Politechnika Śląska
Wydział Mechaniczny Technologiczny
Katedra Podstaw Konstrukcji Maszyn

Dominik Wachla

Identyfikacja dynamicznych modeli diagnostycznych metodami odkryć wiedzy w bazach danych

Gliwice 2006

Recenzenci
Dr hab. inż. Andrzej Pieczyński, prof. Uniwersytetu Zielonogórskiego
Prof. dr hab. inż. Wojciech Cholewa, Politechnika Śląska w Gliwicach

Redaktor zeszytów
Wojciech Cholewa

Redaktor techniczny
Marek Wyleżoł

Projekt okładki
Wojciech Cholewa, Marek Wyleżoł

ISBN 83–916957–9–4

Wydawca
Katedra Podstaw Konstrukcji Maszyn
Wydział Mechaniczny Technologiczny
Politechnika Śląska
ul. Konarskiego 18a, 44-100 Gliwice
tel. (32) 237-14-67, fax (32) 237-13-60
https://kpkm.polsl.pl

Druk i oprawa (z dostarczonych materiałów)
Drukarnia Internetowa i-Print.com.pl, 44-100 Gliwice

Od autora

Zeszyt został opracowany na podstawie mojej rozprawy doktorskiej, wykonanej pod kierunkiem prof. dra hab. Wojciecha Moczulskiego. Publiczna obrona rozprawy odbyła się 21 lutego 2006 roku przed Komisją powołaną przez Radę Wydziału Mechanicznego Technologicznego. W opracowaniu zostały uwzględnione uwagi recenzentów rozprawy doktorskiej: prof. dra hab. inż. Wojciecha Cholewy i dra hab. inż. Andrzeja Pieczyńskiego.

Niniejszą książkę dedykuję moim rodzicom i rodzeństwu w podzięce za wyrozumiałość i cierpliwość. Składam serdeczne podziękowania Koleżankom i Kolegom z Katedry Podstaw Konstrukcji Maszyn Politechniki Śląskiej za okazaną mi pomoc i życzliwość w czasie wykonywania pracy. Szczególnie dziękuję prof. dr. hab. Wojciechowi Moczulskiemu za pomoc w realizacji badań oraz cenne uwagi krytyczne.

Gliwice, marzec 2006 *Dominik Wachla*

Część badań opisanych w pracy wykonano w ramach projektu promotorskiego KBN 4 T07B 059 26 pod tytułem „Identyfikacja dynamicznych modeli diagnostycznych metodami odkryć wiedzy w bazach danych" finansowanego przez ówczesny Komitet Badań Naukowych, obecnie: Ministerstwo Nauki i Szkolnictwa Wyższego.

Spis treści

Wykaz ważniejszych oznaczeń

a, a_i	atrybut
$a()$, $a_i()$	wartość atrybutu
$\mathrm{acc}(\,)$	dokładność predykcji
$\arg\min f(\,)$	wartość minimalizująca $f(\,)$
A, A_i	zbiór (przestrzeń) atrybutów, podzbiór atrybutów
$\mathrm{card}(\,)$	moc zbioru
$f(\,)$	funkcja docelowa
h, $h(\,)$	wzorzec, model
$\mathbb{H}$	zbiór wzorców, modeli
L	zbiór przykładów uczących
$\mathbb{N}$	zbiór liczb naturalnych
$\mathbb{R}$	zbiór liczb rzeczywistych
θ	parametr, zbiór parametrów
t	czas, wartość statystyki t-Studenta
T	zbiór przykładów testowych, weryfikacyjnych, walidacyjnych
u_i	zmienna wejściowa
U	zbiór przykładów, $U \subseteq X$
U_i	podzbiór zbioru przykładów, $U_i \subset U$
v_i	wartość atrybutu
V, $V(\,)$	zbiór wartości atrybutów
w	waga
x, x_i	przykład $x \in X$, zmienna stanu
$\dot{x}$	pochodna zmiennej x
X	dziedzina
y, y_i	wartości funkcji docelowej
y_i	zmienna wyjściowa

Rozdział 1

Wstęp

W obszarze związanym z diagnostyką techniczną i eksploatacją maszyn istnieje wiele baz danych, które mogą być źródłem użytecznej wiedzy diagnostycznej. Próby zastosowania klasycznych metod analizy danych w przypadku tego typu zbiorów napotykają przeszkody związane z dużą liczbą danych, jak również z brakiem i niepoprawnymi wartościami zgromadzonych danych. Odpowiedzią na to zapotrzebowanie są metody rozwijane w ramach nowej dziedziny inżynierii wiedzy jaką jest odkrywanie wiedzy w bazach danych (ang. *Knowledge Discovery in Databases*, KDD).

Większość z metod KDD przeznaczona jest do analizy danych opisujących statyczne cechy systemów, procesów lub zjawisk fizycznych, podczas gdy w wielu bazach danych gromadzone są dane opisujące cechy dynamiczne. Możliwość zastosowania wybranej metody KDD do analizy danych opisujących dynamikę systemów jest w tym przypadku determinowana przez odpowiednie przekształcenie tych danych.

W przypadku baz danych, przekształcenie tego typu odniesione do określonej grupy atrybutów prowadzi do utworzenia zbioru *nowych* atrybutów. Negatywnym efektem dokonanej transformacji jest znaczące zwiększenie liczby *nowych* atrybutów, spośród których tylko nieliczne mają istotne znaczenie w procesie indukcji wiedzy wybraną metodą KDD. Celowa jest zatem redukcja tego zbioru do zbioru atrybutów relewantnych. Pomocne w tym zakresie są również metody stosowane w ramach KDD.

Jednym z zadań KDD jest zadanie poszukiwania zależności funkcyjnych nazywane aproksymacją funkcji. W takim przypadku selekcja atrybutów relewantnych prowadzona jest za pomocą rozwiązań łączących metody KDD z metodami rozwijanymi w dziedzinie identyfikacji systemów.

W ramach pracy zaproponowano nową metodę identyfikacji modeli obiektów i procesów, przydatną diagnostycznie. Opracowana metoda analizy ilościowych danych opisujących dynamikę procesu lub obiektu bazuje na zastosowaniu dostępnych metod odkrywania wiedzy w bazach danych. Jej istotą jest projekcja wartości rozpatrywanego zbioru atrybutów w tzw. wielowymiarową przestrzeń regresorów. Do selekcji podzbioru cech relewantnych zastosowano algorytm genetyczny, a do indukcji wiedzy metodę wektorów wspomagających (ang. *Support Vector Machines*, SVM). Jako kryterów oceniających użyto miar AIC, MDL oraz własnej funkcji heurystycznej. Opracowaną metodę poddano weryfikacji dla dwóch różnych zbiorów danych: wygenerowanych w ramach eksperymentu

numerycznego oraz zgromadzonych przez przemysłowy system rejestracji danych na specyficznym obiekcie, jakim była pompownia wód głębinowych. Zrealizowano obszerny plan badań weryfikacyjnych, obejmujący wiele wariantów parametrów sterujących przebiegiem obliczeń. Wyniki weryfikacji potwierdzają skuteczność opracowanej metody oraz przydatność uzyskanych modeli do zastosowań w detekcji uszkodzeń bazującej na modelu obiektu.

Tematyka pracy wpisuje się w rozwój nowoczesnej dziedziny naukowej, jaką jest diagnostyka procesów przemysłowych. W pracy dokonano połączenia wielu wątków, przedstawiając zastosowanie nowych metod sztucznej inteligencji, związanych z tzw. odkrywaniem wiedzy w bazach danych, gromadzonych przez systemy diagnostyczne. Uzyskane wyniki w zakresie metodyki identyfikacji modeli mają znaczenie aplikacyjne i mogą stanowić przedmiot potencjalnego wdrożenia.

1.1. Zakres rozprawy

Praca obejmuje opis metody odkrywania modeli opisujących dynamikę procesów i obiektów z wykorzystaniem metod rozwijanych w ramach nowej dziedziny jaką jest odkrywanie wiedzy w bazach danych. Pierwszy rozdział pracy jest krótkim wstępem będącym wprowadzeniem do tematyki pracy.

W rodziale drugim sformułowano problem badawczy, jakim jest potrzeba opracowania metod identyfikacji zależności, które mogą występować w istniejących i dostępnych bazach danych gromadzących wyniki diagnostycznych i eksploatacyjnych obserwacji procesów przemysłowych zachodzących w różnego rodzaju obiektach technicznych, w tym w instalacjach przemysłowych. Potrzeba identyfikacji takich zależności wynika m.in. z dużej ilości dostępnych danych pomiarowych a także jest związana z trudnościami w budowaniu modeli diagnostycznych z zastosowaniem klasycznych technik modelowania, co wynika ze złożoności obserwowanych procesów jak i obiektów.

Rozdział trzeci zawiera przegląd modeli znajdujących szerokie zastosowanie w diagnostyce procesów przemysłowych i obiektów. Wśród omawianych modeli wyróżniono modele do detekcji uszkodzeń oraz modele do lokalizacji uszkodzeń i rozpoznawania stanu technicznego obiektu lub procesu. Podział modeli przyjęto zgodnie z obowiązującą klasyfikacją modeli w dziedzinie diagnostyki procesów.

W rozdziale czwartym opisano wybrane problemy nowej dziedziny inżynierii wiedzy, jaką jest KDD. Podano definicję odkrywania wiedzy w bazach danych jako procesu składającego się z kilku powiązanych ze sobą etapów, których realizacja może doprowadzić do odkrycia nowej i interesującej wiedzy. W dalszej części rozdziału omówiono poszczególne stadia procesu KDD. Następnie scharakteryzowano zadania odkrywania wiedzy oraz szczegółowo omówiono metody stosowane na poszczególnych etapach procesu odkrywania wiedzy. Rozdział zakończono krótkim przeglądem zastosowań metod KDD w zakresie diagnostyki technicznej.

W rozdziale szóstym opisano zaproponowaną metodę odkrywania ilościowych zależności funkcyjnych. Przedstawiono pojęcia *system informacyjny* oraz *temporalny system*

informacyjny służące do formalnego opisu danych. Na bazie tych dwóch pojęć przedstawiono algorytm przekształcający temporalny system informacyjny w „*zwykły*" system informacyjny. Dzięki temu przekształceniu dane opisujące własności dynamiczne procesu lub obiektu mogą być analizowane z zastosowaniem metod przewidzianych do analizy danych opisujących własności statyczne. W kolejnej części rozdziału opisano istotę indukcji zależności funkcyjnych na podstawie danych. Dokonano przeglądu miar stosowanych do oceny jakości pozyskanych modeli oraz sposobów ich wyznaczania.

Przekształcenie temporalnego systemu informacyjnego w „*zwykły*" system informacyjny powoduje zwiększenie liczby rozpatrywanych atrybutów. Nie wszystkie z tych atrybutów mają istotne znaczenie dla jakości zidentyfikowanej zależności funkcyjnej. W związku z tym w dalszej części rozdziału szóstego opisano przyjętą metodę selekcji atrybutów relewantnych z zastosowaniem algorytmu indukcji wiedzy. W szczególności przedstawiono kryteria selekcji atrybutów relewantnych oraz istotne elementy algorytmu genetycznego, który wybrano jako metodę przeszukiwania przestrzeni atrybutów.

Rozdział siódmy zawiera opis badań weryfikacyjnych prowadzonych dla opisanej w rozdziale 6 metody. Rozdział ten składa się z dwóch zasadniczych części. W części pierwszej przedstawiono przebieg jak i wyniki weryfikacji opracowanej metody dla danych pozyskanych w wyniku przeprowadzonych eksperymentów numerycznych, które obejmowały symulacje testowych systemów dynamicznych. W szczególności opisano wybrane do symulacji systemy dynamiczne, jak również parametry i przebieg symulacji. Na podstawie pozyskanych danych przeprowadzono badania weryfikacyjne, na które złożyło się m.in.: przyjęcie planu eksperymentu a także ustalenie wartości parametrów proponowanej metody. W dalszej części przedstawiono w formie tabelarycznej wyniki badań. Wybrane modele opisano w szerszej formie, która obejmowała dodatkowo (poza wynikami zamieszczonymi w tabelach): zbiór wyselekcjonowanych zmiennych wejściowych, wykres przewidywanych za pomocą modelu wartości jakie powinny pojawić się na wyjściu systemu testowego oraz wykres szeregu reszt będcy różnicą pomiędzy wartościami pojawiającymi się na wyjściu systemu a wartościami generowanymi przez modele. Tę część weryfikacji zakończono przeprowadzając dyskusję uzyskanych wyników.

Druga część rozdziału siódmego obejmuje opis badań weryfikacyjnych realizowanych z zastosowaniem wybranej części danych zgromadzonych w bazie danych systemu klasy SCADA, za pomocą którego kontrolowana jest praca pompowni głębinowej. Ta część badań dotyczyła weryfikacji metody pod względem zastosowania praktycznego. W szczególności scharakteryzowano pozyskane do badań dane i dokonano analizy tych danych pod kątem przydatności do odkrywania modeli o znaczeniu diagnostycznym. Następnie opisano proces przygotowania danych oraz przyjęty plan eksploracji danych. Podobnie jak w przypadku weryfikacji prowadzonej dla danych pozyskanych na podstawie eksperymentów numerycznych, również w tym przypadku uzyskane wyniki przedstawiono w postaci tablic, przy czym wybrane przykłady odkrytych modeli przedstawiono w formie poszerzonej o elementy graficzne. Przeprowadzono dyskusję wyników oraz dla wybranych modeli pokazano przykład ich zastosowania diagnostycznego, który polegał na detekcji zmiany stanu technicznego z zastosowaniem testu t-Studenta. Rozdział zakoń-

czono krótkim podsumowaniem obejmującym obydwa zakresy przeprowadzonych badań weryfikacyjnych.

W rozdziale ósmym dokonano podsumowania pracy, przedstawiono wnioski płynące z przeprowadzonych badań oraz wskazano kierunki dalszych badań.

Rozdział 2

Określenie problemu badawczego

We współczesnej diagnostyce technicznej, powszechne jest wnioskowanie bazujące na obserwacji symptomów. Obserwacja symptomów nie pozwala bezpośrednio wnioskować o cechach maszyny (np. wielkościach niewyrównoważeń, postaciach luzów łożyskowych, położenia podpór łożyskowych), ale jest podstawą diagnozowania stanu technicznego obiektu w sposób pośredni, na podstawie mierzonych lub estymowanych cech sygnałów. Taki sposób oceny stanu technicznego maszyn wpływa na okresy międzynaprawcze powodując albo ich skrócenie, co pociąga za sobą zwiększenie kosztów obsługi, albo ich wydłużenie, co skutkuje zwiększeniem prawdopodobieństwa wystąpienia awarii. Szczególnie jest to istotne w przypadku maszyn krytycznych, dla których w tym celu buduje się układy ciągłego nadzoru.

Taki stan rzeczy powoduje, iż w ostatnim czasie wzrosło zainteresowanie diagnostyką opartą na modelach. Pomimo dobrze rozwiniętego aparatu matematycznego, identyfikacja diagnostycznych relacji ilościowych pomiędzy wartościami cech sygnałów a stanem maszyny sprawia wciąż wiele trudności. Należy rozpatrzyć następujące przesłanki wpływające na rozwiązanie tego problemu:

- złożoność współczesnych obiektów technicznych,
- trudność w budowaniu klasycznych modeli matematycznych,
- nadmiar danych, przejawiający się w dużej liczbie baz danych gromadzących wyniki eksperymentów diagnostycznych lub wyniki monitorowania, a także w ilości gromadzonych danych (np. przez systemy monitorowania maszyn krytycznych),
- brak zrozumienia danych - ludzie (specjaliści) nie potrafią interpretować danych, nie mają czasu, nie potrafią przekazywać wiedzy,
- fakt, że wiele zmian stanu obiektów technicznych zachodzi w bardzo długim czasie i dla człowieka mogą być niezauważalne, a także mogą nie być objęte horyzontem czasowym rozpatrywanej bazy danych.

Problemem wymagającym rozwiązania jest zgromadzenie i/lub opracowanie nowych, bardziej uniwersalnych i skutecznych metod identyfikacji ilościowych modeli na podstawie danych gromadzonych w bazach danych diagnostycznych i eksploatacyjnych. Podstawą do opracowania takich metod mogłyby być stosowane coraz częściej w świecie metody odkrywania wiedzy w bazach danych,

a w szczególności bazujące na nich metody odkrywania zależności funkcyjnych. Metody te pozwalają na ilościowy opis statycznych i/lub dynamicznych cech obiektu lub procesu, które mogą występować w danych będących wynikiem obserwacji tegoż obiektu lub procesu. Charakterystyczne cechy metod KDD pozwalają sądzić, że ich zastosowanie skróci czas identyfikacji modeli, polepszy ich jakość, a także pozwoli na pozyskanie nowych, niezidentyfikowanych jeszcze zależności diagnostycznych.

Rozdział 3

Modele w diagnostyce procesów

Celem istnienia środków technicznych m.in. takich jak maszyny i urządzenia (mechaniczne lub elektryczne), instalacje przemysłowe i megaukłady maszynowe jest realizacja określonych działań zwanych *procesami*. W uproszczeniu można przyjąć, że proces realizowany przez dany środek techniczny jest *ciągiem stanów*, z których każdy stanowi *statyczny opis własności i właściwości środka technicznego* [61]. Istotą procesu jest zmiana [61].

Do procesów związanych ze środkami technicznymi można m.in. zaliczyć szeroko rozumiane:

- *procesy wytwórcze* (produkcyjne),
- *procesy eksploatacyjne*.

W wielu przypadkach badanie właściwości tych procesów wymaga opracowania modelu procesu. Potrzeba modelowania procesów w szczególności jest widoczna w dziedzinie diagnostyki technicznej, której celem jest „*określenie szeroko rozumianego stanu technicznego urządzeń za pomocą obiektywnych metod i środków*" [12].

3.1. Diagnozowanie z zastosowaniem modelu procesu

Diagnozowanie z zastosowaniem modelu jest typowym podejściem stosowanym w diagnostyce procesów przemysłowych [43, 73]. W dziedzinie tej rozpatruje się dwa odrębne zadania [43]:

- *Detekcję uszkodzenia* (ang. *fault detection*) – której celem jest stwierdzenie, czy w nadzorowanym układzie występuje niesprawność/uszkodzenie (tzw. diagnostyka dwustanowa dla klas stanów {"zdatny", "niezdatny"}).
- *Diagnozę uszkodzenia* (ang. *fault diagnosis*) – obejmuje lokalizację uszkodzenia (ang. *fault isolation*), której celem jest określenie rodzaju, miejsca i czasu wystąpienia uszkodzenia, oraz identyfikację uszkodzenia (ang. *fault identification*), która dotyczy określenia rozmiaru uszkodzenia i charakteru jego zmienności w czasie [36], a nawet przyczyny uszkodzenia [61].

3.2. Modele procesów do detekcji uszkodzeń

Detekcja uszkodzeń jest procesem generacji sygnałów diagnostycznych na podstawie zmiennych procesowych w celu wykrywania uszkodzeń [43]. Rozróżnia się dwie grupy metod detekcji uszkodzeń [43]:

- metody bazujące na związkach występujących między zmiennymi procesowymi,
- metody bazujące na kontroli zmiennych procesowych.

W pierwszej z grup stosowane są m.in. następujące rodzaje modeli procesów [43,46]:

- modele analityczne (równania fizyczne, równania stanu, transmitancja operatorowa),
- modele neuronowe,
- modele rozmyte,
- modele neuronowo–rozmyte.

W następnych podpunktach pracy krótko scharakteryzowano wymienione klasy modeli.

3.2.1. Równania fizyczne

Równania fizyczne są wynikiem *modelowania matematycznego* [81] dynamiki obiektów lub procesów. Dynamikę obiektu lub procesu o jednym wejściu i jednym wyjściu można opisać za pomocą następującego nieliniowego równania różniczkowgo:

$$f\left(y, \frac{dy}{dt}, \frac{d^2y}{dt^2}, \ldots, \frac{d^ny}{dt^n}, u, \frac{du}{dt}, \frac{d^2u}{dt^2}, \ldots, \frac{d^mu}{dt^m}\right) = 0. \tag{3.1}$$

Przykładem równań fizycznych może być układ równań różniczkowych określających bilans przepływów w układzie trzech zbiorników [43, 46].

Model (3.1) można zastosować do wykrywania uszkodzeń obiektu na podstawie oceny wartości residuów wyliczanych w następujący sposób:

$$r = f\left(y, \frac{dy}{dt}, \frac{d^2y}{dt^2}, \ldots, \frac{d^ny}{dt^n}, u, \frac{du}{dt}, \frac{d^2u}{dt^2}, \ldots, \frac{d^mu}{dt^m}\right). \tag{3.2}$$

Modele zbudowane na podstawie równań fizycznych najpełniej opisują związki między zmiennymi procesowymi. Ta cecha sprawia, że niewielkie odchylenia parametrów obserwowanego procesu będzie można wykryć stosując ten typ modeli.

Dla wielu obiektów opracowanie modelu w postaci równań fizycznych jest bardzo trudne lub wręcz niemożliwe, a identyfikacja ich parametrów sprawia wiele trudności.

3.2.2. Transmitancja i liniowe równania stanu obiektu

W przypadku kiedy opisanie układu dynamicznego za pomocą równań fizycznych sprawia trudności, sięga się do metod opisu właściwości dynamicznych bazujących na *transmitancji operatorowej* lub *liniowych równaniach stanu*. Ponieważ struktura tych

modeli jest niezmienna zadanie identyfikacji polega na określeniu wartości parametrów tych modeli. W tym celu stosuje się metody rozwijane w ramach identyfikacji systemów [56, 81]. Zarówno *transmitancja operatorowa* jak i *liniowe równania stanu* są stosowane do opisu stacjonarnych modeli liniowych. W przypadku układów nieliniowych służą do modelowania ich własności dynamicznych w otoczeniu *punktu pracy*.

Równania stanu obiektu liniowego

Dynamiczny stacjonarny układ liniowy o p wejściach:

$$\mathbf{u}(t) = [u_1(t), u_2(t), \ldots, u_p(t)]^T, \tag{3.3}$$

oraz o q wyjściach:

$$\mathbf{y}(t) = [y_1(t), y_2(t), \ldots, y_q(t)]^T, \tag{3.4}$$

można opisać za pomocą *równań stanu* z czasem ciągłym:

$$\begin{aligned} \dot{\mathbf{x}}(t) &= \mathbf{A}\mathbf{x}(t) + \mathbf{B}\mathbf{u}(t), \\ \mathbf{y}(t) &= \mathbf{C}\mathbf{x}(t) + \mathbf{D}\mathbf{u}(t), \end{aligned} \tag{3.5}$$

lub z czasem dyskretnym:

$$\begin{aligned} \mathbf{x}(k+1) &= \mathbf{A}\mathbf{x}(k) + \mathbf{B}\mathbf{u}(k), \\ \mathbf{y}(k) &= \mathbf{C}\mathbf{x}(k) + \mathbf{D}\mathbf{u}(k), \end{aligned} \tag{3.6}$$

gdzie: $\mathbf{x}(t)$ – n wymiarowy wektor stanu,
$\mathbf{A}$ – macierz układu (procesu) o wymiarze $n \times n$,
$\mathbf{B}$ – macierz sterowania (wejścia) o wymiarze $n \times p$,
$\mathbf{C}$ – macierz odpowiedzi (wyjścia) o wymiarze $q \times n$,
$\mathbf{D}$ – macierz o wymiarze $q \times p$.

Transmitancja obiektu liniowego

W dziedzinie automatyki powszechnie stosowanym opisem liniowych układów dynamicznych jest *transmitancja operatorowa* [38, 46]:

$$G(s) = \frac{y(s)}{u(s)}, \tag{3.7}$$

gdzie: $y(s)$ i $u(s)$ są transformatami Laplace'a odpowiednio sygnału wyjściowego $y(t)$ oraz sygnału wejściowego $u(t)$ przy zerowych warunkach początkowych. Własności dynamiczne wielowymiarowych stacjonarnych układów liniowych o p wejściach i q wyjściach są opisywane za pomocą macierzy *transmitancji operatorowych*:

$$\mathbf{G}(s) = \frac{\mathbf{y}(s)}{\mathbf{u}(s)} = \begin{bmatrix} G_{11}(s) & G_{12}(s) & \cdots & G_{1p}(s) \\ G_{21}(s) & G_{22}(s) & \cdots & G_{2p}(s) \\ \vdots & \vdots & & \vdots \\ G_{q1}(s) & G_{q2}(s) & \cdots & G_{qp}(s) \end{bmatrix} \tag{3.8}$$

gdzie: $G_{ij} = \frac{y_i(s)}{u_j(s)}$, $i = 1, 2, \ldots, q$, $j = 1, 2, \ldots, p$, jest transmitancją operatorową między i-tym wyjściem a j-tym wejściem układu.

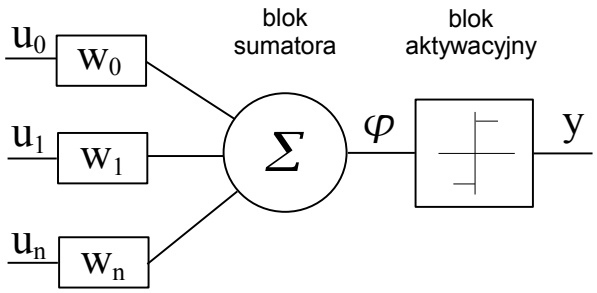
W przypadku układu z czasem dyskretnym wyznaczana jest *transmitancja dyskretna*, która jest ilorazem transformaty $\mathbf{Z}$ sygnału wyjściowego $y(z)$ do transformaty $\mathbf{Z}$ sygnału wejściowego $u(z)$ przy zerowych warunkach początkowych:

$$G(z) = \frac{y(z)}{u(z)}. \tag{3.9}$$

Zastosowanie diagnostyczne modeli w postaci *transmitancji operatorowych* lub *liniowych równań stanu* jest ograniczone z uwagi na trudności związane z uzyskaniem odpowiedniej dokładności generowania residuów, które powinny być czułe na uszkodzenia a niewrażliwe na naturalne zakłócenia procesu, szumy pomiarowe, błędy modelowania, zmiany punktu pracy [46] itp.

3.2.3. Modele neuronowe

Sztuczna sieć neuronowa jest zbiorem wielu połączonych i równolegle działających elementów zwanych *neuronami* (Rys. 3.1).



Rys. 3.1: Schemat neuronu [46]

Podstawowymi elementami neuronu są:

- bloki mnożenia sygnałów wejściowych u_i,
- blok sumatora sygnałów wejściowych,
- blok aktywacji neuronu, który generuje sygnał wyjściowy neuronu y.

