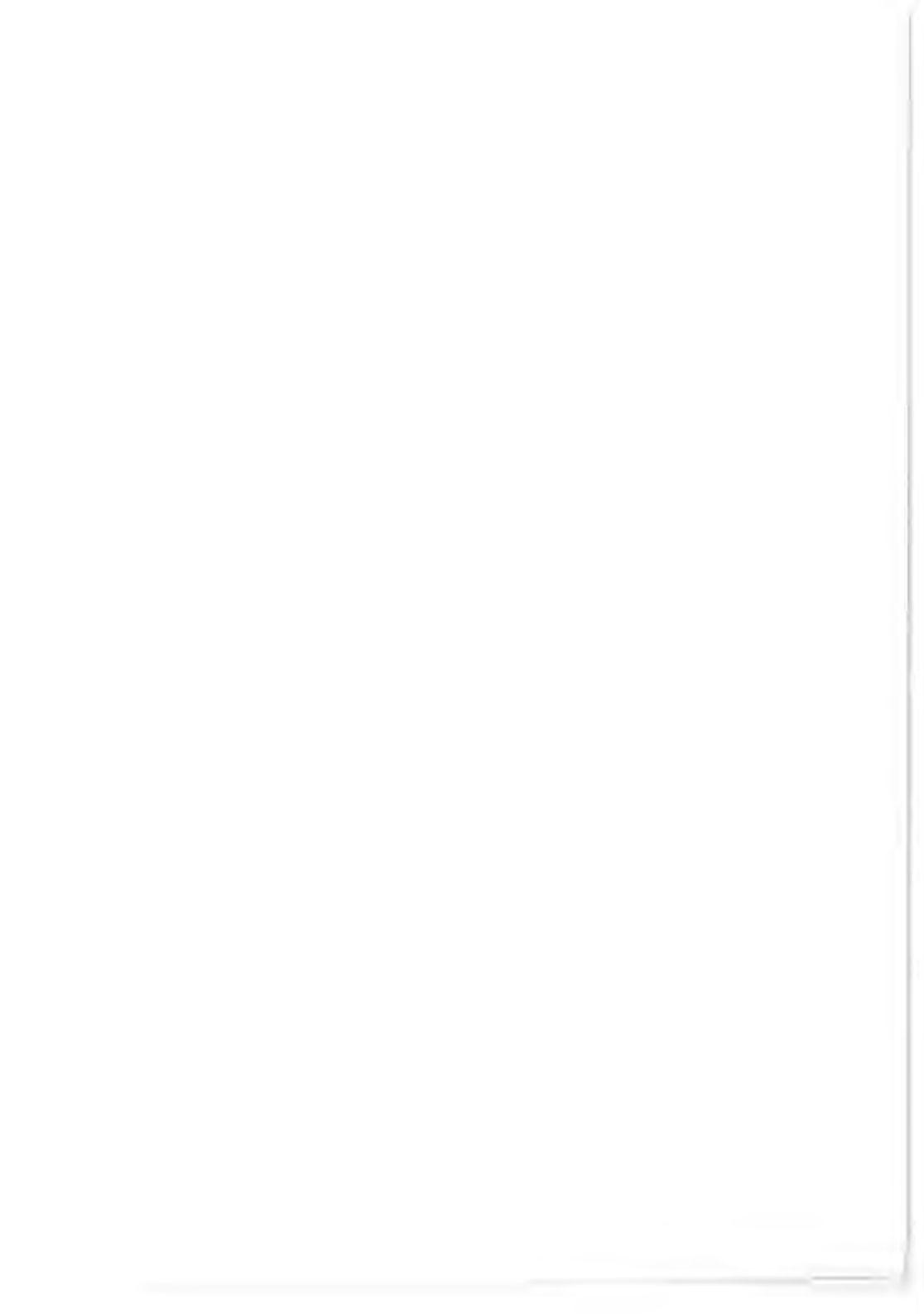
$(a4=5) \wedge (a5=1) \Rightarrow (a6=3)$	(7 przykładów)
$(a1=7) \wedge (a5=1) \Rightarrow (a6=3)$	(6 przykładów)
$(a1=5) \wedge (a2=4) \Rightarrow (a6=3)$	(1 przykład)
$(a4=2) \Rightarrow (a6=3)$	(2 przykłady)
$(a1=6) \wedge (a4=8) \wedge (a5=2) \Rightarrow (a6=3)$	(1 przykład)
$(a3=3) \wedge (a4=5) \Rightarrow (a6=3)$	(1 przykład)
$(a4=4) \wedge (a5=1) \Rightarrow (a6=3)$	(2 przykłady)
$(a1=7) \wedge (a3=3) \Rightarrow (a6=3)$	(1 przykład)
$(a3=3) \wedge (a5=3) \Rightarrow (a6=3)$	(1 przykład)
$(a4=3) \Rightarrow (a6=3)$	(1 przykład)
$(a1=9) \Rightarrow (a6=3)$	(1 przykład)

Wygenerowane reguły w 100 % poprawnie opisują wszystkie przykłady ze zbioru uczącego (25 przykładów z klasy 1, 25 przykładów z klasy 2 i 25 przykładów z klasy 3).