Identyfikacja systemu z zastosowaniem sieci neuronowych wymaga określenia struktury (liczby warstw sieci oraz liczby neuronów w każdej warstwie), postaci i wartości parametrów funkcji aktywacji oraz doboru wartości wag. Wartości wag są ustalane w trakcie procesu uczenia sieci, który może być prowadzony z nadzorem lub bez nadzoru.

Do modelowania obiektów dynamicznych stosuje się m.in.:

- sieci neuronowe typu perceptron wielowarstwowy,
- sieci radialne RBF (ang. *Radial Basis Function*),
- sieci typu GMDH (ang. *Group Method Data Handling*),

- sieci rekurencyjne,
- sieci dynamiczne.

Podstawowe informacje o sieciach neuronowych można znaleźć w następujących pozycjach literaturowych [24, 42–44, 83].

3.2.4. Modele rozmyte

Modelowanie rozmyte bazuje na logice rozmytej będącej częścią teorii zbiorów rozmytych zaproponowanej przez L. Zadeha (1965). Modele rozmyte pozwalają w sposób naturalny dla człowieka zapisać działanie danego systemu w postaci zbioru reguł *jeżeli – to*.

Identyfikacja modeli rozmytych wymaga połączenia wiedzy eksperta oraz dostępnych danych pomiarowych. Wiedza ekspercka służy do określenia struktury oraz początkowych wartości parametrów modelu (rozmieszczenie funkcji przynależności), natomiast dane pomiarowe do strojenia modelu.

Wynikiem modelowania rozmytego jest baza reguł (model rozmyty), która ujmuje związki pomiędzy stanami i wyjściami identyfikowanego obiektu. W ogólnym przypadku baza ta jest n–wymiarowa, gdzie n oznacza liczbę zmiennych wejściowych.

Jedną z najczęściej stosowanych metod budowy modeli rozmytych jest metoda Wanga i Mendla (WM), która pozwala identyfikować rozmyte modele dynamiczne. Proces identyfikacji modelu rozmytego za pomocą metody WM składa się z następujących etapów:

1. zdefiniowanie zmiennych lingwistycznych oraz ich podział na obszary,
2. generacja reguł na podstawie eksperymentu,
3. przyporządkowanie wag regułom,
4. utworzenie bazy reguł.

Szersze omówienie tej metody zostało zawarte m.in. w pracy [75].

3.2.5. Modele neuronowo–rozmyte

Rozmyte sieci neuronowe (ang. *Fuzzy Neural Network*, FNN) stanowią połączenie techniki modelowania rozmytego z metodami uczenia sieci neuronowych. Struktura FNN reprezentuje proces wnioskowania rozmytego, a więc pozwala w trakcie budowy modelu uwzględnić wiedzę eksperta (liczba reguł, funkcje przynależności). FNN zostały zaproponowane przez Hirokawę.

W FNN można wyróżnić dwie części [46]:

1. pierwszą, odpowiadającą przesłankom reguł rozmytych, która jest odpowiedzialna za obliczenie poziomów „*zapłonów*" reguł,
2. drugą, odpowiadającą konkluzjom reguł rozmytych, której zadaniem jest obliczenie wyjścia sieci na podstawie wypracowanych wartości przesłanek.

W [46] wyróżniono trzy rodzaje rozmytych sieci neuronowych:

- z wyjściami w postaci singletonów,
- z wyjściami stanowiącymi liniową kombinację wejść (model TSK),
- z wyjściami będącymi liczbą rozmytą.

Do trenowania FNN stosowana jest najczęściej metoda wstecznej propagacji błędów.

3.3. Modele do lokalizacji uszkodzeń

Lokalizacja uszkodzeń prowadzona jest na podstawie sygnałów diagnostycznych generowanych przez algorytmy detekcyjne. Wynikiem lokalizacji jest diagnoza wskazująca uszkodzenia lub klasy stanu obiektu, które mogą obejmować podzbiory stanów z różnymi uszkodzeniami. Do lokalizacji uszkodzeń niezbędna jest znajomość relacji między sygnałami diagnostycznymi a uszkodzeniami lub ewentualnie stanami technicznymi.

Sygnałami wejściowymi w procesie lokalizacji uszkodzeń lub rozpoznawania stanów obiektu są [43]:

- residua generowane na podstawie modeli obiektów,
- binarne lub wielowartościowe sygnały powstałe w wyniku kwantowania wartości residuów,
- binarne lub wielowartościowe sygnały generowane z zastosowaniem klasycznych i heurystycznych metod detekcji uszkodzeń,
- parametry statystyczne (cechy) opisujące właściwości sygnałów losowych,
- zmienne procesowe tzn. mierzone lub wyliczane sygnały wielkości fizycznych.

Zasadniczą cechą modeli do lokalizacji uszkodzeń lub rozpoznawania stanu jest to, że odwzorowują one przestrzeń wartości sygnałów diagnostycznych w dyskretną przestrzeń uszkodzeń lub stanów obiektu.

W pracy [46] jako podstawowe kryterium klasyfikacji modeli do lokalizacji uszkodzeń przyjęto sposób pozyskania wiedzy o relacji *sygnały diagnostyczne* $\rightarrow$ *uszkodzenie*.

Zgodnie z przyjętą w [46] klasyfikacją, wśród modeli do lokalizacji uszkodzeń można wyróżnić modele realizujące następujące odwzorowania:

- binarne sygnały diagnostyczne $\Rightarrow$ uszkodzenia lub stany obiektu,
- wielowartościowe sygnały diagnostyczne $\Rightarrow$ uszkodzenia lub stany obiektu,
- ciągłe sygnały diagnostyczne $\Rightarrow$ uszkodzenia lub stany obiektu.

3.3.1. Modele odwzorowujące binarne sygnały diagnostyczne

Binarne sygnały diagnostyczne powstają w wyniku dwuwartościowej oceny residuów, cech sygnałów diagnostycznych lub zmiennych procesowych [43]. Można je również uzyskać poprzez kontrolę ograniczeń lub poprzez analizę związków występujących pomiędzy zmiennymi procesowymi. Wśród modeli służących do lokalizacji uszkodzeń oraz wykorzystujących binarne sygnały diagnostyczne można wymienić:

- binarną macierz diagnostyczną,
- drzewa i grafy diagnostyczne,
- reguły i funkcje logiczne.

Binarna macierz diagnostyczna

Binarna macierz diagnostyczna jest relacją określoną na iloczynie kartezjańskim zbiorów uszkodzeń $F = \{f_k,\, k = 1, 2, \ldots, K\}$ oraz sygnałów diagnostycznych $S = \{s_j, 1, 2, \ldots, J\}$:

$$R_{FS} \subset F \times S. \tag{3.10}$$

Binarną macierz diagnostyczną można określić na podstawie równań residuów uwzględniających wpływ uszkodzeń lub na podstawie wiedzy eksperckiej poprzez analizę wpływu uszkodzeń na wartości sygnałów diagnostycznych. Tablica 3.1 jest przykładem binarnej macierzy diagnostycznej dla zespołu trzech zbiorników [43, 46].

Tab. 3.1: Przykład binarnej macierzy diagnostycznej dla zespołu trzech zbiorników [43]

S/F	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	f_{10}	f_{11}	f_{12}	f_{13}	f_{14}
s_1	1				1	1	1	1						
s_2	1	1	1						1			1		
s_3		1	1	1					1	1			1	
s_4			1	1						1	1			1

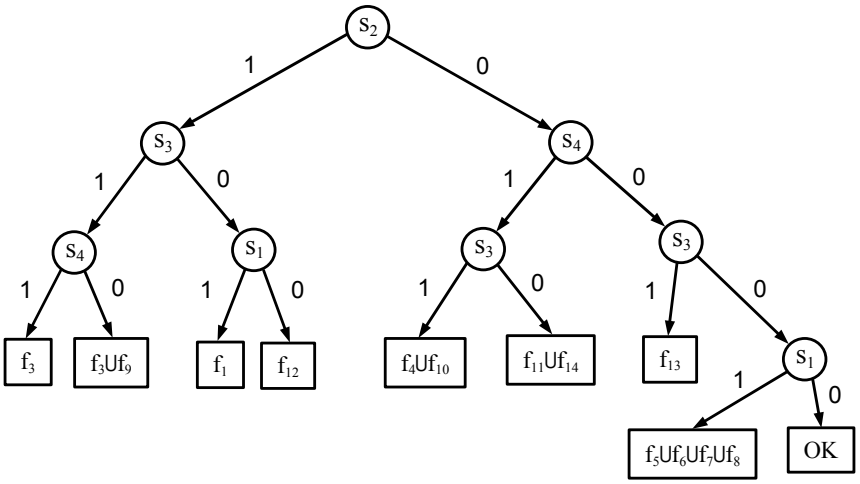
Binarne drzewa diagnostyczne

Binarne drzewa diagnostyczne stanowią dogodną formę reprezentacji sposobu wnioskowania diagnostycznego. Budowa drzewa diagnostycznego jest analogiczna do tej, która stosowana jest do organizowania danych w systemach komputerowych. W binarnym drzewie diagnostycznym *węzły* odpowiadają sygnałom diagnostycznym (testom). Z każdego *węzła* wychodzą dwie gałęzie odpowiadające dwóm wartościom sygnału diagnostycznego, tj. pozytywnemu i negatywnemu. Korzeniem drzewa jest węzeł do którego przypisany jest sygnał, którego wartość analizowana jest jako pierwsza. *Liście* drzewa odpowiadają diagnozom. Na rysunku 3.2 przedstawiono przykład binarnego drzewa diagnostycznego.

Reguły diagnostyczne

Reguły diagnostyczne, podobnie jak i binarne drzewa diagnostyczne, są wyznaczane na podstawie binarnej macierzy diagnostycznej. Do opisu związku pomiędzy uszkodzeniami a wartościami sygnałów diagnostycznych można użyć następujących reguł:

$$\textbf{If } (s_1 = 0) \wedge \ \ldots \ \wedge (s_j = 1) \wedge \ \ldots \ \wedge (s_J = 1) \textbf{ then } f_k, \tag{3.11}$$



Rys. 3.2: Przykładowe drzewo diagnostyczne [46]

$$\textbf{If } (s_j = 1) \textbf{ then } f_a \vee \ \ldots \ \vee f_k \vee f_n, \tag{3.12}$$

gdzie: $\wedge$ – operator koniunkcji, $\vee$ – operator alternatywy.

Reguła typu (3.11) określa jedną kolumnę binarnej macierzy diagnostycznej, natomiast reguła (3.12) jeden wiersz tej macierzy.

3.3.2. Modele odwzorowujące wielowartościowe sygnały diagnostyczne

Wielowartościowe sygnały diagnostyczne $s_j \in V_j$ powstają w wyniku kwantyzacji wartości residuów lub cech sygnałów. Mogą być także wynikiem kontroli ograniczeń zmiennych procesowych przy zastosowaniu kilku wartości granicznych. Każdemu sygnałowi diagnostycznemu może odpowiadać inny podzbiór wartości V_j.

Podobnie jak w przypadku binarnych sygnałów diagnostycznych również dla sygnałów wielowartościowych można stosować te same modele do lokalizacji uszkodzeń tj. drzewa diagnostyczne, reguły itd., przy czym podstawowym modelem opisującym odwzorowanie *wielowartościowe sygnały diagnostyczne* $\rightarrow$ *uszkodzenia* jest system informacyjny. System informacyjny stanowi tu podstawę do budowania wielowartościowych drzew diagnostycznych oraz reguł diagnostycznych.

System informacyjny

System informacyjny został szczegółowo omówiony w punkcie 6.1.1 niniejszej pracy. Dla potrzeb lokalizacji uszkodzeń jako system lokalizacji uszkodzeń (ang. *Fault Isolation System*, FIS) [46] został zaadaptowany system informacyjny przybliżony [46], tj. system informacyjny [71], w którym wartość atrybutu dla danego obiektu nie jest dokładnie znana. W systemie FIS zbiór obiektów jest tożsamy ze zbiorem uszkodzeń

$X \equiv F = \{f_k : \ k = 1, 2, \ldots, K\}$, zbiór atrybutów stanowią sygnały diagnostyczne $A \equiv S = \{s_j : \ j = 1, 2, \ldots, J\}$, a zbiór wartości atrybutów $V_s = \bigcup V_j$ jest sumą zbiorów wartości sygnałów diagnostycznych. Na zbiorze kartezjańskim $F \times S$ jest określona funkcja, która przyporządkowuje parze *uszkodzenie – sygnał diagnostyczny* wartości tego sygnału występujące przy danym uszkodzeniu [46]:

$$r(f_k, s_j) = V_{kj} = \{v_{ji} \in V_j \subset V_j\}. \tag{3.13}$$

System FIS jest zdefiniowany jako następująca czwórka:

$$\text{FIS} = (F, S, V_s, r). \tag{3.14}$$

System FIS jest tablicą określającą wzorcowe wartości sygnałów diagnostycznych dla poszczególnych uszkodzeń. Stanowi on rozszerzenie binarnej macierzy diagnostycznej o następujące elementy [43]:

- dla każdego sygnału diagnostycznego może istnieć indywidualny zbiór jego wartości;
- zbiór V_j wartości j-tego sygnału diagnostycznego może być wieloelementowy,
- dowolny element systemu FIS (komórka w tablicy 3.2) zawierać może jedną wartość sygnału diagnostycznego lub ich podzbiór.

W tablicy 3.2 przedstawiono przykładowy system FIS.

Tab. 3.2: Przykład systemu lokalizacji uszkodzeń [43]

S/F	f_1	f_2	f_3	f_4	f_5	f_6	V_j
s_1	1	0	1	0	0	1	$\{0, 1\}$
s_2	0	-1	0	$+1$	-1	0	$\{0, +1, -1\}$
s_3	-1	$+1$	$+1, -1$	0	$+1$	$+1$	$\{0, +1, -1\}$
s_4	0	$1, 2$	$0, 1$	0	$1, 2$	$1, 1$	$\{0, 1, 2\}$
s_5	$+1$	0	$+1$	$+1$	0	$+1, -1$	$\{0, +1, -1\}$

3.3.3. Modele odwzorowujące ciągłe sygnały diagnostyczne

Do sygnałów niosących informacje diagnostyczne należą przede wszystkim residua generowane na podstawie modelu lub modeli obiektu a także wartości cech sygnałów oraz zmienne procesowe. W przestrzeni tych sygnałów występują obszary wzorcowe, które odpowiadają poszczególnym uszkodzeniom lub stanom obiektu. Modele opisujące tego typu zależności należą do grupy modeli klasyfikujących, które zazwyczaj są pozyskiwane za pomocą metod uczenia maszynowego, rozpoznawania obrazów, modelowania neuronowego i neuronowo-rozmytego itp. Metodom tym jak i ich zastosowaniom w diagnostyce poświęcono wiele publikacji opisujących szczegółowo te metody [43, 44, 46, 73, 74].

3.4. Podsumowanie

W rozdziale przedstawiono podstawowe grupy modeli stosowane w diagnostyce procesów do detekcji i lokalizacji uszkodzeń oraz rozpoznawania stanów obiektu. W skrócie opisano najważniejsze rodzaje modeli procesów wykorzystywanych do generacji residuów oraz modele opisujące związki między sygnałami diagnostycznymi i uszkodzeniami.

Modele do detekcji uszkodzeń wymagają określenia struktury modelu np. na podstawie równań fizycznych, wiedzy eksperckiej (np. modele rozmyte, sieci neuronowe). Do określenia parametrów modelu stosowane są metody identyfikacji (uczenia) na podstawie danych eksperymentalnych.

Modele do lokalizacji uszkodzeń lub rozpoznawania stanów obiektu mogą zostać określone na podstawie: uczenia, modelowania wpływu uszkodzeń na wartości residuów oraz wiedzy eksperckiej. Sposób pozyskania tej wiedzy zależy od specyfiki obiektu diagnozowania. Dla obiektów jednostkowych o dużej złożoności lokalizacja uszkodzeń i rozpoznanie stanów są prowadzone z zastosowaniem modeli bazujących na wiedzy eksperckiej. W przypadku dobrze rozpoznanych obiektów np. turbozespołów budowane są modele fizyczne lub analityczne, za pomocą których pozyskiwane są dane uczące dla stanów z uszkodzeniami. Dla obiektów produkowanych seryjnie możliwe jest w niektórych przypadkach zebranie danych pomiarowych w stanach z uszkodzeniami poprzez prowadzenie czynnych eksperymentów diagnostycznych.

Różnorodność modeli stosowanych w diagnostyce procesów odzwierciedla różny stopień wiedzy o obiekcie diagnozowania oraz różne sposoby pozyskiwania tej wiedzy.

Rozdział 4

Metody odkrywania wiedzy w bazach danych

Odkrywanie wiedzy w bazach danych (ang. *Knowledge Discovery in Databases*, KDD) jest stosunkowo nową dziedziną inżynierii wiedzy. Początki KDD sięgają lat 90. ubiegłego wieku. Inspiracją dla rozwoju tej dziedziny stał się gwałtowny rozwój technologii pozwalających gromadzić duże zbiory danych, których rozmiary aktualnie osiągają wartości rzędu kilku terabajtów. Stosowane dotychczas tradycyjne metody analizy danych (np. metody wnioskowania statystycznego) [45, 70, 93] okazały się niewystarczające w przypadku analizy tak dużych zbiorów danych. Zaistniała więc potrzeba opracowania nowych metod, za pomocą których ukrytą w danych wiedzę będzie można wydobyć (odkryć). Metody te stanowią połączenie metod stosowanych w statystyce matematycznej, identyfikacji systemów, uczeniu maszynowym oraz wizualizacji danych, przy czym główny nacisk położony jest na:

- szybkość przetwarzania danych,
- ilość przetwarzanych danych,
- możliwość przetwarzania danych, w których występują niepoprawne i brakujące wartości.

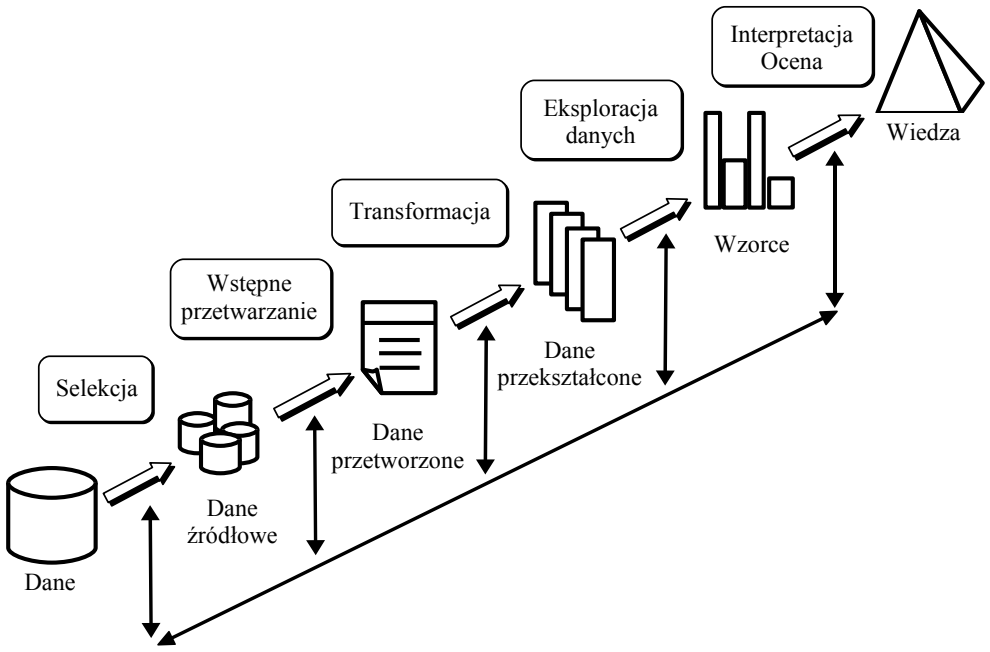
Jedni z twórców tej dziedziny (Frawley, Piatetsky-Shapiro i Matheus) sformułowali w 1991 r. następującą definicję KDD [28]:

> *Odkrywanie wiedzy w bazach danych* jest nietrywialnym procesem identyfikowania ważnych, nowatorskich, potencjalnie użytecznych i zrozumiałych wzorców w danych.

4.1. Proces odkrywania wiedzy w bazach danych

Odkrywanie wiedzy w bazach danych ma charakter procesu o wielu wyraźnie wyodrębnionych stadiach, który schematycznie został przedstawiony na rysunku 4.1.

W większości przypadków proces KDD ma charakter interaktywny i iteracyjny. Wymaga to od osoby prowadzącej analizy wielu umiejętności oraz podejmowania różnych decyzji [28]. Pogląd ten nie jest jednak powszechny. Inni autorzy (np. J. Żytkow



Rys. 4.1: Stadia procesu odkrywania wiedzy w bazach danych [28]

i R. Zembowicz [100]) zwracają uwagę na to, by proces KDD przebiegał autonomicznie, bez konieczności interweniowania użytkownika.

Na proces KDD składają się następujące etapy [28, 29]:

1. Utworzenie zbioru danych źródłowych, dla którego zostanie przeprowadzone odkrywanie wiedzy; w tym celu można wyselekcjonować podzbiór rekordów i/lub podzbiór atrybutów (zmiennych).
2. Wstępne przetwarzanie danych, polegające na usunięciu szumu, usunięciu danych o wartościach znacząco odbiegających od innych (ang. *outliers*), zgromadzenie informacji dotyczących wyboru odpowiedniego modelu błędu/szumu w danych, przyjęcie sposobu postępowania w odniesieniu do brakujących danych.
3. Redukcja i projekcja danych, polegająca na selekcji takich atrybutów (źródłowych lub przekształconych), których wartości będą w odpowiedni sposób reprezentowały dane. Stadium to obejmuje także redukcję ilości informacji polegającą na zmniejszeniu liczby rozróżnianych wartości poszczególnych zmiennych. Celem jest znalezienie reprezentacji danych właściwej w odniesieniu do celu danego zadania KDD.
4. Wybór zadania eksploracji danych (ang. *Data Mining*, DM), polegający na określeniu wyniku prowadzonego procesu: klasyfikacja, regresja, grupowanie, poszukiwanie zależności jakościowych, poszukiwanie zależności funkcyjnych i in.
5. Wybór metody eksploracji danych, która zostanie zastosowana do poszukiwania wzorców w danych. Stadium to obejmuje wybór określonej klasy modeli opisujących dane. Przy wyborze metod eksploracji danych należy uwzględnić wymagania

i oczekiwania końcowego użytkownika.

6. Właściwa eksploracja danych (Data Mining), której istotą jest poszukiwanie wzorców lub regularności [100] w zbiorze/bazie danych, odpowiadających wybranemu sposobowi reprezentacji. Na uzyskane w tym stadium wyniki istotny wpływ mają poprzednie stadia procesu KDD.
7. Zinterpretowanie i ocena odkrytych wzorców, regularności i zależności; w wyniku tego postępowania może nastąpić powrót do wcześniejszych kroków $1 \div 6$ procesu, który ma charakter iteracyjny.
8. Połączenie odkrytej wiedzy, co może być zrealizowane w różny sposób, począwszy od właściwego udokumentowania odkrytej wiedzy, a skończywszy na dołączeniu nowej porcji wiedzy do (być może już istniejącej) bazy wiedzy jakiegoś inteligentnego systemu informacyjnego. Należy podkreślić, że dopiero pomyślne zakończenie tego stadium pozwala uznać, że wyniki procesu KDD zostały poddane walidacji (przez końcowego użytkownika).

W wielu publikacjach eksploracja danych utożsamiana jest z odkrywaniem wiedzy w bazach danych. Takie podejście jest jednak niepoprawne, ponieważ jak pokazano na rysunku 4.1, eksploracja danych jest jednym z etapów procesu odkrywania wiedzy. Etapy procesu KDD poprzedzające właściwą eksplorację danych oraz odpowiednia interpretacja odkrytej wiedzy mają istotny wpływ na końcowy rezultat. Stadia te wymagają dużej uwagi i staranności co powoduje, że pochłaniają one znaczną część czasu przeznaczonego na realizację kompletnego procesu.

W dalszej części rozdziału omówiono istotne z punktu widzenia pracy elementy procesu odkrywania wiedzy w bazach danych.

4.2. Zadania odkrywania wiedzy

Jednym ze sposobów usystematyzowania metod i środków stosowanych w ramach eksploracji danych jest kryterium celu ich zastosowania. W literaturze przedmiotu [28, 35, 85] wyróżnia się dwa zasadnicze zadania eksploracji danych: analizę predykcyjną oraz analizę opisową. Analiza predykcyjna dotyczy przewidywania nieznanych lub przyszłych wartości pewnej określonej zmiennej na podstawie pozostałych zmiennych występujących w rozpatrywanym zbiorze danych. Z kolei analiza opisowa dotyczy wyszukiwania wzorców opisujących i podsumowujących dane. Postać wyszukanych wzorców pozwala na dogodną interpretację odkrytych regularności.

Na zasadnicze zadania eksploracji danych składają się zadania szczegółowe. W szczególności są to [35, 85]:

1. analiza predykcyjna,
   - klasyfikacja,
   - aproksymacja,
2. analiza opisowa:

- analiza skupień (grupowanie),
- odkrywanie wzorców i reguł asocjacyjnych.

W dalszej kolejności zostaną omówione niektóre z wymienionych wyżej zadań.

4.2.1. Klasyfikacja

W zadaniu klasyfikacji dane wejściowe stanowią zbiór rekordów nazywany czasami zbiorem instancji lub zbiorem przykładów. Każdy rekord jest charakteryzowany za pomocą pary $(\mathbf{x}, C)$, gdzie $\mathbf{x}$ jest wektorem, którego liczba elementów wynika z liczby rozpatrywanych atrybutów *wejściowych*, natomiast C jest atrybutem *decyzyjnym*. Za pomocą atrybutu C określana jest przynależność każdego z przykładów $\mathbf{x}$ do jednej z predefiniowanych klas (kategorii). Zbiorem wartości atrybutu C jest skończony zbiór elementów będących *etykietami* (opis symboliczny) klas. Z kolei elementy składowe wektora $\mathbf{x}$ mogą przyjmować wartości rzeczywiste lub dyskretne (numeryczne lub symboliczne).

Zadanie identyfikacji klasyfikatora polega na wyznaczeniu funkcji (odwzorowania) $f(\,)$, za pomocą której każdy z przykładów $\mathbf{x}$ jest przypisywany do jednej z predefiniowanych klas C. Funkcja $f(\,)$ jest często określana mianem *klasyfikatora* lub *modelu klasyfikacji*.

Funkcją klasyfikującą $f(\,)$ można się posłużyć jako narzędziem wyjaśniającym do rozróżnienia przykładów należących do różnych klas. Kolejnym ważnym zastosowaniem *modelu klasyfikującego* jest możliwość przypisania przykładów do określonej klasy, w przypadku gdy nie są znane kategorie do jakich te przykłady należą.

Do metod stosowanych w zadaniach klasyfikacji danych oraz identyfikacji klasyfikatora zalicza się m.in.:

- metody indukcji drzew decyzyjnych,
- metody indukcji reguł,
- klasyfikację bayesowską,
- klasyfikację z zastosowaniem sieci neuronowych,
- metodę wektorów wspomagających.

4.2.2. Aproksymacja funkcji

Podobnie jak w zadaniu identyfikacji klasyfikatora, w zadaniu aproksymacji funkcji dane wejściowe stanowią zbiór przykładów, gdzie każdy przykład jest charakteryzowany za pomocą pary (x_i, y). Jednakże w tym przypadku wektor x_i, $i = 1, \ldots, p$ składa się z elementów, których wartości są przede wszystkim liczbami rzeczywistymi. Atrybuty tworzące wektor x_i są nazywane *zmiennymi niezależnymi*. Z kolei wartości jakie przyjmuje atrybut y mogą być wyłącznie liczbami rzeczywistymi. W zadaniach aproksymacji funkcji i odkrywania równań atrybut y określany jest jako *zmienna zależna*.

Zadanie aproksymacji polega na wyznaczeniu zależności $y = f(x_i; \theta)$ opisującej związek funkcyjny występujący pomiędzy *zmienną zależną* y a *zmiennymi niezależnymi* x_i.

Przy czym $f(\,)$ jest strukturą wzorca (modelu) a θ jest zbiorem parametrów tej struktury, których wartości wyznaczane są poprzez minimalizację pewnej ustalonej funkcji oceny (np. *kryterium najmniejszych kwadratów*).

W zadaniu aproksymacji, struktura funkcji $f(\,)$ jest przyjmowana *a priori*, natomiast wyznaczane są jedynie wartości parametrów charakterystycznych dla tej struktury. Struktura wzorca jest zwykle określona poprzez zastosowaną metodę aproksymacji funkcji. W przypadku regresji [45, 70, 93] strukturą będzie sparametryzowane wyrażenie algebraiczne, dla metod modelowania neuronowego będzie to sieć neuronów określonego typu [24, 37, 42, 83]. Oprócz wymienionych metod, w aproksymacji funkcji zastosowanie znajduje również metoda wektorów wspomagających [23, 37, 78, 91]. Metody aproksymacji funkcji znajdują zastosowanie w tych analizach, w których duże znaczenie ma dokładność identyfikowanej zależności funkcyjnej. W przypadku gdy ważniejsza jest łatwość interpretacji opisu poszukiwanej zależności funkcyjnej należy rozpatrzeć zadanie odkrywania równań.

4.2.3. Analiza skupień

Analiza skupień polega na dekompozycji i podziale zbioru danych na grupy w taki sposób, by elementy w tej samej grupie były do siebie podobne, a jak najbardziej odmienne od elementów z pozostałych grup [35]. Utworzonym grupom można w dalszej kolejności przypisać pewien opis (*etykietę klasy*) dzięki któremu będzie możliwe klasyfikowanie nowych przykładów.

W przeciwieństwie do poprzednio omówionych zadań eksploracji danych, w analizie skupień zbiór danych wejściowych nie zawiera atrybutu *decyzyjnego* lub *zmiennej zależnej*.

W analizie skupień można wyróżnić następujące rodzaje algorytmów grupowania [35]:

- oparte na podziale na określoną liczbę grup,
- oparte na hierarchii struktury skupień,
- oparte na modelu probabilistycznym.

Zadaniem grupowania opartego na podziale jest dokonanie dekompozycji zbioru danych na k rozłącznych podzbiorów elementów w taki sposób, aby elementy każdej z grup były jak najbardziej jednorodne. Jednorodność jest identyfikowana przez zastosowaną *funkcję oceny*, którą jest zazwyczaj *odległość* (np. odległość euklidesowa [16, 17]) między każdym przykładem x_i przypisanym do określonej grupy a „*środkiem ciężkości*" tej grupy.

Istotą grupowania hierarchicznego jest stopniowe łączenie przykładów w grupy lub stopniowe dzielenie nadskupień. Prowadzi to do podziału metod grupowania hierarchicznego na metody *aglomeracyjne* (łączące) i *rozdzielające* [35, 85]. Najczęściej stosowane jest grupowanie aglomeracyjne oparte na miarach odległości między skupieniami. Grupowanie to rozpoczyna się od utworzenia grup składających się z pojedynczych przykładów, które w kolejnych etapach są łączone parami w coraz to bardziej liczne grupy aż

do momentu gdy powstanie jedno skupienie zawierające wszystkie przykłady. Wynikiem grupowania hierarchicznego jest *dendrogram* kształtem przypominający drzewo.

Do najczęściej stosowanych algorytmów grupowania należą: algorytm *K–średnich*, algorytm aglomeracyjnego grupowania hierarchicznego i bazujący na nim algorytm *COBWEB* [21] oraz algorytm *DBSCAN*, w którym podział na grupy wynika z rozkładu prawdopodobieństwa zbioru danych wejściowych.

4.3. Dane w procesie odkrywania wiedzy

Dane stanowią symboliczną reprezentację własności i właściwości obiektów będących przedmiotem zainteresowania określonej dziedziny pojęciowej [35]. Np. w diagnostyce maszyn są wartościami cech sygnałów, za pomocą których są obserwowane własności i właściwości obiektu badań [60]. Dane gromadzi się w bazach danych [5, 90] w celu prowadzenia różnych analiz m.in. za pomocą metod odkrywania wiedzy w bazach danych.

4.3.1. Właściwości danych

Dane gromadzone w bazach danych mogą być niekompletne, błędne, obciążone niepewnością, niedokładne [61]. W szczególności [60, 61]:

- *niekompletność* danych może być spowodowana brakiem uwzględnienia ważnych cech opisujących obiekt lub brakiem wartości tych cech,
- *niepewność* danych wynika z wpływu szumów na wejścia i wyjścia obserwowanego obiektu,
- *niedokładność* danych spowodowana jest przez własności systemów pomiarowych oraz technik określania wartości cech,
- *dane błędne* mogą zawierać błędy dotyczące jednej wartości danej cechy, lub mogą być sprzeczne.

4.3.2. Reprezentacja danych

W metodach odkrywania wiedzy w bazach danych najczęściej stosowanym sposobem reprezentacji danych jest *model atrybutowy*. Zbiór danych charakteryzujący obiekt zainteresowania jest wówczas reprezentowany w postaci macierzy [60, 61]:

$$\begin{bmatrix} a_{11} & \cdots & a_{1m} & d_{11} & \cdots & d_{1n} \\ & \cdots & & & \cdots & \\ a_{N1} & \cdots & a_{Nm} & d_{N1} & \cdots & d_{Nn} \end{bmatrix} \tag{4.1}$$

gdzie: $a_{i,j}$, $i = 1, \ldots, N;\ j = 1, \ldots, m$ są wartościami m atrybutów warunku, $d_{i,j}$, $i = 1, \ldots, N;\ j = 1, \ldots, n$ są wartościami n atrybutów decyzyjnych. W przypadku gdy $n \neq 0$ to zadanie odkrywania wiedzy dotyczy predykcji, natomiast gdy $n = 0$ prowadzona może być analiza opisowa.

4.4. Metody wstępnego przetwarzania danych

Wstępne przetwarzanie obejmuje wiele zabiegów, których celem jest przygotowanie zbioru danych do etapu eksploracji. Odpowiednio przygotowane dane są kluczem do pozyskania wiarygodnej wiedzy. Szacuje się, że etap przygotowania danych zajmuje od 70% do 80% czasu związanego z procesem odkrywania wiedzy.

Do zadań wstępnego przetwarzania danych m.in. zalicza się:

- czyszczenie danych,
- transformację atrybutów,
- redukcję informacji,
- selekcję atrybutów relewantnych.

4.4.1. Czyszczenie danych

Czyszczenie danych jest zadaniem związanym z usunięciem nadmiernego szumu zawartego w danych. Zadanie to obejmuje również usunięcie wartości znacząco odbiegających od wartości pozostałych danych oraz uzupełnienie brakujących wartości. Zastosowanie konkretnej metody usuwania szumu zależy od typu przetwarzanych danych. Przykładowo dla atrybutów ciągłych dobrym rozwiązaniem jest zastosowanie metod wygładzania i filtracji [57]. Z kolei dla danych jakościowych będą to metody grupowania.

W celu uzupełnienia brakujących wartości atrybutów można rozważyć następujące podejścia [21]:

- ignorowanie przykładów z brakującymi wartościami,
- traktowanie brakujących wartości jako specjalnej dodatkowej wartości,
- uzupełnienie brakującej wartości:
  - najczęściej występującą wartością atrybutu a_i w zbiorze trenującym,
  - wartością wyznaczoną na bazie modelu zbudowanego na podstawie znanych wartości innych atrybutów.

4.4.2. Dyskretyzacja atrybutów

Dyskretyzacja polega na przekształceniu atrybutów ciągłych w atrybuty o wartościach dyskretnych, które odpowiadają pewnym przedziałom wartości atrybutów oryginalnych. Dzięki przekształceniu atrybutów ciągłych w atrybuty porządkowe możliwe jest zastosowanie metod odkrywania wiedzy przewidzianych do analizy danych jakościowych.

Zasadniczym problemem w dyskretyzacji atrybutów jest ustalenie wartości i liczby progów przedziałów dyskretyzacji. Opracowano wiele metod rozwiązujących ten problem. Obszerne badania w tym zakresie przeprowadził K. Ciupke [22], dokonując bardzo szczegółowego porównania przydatności następujących metod dyskretyzacji [22]:

1. Metody *jednoatrybutowe*:

(a) Metoda równej szerokości przedziałów;

(b) Metoda równej częstości w przedziałach dyskretyzacji;

(c) Metoda *ChiMerge*: Istotą tej metody jest wstępny podział dziedziny atrybutu na podzbiory, a następnie wybrane podprzedziały łączy się do momentu, w którym zostanie spełnione kryterium zatrzymania określone poprzez minimalną liczność w danych podprzedziałach oraz progową wartość statystyki χ^2;

(d) Metoda minimum entropii (obliczanej dla zbioru przykładów po jego podziale na podzbiory);

(e) Metoda grupowania jednoatrybutowego: Wartości progowe określane są po przeprowadzeniu grupowania wartości atrybutów z zastosowaniem różnych metod, jak np. *k najbliższych sąsiadów*.

2. Metody *wieloatrybutowe*:

(a) Metoda grupowania wieloatrybutowego: Istota metody jest podobna jak w przypadku opisanej powyżej metody grupowania jednoatrybutowego;

(b) Metoda heurystyczna (bazująca na częściowym zastosowaniu zbiorów przybliżonych).

4.4.3. Selekcja atrybutów relewantnych

Wielkość współczesnych baz danych nie tylko zależy od liczby rekordów lecz również od liczby atrybutów. Liczba ta w niektórych przypadkach może sięgać setek a nawet tysięcy. Analiza takich zbiorów danych nastręcza wiele trudności. W związku z tym jednym z etapów procesu odkrywania wiedzy jest etap selekcji atrybutów relewantnych. Polega on na wyborze takiego podzbioru atrybutów, który pozwoli na bardziej efektywne prowadzenie analizy z uwzględnieniem celu tych analiz oraz przyjętego układu kryteriów. Powody, dla których warto i należy stosować selekcję atrybutów relewantnych to m.in.:

- Poprawa wydajności stosowanych algorytmów odkrywania wiedzy.
- Dokładność predykcji pozyskanych modeli na podstawie mniejszego zbioru atrybutów może w wielu wypadkach być porównywalna z modelami pozyskanymi dla zbioru wszystkich atrybutów.
- Zastosowanie mniejszej liczby atrybutów może prowadzić do pozyskiwania *prostszych* modeli, co z kolei ułatwia interpretację wiedzy reprezentowanej za pomocą tych modeli.

W pracach [40, 41] rozpatrywano dwa podejścia do problemu selekcji atrybutów relewantnych. Pierwsze z nich, zwane *filtracją cech*, stanowi próbę oceny atrybutów poprzez pryzmat wartości przyjmowanych przez te atrybuty. Metody filtracji cech są zazwyczaj stosowane na etapie przygotowywania danych. Główną wadą tego podejścia jest pominięcie wpływu wybranego podzbioru atrybutów na osiągi algorytmu indukcji wiedzy zastosowanego w późniejszych etapach procesu odkrywania wiedzy.

Wśród metod selekcji atrybutów zaliczanych do filtracji cech należy wymienić metody [22]:

- bazujące na miarach prawdopodobieństwa (analiza macierzy kowariancji/korelacji),
- oparte na teorii zbiorów przybliżonych,
- oparte na minimalizacji entropii zbioru przykładów.

Ciekawym przykładem rozwiązania zadania ograniczenia liczności zbioru rozpatrywanych atrybutów diagnostycznych jest metoda zaproponowana przez W. Cholewę [16]. Polega ona na wyznaczeniu osi głównych przestrzeni wartości atrybutów, a następnie na określeniu ograniczonej przestrzeni wartości atrybutów głównych.

Drugie podejście do problemu selekcji atrybutów polega na zastosowaniu algorytmu indukcji wiedzy (ang. *wrapper approach*). Algorytm indukcji wiedzy traktowany jest jako *czarna skrzynka*, której na wejście podaje się określony podzbiór atrybutów aby na wyjściu otrzymać model. Wygenerowany model stanowi podstawę oceny wejściowego zestawu atrybutów. Zaprezentowane podejście wymaga określenia trzech elementów. Pierwszym istotnym elementem jest algorytm indukcji wiedzy, który należy dobrać ze względu na rodzaj rozpatrywanego zadania odkrywania wiedzy. Drugim równie ważnym elementem jest metoda przeszukiwania *przestrzeni stanów*, gdzie stany wyznaczające tę przestrzeń określone są przez wszystkie podzbiory jakie można utworzyć dla danego zbioru atrybutów. W tym celu można m.in. zastosowa,ć [35, 40, 41, 66]:

- strategię przeszukiwania wszerz,
- strategię przeszukiwania w głąb,
- strategię *zachłanną* (ang. *hill–climbing*),
- strategię *wpierw najlepszy* (ang. *best–first*),
- *symulowane wyżarzanie*,
- przeszukiwanie genetyczne.

Ostatnim elementem tej metody jest funkcja oceny, którą może być jedna z miar dokładności predykcji np.: korelacja, odchylenie średniokwadratowe, liczba poprawnie klasyfikowanych przykładów itp. Jako funkcję oceny można również zastosować jedno z kryteriów informacyjnych np. AIC (ang. *Akaike Information Criterion*), BIC (ang. *Bayesian Information Criterion*), MDL (ang. *Minimum Description Length*) itp.

4.5. Odkrywanie wiedzy

Automatyczne odkrywanie wiedzy różni się w istotny sposób od *uczenia maszynowego*. Główna różnica polega na tym, że metodami odkryć wiedzy pozyskiwana jest *nowa porcja wiedzy*, natomiast za pomocą metod *uczenia maszynowego* pozyskiwana jest wiedza, która została już wcześniej odkryta [101]. Celem procesu uczenia maszynowego jest zwięzły i przejrzysty sposób reprezentacji odkrytej wcześniej wiedzy na potrzeby bazy wiedzy systemu doradczego.

Celem *odkrywania wiedzy* jest *identyfikacja regularności* jakie mogą występować w zbiorze danych [28]. *Regularność* określana jest poprzez pewien *wzorzec* (ang. *pattern*) oraz zakres, w którym ten wzorzec występuje [100]. Przykładami wzorców są *tablice kontyngencji*, *równania*, *reguły asocjacyjne* i *równoważności logiczne*. Zakres występowania regularności jest ustalony w wyniku procesu dekompozycji bazy danych na fragmenty, w których mogą występować istotne wzorce. Dekompozycja taka może polegać na [21,90] *rzutowaniu* (pomijanie pewnych atrybutów) i/lub *selekcji* rekordów, dla których atrybuty przyjmują określone wartości.

4.5.1. Elementy systemu odkrywania wiedzy

System dokonywania odkryć w bazach danych winien działać w sposób autonomiczny. System ten dobiera właściwy sposób reprezentacji wiedzy oraz steruje procesem poszukiwania.

Architektura systemu odkrywania wiedzy jest dwuwarstwowa. Pierwsza warstwa jest odpowiedzialna za dekompozycję bazy danych na fragmenty, w których mogą występować interesujące wzorce. Dekompozycja ta może polegać na wcześniej wspomnianym rzutowaniu i/lub selekcji rekordów, dla których wartości pewnych atrybutów znajdują się w określonym zakresie.

Druga warstwa jest związana z poszukiwaniem określonego typu wzorców w podzbiorze danych wyznaczonych przez pierwszą warstwę systemu. Etap ten wymaga określenia algorytmu, za pomocą którego będzie odkrywany wzorzec. W szczególności w zastosowanym algorytmie powinny zostać określone następujące elementy [35]:

1. *Struktura wzorca*, która jest wyszukiwana w danych (np. tablice kontyngencji, równania itp.).
2. *Funkcja oceny*, za pomocą której określa się jakość wzorców występujących w danych (np. błąd średniokwadratowy, miara istotności Q [100]).
3. *Metoda przeszukiwania* i/lub *optymalizacji* stosowana do przeszukiwania przestrzeni struktur i parametrów. Jej celem jest wyszukanie wzorca, dla którego wartość zastosowanej funkcji oceny jest maksymalna. Jeśli struktura wzorca jest pojedynczą ustaloną strukturą (np. struktura sieci neuronowej), przeszukiwanie jest prowadzone wyłącznie w przestrzeni parametrów.

4.5.2. Tablice kontyngencji

Do identyfikacji regularności występujących w bazach danych w pierwszej kolejności są stosowane *tablice kontyngencji* [100]. Są one znaną ze statystyki formą reprezentacji zależności pomiędzy dwoma zmiennymi losowymi, którymi przy dokonywaniu odkryć są atrybuty. Inne środki reprezentacji wiedzy takie jak reguły mogą być uznane za szczególne przypadki tych tablic.

W przypadku dwóch atrybutów a_1 i a_2 o niewielkiej liczbie wartości dyskretnych tablica kontyngencji jest złożona z wierszy odpowiadających wszystkim wartościom atry-

butu a_1 (ze zbioru A_1) oraz z kolumn odpowiadających wszystkim wartościom atrybutu a_2 (ze zbioru A_2). Element tablicy na przecięciu wiersza odpowiadającego wartości $v_1 \in A_1$ i kolumny odpowiadającej wartości $v_2 \in A_2$ stanowi liczbę rekordów, dla których a_1 ma wartość v_1 i a_2 ma wartość v_2. W przypadku atrybutów ciągłych ich wartości są dyskretyzowane. W najprostszym przypadku, gdy dla obu atrybutów dyskretyzacja dzieli zakres wartości na dwa przedziały (małe i duże wartości), tablica kontyngencji jest czteroelementową tablicą typu 2×2.

Na rysunku 4.2 przedstawiono przykładową tablicę kontyngencji wyznaczoną dla dwóch atrybutów: atrybutu porządkowego $\text{Unb}_1 \in \{90, 203, 360\}$ oraz atrybutu A04D1 $\in \mathbb{R}^+$. Atrybuty te stanowią część bazy danych [62], w której zgromadzono dane wejściowe oraz wyniki eksperymentu numerycznego. Celem tego eksperymentu było określenie relacji diagnostycznych pomiędzy różnymi typami niewyrownoważenia wirnika maszyny a cechami sygnałów obserwowanych w podporach łożyskowych tegoż wirnika [65]. Za pomocą atrybutu Unb_1 opisano wartości niewywag jakie przykładano do jednej z dwóch tarcz wirnika. Z kolei atrybut A04D1 zawiera wartości jednej z wyznaczanych cech. Cechą tą była długość większej półosi trajektorii (w kształcie elipsy), która została zbudowana na podstawie sygnałów mierzonych w dwóch prostopadłych do siebie kierunkach.

A04D1			
3.0-3.5	0	0	12
2.5-3.0	0	8	**32**
2.0-2.5	12	8	**64**
1.5-2.0	4	**40**	12
1.0-1.5	**40**	**64**	0
0.5-1.0	**64**	0	0
	90	203	360
			Unb_1

Rys. 4.2: Przykładowa tablica kontyngencji [94]

Występowanie istotnej zależności w tablicy kontyngencji jest identyfikowane poprzez prawdopodobieństwo Q zdarzenia, że rozpatrywana zależność ma charakter statystycznej fluktuacji atrybutów o rozkładzie losowym. Im mniejsze to prawdopodobieństwo, tym bardziej istotna jest wykryta zależność [100]. Do wyznaczenia Q stosowana jest wartość statystyki χ^2:

$$\chi^2 = \sum_{i,j} \frac{(A_{ij} - E_{ij})^2}{E_{ij}}, \tag{4.2}$$

gdzie: A_{ij} są częstościami z próby, zaś E_{ij} są częstościami oczekiwanymi w przypadku prawdziwości hipotezy zerowej o braku jakiejkolwiek zależności pomiędzy obydwoma atrybutami. Rozkład E_{ij} z próby jest wyznaczany w następujący sposób:

$$E_{i,j} = \frac{n_{x_i} \cdot n_{y_j}}{N}, \tag{4.3}$$

gdzie: n_{x_i} – suma liczb rekordów dla i-tej kolumny tablicy kontyngencji, n_{y_j} – suma liczb rekordów dla j-tego wiersza tablicy kontyngencji, N – całkowita liczba rekordów.

Na podstawie badań [100] stwierdzono, że dla wartości $Q < 10^{-5}$ istnieje bardzo małe prawdopodobieństwo tego, że wykryte wzorce mogły powstać w sposób losowy.

Do oceny *mocy predykcji* danej tablicy kontyngencji w sposób niezależny od liczby stopni swobody tej tablicy oraz liczby rekordów mogą być zastosowane następujące miary:

- miara V Cramera [100]:

$$V = \sqrt{\frac{\chi^2}{N \cdot min\left(M_{row} - 1, M_{col} - 1\right)}}, \tag{4.4}$$

  gdzie: M_{row} – liczba wierszy, M_{col} – liczba kolumn w danej tablicy kontyngencji.

- współczynnik kontyngencji C [100]:

$$C = \sqrt{\frac{\chi^2}{\chi^2 + N}}. \tag{4.5}$$

Dla $V > 0.9$ lub $C > 0.9$, zależność reprezentowaną przez daną tablicę kontyngencji można uznać za równoważność [100].

4.5.3. Odkrywanie równań

Tablice kontyngencji, dla których wartość miary V jest dostatecznie duża, są przedmiotem dalszego przeszukiwania w celu znalezienia równań. Równania są poszukiwane jedynie w takim zbiorze danych, w którym zachodzi *relacja funkcjonalności*:

$$D = \{(x_i, y_i) | x_i \in X \wedge i = 1, \ldots, N\}, \tag{4.6}$$

przy czym y jest funkcją x wtedy i tylko wtedy, gdy dla każdego $x_0 \in X$ istnieje tylko jedno y_0, takie że $(x_0, y_0) \in D$.

Równania poszukiwane są zazwyczaj w sposób automatyczny. Automatyzacja procesu odkrywania równań wymaga uwzględnienia w zastosowanej metodzie poszukiwania następujących elementów:

- zbioru reguł heurystycznych służących do automatycznego generowania parametrycznych struktur równań dopasowywanych do danych, oraz
- metody, za pomocą której wyznaczane są wartości parametrów równania w taki sposób aby uzyskać najwiekszą wartość przyjętego kryterium oceny (np. *kryterium najmniejszych kwadratów*).

Wśród wymienionych elementów kluczową rolę stanowią heurystyki do automatycznego generowania parametrycznych struktur równań. Jednym z pierwszych systemów automatycznego odkrywania równań był system odkryć *BACON* [50–53]. W systemie tym zastosowano cztery następujące reguły heurystyczne [10, 21]:

1. *jeśli zmienna y ma wartość v dla odpowiednio dużej liczby przykładów, to należy przyjąć, że y ma stałą wartość v*,

2. *jeśli zmienne x i y są liniowo zależne dla odpowiednio dużej liczby przykładów to należy przyjąć, że ten związek zachodzi zawsze*,
3. *jeśli x rośnie gdy y maleje, a x i y nie są liniowo zależne, to należy utworzyć nową zmienną $t = xy$*,
4. *jeśli x rośnie gdy y rośnie, a x i y nie są liniowo zależne, to należy utworzyć nową zmienną $t = x/y$.*

Za pomocą powyższych heurystyk można odkrywać w efektywny sposób nietrywialne prawa takie jak np. trzecie prawo Kepplera. Pewnym ograniczeniem tych heurystyk jest to, że nie pozwalają na odkrycie związków wielomianowych stopnia 2 lub wyższych (np. $y = ax^2 + b$).

W kolejnych wersjach systemu *BACON* wprowadzono bardziej zaawansowane heurystyki, m.in. heurystykę znajdowanie stałych pochodnych oraz tzw. *rekursję na wyższy poziom opisu* [10,21]. Pierwsza z nich umożliwia znajdowanie zależności wielomianowych pomiędzy parą zmiennych. Z kolei druga pozwala na identyfikację złożonych zależności w przypadku gdy liczba uwzględnianych zmiennych niezależnych jest większa od 2.

4.5.4. Odkrywanie reguł asocjacyjnych

Reguły asocjacyjne reprezentują wiedzę o tym, że pewne wartości niektórych atrybutów zazwyczaj występują łącznie z pewnymi wartościami innych atrybutów. Każda reguła asocjacyjna zawiera dwie listy wartości atrybutów [21]:

- listę wartości *warunkujących*, i
- listę wartości *warunkowanych*.

Regułę ze zbiorem wartości warunkujących $X \subset \mathbb{V}$ i zbiorem wartości warunkowanych $Y \subset \mathbb{V}$ zapisuje się jako:

$$X \to Y, \tag{4.7}$$

przy czym $X \cap Y = \emptyset$.