Literatura

1. Alcock, R. J., Manolopoulos, Y. (1999) Time-Series Similarity Queries Employing a Feature-Based Approach. 7th Hellenic Conference on Informatics, Ioannina, Greece.
2. 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.
3. Kacprzyk J., Szkatuła G. (1998) *IPI - An Improved Inductive Learning Procedure with a Preprocessing of Data*. Proceedings of IDEAL'98 (Hong Kong), Springer-Verlag.
4. Kacprzyk J., Szkatuła G. (1999) An inductive learning algorithm with a preanalysis of data. *International Journal of Knowledge - Based Intelligent Engineering Systems*, vol. 3, 135-146.
5. 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.
6. Krawczak, M., Miklewski, A., Jakubowski, A., Konieczny, P. (2000) Investment Risk Management, (in Polish). Polish Academy of Sciences, Systems Research 25.
7. Kumar N., Lolla N., Keogh E., Lonardi S., Ratanamahatana C., Wei L. (2005) Time-Series Bitmaps: A Practical Visualization Tool for Working with Large Time Series Databases. In proceedings of SIAM International Conference on Data Mining (SDM '05), Newport Beach, CA, April 21-23, 2005.
8. 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.
9. Nanopoulos, A., Alcock, R., & Manolopoulos, Y. (2001): Feature-based Classification of Time-series Data. *International Journal of Computer Research*, 49-61.
10. Oja E. (1992) Principal components, minor components and linear neural networks. *Neural Networks*, vol.5, ss 927-935.

11. Pawlak Z. (1982): Rough Set. *International Journal of Computer and Information Sciences*, Vol. 11, No 5, 341-356.
12. Pawlak Z. (1991): *Rough Set. Theoretical Aspect of Reasoning about Data*. Kluwer Academic Publishers.
13. 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.
14. 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.
15. Szkatuła G. (1995) *Machine learning from examples under errors in data*, Ph.D. thesis, SRI PAS Warsaw, Poland.
16. Szkatuła G. (2002). *Zastosowanie zmodyfikowanego zadania pokrycia w uczeniu maszynowym*. W: Gutenbaum J. (eds.): *Automatyka Sterowanie Zarządzanie*. SRI PAS, Warszawa, 431-445.
17. 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.
18. Wu, Y. & Chang, E.Y. (2004): Distance-function design and fusion for sequence data. *CIKM '04*, 324-333.
19. X. Xi, E. J. Keogh, C. R. Shelton, L. Wei, and C. A. Ratanamahatana. Fast time series classification using numerosity reduction. In *ICML, 2006*.





the 1990s, the number of people who have been employed in the public sector has increased in all countries. The increase has been particularly rapid in the United Kingdom, where the public sector has grown from 10.5% of the economy in 1970 to 17.5% in 1995 (see Figure 1).

There are a number of reasons for the increase in public sector employment. One of the main reasons is the increasing demand for public services, particularly in the areas of health care and education. Another reason is the increasing need for public services to be financed, particularly in the areas of health care and education. A third reason is the increasing need for public services to be provided, particularly in the areas of health care and education.

The increase in public sector employment has led to a number of problems. One of the main problems is the increasing cost of public services, particularly in the areas of health care and education. Another problem is the increasing need for public services to be financed, particularly in the areas of health care and education. A third problem is the increasing need for public services to be provided, particularly in the areas of health care and education.

The increase in public sector employment has also led to a number of other problems. One of the main problems is the increasing need for public services to be financed, particularly in the areas of health care and education. Another problem is the increasing need for public services to be provided, particularly in the areas of health care and education. A third problem is the increasing need for public services to be provided, particularly in the areas of health care and education.

The increase in public sector employment has also led to a number of other problems. One of the main problems is the increasing need for public services to be financed, particularly in the areas of health care and education. Another problem is the increasing need for public services to be provided, particularly in the areas of health care and education. A third problem is the increasing need for public services to be provided, particularly in the areas of health care and education.

The increase in public sector employment has also led to a number of other problems. One of the main problems is the increasing need for public services to be financed, particularly in the areas of health care and education. Another problem is the increasing need for public services to be provided, particularly in the areas of health care and education. A third problem is the increasing need for public services to be provided, particularly in the areas of health care and education.

The increase in public sector employment has also led to a number of other problems. One of the main problems is the increasing need for public services to be financed, particularly in the areas of health care and education. Another problem is the increasing need for public services to be provided, particularly in the areas of health care and education. A third problem is the increasing need for public services to be provided, particularly in the areas of health care and education.

The increase in public sector employment has also led to a number of other problems. One of the main problems is the increasing need for public services to be financed, particularly in the areas of health care and education. Another problem is the increasing need for public services to be provided, particularly in the areas of health care and education. A third problem is the increasing need for public services to be provided, particularly in the areas of health care and education.