W celu sprecyzowania interpretacji reguł asocjacyjnych określa się dla nich tzw. *wsparcie* i *wiarygodność*. *Wsparcie* mówi o tym, jak często w rozważanym zbiorze przykładów występuje sytuacja opisana przez regułę, a *wiarygodność* o tym, jak często wystąpienie wartości warunkujących faktycznie pociąga za sobą wystąpienie wartości warunkowanych. Formalne definicje tych metryk są następujące:

- *wsparcie*

$$s_P(X \to Y) = \frac{\mathrm{card}(P_{X \cup Y})}{\mathrm{card}(P)}, \tag{4.8}$$

- *wiarygodność*

$$c_P(X \to Y) = \frac{\mathrm{card}(P_{X \cup Y})}{\mathrm{card}(P_X)}, \tag{4.9}$$

gdzie: $P_{X \cup Y}$ jest zbiorem przykładów „*pokrywanych*" jednocześnie przez wartości warunkujące i warunkowane, P_X jest zbiorem przykładów pokrywanych przez wartości warunkujące, a P jest rozpatrywanym zbiorem przykładów.

Odkrywanie reguł asocjacyjnych na podstawie zbioru danych (zbioru przykładów) jest realizowane w dwóch etapach:

1. wyszukanie wszystkich zestawów wartości atrybutów często występujących w rozpatrywanym zbiorze przykładów trenujących,
2. utworzenie dla każdej pary zestawów wartości atrybutów (z których jeden jest podzbiorem drugiego) odpowiedniej reguły asocjacyjnej.

Pierwszym istotnym problemem w procesie indukcji reguł asocjacyjnych jest wyszukanie wszystkich zbiorów wartości, których wsparcie przekracza zadane minimum. Takie zbiory są nazywane *częstymi zbiorami wartości*. Efektywne znajdowanie częstych zbiorów wartości stanowi główną trudność przy generowaniu reguł asocjacyjnych. Rozwiązanie tego problemu opiera się na obserwacji, że każdy zbiór wartości zawierający się w pewnym częstym zbiorze wartości, także jest częstym zbiorem. Metoda opierająca się na tej obserwacji, rozpoczyna proces wyszukiwania począwszy od częstych zbiorów jednoelementowych, generując w systematyczny sposób ich nadzbiory, każdorazowo zawierające jedną dodatkową wartość, gdyż tylko nadzbiory częstych zbiorów mogą być częstymi zbiorami. Realizacją tej koncepcji jest algorytm *Apriori* [1].

W drugim etapie, który obejmuje tworzenie reguł asocjacyjnych na podstawie znalezionych częstych zbiorów wartości atrybutów, dla dowolnych dwóch częstych zbiorów wartości X i Y przy wymaganym *wsparciu* oraz relacji $X \subset Y$, może zostać utworzona reguła asocjacyjna:

$$X \rightarrow Y - X, \tag{4.10}$$

dla której *wsparcie* i *wiarygodność* na dowolnym zbiorze przykładów P można bezpośrednio określić na podstawie wsparcia zbiorów X i Y na P w następujący sposób:

$$s_P(X \Rightarrow Y - X) = s_P(Y) \tag{4.11}$$

$$c_P(X \Rightarrow Y - X) = \frac{s_P(Y)}{s_P(X)} \tag{4.12}$$

uwzględniając, że $X \cup (Y - X) = Y$.

Reguły asocjacyjne mogą być również uzyskane z tablic kontyngencji, które wykazują występowanie statystycznie istotnej zależności pomiędzy dwoma atrybutami. W takim przypadku tablica kontyngencji umożliwia formułowanie reguły asocjacyjnej z jednym atrybutem warunkującym i z jednym warunkowanym.

4.5.5. Odkrywanie zależności dynamicznych

Bardzo ciekawym kierunkiem w odkrywaniu równań jest odkrywanie równań różniczkowych. Odkrywanie równań fizycznych (równań różniczkowych) jest możliwe wtedy gdy w algorytmie zostanie zastosowany jeden z następujących elementów:

- numeryczne różniczkowanie danych [25, 26],
- numeryczne całkowanie danych [68, 86, 87].

Najbardziej naturalnym podejściem jest zastosowanie różniczkowania [25, 26]. Różniczkowanie można stosować w przypadku baz danych, w których wartości poszczególnych atrybutów są szeregami czasowymi. Zasadniczą wadą różniczkowania jest jego duża wrażliwość na występowanie zakłóceń w danych. Powoduje to konieczność uprzedniego zastosowania metod umożliwiających filtrację tych zakłóceń.

Znacznie mniej wrażliwa na zakłócenia występujące w danych jest operacja całkowania numerycznego. Zastosowanie całkowania sprowadza się w tym przypadku do wyznaczenia współczynników równania różniczkowego 1. rzędu o znanej strukturze w określonym przedziale czasu [68, 86, 87].

Algorytm LAGRANGE

Koncepcja algorytmu *LAGRANGE* [25, 26] bazuje na analizie szeregów czasowych będących wynikami pomiarów określonych wielkości fizycznych estymowanych w regularnych odstępach czasu. Głównym elementem algorytmu *LAGRANGE* jest procedura różniczkowania numerycznego bazująca na równaniu (4.13):

$$\dot{x}_i = \frac{1}{12h}\left(x_{i-2} - 8\,x_{i-1} + 8\,x_{i+1} - x_{i+2}\right). \tag{4.13}$$

Proces odkrywania równań różniczkowych z zastosowaniem algorytmu *LAGRANGE* obejmuje trzy etapy:

1. Wyznaczenie wszystkich pochodnych szeregów czasowych podanych na wejście algorytmu począwszy od pochodnych rzędu 0 a skończywszy na pochodnych rzędu o, który jest jednym z argumentów wejściowych algorytmu. Pochodne obliczane są zgodnie z wyrażeniem 4.13.
2. Druga faza działania algorytmu obejmuje wprowadzenie nowych zmiennych (tzw. *termów*), które zostają utworzone poprzez wielokrotne przemnażanie oryginalnych szeregów czasowych oraz pochodnych utworzonych w poprzednim etapie. O liczbie przeprowadzonych mnożeń decyduje wartość parametru d, liczba uwzględnianych szeregów czasowych oraz liczba wyznaczonych pochodnych.
3. W trzecim kroku, algorytm generuje i testuje równania. Tworzone są podzbiory zawierające kombinacje wszystkich wprowadzonych wcześniej zmiennych, ich pochodnych oraz *termów*. Maksymalna liczebność tak utworzonych zbiorów jest określona przez wartość parametru r. Każdy z utworzonych podzbiorów o liczbie elementów co najwyżej $r + 1$ jest użyty do wygenerowania równania liniowego, gdzie jeden z *termów* jest wyrażony jako liniowa kombinacja pozostałych termów. Parametry równania liniowego są wyznaczane z zastosowaniem metody najmniejszych kwadratów, a jakość równania jest oceniana za pomocą współczynnika korelacji *R*.

Kompletny algorytm *LAGRANGE* został przedstawiony na rysunku 4.3.

Algorytm LAGRAMGE

W systemie *LAGRANGE* równania różniczkowe odkrywane są za pomocą numerycznego różniczkownia szeregów czasowych. Znaną cechą numerycznego różniczkowania da-

argumenty:

S – uporządkowany zbiór szeregów czasowych

o – rząd modelu

d – stopień *zagłębienia* termów

r – liczba niezależnych zmiennych regresyjnych

zwracane:

M – zbiór równań różniczkowych

LAGRANGE (S, o, d, r)
  /* WYZNACZANIE POCHODNYCH */
    $V \leftarrow S$
    forall $v \in S$ **do**
      $v_0 \leftarrow v$
      for $i \leftarrow 1, 2, \ldots, o$ **do**
        $V \leftarrow V \cup \left\{ \frac{d}{dt} v_{i-1} \right\}$
      end
    end
  /* BUDOWANIE NOWYCH ZMIENNYCH (TERMÓW) */
    $V_1 \leftarrow V$
    for $k \leftarrow 2, 3, \ldots, d$ **do**
      $V_k \leftarrow \emptyset$
      forall $(v_i, v_j) \in V_1 \times V_{k-1}$ **do**
        $V_k \leftarrow V_k \cup \{v_i \cdot v_j\}$
      end
      $V \leftarrow V \cup V_k$
    end
  /* GENEROWANIE I TESTOWANIE RÓWNAŃ */
    $M \leftarrow \emptyset$
    for $i \leftarrow 0, 1, \ldots, r$ **do**
      forall $R \in \mathcal{P}(V)$ takich, że $\mathbf{card}(R) = i + 1$ **do**
        Wybierz zmienną zależną $y \in R$
        if $[\mathbf{c}, \sigma_c] =$ **regresja-liniowa** $(y, R \backslash \{y\})$ jest istotne **then**
          $$M \leftarrow M \cup \left\{ y = c_0 + \sum_{x \in R \backslash \{y\}} c_x x \right\}$$
        end
      end
    end
  return M

Rys. 4.3: Algorytm *LAGRANGE* [26]

nych jest brak odporności na występujący w danych szum co skutkuje pojawieniem się dużych zakłóceń w wyznaczonych pochodnych. W takim przypadku identyfikacja zależności w postaci równań różniczkowych jest bardzo utrudniona. Alternatywnym podejściem może być zastąpienie numerycznego różniczkowania opracją numerycznego całkowania, która cechuje się mniejszą wrażliwością na występowanie szumu. Pewnym ograniczeniem związanym z zastosowaniem numerycznego całkowania danych jest to, że pozwala ono jedynie na odkrywanie równań różniczkowych pierwszego stopnia.

Odrębny aspekt stanowi możliwość uwzględnienia wiedzy o obiekcie lub procesie, dla którego identyfikowany jest model. Pomocne w tym zakresie mogą być gramatyki bezkontekstowe. Gramatyki bezkontekstowe są narzędziem bardzo często stosowanym w dziedzinie informatyki. W szczególności są stosowane do definiowania składni języków programowania oraz stanowią podstawę do opracowania kompilatorów tych języków.

Z uwagi na swoje właściwości, gramatyki bezkontekstowe zostały wykorzystane w systemie *LAGRAMGE* [27, 86, 87] jako narzędzie umożliwiające wprowadzenie wiedzy dziedzinowej do procesu odkrywania równań (w tym również równań różniczkowych).

Gramatyki bezkontekstowe są zapisywane jako następująca krotka [27, 86]:

$$G = (N, T, P, S), \tag{4.14}$$

gdzie:

- N – skończony zbiór zmiennych zwanych *nieterminalami* lub kategoriami syntaktycznymi. Każdy element takiego zbioru reprezentuje wyrażenie lub frazę języka opisywanego przez gramatykę.
- T – zbiór symboli pomocniczych (alfabet pomocniczy) zwanych *terminalami*. Zbiory N i T są zbiorami rozłącznymi, tj. zachodzi relacja $N \cap T = \emptyset$.
- P – skończony zbiór relacji zwanych *produkcjami*. Produkcje w zbiorze P są oznaczane jako $A \rightarrow \alpha$, gdzie $A \in N$ i $\alpha \in (N \cup T)$ są nazywane odpowiednio lewą i prawą stroną produkcji, $\rightarrow$ jest metasymbolem.
- S – wyróżniony symbol będący symbolem początkowym $S \in N$.

Na rysunku 4.4 przedstawiono przykładową gramatykę bezkontekstową jaka została zastosowana do odkrywania modelu prostego układu mechanicznego [96].

Wyrażenia języka definiowanego przez gramatykę (4.14) są wyprowadzane z zastosowaniem drzew derywacji (ang. *derivation trees*). Każde drzewo derywacji musi spełniać pewien układ warunków [27, 86].

W przypadku gdy etykietami wszystkich liści drzewa derywacji są elementy zbioru T, staje się ono wyrażeniem reprezentującym strukturę równania, którego parametry wyznaczane są za pomocą metody *Levenberga - Marquardta* [76] lub metody *downhill simplex* [76].

Opisane cechy gramatyk bezkontekstowych powodują, że są one predysponowane jako narzędzie do automatycznego odkrywania równań, a w połączeniu z całkowaniem numerycznym pozwalają odkrywać równania różniczkowe.

$$
\begin{aligned}
N &= \{E, F, G, H, M, v\} \\
T &= \{+, \cdot, const, \texttt{exp}, \texttt{sin}, \texttt{cos}, (,), t, x\} \\
P &= \left\{ \begin{array}{lcl} E & \rightarrow & const * H + const * H \\ H & \rightarrow & M \mid M * F \\ M & \rightarrow & \texttt{exp}(G) \mid \texttt{exp}(G) * V \\ F & \rightarrow & \texttt{sin}(G) \mid \texttt{cos}(G) \\ G & \rightarrow & const * V \end{array} \right\} \\
S &= E
\end{aligned}
$$

Rys. 4.4: Przykład gramatyki bezkontekstowej służącej do odkrywania modelu prostego układu mechanicznego [96]

4.5.6. Odkrywanie zależności jakościowych

Odkrywanie zależności jakościowych jest związane z zaproponowaną przez B. Kuipersa metodą *QSIM* (ang. *Qualitative Simulation*) [48, 49]. Metoda ta bazuje na jakościowych równaniach różniczkowych *QDE* (ang. *Qualitative Differential Equations*) [30, 48, 49], które są budowane z zastosowaniem ograniczeń jakościowych reprezentowanych za pomocą predykatów (Tab. 4.1) oraz na podstawie zadanego układu „*zwykłych*" równań różniczkowych opisujących modelowany system.

Tab. 4.1: Ograniczenia stosowane w modelowaniu jakościowym [26]

Predykat	Znaczenie
$const(v)$	zmienna v jest stała w rozpatrywanym przedziale czasu
$deriv(v_1, v_2)$	v_2 jest pochodną v_1, tj. $v_2(t) = \frac{d}{dt} v_1(t)$
$minus(v_1, v_2)$	$v_1 = -v_2$
$add(v_1, v_2, v_3)$	$v_1 + v_2 = v_3$
$mult(v_1, v_2, v_3)$	$v_1 * v_2 = v_3$
$M^+(v_1, v_2)$	v_1 monotonicznie wzrasta wraz z v_2
$M^-(v_1, v_2)$	v_1 monotonicznie maleje wraz z v_2

Oprócz układu jakościowych równań różniczkowych do jakościowego opisu systemu wymagana jest zamiana reprezentacji dziedziny z ilościowej na jakościową dla każdej z uwzględnianych zmiennych systemowych. Zmienna jakościowa v jest funkcją [31, 48]:

$$v : [a, b] \rightarrow \mathbb{R}^*, \tag{4.15}$$

gdzie: $\mathbb{R}^* = (-\infty, \infty)$ oraz $[a, b] \subseteq \mathbb{R}^*$, przy czym:

- v jest ciągłe na przedziale $[a, b]$,

- v jest różniczkowalne w każdym punkcie przedziału (a, b),
- v posiada skończoną liczbę wartości charakterystycznych (ang. *landmark value*),
- w zbiorze $\mathbb{R}^*$ istnieją granice:

$$\lim_{t \to a^+} v'(t) = v'(a), \tag{4.16}$$

$$\lim_{t \to b^-} v'(t) = v'(b). \tag{4.17}$$

Każda zmienna jakościowa v posiada uporządkowany zbiór punktów granicznych $\{g_1, g_2, \ldots, g_n\}$ wyróżnionych spośród pozostałych wartości dziedziny, który oprócz wartości zmiennej v w każdym z tych punktów musi zawierać również punkty: 0, $v(a)$ i $v(b)$. Zbiór punktów granicznych wyznacza zbiór rozłącznych przedziałów $\{(-\infty, g_1), g_1, (g_1, g_2, g_2, \ldots, g_n, (g_n, \infty))\}$, z którego zmienna jakościowa v przyjmuje wartości.

Dla potrzeb symulacji dynamicznych zależności jakościowych zostało wprowadzone pojęcie stanu jakościowego [48], który dla zbioru punktów granicznych $g_1, g_2, \ldots, g_n$ w danej chwili t ($t \in [a, b]$) jest następującą parą:

$$QS(v, t) = \langle qmag, qdir \rangle, \tag{4.18}$$

gdzie:

$$qmag = \begin{cases} g_i & \text{dla} \quad v(t) = g_i \\ (g_i, g_{i+1}) & \text{dla} \quad v(t) \in (g_i, g_{i+1}), \end{cases} \tag{4.19}$$

$$qdir = \begin{cases} + & \text{dla} \quad v'(t) > 0 \\ 0 & \text{dla} \quad v'(t) = 0 \\ - & \text{dla} \quad v'(t) < 0. \end{cases} \tag{4.20}$$

Sekwencja stanów jakościowych $QS(v, t_0), QS(v, t_0, t_1), QS(v, t_1), \ldots, QS(v, t_{n-1}, t_n), QS(v, t_n)$ określa zachowanie zmiennej jakościowej v w przedziale $[a, b]$. Przeprowadzenie symulacji za pomocą metody *QSIM* wymaga dodatkowo podania reguł definiujących sposób przejścia zmiennej jakościowej v z jednego stanu jakościowego QS do drugiego. Ponieważ dany stan jakościowy może być punktem lub przedziałem, wyróżnione zostały dwa typy przejść [31, 48]: przejście typu **P** z punktu do przedziału oraz przejście typu **I** z przedziału do punktu.

Wszystkie z przedstawionych powyżej elementów pozwalają w końcowym efekcie na przeprowadzenie symulacji jakościowych zgodnie z algorytmem *QSIM* [48, 49, 79]:

1. Ustal stan początkowy systemu dla t_0 lub (t_0, t_1).
2. Dla każdej zmiennej jakościowej określ zbiór wszystkich możliwych przejść typu **P** lub **I** pomiędzy bieżącym a następnym stanem jakościowym.

argumenty:

S – uporządkowany zbiór szeregów czasowych
o – rząd modelu
d – stopień *zagłębienia* termów
r – liczba niezależnych zmiennych regresyjnych

zwracane:

M – zbiór jakościowych równań różniczkowych

QMN (S, o, d, r)
  $V \leftarrow S$
  forall $v \in S$ **do**
    $v_0 \leftarrow v$
    for $i \leftarrow 1, 2, \ldots, o$ **do**
      $V \leftarrow V \cup \left\{\frac{d}{dt} v_{i-1}\right\}$
    end
  end
  $V_1 \leftarrow V$
  for $k \leftarrow 2, 3, \ldots, d$ **do**
    $V_k \leftarrow \emptyset$
    forall $(v_i, v_j) \in V_1 \times V_{k-1}$ **do**
      $e \leftarrow v_i + v_j$
      $f \leftarrow v_i - v_j$
      $g \leftarrow v_i * v_j$
      $h \leftarrow v_i / v_j$
      $V_k \leftarrow V_k \cup \{e, f, g, h\}$
    end
    $V \leftarrow V \cup V_k$
  end
  forall $v \in V$ **do**
    if $TEST(const(v), \epsilon, \delta)$ **then**
      $M \leftarrow M \cup \{const(v)\}$
    end
  end
  forall $(v_i, v_j) \in V \times V$ **do**
    if $TEST(deriv/minus/M^+/M^-(v_i, v_j), \epsilon, \delta)$ **then**
      $M \leftarrow M \cup \{deriv/minus/M^+/M^-(v_i, v_j)\}$
    end
  end
  forall $(v_i, v_j, v_k) \in V \times V \times V$ **do**
    if $TEST(add/mult(v_i, v_j, v_k), \epsilon, \delta)$ **then**
      $M \leftarrow M \cup \{add/mult(v_i, v_j, v_k)\}$
    end
  end
return M

Rys. 4.5: Algorytm *QMN* [26]

3. Usuń zmienne, które nie spełniają poszczególnych równań modelu.
4. Usuń zmienne, które nie spełniają wszystkich równań modelu.
5. Usuń te stany zmiennych jakościowych, które są identyczne ze stanem poprzednim. Pozostały zbiór stanów jakościowych wyznacza nowy stan systemu.
6. Jeżeli zmienne jakościowe nie wykazują zmian lub jedna ze zmiennych osiągnęła granicę przedziału wartości, zakończ symulację i podaj zbiór wszystkich stanów systemu. W przeciwnym przypadku przejdź do 2.

W wyniku symulacji metodą *QSIM* powstaje drzewo stanów opisujące wszystkie możliwe jakościowe *zachowania* systemu. Zastosowanie metody *QSIM* do modelowania działania złożonych obiektów powoduje, że w generowanym drzewie stanów liczba gałęzi rośnie kombinatorycznie. W przypadku wielu gałęzi nie można wykluczyć zasadności ich wystąpienia ze względu na brak danych [60].

Pewne elementy metody *QSIM* mogą być zastosowane do odkrywania jakościowych równań różniczkowych. W szczególności równania tego typu mogą być odkrywane w przypadku gdy dana jest sekwencja stanów jakościowych oraz układ relacji jakościowych reprezentowanych za pomocą predykatów (Tab. 4.1). Idea ta została zrealizowana za pomocą algorytmu *QMN* [26], który został przedstawiony na rysunku 4.5.

Algorytm *QMN* (Rys. 4.5) w pierwszym etapie wyznacza wszystkie pochodne zmiennych ze zbioru S. Podobnie jak w algorytmie *LAGRANGE* (Rys. 4.3) pochodne te są wyznaczane zgodnie z wyrażeniem (4.13). Maksymalny rząd pochodnych określany jest za pomocą parametru o. W drugim etapie generowane są nowe zmienne poprzez zastosowanie podstawowych operacji arytmetycznych względem zmiennych ze zbioru S oraz ich pochodnych. Ostatni etap obejmuje generowanie i testowanie (procedura $TEST()$) wszystkich możliwych relacji jakościowych jakie mogą występować w zbiorze zmiennych rozpatrywanych w algorytmie.

Szczegółowy opis algorytmu *QMN* wraz z przykładami jego zastosowania został przedstawiony w [26].

4.6. Przykłady zastosowań w diagnostyce technicznej

Pomimo krótkiego okresu rozwiajania metod odkrywania wiedzy w bazach danych, w kilku przypadkach znalazły one już zastosowania w obszarze diagnostyki technicznej.

Wśród przykładów zastosowania metod KDD, w pierwszej kolejności należy wymienić system *ENIGMA* [33] autorstwa A. Giordana, L. Saitta oraz F. Bergadano. System ten zastosowano do odkrywania zależności przyczynowo-skutkowych pomiędzy uszkodzeniami występującymi w agregatach elektromechanicznych [33, 60].

Kolejny przykład to badania jakie były prowadzone przez J. Żytkowa oraz W. Moczulskiego [65]. Ich celem było wykrycie regularności oraz zależności funkcyjnych występujących w bazie danych, w której zgromadzono wyniki eksperymentu numerycznego. Eksperyment polegał na wyznaczeniu cech sygnałów obserwowanych w podporach wirnika dla określonego typu niewyrównoważenia przy znanych warunkach działania

układu. Do odkrywania regularności posłużono się metodami analizy danych, w które został wyposażony system *49er* [99,100]. Odkryto m.in. układ 16 równań przyczynowo–skutkowych [63,64] opisujących zależności funkcyjne pomiędzy zmiennymi wejściowymi tj. warunkami działania i stanami układu a zmiennymi wyjściowymi (cechy obserwowanych sygnałów).

Dalsze prace prowadzono w kierunku pozyskania zależności *diagnostycznych* (odwrotnych) [18,19]. Jedno z rozpatrywanych podejść polegało na rozwiązaniu odkrytego układu równań przyczynowo–skutkowych. Zadanie rozwiązano w sposób numeryczny łącząc za pierwszym razem metodę dekompozycji macierzy SVD z metodą *Newtona–Raphsona* [94] a następnie z algorytmami genetycznymi [95].

Bardzo ciekawym przykładem zastosowania metod z pogranicza odkrywania wiedzy, sztucznej inteligencji i identyfikacji systemów jest rozwiązanie zaproponowane przez J. Korbicza i M. Witczaka [98]. Istotą tego rozwiązania jest użycie metod programowania genetycznego do budowy nieliniowych modeli autoregresyjnych klasy NARX. Za pomocą tej metody identyfikowano model ciśnienia oparów w komorze oparowej stanowiącej część stacji wyparnej jednej z polskich cukrowni. Analiza wyników wykazała, że jeden z modeli pozyskanych za pomocą zaproponowanej metody daje dokładniejsze predykcje niż alternatywny model typu ARX, który zbudowano z użyciem klasycznych metod identyfikacji systemów.

4.7. Podsumowanie

Odkrywanie wiedzy w bazach danych jest procesem prowadzącym do odkrycia *nowej* wiedzy na podstawie przykładów, które nie zostały wcześniej sklasyfikowane. Wiedza ta jest pozyskiwana poprzez analizę dużych zbiorów danych gromadzonych w bazach danych. Ilość danych oraz ich niedoskonałość wymusza opracowanie nowych metod analizy danych. Metody te stanowią rozwinięcie metod stosowanych w takich dziedzinach jak statystyka, uczenie maszynowe, sztuczna inteligencja, zarządzanie danymi itp. Z uwagi na wielkość analizowanych zbiorów danych, systemy KDD winny działać w sposób automatyczny i autonomiczny. W tym celu powinny być wyposażone w reguły heurystyczne wspomagające automatyczną analizę danych. Odkryta wiedza powinna być przedstawiona w formie umożliwiającej jej interpretację, np. jako tablica kontyngencji, równanie lub reguła asocjacyjna.

Dokonany przegląd dostępnych autorowi publikacji dotyczących metod odkrywania wiedzy pozwala na stwierdzenie, że metody te mogą być przydatne do odkrywania zależności w bazach danych gromadzonych np. przez systemy SCADA. Zależności te mogą opisywać dynamikę procesów zachodzących w rzeczywistych obiektach. Proces odkrywania zależności dynamicznych powinien dać się zorganizować w sposób automatyczny, bez udziału inżyniera wiedzy. Jeśli udałoby się w wyniku takiego procesu odkrywania wiedzy uzyskać modele w postaci zależności funkcyjnych, to modele te mogłyby prawdopodobnie zostać wykorzystane w diagnostyce tych procesów zgodnie z metodologią diagnostyki wspartej modelowo.

Rozdział 5

Cel i tezy rozprawy

5.1. Cel pracy

Celem prowadzonych badań jest opracowanie metody eksploracji danych procesowych zgromadzonych w bazach danych istniejących w obszarze diagnostyki i eksploatacji maszyn. Za pomocą opracowanej metody będzie można odkrywać występujące w danych związki w postaci zależności funkcyjnych pozwalających opisać dynamikę obserwowanego procesu lub obiektu. Zależności te w dalszej kolejności mogą zostać użyte jako komponenty modeli diagnostycznych takich jak np. modele do detekcji i/lub lokalizacji uszkodzeń. Opracowanie metody wymaga przyjęcia następujących założeń:

1. Metoda stanowi sekwencję logicznie powiązanych ze sobą stadiów, w ramach których stosowane są wybrane metody odkrywania wiedzy w bazach danych i inne metody sztucznej inteligencji, a także opracowane w trakcie realizacji badań metody i algorytmy.
2. Metoda umożliwia eksplorację baz danych w sposób automatyczny lub z ograniczonym udziałem inżyniera wiedzy.
3. Metoda operuje na danych numerycznych, dla których zdefiniowano relację porządkującą, np. za pomocą dedykowanego w tym celu atrybutu.
4. Istnieją i są dostępne dane procesowe będące wynikami pomiarów parametrów procesów zachodzących w obiektach rzeczywiście istniejących lub będące wynikiem numerycznych eksperymentów symulacyjnych, które pozwolą na weryfikację metody.

Przyjęcie ostatniego z wymienionych powyżej założeń jest wynikiem trudności związanych z pozyskaniem baz danych, w których gromadzone są dane będące wynikiem pomiarów i/lub obserwacji rzeczywistych obiektów lub systemów technicznych. Cechy tych baz danych jak np. duża ilość danych, ich niekompletność, niedokładność i/lub niepewność, powodują że stają się one istotnym elementem badań weryfikacyjnych, które pozwalają ocenić opracowaną metodę pod kątem jej przydatności w praktyce.

5.2. Tezy

1. **Odkrywanie zależności dynamicznych w bazach danych może następować z użyciem metod odkrywania zależności statycznych zastosowanych do bazy danych, w której dokonano transformacji atrybutów polegającej na ich projekcji w wielowymiarową przestrzeń regresorów.**
2. **Selekcja atrybutów poprzez łączne zastosowanie algorytmu genetycznego i metody wektorów wspomagających umożliwia odkrywanie modeli dynamicznych o małej złożoności i wystarczającej do zastosowań diagnostycznych dokładności.**

Rozdział 6

Metoda odkrywania modeli procesów

6.1. Reprezentacja danych w systemach odkrywania wiedzy

Stosowanie systemów odkrywania wiedzy wymaga przygotowania opisu rozwiązywanego problemu [47]. Oznacza to przyjęcie odpowiedniego sposobu reprezentacji danych. W zagadnieniach związanych z praktycznym odkrywaniem wiedzy do reprezentacji danych stosowany jest zazwyczaj *model atrybutowy* [22]. Formalnym zapisem takiego modelu jest *system informacyjny* [71, 72].

6.1.1. System informacyjny

Pojęcie *systemu informacyjnego* zostało upowszechnione przez Z. Pawlaka, który stosował je do opisu teorii zbiorów przybliżonych (ang. *rough sets*) [71]. System informacyjny jest określany za pomocą następującej pary:

$$\mathrm{SI} = (U, A), \tag{6.1}$$

gdzie:
$U = \{x_1, x_2, \ldots, x_N\}$ – niepusty i skończony zbiór obiektów zwanych *uniwersum*,
A – niepusty i skończony zbiór atrybutów $a_i \in A$ opisujących obiekty, taki że:

$$\forall_{a \in A} \left[g \colon U \to V_a \right], \tag{6.2}$$

V_a – dziedzina atrybutu a,
g – funkcja informacyjna.

W każdym systemie informacyjnym występuje skończony zbiór obiektów, zdarzeń itp. Obiekty systemu informacyjnego są charakteryzowane przez ich cechy tj. atrybuty i ich wartości. Z każdym atrybutem a należącym do zbioru atrybutów A związany jest zbiór jego wartości V_a. Zbiór V_a nazywany jest dziedziną atrybutu a. Dla każdego obiektu $x \in U$ oraz atrybutu $a_i \in A$ dana jest funkcja g, która obiektowi x przyporządkowuje wartość v należącą do dziedziny V_a atrybutu a.

System informacyjny zapisuje się zwykle w postaci tablicy, której kolumny odpowiadają atrybutom, a wiersze zgromadzonym przypadkom (przykładom) reprezentowanym przez wartości wszystkich rozpatrywanych atrybutów.

6.1.2. Temporalny system informacyjny

Dla atrybutów, których wartości zmieniają się w czasie, zostało wprowadzone pojęcie *Temporalnego Systemu Informacyjnego* (TSI) [7, 8]. Temporalny system informacyjny można określić poprzez podanie elementów następującej piątki:

$$\mathrm{TSI} = (U, A \cup \{t\}, \prec, \Delta t), \tag{6.3}$$

gdzie:
U, A – przypisywane jest takie samo znaczenie jak w przypadku systemu informacyjnego,
t – jest atrybutem określającym kolejność zdarzeń: $t \notin A$,
$\prec$ – jest relacją porządkującą dla atrybutu t:

$$\prec = \{(x, y) \colon x, y \in \mathbb{N} \text{ oraz } x < y\}, \tag{6.4}$$

$\Delta t = \mathrm{idem}$ – interwał (krok) czasu rozumiany jako *odległość* dwóch kolejnych elementów w t.

Dziedziną atrybutu porządkującego t jest dziedzina czasu, która niekoniecznie musi być opisywana za pomocą „*jawnych*" jednostek czasu, takich jak sekundy, godziny, lata itp. Przykładowo, w obszarze związanym z eksploatacją maszyn, atrybut t może być wielkością opisującą liczbę kilometrów przejechanych przez pojazd od chwili opuszczenia taśmy montażowej w fabryce [17].

Tablica 6.1 przedstawia przykład temporalnego systemu informacyjnego.

Tab. 6.1: Przykład temporalnego systemu informacyjnego

U	t	$u1$	$u2$	$u3$	$u4$	$y1$	$y2$	$y3$	$y4$
x_1	0	0,724	0,692	–2,28	1,80e–2	320	2,51	0,03	9,3
x_2	3	0,527	0,383	–3,72	1,87e–2	321	2,55	0,28	9,7
x_3	6	0,590	0,706	–1,53	2,08e–2	320	2,36	0,20	11,0
x_4	9	0,365	0,713	–3,17	2,29e–2	325	0,03	0,33	12,4
x_5	12	0,576	0,362	–1,80	2,04e–2	326	0,29	0,75	13,7
x_6	15	0,643	0,560	–2,80	2,32e–2	326	2,63	1,85	14,6
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮
x_N	28797	0,386	0,241	–1,52	0,46e–2	246	2,05	–1,24	7,8

6.1.3. Przekształcanie TSI w SI

Przy zastosowaniu metod odkrywania wiedzy w bazach danych dopuszcza się możliwość modyfikacji przestrzeni atrybutów tworzących dany system informacyjny. Celem prowadzenia tych modyfikacji jest poprawienie dokładności uzyskiwanych rozwiązań, zmniejszenie stopnia ich złożoności oraz zwiększenie czytelności. Przeprowadzenie modyfikacji polega na usunięciu, dodaniu nowych lub zastąpieniu istniejących atrybutów nowymi [21]. Operacja usunięcia istniejących atrybutów i/lub dodania nowych jest dozwolona pod warunkiem, że nie poszerzy przestrzeni rozwiązań. Wiedza o tym, który z atrybutów usunąć, a który dodać, wynika z wcześniejszej analizy zmian wartości atrybutów (danych), dostępnej wiedzy o dziedzinie reprezentowanej przez atrybuty lub jest wynikiem analizy rozwiązań generowanych przez zastosowany algorytm indukcji wiedzy.

W przypadku temporalnego systemu informacyjnego analiza występujących w nim regularności może polegać m.in. na:

- poszukiwaniu zależności funkcyjnych pomiędzy wartościami każdego z atrybutów a_1, a_2, ..., a_m a atrybutem porządkującym t, który w tym przypadku staje się tzw. zmienną niezależną,
- wyszukiwaniu podobnych sekwencji [15, 35],
- wyszukiwaniu podobieństw atrybutów [82],
- wyszukiwaniu zależności pomiędzy wartościami atrybutu $a_i(t)$ a wartościami, które atrybut a_i oraz pozostałe atrybuty przyjmowały we wcześniejszych chwilach czasu.

Z punktu widzenia analizy zależności dynamicznych występujących w temporalnym systemie informacyjnym (6.3) najbardziej interesującą możliwością jest ostatnia z nich. Identyfikacja zależności należących do tej klasy jest możliwa w przypadku utworzenia zbioru odpowiednio przygotowanych przykładów trenujących. Przykładowo, dla temporalnego systemu informacyjnego (Tab. 6.1) oraz ustalonej chwili czasu t, przykład trenujący x_i powinien składać się z wartości wybranego atrybutu zależnego np. $y1(t)$, z wartości jakie atrybut ten przyjmował we wcześniejszych chwilach czasu $y1(t - n\Delta t)$ oraz z wartości jakie przyjmowały pozostałe atrybuty $u1(t - n\Delta t)$, $u2(t - n\Delta t)$, ..., $y4(t - n\Delta t)$. Stała Δt jest różnicą pomiędzy kolejnymi chwilami czasu, w których zapisywano wartości atrybutów należących do rozpatrywanego TSI. Natomiast za pomocą parametru n zostały określone chwile czasu, dla których należy uwzględnić wartości rozpatrywanych atrybutów, przy czym $n \in \mathbb{N}$. Utworzenie zbioru nowych przykładów zgodnie z przedstawioną powyżej ideą prowadzi do przekształcenia temporalnego systemu informacyjnego w „*zwykły*" system informacyjny. Przekształcenie to konkretyzuje przedstawiony na rysunku 6.1 algorytm.

Działanie przedstawionego na rysunku 6.1 algorytmu jest następujące. W pierwszej kolejności zostają utworzone dwa puste zbiory wyjściowego systemu informacyjnego (wiersze 2-3), tj. zbiór przykładów $\hat{U}$ oraz zbiór atrybutów $\hat{A}$. W kolejnym kroku (wiersze 5-7), dla atrybutu ustalonego za pomocą głównej pętli algorytmu (wiersz nr 3), definiowane są bazujące na tym atrybucie nowe atrybuty, które następnie są dołączone do zbioru atrybutów $\hat{A}$. W wierszu 8. zostaje otwarta pętla, dzięki której w następnym

argumenty:

$\mathrm{TSI} = (U, A \cup \{t\}, \prec, \Delta t)$ – temporalny system informacyjny

$m = \mathbf{card}(A)$ – liczba atrybutów w TSI

$N = \mathbf{card}(U)$ – liczba obiektów (przykładów) w TSI

n – horyzont czasowy

zwracane:

$\mathrm{SI} = (\hat{A}, \hat{U})$ – system informacyjny

```
1:  TSI2SI (TSI, m, N, n)
2:      Â ← ∅
3:      Û ← ∅
4:      for i ← 1, 2, ..., m
5:          for l ← 0, 1, 2, ..., n
6:              Â ← Â ∪ a_il
7:          end
8:          for j ← 1, 2, ..., N − nΔt
9:              Û ← Û ∪ y_j
10:             l ← n
11:             for k ← j, j + Δt, j + 2Δt, ..., j + nΔt
12:                 a_il(y_j) ← a_i(x_k)
13:                 l ← l − 1
14:             end
15:         end
16:     end
17: return SI = (Â,Û).
```

Rys. 6.1: Algorytm TSI2SI

wierszu (wiersz nr 9) tworzone są przykłady należące do zbioru $\hat{U}$. W kolejnym kroku definiowana jest zmienna l. Jej celem jest odwrócenie *kierunku* indeksowania atrybutów w $\hat{A}$ tak aby atrybut, do którego zostanie dopisana pierwsza występująca wartość aktualnie rozpatrywanego atrybutu ze zbioru A, miał najwyższy indeks l. W wierszach 11-12 dla ustalonego za pomocą głównej pętli algorytmu (wiersz 4) oraz zmiennej l (wiersz 10) atrybutu oraz ustalonemu za pomocą pętli z wiersza 8 przykładowi uczącemu przypisywana jest odpowiednia wartość z wejściowego TSI. Algorytm kończy działanie zwracając zbiory $\hat{U}$ i $\hat{A}$ tworzące wymagany system informacyjny.

W tablicy 6.2 przedstawiono wynik jaki zostanie osiągnięty w przypadku przekształcenia temporalnego systemu informacyjnego (Tab. 6.1) w *zwykły* system informacyjny za pomocą zaproponowanego algorytmu (Rys. 6.1). W prezentowanym przykładzie przyjęto następujące wartości parametrów przekształcenia: $\Delta t = 3$ oraz $n = 2$.

Tab. 6.2: Wynik przekształcenia temporalnego systemu informacyjnego (Tab. 6.1) za pomocą algorytmu **TSI2SI** (Rys. 6.1)

U	$u1(t)$	$u1(t-\Delta t)$	$u1(t-2\Delta t)$	$\cdots$	$y1(t)$	$y1(t-\Delta t)$	$y1(t-2\Delta t)$	$\cdots$
x_1	0,724	0,527	0,590	$\cdots$	320	322	321	$\cdots$
x_2	0,527	0,590	0,365	$\cdots$	322	321	325	$\cdots$
x_3	0,590	0,365	0,577	$\cdots$	321	325	327	$\cdots$
x_4	0,365	0,577	0,643	$\cdots$	325	327	326	$\cdots$
x_5	0,577	0,643	0,495	$\cdots$	327	326	327	$\cdots$
x_6	0,643	0,495	0,585	$\cdots$	326	327	329	$\cdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
x_{N-2}	0,145	0,274	0,386	$\cdots$	247	248	247	$\cdots$

6.2. Indukcja zależności funkcyjnych

Gdy w systemie informacyjnym SI zbiór wartości V atrybutów A należy do zbioru liczb rzeczywistych $\mathbb{R}$, proces odkrywania wiedzy najczęściej dotyczy zadania aproksymacji funkcji [21]. Zadanie aproksymacji polega na poszukiwaniu zależności funkcyjnej wybranego atrybutu ciągłego od innych atrybutów, również ciągłych. Odkryta zleżność funkcyjna ma *dobrze* przybliżyć funkcję docelową $f(\,)$ zadaną przez wartości rozpatrywanych atrybutów. Warunek ten wynika bezpośrednio z potrzeby jaką jest prognozowanie wartości wybranego atrybutu zależnego. Główną trudnością w zadaniu aproksymacji funkcji jest pogodzenie czytelności formy, za pomocą której reprezentowana jest zależność funkcyjna, z wymaganą dokładnością. W przypadku gdy dokładność jest ważniejsza niż jawne reprezentowanie zależności za pomocą równania, w aproksymacji funkcji docelowej $f(\,)$ znajdują zastosowanie m.in. takie metody, jak:

- metoda modelowania neuronowego [24, 37, 42, 44, 83],
- metoda wektorów wspomagających [37, 77, 91],

Gdy nadrzędny priorytet stanowi czytelność, w zadaniu aproksymacji stosuje się takie metody, jak:

- metody regresji [45, 70, 93],
- metody odkrywania równań [26, 54, 97, 100].

Wybór jednej z wymienionych metod aproksymacji funkcji powinien wynikać z przewidywanego celu zastosowania odkrytej zależności funkcyjnej. Sterowanie układami technicznymi wymaga dużej dokładności, a więc do aproksymacji funkcji należy użyć metod gwarantujących jej uzyskanie. W przypadku opisywania praw fizyki, związków występujących w systemach biologicznych i in. duży nacisk kładzie się na czytelną formę reprezentacji aproksymowanych zależności. Należy wówczas stosować metody należące do drugiej grupy.

6.2.1. Zadanie aproksymacji funkcji

Zadanie aproksymacji funkcji jest formułowane w następujący sposób. Dla danej dziedziny V, przestrzeni funkcji aproksymujących (aproksymatorów) $\mathbb{H}$ oraz zbioru trenującego $L \subseteq U$ reprezentującego funkcję docelową $f(\,)$ należy znaleźć taką funkcję aproksymującą $h(\,)$, która będzie w świetle ustalonego układu kryteriów najlepszym przybliżeniem funkcji $f(\,)$. W przypadku gdy znany jest rozkład prawdopodobieństwa Ω na dziedzinie V oraz zbiór trenujący składa się z przykładów trenujących wybranych zgodnie z tym rozkładem, zadanie aproksymacji polega wówczas na wyborze takiego aproksymatora $h \in \mathbb{H}$, który będzie minimalizować błąd rzeczywisty Err_{gen}.

6.2.2. Funkcja docelowa

Na rozważanej dziedzinie V jest określona pewna funkcja docelowa $f : V \to \mathbb{R}$ o wartościach rzeczywistych.

6.2.3. Modele w aproksymacji funkcji

Dla zadania uczenia się aproksymacji funkcji, aproksymatory tworzące przestrzeń modeli $\mathbb{H}$ są funkcjami przekształcającymi przykłady z dziedziny V w zbiór liczb rzeczywistych. Dla $h \in \mathbb{H}$ mamy $h : V \to \mathbb{R}$.

W przypadku aproksymacji funkcji modele są funkcjami, które reprezentowane są za pomocą następujących elementów:

- s – struktura,
- w – zbiór wag (parametry),
- A_i – zbiór atrybutów (zmienne) $A_i \subseteq A$.

Zatem, formalny zapis modelu dla zadania aproksymacji funkcji ma postać:

$$h = \{s, w, A_i\}. \tag{6.5}$$

6.2.4. Zbiór trenujący

W najprostszym przypadku zbiór trenujący $L_A^{f(\,)}$, na podstawie którego dokonywana jest aproksymacja funkcji docelowej $f : V \to \mathbb{R}$, jest zbiorem par złożonych z przykładu z dziedziny, opisanego przez wartości atrybutów na niej określonych, i wartości funkcji docelowej dla tego przykładu, czyli

$$L_A^{f(\,)} = \{(\langle x \rangle_A, f(x)) | x \in L \subseteq U\}. \tag{6.6}$$

6.3. Ocena jakości predykcji aproksymatorów funkcji

Aproksymator $h(\,)$ przybliżający funkcję docelową $f(\,)$ wyznaczany jest na podstawie skończonego zbioru danych. Weryfikację aproksymatora $h(\,)$ przeprowadza się stosując

w najprostszym przypadku dwa zbiory przykładów:

$$U = L \cup T, \tag{6.7}$$

gdzie: podzbiór L nazywany jest zbiorem przykładów trenujących (lub zbiorem trenującym), podzbiór T nazywany jest zbiorem przykładów testowych (lub zbiorem testowym). Merytoryczna ocena dokładności wygenerowanego na podstawie zbioru przykładów trenujących L aproksymatora $h(\,)$ dokonywana jest poprzez wyznaczenie wartości estymat ilościowych miar dokładności predykcji [6, 84] i/lub estymat funkcyjnych (np. funkcja autokorelacji) [81] w odniesieniu do odpowiedniego zbioru przykładów testowych T.

Błąd predykcji aproksymatora $h(\,)$ dla dowolnego przykładu $x \in T$ definiowany jest jako następująca różnica:

$$e_{x \in T}(h) = f(x) - h(x). \tag{6.8}$$

Błąd (6.8) jest często określany mianem *residum* lub *reszty* [45, 70].

6.3.1. Standardowe miary dokładności predykcji

Wyznaczone dla kolejnych przykładów x ze zbioru testowego T błędy predykcji $e_{x \in T}(h)$ tworzą ciąg wartości, na podstawie którego wyznacza się następujące standardowe miary dokładności predykcji [6, 84]:

- błąd średni (ang. *Mean Error*, ME)

$$ME = \frac{1}{\mathrm{card}(T)} \sum_{x \in T} \left(f(x) - h(x) \right), \tag{6.9}$$

- średni moduł błędu (ang. *Mean Absolute Error*, MAE)

$$MAE = \frac{1}{\mathrm{card}(T)} \sum_{x \in T} |f(x) - h(x)|, \tag{6.10}$$

- suma kwadratów błędów (ang. *Sum of Squared Errors*, SSE)

$$SSE = \sum_{x \in T} \left(f(x) - h(x) \right)^2, \tag{6.11}$$

- błąd średniokwadratowy (ang. *Mean Squared Error*, MSE)

$$MSE = \frac{1}{\mathrm{card}(T)} \sum_{x \in T} \left(f(x) - h(x) \right)^2, \tag{6.12}$$

- współczynnik korelacji r

$$r = \frac{\sum_{x \in T} \left(h(x) - \bar{h}(x) \right) \left(f(x) - \bar{f}(x) \right)}{\sqrt{\sum_{x \in T} \left(h(x) - \bar{h}(x) \right)^2 \sum_{x \in T} \left(f(x) - \bar{f}(x) \right)^2}}, \tag{6.13}$$

gdzie:

$$\bar{h}(x) = \frac{1}{\mathrm{card}(T)} \sum_{x \in T} h(x), \tag{6.14}$$

$$\bar{f}(x) = \frac{1}{\mathrm{card}(T)} \sum_{x \in T} f(x), \tag{6.15}$$

- współczynniki determinacji r^2

$$r^2 = 1 - \frac{\sum_{x \in T} \left(f(x) - h(x)\right)^2}{\sum_{x \in T} \left(f(x) - \bar{f}(x)\right)^2}. \tag{6.16}$$

Wymienione powyżej miary są bardzo często stosowane pomimo szeregu ograniczeń jakie się wiążą z ich stosowaniem. Przykładowo zastosowanie miary MSE w procesie indukcji aproksymatora funkcji będzie prowadzić do wyznaczenia bardziej złożonych modeli. Ponadto miara MSE, jako miara bezwzględna, nie pozwala porównywać wyników predykcji w przypadku stosowania wyznaczonego aproksymatora dla różnych zbiorów testowych.

6.3.2. Względne miary dokładności predykcji

Alternatywą w stosunku do standardowych miar predykcji są miary względne. Miary względne dostarczają bardziej istotnych informacji, które pozwalają ocenić różne modele $h(\,)$. Najczęściej stosowanymi miarami względnymi są:

- błąd procentowy (ang. *Percentage Error*, PE)

$$PE_{x \in T} = 100 \left(\frac{f(x) - h(x)}{f(x)} \right), \tag{6.17}$$

- średni błąd procentowy (ang. *Mean Percentage Error*, MPE)

$$MPE = \frac{100}{\mathrm{card}(T)} \sum_{x \in T} \left(\frac{f(x) - h(x)}{f(x)} \right), \tag{6.18}$$

- bezwzględny błąd procentowy (ang. *Absolute Percentage Error*, ABE)

$$APE_{x \in T} = 100 \left| \frac{f(x) - h(x)}{f(x)} \right|, \tag{6.19}$$

- średni bezwzględny błąd procentowy (ang. *Mean Absolute Percentage Error*, MAPE)

$$MAPE = \frac{100}{\mathrm{card}(T)} \sum_{x \in T} \left| \frac{f(x) - h(x)}{f(x)} \right|. \tag{6.20}$$

6.3.3. Test istotności t Studenta

Ocena jakości modeli predykcyjnych bazuje na analizie szeregu reszt stanowiących różnice pomiędzy wartościami rzeczywistymi a wartościami prognozowanymi. Analiza reszt ma na celu m.in. sprawdzenie czy [84]:

- prognozy generowane przez model nie są *obciążone*,
- średnia wartość prognoz przyjmuje zbliżone wartości.

Jednym z zasadniczych etapów analizy reszt jest sprawdzenie czy prognozy generowane za pomocą modelu nie są *obciążone*, tzn. czy nie występuje istotna przewaga reszt dodatnich lub ujemnych. W poprawnie zbudowanym modelu reszty powinny mieć rozkład normalny o średniej równej zero.

Formalną ocenę zgodności rozkładu reszt z rozkładem normalnym można przeprowadzić za pomocą testów statystycznych. W szczególności można zastosować test statystyczny *t* Studenta. Za pomocą tego testu można m.in. weryfikować hipotezę zerową $H_0 : \mu = 0$, która w przypadku modeli predykcyjnych pozwala sprawdzić czy wartość średnia szeregu reszt różni się w sposób istotny od wartości zerowej.

Weryfikacja hipotezy zerowej $H_0 : \mu = 0$ polega na obliczeniu wartości statystyki *t*-Studenta:

$$t = \frac{|\bar{x} - \mu|}{s(\bar{x})}, \tag{6.21}$$

gdzie: μ – prawdziwa średnia z populacji, $\bar{x}$ – średnia wartość z próby, $s(\bar{x})$ – oszacowany błąd standardowy średniej z próby.

Następnie wartość t jest porównywana z wartością graniczną t_α rozkładu *t*-Studenta przy określonym poziomie istotności α i danej liczbie stopni swobody, która odpowiada pomniejszonej o jeden liczbie pomiarów, których użyto do wyznaczenia wartości t.

W przypadku gdy $t \leq t_\alpha$ wówczas nie ma podstaw do odrzucenia hipotezy zerowej H_0. W przeciwnym przypadku hipotezę H_0 można odrzucić z przyjętym ryzykiem błędu tj. poziomem istotności α. Istnieje wówczas istotna różnica między średnią wartością $\bar{x}$ a zerem.

Istnieje możliwość, że zidentyfikowany na podstawie danych pomiarowych model nie będzie dobrze opisywał pewnych obszarów modelowanego procesu, natomiast w pozostałych obszarach wartości wyjść generowanych przez model będą spełniały wymagania. Przedstawiona sytuacja ma najczęściej miejsce w przypadku identyfikacji modeli globalnych [18, 39, 46]. Przeprowadzenie weryfikacji nieobciążoności prognoz generowanych przez modele globalne z uwzględnieniem całego zbioru reszt może prowadzić w wielu przypadkach do negatywnej oceny modelu pomimo jego przydatności w pewnych określonych obszarach.

W związku z powyższym autor pracy zaproponował aby oprócz formalnej weryfikacji jakości modelu na poziomie globalnym prowadzić również weryfikację określonych przedziałów prognoz. W tym celu stosowany jest również test *t*-Studenta. Zakres (przedział) prognoz, na podstawie których prowadzona jest weryfikacja, wyznaczany jest za pomocą *okna* przesuwanego wzdłuż osi zmiennej określającej porządek generowanych przez model

prognoz. Szerokość okna odpowiada wybranej liczbie stopni swobody. Efektem opisanej techniki weryfikacji jest pewna funkcja $t()$, na podstawie której można określić obszary zastosowania weryfikowanego modelu.

6.3.4. Metody wyznaczania błędu predykcji aproksymatorów funkcji

Do wyznaczenia wymienionych w poprzednim podrozdziale błędów predykcji stosuje się następujące metody [85]:

- *Hold-out* polega na jednorazowym podziale zbioru przykładów U na dwa podzbiory $U = L \cup T$: podzbiór L nazywany zbiorem przykładów trenujących (lub zbiorem trenującym), podzbiór T nazywany zbiorem przykładów testowych (lub zbiorem testowym). Na podstawie zbioru trenującego L system odkryć wyznacza model $h(\,)$ będący przybliżeniem funkcji docelowej $f(\,)$. Z kolei zbiór T służy do ilościowej i jakościowej oceny wygenerowanego modelu z zastosowaniem przedstawionych w poprzednim podrozdziale miar.
- *Random subsampling* polega na N-krotnym powtórzeniu metody *Hold-out*, przy czym dla każdej iteracji podział dokonywany jest niezależnie. Ocena jest średnią arytmetyczną wartości zastosowanej miary dokładności predykcji:

$$\bar{\eta} = \frac{1}{N} \sum_{i=1}^{N} \eta_i. \tag{6.22}$$

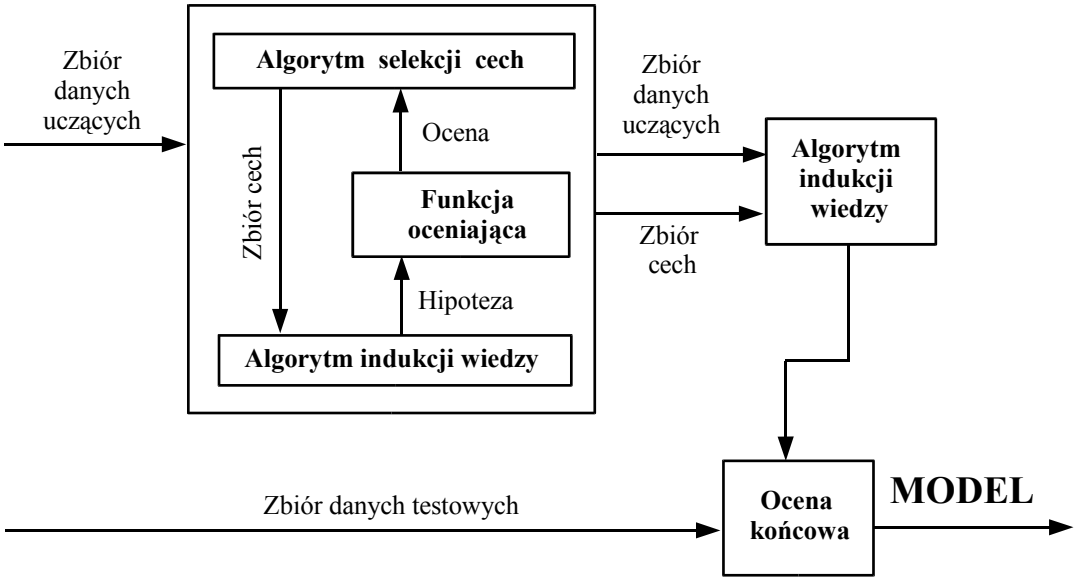
- *Leave-one-out* jest techniką, w której liczba iteracji jest równa liczbie przykładów uczących ($N = \mathrm{card}(U)$). W kolejnych iteracjach o numerach i, $i = 1, 2, \ldots, N$ ze zbioru przykładów U usuwa się jeden przykład x_i, tworząc zbiór uczący $L_i = E \backslash \{x_i\}$ oraz jednoelementowy zbiór testowy $T_i = x_i$. Technika ta jest obliczeniowo najbardziej kosztowna.
- *k-fold cross-validation* jest techniką podobną do techniki *leave-one-out*. Jednak w kolejnych iteracjach ze zbioru U usuwa się nie jeden przykład, ale wieloelementowy zbiór przykładów U_i. Wówczas $L_i = U \backslash U_i$ oraz $T_i = U_i$, natomiast $U = U_1 \cup \ldots \cup U_k$. Ocenę stanowi średnia arytmetyczna (6.22) wartości zastosowanej miary dokładności w każdej iteracji.

Wybór odpowiedniej techniki zależy od liczności zbioru przykładów.

6.4. Selekcja zbioru atrybutów relewantnych

Selekcja zbioru atrybutów relewantnych jest jednym z ważnych etapów procesu odkrywania wiedzy. Jej głównym celem jest określenie takiego podzbioru atrybutów, na podstawie którego wyznaczony model będzie się cechował dużą dokładnością predykcji oraz wysokim poziomem generalizacji.

Problemowi selekcji atrybutów relewantnych poświęcono wiele prac [22,41]. W większości przypadków dotyczą one selekcji atrybutów w zadanich klasyfikacji przykładów. W pracy [41] rozpatrzono dwa podejścia do problemu selekcji atrybutów: tzw. filtrację oraz selekcję z zastosowaniem algorytmu indukcji wiedzy (ang. *wrapper approach*). Schemat tego ostatniego podejścia przedstawiono na rysunku 6.2.



Rys. 6.2: Schemat procesu selekcji atrybutów z zastosowaniem algorytmu indukcji wiedzy [41]

Obydwa z prezentowanych w [41] podejść mają szereg wad i zalet. Filtracja[1] bazuje wyłącznie na ocenie atrybutów poprzez pryzmat zbioru wartości tych atrybutów. Nie jest brany pod uwagę stosowany w dalszej kolejności algorytm indukcji wiedzy oraz sprawność modelu zbudowanego na wyselekcjonowanym podzbiorze atrybutów. Z kolei selekcja atrybutów z zastosowaniem algorytmu indukcyjnego wymaga wielokrotnego powtarzania procesu indukcji wiedzy. Wpływa to na czas trwania procesu selekcji. Istotnym czynnikiem wpływającym na jakość uzyskanego podzbioru atrybutów jest funkcja oceny. Niewłaściwy dobór tej funkcji będzie skutkował wyznaczaniem podzbioru atrybutów, który nie zapewni identyfikacji modelu cechującego się wymienionymi wcześniej właściwościami.

Z uwagi na zdecydowanie większe możliwości dopasowania pod względem kompozycji różnorodnych algorytmów oraz użycia w różnorodnych problemach selekcji cech przyjęto, że w podjętych badaniach stosowana będzie metoda selekcji z zastosowaniem algorytmu indukcji wiedzy. Zgodnie z rysunkiem 6.2 stosowanie tej metody selekcji wymaga określenia trzech składników:

- algorytmu przeszukiwania,

[1]Do metod filtracji cech zalicza się wszelkiego rodzaju metody korelacyjne, analizę dyskryminacyjną czy też metody wykorzystujące ważenie cech.

- funkcji oceniającej,
- algorytmu indukcji wiedzy.

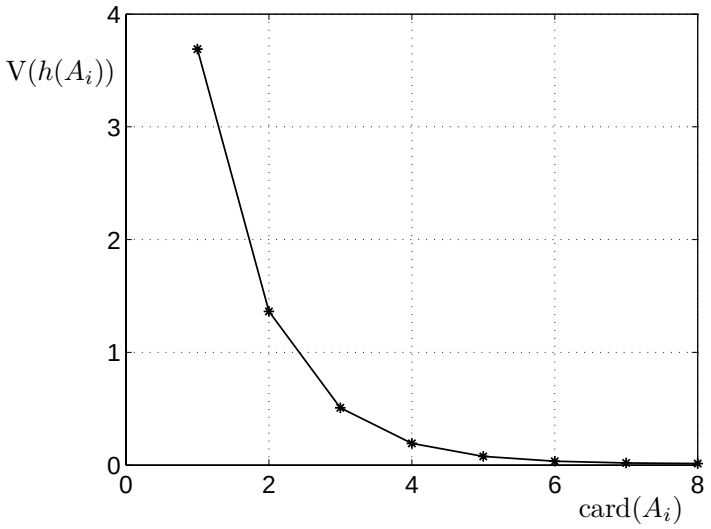
6.5. Kryteria selekcji cech

Elementem decyzyjnym opisanej w rozdz. 6.4 metody selekcji atrybutów jest funkcja oceniająca (Rys. 6.2). Od jej właściwego doboru zależy jakość modelu zidentyfikowanego na bazie wyselekcjonowanych atrybutów.

W przyjętej metodzie selekcji, o jakości wskazanego podzbioru atrybutów decyduje efekt zastosowania w postaci modelu wyznaczonego na tym podzbiorze. W takim przypadku ogólne wyrażenie na funkcję oceniającą dane jest w postaci następującego wyrażenia:

$$\tilde{A} = \arg\min_{A_i \subseteq A} J(h(A_i)), \tag{6.23}$$

gdzie: A_i – podzbiór zbioru atrybutów A, $J(\,)$ – funkcja oceniająca.



Rys. 6.3: Przykładowa zależność dokładności modelu od liczby atrybutów użytych do zbudowania modelu [81]

Podstawą do zbudowania funkcji oceniającej $J(\,)$ (6.23) jest zależność (Rys. 6.3) występująca pomiędzy dokładnością $V(h(A_i))$ identyfikowanych modeli a ich złożonością. Z analizy tej zależności wynika, że na krzywej (Rys. 6.3) znajduje się pewien punkt, który spełnia wymagania stawiane dokładności i złożoności dla identyfikowanego modelu. W związku z tym zadanie doboru kryterium $J()$ polega na zbudowaniu funkcji jednomodalnej, której optimum odpowiada lub jest zbliżone do wspomnianego wcześniej punktu. Rozwiązanie tego zadania jest trudne, co wynika z różnorodności zbiorów danych identyfikacyjnych oraz stosowanych metod identyfikacji systemów.

6.5.1. Kryterium informacyjne

W dziedzinie identyfikacji systemów wypracowano zbiór statystyk bazujących na teorii informacji, które pozwalają oceniać pozyskane modele pod względem złożoności ich struktur oraz dokładności. Ogólna postać tego typu kryteriów ma następującą formę [81]:

$$J = N \log acc(h(A_i)) + \beta(N, \mathrm{card}(A_i)). \tag{6.24}$$

W równaniu 6.24 dodatkowy człon $\beta(N, \mathrm{card}(A_i))$ „*karze*" za modele o zbyt dużej złożoności. Przyjęcie $\beta(N, \mathrm{card}(A_i)) = 2\mathrm{card}(A_i)$ prowadzi do kryterium informacyjnego Akaike (ang. *Akaike Information Criterion*, AIC) [2, 11, 81, 89]:

$$W_N = N \log acc(h(A_i)) + 2\mathrm{card}(A_i). \tag{6.25}$$

Zastosowanie kryterium AIC daje zazwyczaj modele cechujące się większą złożonością [89].

6.5.2. Funkcja heurystyczna

Innym podejściem do oszacowania wartości funkcji strat jest podejście heurystyczne, bardzo często stosowane w dziedzinie metod sztucznej inteligencji. W przypadku porównywania modeli różniących się znacznie pod względem złożoności i nieznacznie pod względem dokładności predykcji ogólnym zaleceniem jest wybór modelu o mniejszej złożoności (istnieją przesłanki, że taki model będzie posiadał większą zdolność do uogólniania). Korzystając z tej zasady można zbudować heurystyczną funkcję wyboru modelu o następującej ogólnej postaci:

$$W_N = acc(h(A_i)) \pm \rho \left(\frac{\mathrm{card}(A) - \mathrm{card}(A_i)}{\mathrm{card}(A)} \right). \tag{6.26}$$

Funkcja (6.26) składa się z dwóch członów. Pierwszy człon tej funkcji jest jedną z miar dokładności predykcji (rozdz. 6.3) , natomiast drugi człon pełni rolę czynnika „*karzącego*" za zbudowanie modelu na zbyt dużym zbiorze atrybutów.

Zadaniem mnożnika ρ występującego w równaniu (6.26) jest określenie ważności jednego z dwóch kryteriów uwzględnianych w tym równaniu, tj. kryterium dokładności oraz złożoność modeli. Określenie wartości tego współczynnika zależy od wielu czynników, m.in. wartości uzyskiwanych za pomocą zastosowanej miary dokładności predkcji modeli $acc(\,)$. Wyznaczenie *a priori* wartości współczynnika ρ w wielu przypadkach może sprawiać trudności. Na taki stan rzeczy mają wpływ m.in. różne przedziały wartości rozpatrywanych atrybutów jak również stosowany algorytm indukcji wiedzy. W celu minimalizacji problemu doboru wartości współczynnika ρ autor zaproponował aby miara $acc(\,)$ była jedną ze względnych miar ocen dokładności predykcji (rozdz. 6.3.2).

Dodatkowym aspektem jaki należało rozpatrzyć w zaproponowanej funkcji oceniającej (6.26) jest ustalenie znaku łączącego obydwa człony tej funkcji. W przypadku zastosowania miary dokladności predykcji, dla której mniejsze wartości oznaczają większą

dokładność (np. odchylenie średniokwadratowe), w równaniu (6.26) należy użyć operatora różnicy „−". W przypadku przeciwnym (np. kiedy miarą $acc(\,)$ będzie korelacja), operatorem tym powinien być operator sumy „+".

Przy uwzględnieniu powyższych spostrzeżeń jedna z możliwych form funkcji oceny modeli (6.26) może mieć następującą postać:

$$J(h(A_i)) = \mathrm{MAPE}(h(A_i)) - \rho\left(\frac{\mathrm{card}(A) - \mathrm{card}(A_i)}{\mathrm{card}(A)}\right). \tag{6.27}$$

gdzie: MAPE - średni bezwzględny błąd procentowy.

6.6. Przeszukiwanie przestrzeni atrybutów

Przeszukiwanie przestrzeni atrybutów A dla przyjętej metody selekcji można przeprowadzić z zastosowaniem różnych metod. W pracach [40, 41] zastosowano dwie strategie przeszukiwania przestrzeni cech:

- strategię zachłanną (ang. *hill-climbing*),
- strategię *najpierw najlepszy* (ang. *best-first*).

Inną możliwością jest zastosowanie algorytmów ewolucyjnych, w tym prostego algorytmu genetycznego [3, 34, 59]. Wyniki wielu prac, w których do selekcji atrybutów stosowano algorytmy genetyczne potwierdzają skuteczność podejścia ewolucyjnego.

6.6.1. Reprezentacja podzbiorów atrybutów

Zastosowanie algorytmu genetycznego w zadaniu selekcji atrybutów wymaga przyjęcia odpowiedniego sposobu kodowania „*osobników*". Najbardziej naturalne jest tu zastosowanie kodowania binarnego. Istota tego kodowania polega na utworzeniu chromosomu o długości (liczbie cyfr binarnych) odpowiadającej liczbie atrybutów występujących w przestrzeni atrybutów A. Następnie do każdego z genów w chromosomie przypisywany jest jeden z atrybutów a_i należących do rozważanej przestrzeni atrybutów A. Informacja o tym czy dany atrybut a_i ma wejść w skład analizowanego podzbioru atrybutów jest określana na podstawie wartości genu odpowiadającej temu atrybutowi. I tak w przypadku gdy wartość genu wynosi 1 uznaje się, że dany atrybut ma wejść w skład tworzonego podzbioru atrybutów, w przeciwnym przypadku, tj. gdy gen przyjmuje wartość 0, oznacza to że atrybut nie jest uwzględniany. Ideę takiego sposobu kodowania przedstawia rysunek 6.4.

$$\begin{array}{cccccccccc} a_1 & a_2 & a_3 & a_4 & a_5 & a_6 & a_7 & a_8 & a_9 & a_{10} \\ \big[\ 1 & 0 & 0 & 1 & 0 & 0 & 1 & 1 & 0 & 0\ \big] \end{array}$$

Rys. 6.4: Schemat kodowania rozwiązań dla zadania selekcji atrybutów z zastosowaniem algorytmu genetycznego

6.6.2. Modyfikacja reprodukcji proporcjonalnej

Dla algorytmów genetycznych, na etapie selekcji istnieje możliwość zastosowania różnych metod reprodukcji, m.in. [3]:

- reprodukcji proporcjonalnej,
- reprodukcji rangowej,
- reprodukcji turniejowej,
- reprodukcji progowej.

Do najchętniej stosowanych, należy reprodukcja proporcjonalna zwana także *metodą ruletki* [3,34]. W reprodukcji proporcjonalnej prawdopodobieństwo wylosowania *osobnika* jest wprost proporcjonalne do jego wartości funkcji przystosowania. Prawdopodobieństwo to jest wyznaczane wg następującej zależności:

$$p_r(\mathbf{X}) = \frac{\Phi(\mathbf{X})}{\sum_{Y \in P^t} \Phi(\mathbf{Y})}. \tag{6.28}$$

Zastosowanie metody selekcji proporcjonalnej wymaga aby:

- wartości funkcji przystosowania *osobników* wyznaczanych w całym cyklu algorytmu genetycznego były dodatnie,
- rozwiązywane za pomocą algorytmu genetycznego zadanie optymalizacji polegało na wyznaczeniu *osobnika*, dla którego wartość funkcji przystosowania byłaby maksymalna.

Drugi z wymienionych powyżej postulatów dotyczy sytuacji, w której zadanie optymalizacji jest zadaniem minimalizacji pewnej funkcji celu. Przejście od zadania minimalizacji do zadania maksymalizacji wymaga przemnożenia funkcji przystosowania przez wartość -1. Efekt przeprowadzenia takiej operacji w ogólnym przypadku powoduje, że wartości funkcji przystosowania będą wartościami ujemnymi. Jest to w sprzeczności z pierwszym z wymienionych powyżej postulatów. Problem ujemnych wartości funkcji przystosowania jest rozwiązywany poprzez dodanie do funkcji przystosowania pewnej stałej wartości Φ_0.

Dodanie stałej wartości do funkcji przystosowania powoduje zmianę wartości prawdopodobieństwa wyznaczanego za pomocą wyrażenia (6.28).

W celu ograniczenia wpływu wymienionych problemów zaproponowano algorytm przedstawiony na rysunku 6.5 modyfikujący w odpowiedni sposób wartości funkcji przystosowania. Modyfikacja wartości funkcji przystosowania jest przeprowadzana dla każdej iteracji algorytmu genetycznego. Reprodukcja proporcjonalna jest wówczas prowadzona z zastosowaniem zmodyfikowanych wartości funkcji przystosowania.

argumenty:

$\mathbf{P}^k$ – populacja bazowa pokolenia k

k – numer pokolenia

N – rozmiar populacji (liczba osobników)

zwracane:

$\Phi'(\mathbf{P}^k)$ – zmodyfikowana wartość przystosowania

1: **MODYFIKACJA** $(\mathbf{P}^k, k, N)$

2: $\Phi_{\min} = \min(\Phi(\mathbf{P}^k))$;

3: $\Phi_{\max} = \max(\Phi(\mathbf{P}^k))$;

4: $\Phi_0 = \Phi_{\min} - \dfrac{(\Phi_{\max} - \Phi_{\min})}{N}$;

5: $\Phi'(\mathbf{P}^k) = \Phi(\mathbf{P}^k) - \Phi_0$;

6: **return** $\Phi'(\mathbf{P}^k)$.

Rys. 6.5: Algorytm modyfikacji wartości funkcji przystosowania dla reprodukcji proporcjonalnej

Rozdział 7

Weryfikacja metody

Celem rozdziału jest przedstawienie i omówienie badań, jakie przeprowadzono w celu weryfikacji zaproponowanej metody odkrywania modeli opisujących dynamikę procesów lub obiektów. Prowadzone badania obejmowały takie elementy, jak przygotowanie oprogramowania działającego wg zaproponowanej metody, pozyskanie odpowiednich baz danych, przygotowanie zbiorów przykładów uczących, odkrywanie i testowanie modeli.

Badania weryfikacyjne przeprowadzono na podstawie danych pochodzących z dwóch różnych źródeł:

- eksperymentów symulacyjnych,
- bazy danych zawierającej wyniki pomiarów wybranych parametrów pracy pomp głębinowych.

7.1. Plan weryfikacji

Zaproponowana metoda odkrywania zależności dynamicznych w bazach danych weryfikowana jest w dwóch zakresach. Pierwszy zakres obejmuje weryfikację dla danych pozyskanych poprzez numeryczne symulacje hipotetycznych systemów dynamicznych. Znajomość cech symulowanych systemów dynamicznych pozwala ocenić metody pod względem poprawności uzyskiwanych wyników. Z kolei drugi obszar, w którym do weryfikacji metody stosowane są dane będące wynikami pomiarów rzeczywistego układu technicznego, pozwala ocenić przydatność metody w zastosowaniach praktycznych. Przebieg działań wykonywanych na obu etapach jest zbliżony. Główne różnice dotyczą sposobu pozyskania oraz przygotowania danych do analizy.

W obu przypadkach na plan weryfikacji składają się następujące kroki:

1. planowanie eksperymentu,
2. pozyskanie danych,
3. wstępne przetworzenie danych,
4. przygotowanie środków pomocniczych,
5. określenie parametrów metody,
6. określenie klas odkrywanych modeli,

7. określenie metod oceny pozyskanych modeli,
8. ocena wyników końcowych.

7.2. Dobór metody indukcji zależności funkcyjnych

W zaproponowanej metodzie odkrywania modeli dynamicznych nie określono szczegółowo algorytmu indukcji zależności funkcyjnych. Daje to możliwość doboru metody adekwatnej dla rozwiązywanego problemu. Przekształcenie danych wg algorytmu przedstawionego na rysunku 6.1 pozwala w praktyce na wybór dowolnej metody aproksymacji zależności funkcyjnych. Szerokie spektrum tych metod utrudnia dokonanie właściwego wyboru. W celu weryfikacji zaproponowanej w pracy metody odkrywania zależności funkcyjnych posłużono się następującym układem kryteriów:

- możliwość identyfikacji zależności nieliniowych w szerokim zakresie,
- dokładność wyników,
- czytelność wyników,
- dostępność oprogramowania,
- odporność na występowanie w danych wartości odstających,
- ontogeniczność[1],
- efektywność obliczeniowa (zdolność bazującego na metodzie oprogramowania do przetwarzania dużych ilości danych w relatywnie krótkim czasie).

Na podstawie powyższego układu kryteriów wybrano metodę wektorów wspomagających (ang. *Support Vector Machines*, SVM) [91]. Metoda ta jest z powodzeniem stosowana do indukcji klasyfikatorów w zadaniach klasyfikacyjnych, jak również można ją stosować do aproksymacji zależności funkcyjnych, w szczególności o nieliniowym charakterze.

7.2.1. Metoda Wektorów Wspomagających

W metodzie SVM zadanie aproksymacji funkcji $f(\,)$ jest formułowane następująco [77, 80, 91]:

$$f(x) = w^{\mathrm{T}} x + b, \tag{7.1}$$

Jakość aproksymacji funkcji $f(\,)$ jest mierzona za pomocą następującej ε–niewrażliwej funkcji strat [91]:

$$L\left(y, f(\mathbf{x})\right) = \begin{cases} 0 \text{ dla } |y - f(\mathbf{x})| \leq \varepsilon \\ |y - f(\mathbf{x})| - \varepsilon \text{ w przeciwnym przypadku.} \end{cases} \tag{7.2}$$

W ogólnym przypadku funkcja docelowa $f(\,)$ może być funkcją nieliniową. Wówczas aproksymacja funkcji $f(\,)$ jest dokonywana w pewnej wysokowymiarowej przestrzeni cech $\mathcal{Z}$. Przestrzeń cech $\mathcal{Z}$ jest najczęściej nieliniowym produktem skalarnym pewnych

[1] Ontogeniczność jest zdolnością modeli do zmiany swojej struktury w trakcie procesu uczenia [37].

funkcji bazowych $\phi_i(x)$ określonych w przestrzeni wejściowej. W takim przypadku równanie (7.1) przyjmuje następującą formę:

$$f(x) = \sum_{i=1}^{n} \alpha_i y_i K(x_i, x) + b, \tag{7.3}$$

gdzie $K(x_i, x)$ jest jądrem iloczynu skalarnego funkcji bazowych $\phi_j(x)$, $j = 1, 2, \ldots, m$:

$$K(x, x') = \phi_i(x)^{\mathrm{T}} \phi_i(x'). \tag{7.4}$$

Od funkcji jądra K reprezentujących iloczyn skalarny funkcji bazowych $\phi_i(x)$ wymaga się aby były dodatnio określone (zgodnie z teorią przestrzeni Hilberta). Do najczęściej stosowanych funkcji jądra K zalicza się [37, 77, 80]:

- jądro liniowe
$$K(x, x_i) = x^T x_i, \tag{7.5}$$
- jądro wielomianowe
$$K(x, x_i) = \left[\gamma x^T x_i + \theta\right]^q, \tag{7.6}$$
przy czym K jest dodatnio określone gdy $\theta \in \{0; 1\}$ i $q \in \mathbb{N}$ oraz $\gamma \geq 0$,
- jądro radialne (RBF)
$$K(x, x_i) = \exp\left(-\gamma ||x^T - x_i||^2\right) = \exp\left(-\frac{||x - x_i||^2}{2\sigma^2}\right), \tag{7.7}$$
przy czym $\gamma = 1/2\sigma^2 \geq 0$,
- jądro sigmoidalne
$$K(x, x_i) = \mathrm{tgh}\left(\gamma x^T x_i + \theta\right). \tag{7.8}$$

Liniowa aproksymacja funkcji $f(\,)$ za pomocą równania (7.3) jest dokonywana z jednoczesnym uwzględnieniem wartości funkcji strat (7.2) oraz redukcji złożoności modelu poprzez minimalizację normy $||w||^2$. Problem ten jest rozwiązywany za pomocą dwóch (nieujemnych) zmiennych ζ_i, ζ_i^*, $i = 1, \ldots, n$, które mierzą odchylenie wektora x_i od strefy wyznaczonej przez ε–niewrażliwą funkcję strat. Zadanie aproksymacji funkcji (7.1) z uwzględnieniem funkcji strat (7.2) jest rozpatrywane jako zadanie optymalizacji następującego funkcjonału [37, 77]:

$$\min_{w,\zeta,\zeta^*,\varepsilon,b} \quad \frac{1}{2}||w||^2 + C\sum_{i=1}^{N}\left(\zeta_i + \zeta_i^*\right), \tag{7.9}$$

przy ograniczeniach:

$$f(x_i) - y_i \leq \varepsilon + \zeta_i \; i = 1, \ldots, N, \tag{7.10}$$
$$y_i - f(x_i) \leq \varepsilon + \zeta_i^*, \tag{7.11}$$
$$\zeta_i, \zeta_i^* \geq 0, \tag{7.12}$$

Zadanie optymalizacji (7.9) może zostać przekształcone do problemu dualnego, a jego rozwiązanie dane jest w postaci [37, 77]:

$$f(x) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(x_i, x) + b, \tag{7.13}$$

przy ograniczeniach:

$$\sum_{i=1}^{N} (\alpha_i - \alpha_i^*) = 0 \quad i = 1, \ldots, N, \tag{7.14}$$

$$0 \leq \alpha_i, \alpha_i^* \leq C, \tag{7.15}$$

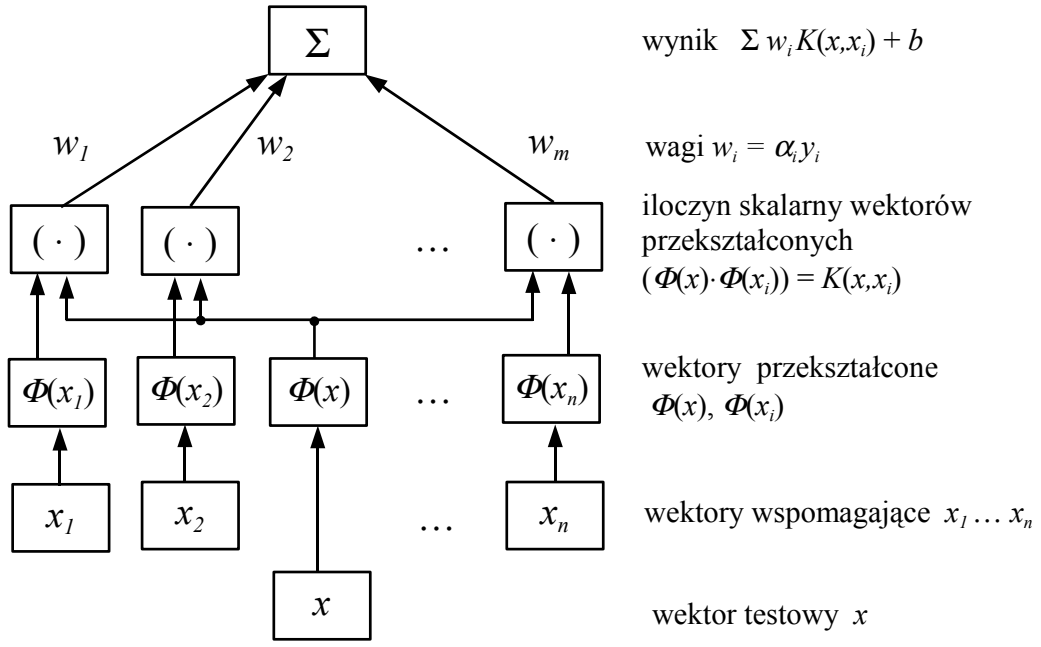
gdzie: α_i^*, α_i – możniki Lagrange'a , $K(x_i, x)$ – funkcja jądra.

Wartość stałej b można wyznaczyć korzystając z nierówności 7.10 i 7.11 dla $\zeta_i, \zeta_i^* = 0$ [80]:

$$b = y_i - f(x_i) - \varepsilon \text{ dla } \alpha_i \in (0, C), \tag{7.16}$$

$$b = y_i - f(x_i) + \varepsilon \text{ dla } \alpha_i^* \in (0, C). \tag{7.17}$$

Najczęściej jednak wartość stałej b jest wyznaczana w procesie selekcji zbioru wektorów wspomagających [37, 80].



Rys. 7.1: Wyznaczanie wartości wyjścia modelu SVM [80]

Efektem zastosowania metody wektorów wspomagających jest model złożony z:

- wybranej funkcji jądra i jej parametrów,
- zbioru wektorów wspomagających,

- wartości stałej b.

Sposób wyznaczania wyjścia modelu zbudowanego z zastosowaniem metody wektorów wspomagających wyjaśnia rysunek 7.1.

Szczegółowe omówienie metody wektorów wspomagających można znaleźć w wielu publikacjach, m.in. w [23, 37, 78, 80, 91].

7.2.2. Dobór parametrów metody wektorów wspomagających

Stopień generalizacji (uogólnienia) w przypadku stosowania metody SVM zależy od dwóch czynników: właściwego doboru wartości metaparametrów C i ε oraz od właściwego doboru funkcji jądra K i jej parametrów. Problem optymalnego zestawienia parametrów metody SVM jest dodatkowo utrudniony poprzez występowanie współzależności pomiędzy tymi parametrami. Dobór parametrów pozostawia się zazwyczaj użytkownikowi, który określa ich wartości bazując na dostępnej *a priori* wiedzy lub własnym doświadczeniu. W przypadku użytkowników nie zaliczających się do grona ekspertów polecane są w literaturze [13, 14, 77] różne metody doboru parametrów bazujące na znanych metodach przeszukiwania przestrzeni stanów [9, 66]. Dobrym rozwiązaniem w tym zakresie jest zastosowanie metody „*wpierw najlepszy*". Pozwala ona zwiększyć efektywność przeszukiwania przestrzeni parametrów poprzez uwzględnienie dodatkowej informacji heurystycznej związanej z rozwiązywanym problemem. Szczegółowy opis tej strategii można znaleźć m.in. w [9, 66].

Parametr C wpływa na relacje występujące pomiędzy złożonością modelu SVM a stopniem w jakim odchylenia większe od ϵ są tolerowane w (7.9). Dla przykładu, jeżeli C jest bardzo duże (nieskończone) to na minimalizację funkcji strat (7.2) ma zasadniczy wpływ drugi człon formuły (7.9). Człon odpowiedzialny w (7.9) za złożoność modelu ma wówczas marginalne znaczenie.

Z kolei za pomocą parametru ε kontrolowana jest szerokość ε–niewrażliwej strefy. Wartość ε wpływa na liczbę wektorów wspomagających będących jednym z elementów modelu (7.3). Im większa wartość ε, tym mniejsza liczebność zbioru wyselekcjonowanych wektorów wspomagających. Z drugiej strony, duża wartość ε prowadzi do bardziej "*płaskich*" estymat. Wynika stąd, że zarówno metaparametr C jak i ε wpływają na złożoność modelu SVM, lecz każdy z nich w inny sposób.

7.3. Weryfikacja metody dla danych symulacyjnych

Celem tego rozdziału jest przedstawienie i omówienie badań jakie prowadzono w zakresie weryfikacji zaproponowanej metody odkrywania ilościowych zależności dynamicznych w bazach danych. W szczególności prowadzone na tym etapie badania dotyczyły odkrywania modeli dla danych pozyskanych poprzez symulacje numeryczne hipotetycznych systemów dynamicznych. Zastosowanie danych pozyskanych na drodze symulacji numerycznych pozwala dokładnie zbadać skuteczność nowej metody. Jest to ważne z uwagi na:

- znajomość *a priori* wszystkich cech (zmiennych, struktury i parametrów) identyfikowanego systemu,
- możliwość kontrolowania parametrów zarówno w przypadku stosowanych systemów, jak również parametrów prowadzonych symulacji.

Z drugiej strony, fakt całkowitego zdeterminowania postaci wybranych systemów symulacyjnych oraz wyników symulacji komputerowych może budzić wątpliwości co do skuteczności metody w przypadku analizy danych będących wynikami obserwacji rzeczywistych systemów.

7.3.1. Plan eksperymentu w przypadku odkrywania modeli dla danych symulacyjnych

Weryfikacja zaproponowanej metody w zakresie analizy danych symulacyjnych wymaga realizacji następujących etapów:

1. Przegląd systemów dynamicznych, które znalazły zastosowanie w weryfikacji różnych metod identyfikacji systemów.
2. Generowanie danych symulacyjnych:
   - sygnał wejściowy,
   - warunki początkowe,
   - organizacja wyników symulacji.
3. Przygotowanie zbiorów przykładów uczących:
   - skalowanie danych,
   - przekształcenie przestrzeni atrybutów w temporalnych systemach informacyjnych utworzonych na podstawie danych symulacyjnych.
4. Określenie procesu odkrywania modeli i jego parametrów:
   - zdefiniowanie klas odkrywanych modeli,
   - dobór parametrów algorytmu genetycznego,
   - określenie metody trenowania i testowania.
5. Ewaluacja i dyskusja wyników eksperymentu.
6. Podsumowanie.

7.3.2. Przegląd testowych systemów dynamicznych

W podrozdziale dokonano przeglądu systemów dynamicznych reprezentowanych za pomocą równań różnicowych, mogących stanowić źródło danych wymaganych do przeprowadzenia weryfikacji metody. Przegląd przeprowadzono biorąc pod uwagę następujący układ kryteriów:

- system powinien być przedstawiony w postaci równania różnicowego,
- system powinien być systemem klasy MISO,
- system powinien cechować się w miarę prostą strukturą,
- system powinien być stabilny,
- system powinien być liniowy względem parametrów.

W wyniku przeprowadzonego przeglądu zidentyfikowano następujący zbiór systemów spełniających przyjęty układ kryteriów [56, 67, 69]:

- system **S1**

$$y(k) = 0.8y(k-1) + u(k-1), \tag{7.18}$$

- system **S2**

$$y(k) = 1.5y(k-1) - 0.7y(k-2) + 0.9u(k-2) + 0.5u(k-3), \tag{7.19}$$

- system **S3**

$$y(k) = \frac{y(k-1)}{1+y(k-1)} + u(k-1)^3, \tag{7.20}$$

- system **S4**

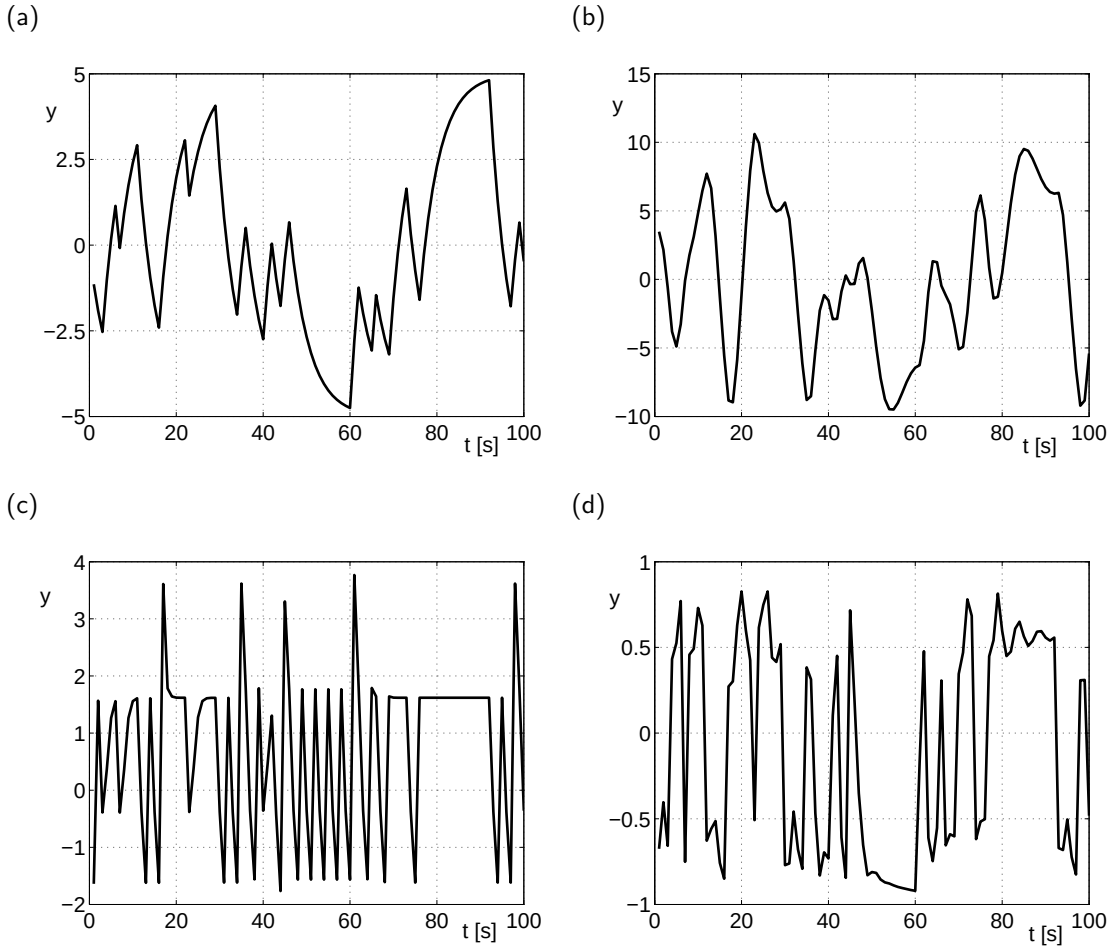
$$y(k) = \frac{y(k-1)y(k-2)y(k-3)u(k-2)\left(y(k-3)-1\right)+u(k-1)}{1+y(k-2)^2+y(k-3)^2}. \tag{7.21}$$

Dwa pierwsze z wymienionych powyżej systemów **S1** (7.18) oraz **S2** (7.19) zostały przedstawione w pracy [56], gdzie posłużyły do zaprezentowania funkcjonalności oprogramowania przeznaczonego do identyfikacji systemów. System **S3** (7.20) zastosowano w pracy [69] do wykazania skuteczności ewolucyjnego algorytmu doboru parametrów sieci neuronów dynamicznych ESSS–FDM (ang. *Evolutionary Search with Soft Selection and Forced Direction of Mutation*). Z kolei system **S4** (7.21) po raz pierwszy opisano w pracy [67], gdzie został użyty jako przykład modelowania systemów dynamicznych za pomocą sztucznych sieci neuronowych. W pracy [92] posłużyły do testowania procedury identyfikacji lokalnych modeli liniowych w postaci równań stanu. Przebiegi sygnałów wyjściowych zidentyfikowanych systemów przedstawiono na rysunku 7.2.

7.3.3. Generowanie danych symulacyjnych

Przeprowadzenie symulacji systemów **S1** (7.18), **S2** (7.19), **S3** (7.20) i **S4** (7.21) opisanych równaniami różnicowymi wymaga określenia następujących elementów:

- postaci sygnału wejściowego $u(t)$ pobudzającego system,
- parametrów addytywnego szumu $e(t)$,
- warunków początkowych.



Rys. 7.2: Przebieg sygnałów wyjściowych: (a) systemu (7.18), (b) systemu (7.19), (c) systemu (7.20) i (d) systemu (7.21)

Sygnał wejściowy

W dziedzinie identyfikacji systemów zastosowanie znajdują następujące typy sygnałów wejściowych [81]:

- funkcja skokowa,
- pseudolosowy ciąg binarny,
- autoregresyjny proces ruchomej średniej,
- suma sygnałów sinusoidalnych.

Wybór określonego typu sygnału wejściowego zależy od wielu czynników, wśród których do istotnych należą m.in. stosowana metoda identyfikacji oraz charakter rzeczywistego sygnału wejściowego oddziaływującego na system. W przypadku systemów technicznych często spotykanymi sygnałami wejściowymi są sygnały binarne, za pomocą których określa się stan pracy układu. Dodatkowym wymogiem stawianym sygnałom wejściowym jest aby sygnał wejściowy był sygnałem trwale pobudzającym.

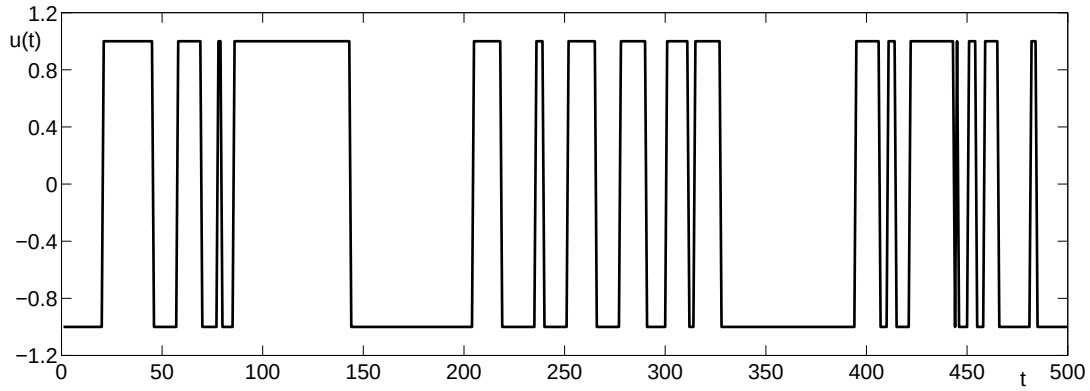
Uwzględniając powyższe uwagi przyjęto, że sygnałem wejściowym $u(k)$ będzie pseudolosowy ciąg binarny (ang. *Pseudorandom Binary Sequence*, PRBS) opisany następu-

jącą zależnością [81]:

$$u(k) = \begin{cases} u(k-1) & \text{z prawdopodobieństwem } \alpha, \\ n(k) & \text{z prawdopodobieństwem } 1-\alpha, \end{cases} \tag{7.22}$$

gdzie: $n(k)$ jest białym szumem o rozkładzie dwupunktowym $P(n(k)=1) = P(n(k) = -1) = 0.5$. Parametr α określa prawdopodobieństwo zmiany poziomu sygnału $u(k)$.

Na rysunku 7.3 przedstawiono przebieg sygnału PRBS, dla którego prawdopodobieństwo α przyjęto na poziomie $0,75$.



Rys. 7.3: Postać sygnału wejściowego stosowanego dla potrzeb symulacji systemów (7.18), (7.19), (7.20) i (7.21)

Zakłócenia

Na każdy rzeczywisty system oprócz znanych i kontrolowanych wejść oddziaływują również inne nieznane czynniki, które zakłócają działanie systemu. Obserwacje danego systemu są prowadzone z zastosowaniem układów pomiarowych, na które podobnie jak w przypadku obserwowanego systemu oddziałują nieznane zewnętrzne czynniki. W dokonanych obserwacjach kumulują się nieznane oddziaływania, które są rozróżniane jako szum. Konieczne jest zatem uwzględnienie wpływu tych czynników w prowadzonych symulacjach. Odbywa się to poprzez dodanie do wyjścia systemu zmiennej losowej $e(t)$, której rozkład wartości jest zazwyczaj rozkładem normalnym. W takim przypadku przyjmuje się, że wartość średnia białego szumu $e(t)$ jest równa 0. Wartość wariancji szumu $e(t)$ zależy od stopnia wpływu nieznanych wejść na system i można ją wyznaczyć za pomocą zależności (7.23) dla zadanej wartość miary SNR (ang. *Signal to Noise Ratio*) [4,20,57].

$$\sigma_e = 10^{(\log \sigma_s - 0.05\,\mathrm{SNR})}, \tag{7.23}$$

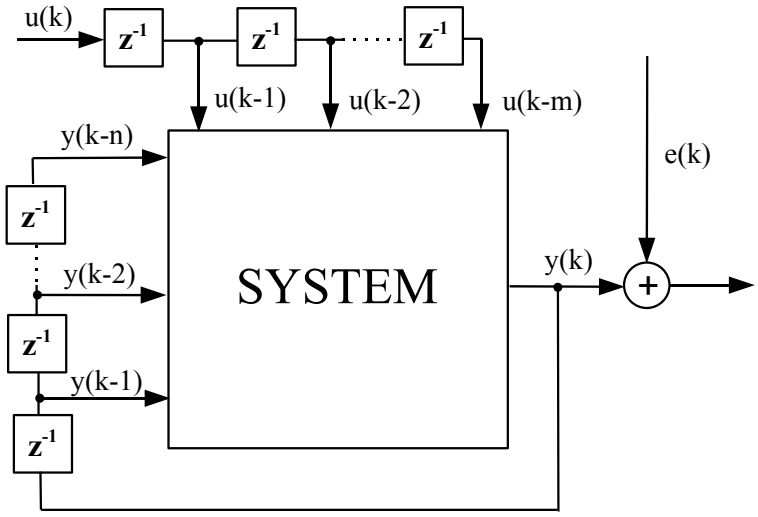
gdzie: σ_s – jest wariancją wybranego sygnału wyjściowego symulowanego systemu, a wartość SNR w [dB] jest przyjmowana w zależności od ustalonego stopnia wpływu nieznanych czynników zewnętrznych na symulowany system.

Warunki początkowe

Przeprowadzenie symulacji dowolnego systemu dynamicznego wymaga określenia liczby i wartości warunków początkowych. Liczba warunków początkowych zależy od maksymalnej wartości opóźnienia w symulowanym systemie. Najczęściej przyjmowane są zerowe wartości warunków początkowych, w przypadku gdy nie są znane *a priori* dodatkowe informacje pozwalające ustalić korzystniejsze wartości tych warunków.

Symulacja

Wejściowo–wyjściowy opis systemów **S1** (7.18), **S2** (7.19), **S3** (7.20) i **S4** (7.21), pozwala w sposób bezpośredni prowadzić proces symulacji. Proces ten jest prowadzony iteracyjnie, zgodnie ze schematem przedstawionym na rysunku 7.4. Liczba opóźnień z^{-1} sygnału wejściowego jak również sygnału wyjściowego, które podawane są na wejście systemu, wynika bezpośrednio z równań opisujących wybrane systemy. W tablicy 7.1 zestawiono parametry sumulacji wybranych systemów.



Rys. 7.4: Ogólny schemat symulacji systemów dynamicznych przyjętych do weryfikacji metody

Tab. 7.1: Parametry symulacji wybranych systemów dynamicznych

System	Warunki początkowe	SNR [dB]	Liczba iteracji
S1	$y(t=0)=0$	25	2048
S2	$y(t=0)=y(t=1)=y(t=2)=0$	25	2048
S3	$y(t=0)=1$	25	2048
S4	$y(t=0)=y(t=1)=y(t=2)=0$	25	2048

7.3.4. Wstępne przetwarzanie danych

Z uwagi na zwiększenie sprawności przetwarzania danych [84] oraz zapewnienie poprawnego działania procedur numerycznych [58] realizujących metodę wektorów wspomagających, pozyskane sygnały poddano operacji skalowania. Celem skalowania [84] jest modyfikacja wartości sygnałów tak aby mieściły się w ściśle określonym przedziale. Skalowanie do dowolnego przedziału $[a; b]$ można przeprowadzić za pomocą następującego przekształcenia [84]:

$$\tilde{z}_i = \frac{[z_i - \min(z)]\,(b-a)}{\max(z) - \min(z)} + a. \tag{7.24}$$

Stosowanie danych przekształconych (7.24) powoduje, że zidentyfikowany model jest dostrojony do wartości tych danych. Wówczas, jego zakres stosowania obejmuje przedział wartości danych, na podstawie którego został zidentyfikowany. Próba zastosowania go dla innego zbioru danych, będzie prowadzić do błędnych wyników. Ponadto, w wielu przypadkach ważny jest *powrót* z wartościami generowanymi przez model do pierwotnego przedziału wartości. Zatem aby odkryty model był w pełni funkcjonalny konieczna jest znajomość parametrów przekształcenia (7.24), które posłużyły do wyznaczenia modelu. Określenie parametrów przekształcenia wymaga redefinicji wyrażenia (7.24). Ponieważ równanie (7.24) jest liniowe, zatem funkcja skalowania danych do przedziału $[a; b]$ przyjmuje następującą postać:

$$\tilde{z}_i = \alpha \cdot z_i + \beta, \tag{7.25}$$

gdzie:

$$\alpha = \frac{b-a}{\max(z) - min(z)}, \tag{7.26}$$

$$\beta = \frac{(2a-b)\cdot\min(z) - a\cdot\max(z)}{\max(z) - \min(z)}. \tag{7.27}$$

Za pomocą zmiennych α i β można zdefiniować przekształcenie odwrotne, pozwalające na powrót do przedziału wartości pierwotnych:

$$z_i = \frac{\tilde{z}_i - \beta}{\alpha}. \tag{7.28}$$

7.3.5. Odkrywanie modeli testowych systemów dynamicznych

Określenie procesu odkrywania testowych systemów symulacyjnych wymaga następujących działań:

- wyboru algorytmu metody SVM, za pomocą którego będą odkrywane modle testowych systemów dynamicznych,
- zdefiniowania klas modeli (określenie wartości parametrów dla wybranego algorytmu metody SVM),
- ustalenia wartości parametrów następujących elementów opracowanej metody odkrywania zależności dynamicznych:
  - parametrów przekształcenia **TSI** w **SI**,

  - parametrów algorytmu genetycznego,
  - sposobu wyznaczania miary pozwalającej ocenić dokładność/jakość odkrywanych modeli,
- przyjęcie planu identyfikacji testowych systemów dynamicznych.

Klasy modeli zastosowanych do odkrywanych testowych systemów dynamicznych

Zastosowanie metody wektorów wspomagających determinuje potrzebę nałożenia ograniczeń w postaci parametrów metody na klasę odkrywanych modeli dynamicznych. Wymaga to uprzedniego wyboru algorytmu SVM. Dla rozpatrywanego problemu badawczego wybrano algorytm ν-SVR [37,77,80], który znajduje zastosowanie w aproksymacji zależności funkcyjnych. W przypadku algorytmu ν-SVR, zdefiniowanie określonej klasy odkrywanych modeli wymaga ustalenia:

- funkcji jądra,
- wartości parametrów funkcji jądra (γ dla RBF (7.7)),
- wartości metaparametru C,
- wartości metaparametru ν.

Na potrzeby odkrywania modeli wybranych systemów symulacyjnych ustalono pięć klas tych modeli (Tab. 7.2). Wartości metaparametrów stosowanego algorytmu SVM oraz parametry funkcji jądra dobrano na podstawie zaleceń literaturowych [13,14].

Tab. 7.2: Klasy modeli stosowane w odkrywaniu testowych systemów dynamicznych

Identyfikator modelu	Algorytm SVM	Funkcja jądra	ν	C	γ
M1	ν-SVR	LIN	0.54	1	—
M2	ν-SVR	RBF	0.54	1	10
M3	ν-SVR	RBF	0.54	1	1
M4	ν-SVR	RBF	0.54	1	0.1
M5	ν-SVR	RBF	0.54	1	0.01

Tab. 7.3: Plan identyfikacji testowych systemów dynamicznych

Kryterium	$acc(\,)$	ρ	System **S1**	System **S2**	System **S3**	System **S4**
AIC	MSE	—	**M1**	**M1**	**M2**, **M3**, **M4**	**M2**, **M3**, **M4**, **M5**
HF	MAPE	0.1				
HF	MAPE	0.5				
HF	MAPE	1.0				
HF	MAPE	5.0	**M1**	**M1**	—	—

W dalszej kolejności przyjęto plan odkrywania modeli symulowanych systemów dynamicznych (Tab. 7.3). W przypadku systemów liniowych **S1** i **S2** założono, że odkrywany model będzie należeć do klasy modeli liniowych **M1**. Z kolei odkrywane modele systemów nieliniowych **S3** i **S4** powinny należeć do klasy modeli opartych na radialnej funkcji jądra RBF z różnymi wartościami parametru γ. Dla każdego z systemów ustalono kryterium selekcji modeli. W szczególności dla kryterium HF przyjęto postać funkcji $acc(\,)$ oraz wartości parametru ρ. Przyjęty plan odkrywania modeli został przedstawiony w tablicy 7.3.

Parametry metody odkrywania zależności dynamicznych

Przyjęto następujące parametry opracowanej metody:

1. parametry transformacji **TSI** w **SI**:
   - $\Delta t = 1$,
   - $n = 10$,
2. parametry algorytmu genetycznego wg [3, 34]:
   - prawdopodobieństwo krzyżowania $p_c = 0.7$,
   - prawdopodobieństwo mutacji $p_m = 0.01$,
   - wielkość populacji: 100,
   - liczba iteracji algorytmu (liczba generacji): 1000,
   - sukcesja z całkowitym zastępowaniem (tzw. sukcesja trywialna).

Dla wyznaczenia wartości przyjętej funkcji oceny błędu predykcji $acc(\,)$ (Tab. 7.3) będącej częścią danego kryterium selekcji, użyto metody *Hold–out* (p. 6.3.4), dla której $\mathrm{card}(L) = \mathrm{card}(T) = 0.5\,\mathrm{card}(U)$.

7.3.6. Wyniki odkrywania testowych systemów dynamicznych

Dane stosowane w procesie identyfikacji systemów są rezultatem pomiarów prowadzonych na rzeczywistych obiektach i procesach. Oprócz tendencji charakterystycznych dla zjawisk występujących w identyfikowanym systemie, dane te zawierają również zakłócenia związane z oddziaływaniem na obserwowany obiekt/proces innych czynników. Obydwa te elementy nie są znane *a priori*. Konieczne jest zatem przyjęcie założeń, które pozwolą oddzielić część deterministyczną związaną z zachowaniem badanego systemu od części stochastycznej związanej z oddziaływaniem czynników losowych. Typowym podejściem stosowanym w takim przypadku jest przyjęcie założenia, że zakłócenia występujące w danych są zmiennymi losowymi o rozkładzie normalnym i wartość oczekiwanej równej zero. W związku z powyższym wyniki przeprowadzonych analiz uzupełniono o wyniki badań, w ramach których weryfikowano hipotezę zerową H_0, że rozkład reszt jest rozkładem normalnym przeciwko hipotezie alternatywnej, że rozkład reszt nie jest rozkładem normalnym. W tym celu zastosowano test H. Lilliefors'a [55].

Z uwagi na dość pokaźny materiał wyników częściowo zrezygnowano z prezentacji wszystkich wyników identyfikacji w postaci wykresów.

Wyniki identyfikacji systemów testowych zestawiono w tablicach: 7.4, 7.5, 7.6, 7.7. Poszczególnym kolumnom jest przypisywane następujące znaczenie:

- kolumna *Kryterium* zawiera informacje o tym jakie zostało zastosowane kryterium selekcji atrybutów (p. 6.5 pracy),
- w kolumnie *b* zestawion wartości stałej b modelu SVM (7.1).
- kolumna *nIN* oznacza liczbę wyselekcjonowanych atrybutów, na podstawie których został zbudowany model SVM (7.1),
- kolumna *nSV* oznacza liczbę wektorów wspomagających, stanowiących jeden z elementów odkrytego model SVM (7.1),
- kolumna *MSE* oznacza średniokwadratową wartość odchyleń. Wartość ta jest wyznaczania wg zależności (6.12).
- kolumna r^2 zawiera wartości współczynnika determinacji (6.16),
- w kolumnie *MAPE* zestawiono wartości średniego względnego błędu prognozy (6.18),
- kolumna *Test* t zawiera wartości statystyki *t*–Studenta obliczonych dla zbioru przykładów testowych T, których liczność $\mathrm{card}(T) = 1014$. Wartości te służą do weryfikacji hipotezy zerowej $H_0 : \mu = 0$, za pomocą której ustala się czy średnia różnica wartości wyjść modelu i testowego systemu różni się istotnie od zera. Poziom istotności α określa poziom prawdopodobieństwa, przy którym odrzuca się hipotezę zerową H_0. Wartość statystyki t przekraczająca wartość krytyczną dla poziomu istotności $\alpha = 0,05$ jest oznaczona jedną gwiazdką; poziom $\alpha = 0,01$ dwiema gwiazdkami.
- kolumna *Test* l zawiera wartości statystyki l H. Lilliefors'a wyznaczone dla reszt uzyskanych na podstawie zbioru przykładów testowych T, którego liczność $\mathrm{card}(T) = 1014$. Wartości te służą do weryfikacji hipotezy zerowej H_0, że rozkład residuów jest rozkładem normalnym przeciwko hipotezie alternatywnej, że rozkład residuów nie jest rozkładem normalnym. Poziom istotności α określa poziom prawdopodobieństwa, przy którym odrzuca się hipotezę zerową H_0. Wartość statystyki l, która przekracza wartość krytyczną dla poziomu istotności $\alpha = 0,05$ jest oznaczona jedną gwiazdką; poziom $\alpha = 0,01$ dwiema gwiazdkami.

Tab. 7.4: Wyniki identyfikacji systemu **S1**

Kryterium	b	nIN	nSV	MSE	r^2	MAPE	Test t	Test l
Model SVM: funkcja jądra: *liniowa*, parametry funkcji jądra: —								
AIC	-0.0063	5	554	$3.262\,\mathrm{E}{-2}$	0.9948	36.18	0.11	$1.24\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=0.1}$	-0.0063	5	557	$3.267\,\mathrm{E}{-2}$	0.9948	37.15	0.24	$1.57\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=0.5}$	-0.0071	3	551	$3.268\,\mathrm{E}{-2}$	0.9948	38.31	0.06	$1.44\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=1.0}$	-0.0014	3	551	$3.217\,\mathrm{E}{-2}$	0.9949	36.67	0.33	$1.48\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=5.0}$	-0.0056	2	549	$3.266\,\mathrm{E}{-2}$	0.9948	38.27	0.02	$1.40\,\mathrm{E}{-2}$

Tab. 7.5: Wyniki identyfikacji systemu **S2**

Kryterium	b	nIN	nSV	MSE	r^2	MAPE	Test t	Test l
Model SVM: funkcja jądra: *liniowa*, parametry funkcji jądra: —								
AIC	-0.2096	8	556	0.3892	0.9896	27.85	0.32	$3.24\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=0.1}$	-0.1784	7	556	0.4092	0.9890	28.06	1.21	$2.25\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=0.5}$	-0.1652	6	555	0.4204	0.9887	27.35	0.11	$1.71\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=1.0}$	-0.1697	6	557	0.4273	0.9885	27.57	0.01	$1.72\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=5.0}$	-0.2083	5	553	0.4219	0.9887	27.50	0.84	$1.93\,\mathrm{E}{-2}$

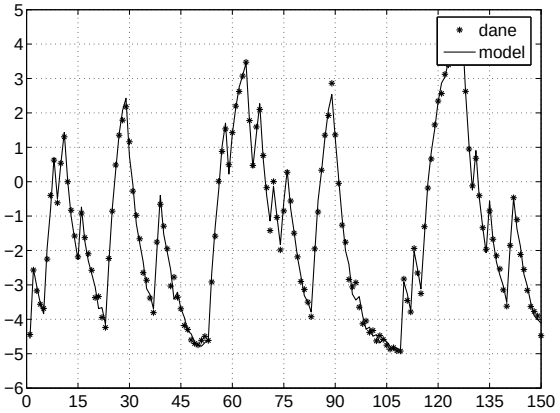
- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	-0.0056
liczba SV	:	549
zmienne wejściowe	:	$y(k-1)$, $u(k-1)$

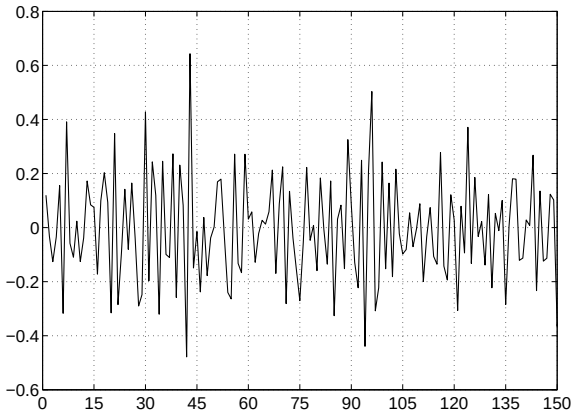
- Oceny modelu

MSE	:	$3.2659\,\mathrm{E}{-2}$
r^2	:	0.9948
MAPE	:	38.27
test t	:	0.02
test l	:	$1.44\,\mathrm{E}{-2}$

(a)



(b)



Rys. 7.5: Wyniki identyfikacji systemu **S1** (7.18) dla modeli klasy **M1** z zastosowaniem kryterium $\mathrm{HF}^{\rho=5.0}$: (a) model, (b) residum

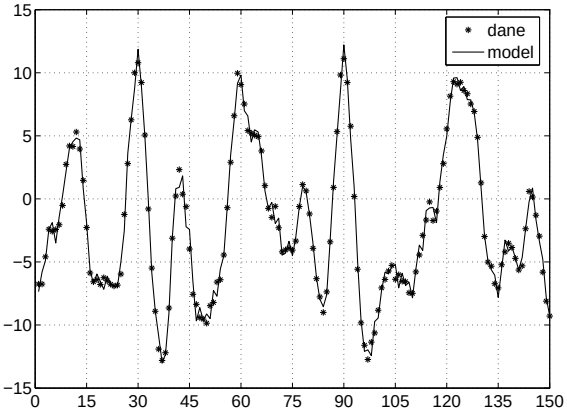
- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	-0.2083
liczba SV	:	553
zmienne wejściowe	:	$y(k-1)$, $y(k-2)$, $y(k-7)$, $u(k-2)$, $u(k-3)$

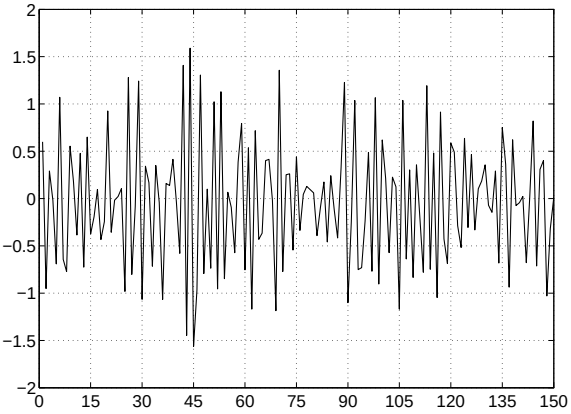
- Oceny modelu

MSE	:	0.4219
r^2	:	0.9887
MAPE	:	27.50
test t	:	0.84
test l	:	$1.71\,\mathrm{E}{-2}$

(a)



(b)



Rys. 7.6: Wyniki identyfikacji systemu **S2** (7.19) dla modeli klasy **M1** z zastosowaniem kryterium $\mathrm{HF}^{\rho=5.0}$: (a) model, (b) residum

Tab. 7.6: Wyniki identyfikacji systemu **S3**

Kryterium	b	nIN	nSV	MSE	r^2	MAPE	Test t	Test l
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 0.1$								
AIC	-1.9109	7	575	0.2005	0.9239	23.72	1.28	0.267^{**}
$\mathrm{HF}^{\rho=0.1}$	-1.8798	7	578	0.2042	0.9225	17.99	1.55	0.271^{**}
$\mathrm{HF}^{\rho=0.5}$	-1.9227	9	595	0.1883	0.9285	26.17	1.09	0.264^{**}
$\mathrm{HF}^{\rho=1.0}$	-1.9246	5	565	0.1776	0.9326	36.05	1.65	0.264^{**}
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 1$								
AIC	-1.9546	7	611	0.1132	0.9570	26.49	0.65	0.253^{**}
$\mathrm{HF}^{\rho=0.1}$	-1.9253	5	581	0.1809	0.9313	28.09	1.16	0.246^{**}
$\mathrm{HF}^{\rho=0.5}$	-1.9664	7	605	0.1387	0.9473	38.30	1.20	0.274^{**}
$\mathrm{HF}^{\rho=1.0}$	-1.9193	6	614	0.1635	0.9379	25.99	2.23^{*}	0.285^{**}
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 10$								
AIC	-1.9064	5	624	0.3147	0.8806	29.12	1.39	0.285^{**}
$\mathrm{HF}^{\rho=0.1}$	-1.9250	5	621	0.2167	0.9177	13.98	0.77	0.302^{**}
$\mathrm{HF}^{\rho=0.5}$	-1.9264	5	608	0.2480	0.9059	21.68	1.43	0.302^{**}
$\mathrm{HF}^{\rho=1.0}$	-1.8819	4	609	0.2083	0.9209	31.56	0.37	0.263^{**}

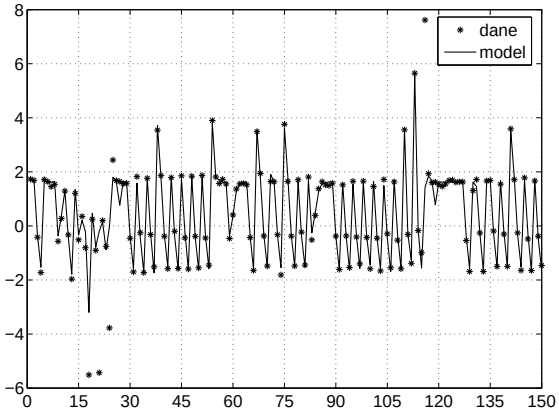
- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 10$
stała b	:	-1.9264
liczba SV	:	608
zmienne wejściowe	:	$y(k-1)$, $y(k-3)$, $u(k-2)$, $u(k-3)$, $u(k-4)$

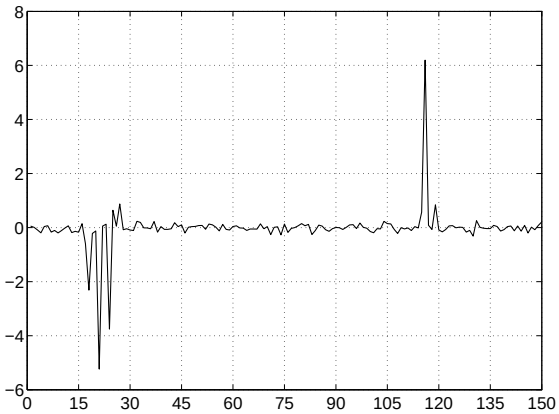
- Oceny modelu

MSE	:	0.2480
r^2	:	0.9059
MAPE	:	21.68
test t	:	1.43
test l	:	0.302^{**}

(a)



(b)



Rys. 7.7: Wyniki identyfikacji systemu **S3** (7.19) dla modeli klasy **M2** z zastosowaniem kryterium $\mathrm{HF}^{\rho=0.5}$: (a) model, (b) residum

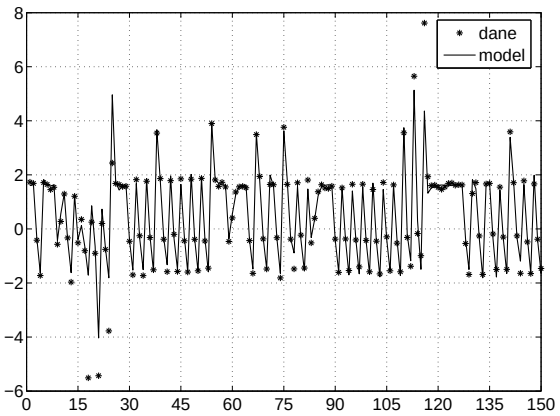
- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 1$
stała b	:	-1.9253
liczba SV	:	581
zmienne wejściowe	:	$y(k-1)$, $y(k-2)$, $y(k-3)$, $u(k-1)$, $u(k-2)$

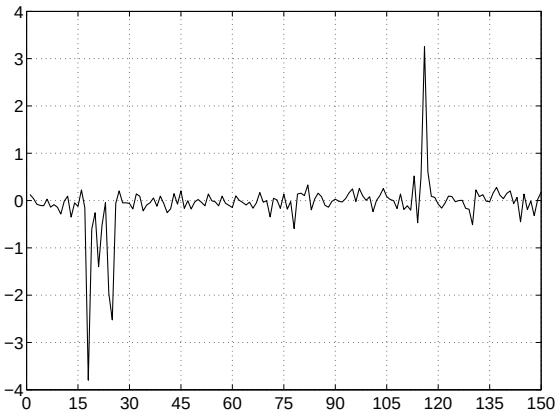
- Oceny modelu

MSE	:	0.1809
r^2	:	0.9313
MAPE	:	28.09
test t	:	1.16
test l	:	0.246^{**}

(a)



(b)



Rys. 7.8: Wyniki identyfikacji systemu **S3** (7.19) dla modeli klasy **M3** z zastosowaniem kryterium $\mathrm{HF}^{\rho=0.1}$: (a) model, (b) residum

Tab. 7.7: Wyniki identyfikacji systemu **S4**

Kryterium	b	nIN	nSV	MSE	r^2	MAPE	Test t	Test l
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 0.01$								
AIC	-1.5577	5	553	$1.837\,\mathrm{E}{-2}$	0.9537	63.93	0.81	$6.44\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=0.1}$	-1.7931	15	573	$1.428\,\mathrm{E}{-2}$	0.9640	48.17	0.40	$7.79\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=0.5}$	-1.6847	10	562	$1.562\,\mathrm{E}{-2}$	0.9606	50.00	0.48	$8.12\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=1.0}$	-1.7295	14	567	$1.489\,\mathrm{E}{-2}$	0.9624	46.59	0.32	$8.03\,\mathrm{E}{-2}^{**}$
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 0.1$								
AIC	-2.0987	9	593	$3.617\,\mathrm{E}{-3}$	0.9909	12.68	0.50	$3.69\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=0.1}$	-2.0221	13	657	$3.577\,\mathrm{E}{-3}$	0.9910	12.56	1.07	$6.35\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=0.5}$	-1.9513	11	654	$3.451\,\mathrm{E}{-3}$	0.9913	12.46	1.60	$6.40\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=1.0}$	-2.0210	12	629	$3.410\,\mathrm{E}{-3}$	0.9914	11.82	0.69	$4.34\,\mathrm{E}{-2}^{**}$
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 1$								
AIC	-1.9792	5	638	$2.018\,\mathrm{E}{-3}$	0.9949	10.33	1.44	$1.45\,\mathrm{E}{-2}$
$\mathrm{HF}^{\rho=0.1}$	-1.9812	5	593	$1.861\,\mathrm{E}{-3}$	0.9953	10.69	0.23	$2.89\,\mathrm{E}{-2}^{*}$
$\mathrm{HF}^{\rho=0.5}$	-1.9812	5	593	$1.861\,\mathrm{E}{-3}$	0.9953	10.69	0.23	$2.89\,\mathrm{E}{-2}^{*}$
$\mathrm{HF}^{\rho=1.0}$	-1.9812	5	593	$1.861\,\mathrm{E}{-3}$	0.9953	10.69	0.23	$2.89\,\mathrm{E}{-2}^{*}$
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 10$								
AIC	-1.9910	4	656	$4.592\,\mathrm{E}{-3}$	0.9884	15.07	0.59	$5.26\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=0.1}$	-1.9789	4	739	$4.770\,\mathrm{E}{-3}$	0.9880	12.98	0.10	$5.26\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=0.5}$	-1.9743	6	737	$5.402\,\mathrm{E}{-3}$	0.9864	13.34	0.25	$9.67\,\mathrm{E}{-2}^{**}$
$\mathrm{HF}^{\rho=1.0}$	-1.9774	5	738	$4.826\,\mathrm{E}{-3}$	0.9878	13.01	0.14	$8.27\,\mathrm{E}{-2}^{**}$

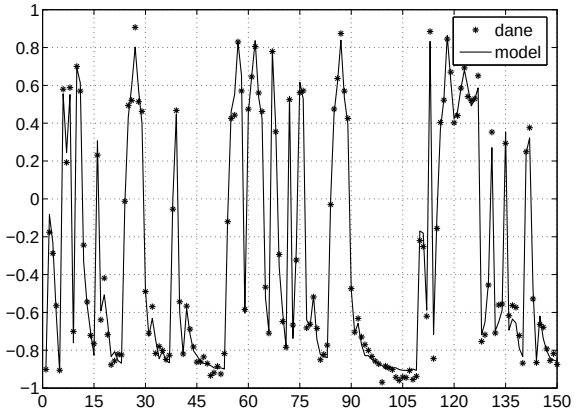
- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 1$
stała b	:	-1.9812
liczba SV	:	593
zmienne wejściowe	:	$y(k-1)$, $y(k-2)$, $y(k-3)$, $u(k-1)$, $u(k-2)$

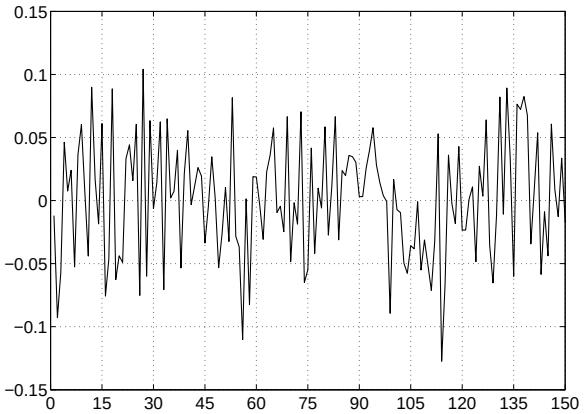
- Oceny modelu

MSE	:	$1.8613\mathrm{E}{-3}$
r^2	:	0.9953
MAPE	:	10.69
test t	:	0.23
test t	:	$2.89\,\mathrm{E}{-2}^*$

(a)



(b)



Rys. 7.9: Wyniki identyfikacji systemu **S4** (7.21) dla modeli klasy **M3** z zastosowaniem kryterium $\mathrm{HF}^{\rho=1.0}$: (a) model, (b) residum

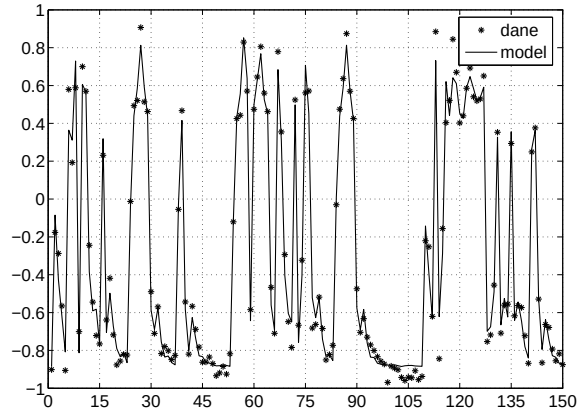
- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 0.1$
stała b	:	-2.0987
liczba SV	:	593
zmienne wejściowe	:	$y(k-2)$, $y(k-3)$, $y(k-4)$, $y(k-5)$, $u(k-1)$, $u(k-2)$, $u(k-3)$, $u(k-4)$, $u(k-7)$

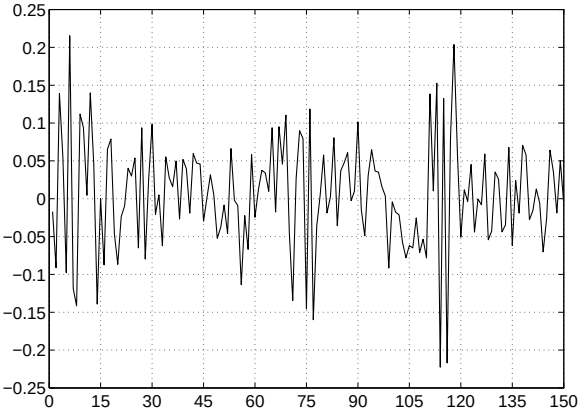
- Oceny modelu

MSE	:	$3.6169\text{E}{-3}$
r^2	:	0.9909
MAPE	:	12.68
test t	:	0.50
test t	:	$3.69\,\text{E}{-2}^{**}$

(a)



(b)



Rys. 7.10: Wyniki identyfikacji systemu **S4** (7.21) dla modeli klasy **M4** z zastosowaniem kryterium AIC: (a) model, (b) residum

7.3.7. Dyskusja wyników

Przedstawiona w rozdziale 6 metoda odkrywania modeli dynamicznych została zastosowana do identyfikacji czterech wybranych systemów dynamicznych, dwóch liniowych: **S1** (7.18) i **S2** (7.19) oraz dwóch nieliniowych: **S3** (7.20) i **S4** (7.21). W celu identyfikacji tych systemów założono 5 klas identyfikowanych modeli: **M1**, **M2**, **M3**, **M4** i **M5**. Pierwszą rozpatrywaną klasą modeli były modele liniowe. Na pozostałe klasy złożyły się modele oparte na radialnej funkcji jądra oraz następującym zbiorze wartości parametru $\gamma = \{10, 1, 0.1, 0.01\}$. Wyniki przeprowadzonych na tym etapie analiz zestawiono w punkcie 7.3.6.

Jednym z istotnych spostrzeżeń będących efektem przeprowadzonej analizy wyników jest fakt zmniejszenia we wszystkich przypadkach liczby zmiennych wejściowych użytych do budowania poszczególnych modeli. Spostrzeżenie to jest o tyle ważne, że jednym z głównych celów opracowanej metody było zmniejszenie złożoności odkrywanych modeli, która m.in. wyraża się w liczbie uwzględnianych w modelu zmiennych wejściowych. Liczba tych zmiennych jest zależna od zastosowanego kryterium selekcji.

W prowadzonych badaniach porównywano dwa kryteria: kryterium AIC oraz zaproponowaną parametryczną funkcję heurystyczną HF. Dla funkcji heurystycznej jako miarę $\mathrm{acc}()$ przyjęto wskaźnik MAPE oraz zbiór wartości parametru $\rho = \{0.1, 0.5, 1.0, 5.0\}$. Porównując modele liniowe otrzymane z zastosowaniem tych kryteriów należy zauważyć występującą prawidłowość, która przejawia się tym, że wraz ze wzrostem wartości parametru ρ odkrywane modele są modelami prostszymi pod względem liczby zmiennych wejściowych. Ponadto należy zauważyć, że wszystkie odkryte modele liniowe zawierają zmienne wejściowe, które zostały użyte do zbudowania wspomnianych wcześniej systemów liniowych.

W przypadku modeli nieliniowych przedstawiona powyżej prawidłowość nie ma miejsca. Można natomiast zauważyć, że dla malejących wartości parametru γ radialnej funkcji jądra zmniejsza się liczba wektorów wspomagających, a rośnie liczba zmiennych wejściowych uwzględnianych w modelach. Jest to zjawisko niezależne od zastosowanego kryterium selekcji. Należy je tłumaczyć właściwościami zastosowanej metody SVM oraz właściwościami funkcji radialnej. Dokonana analiza poprawności wyboru zmiennych wejściowych w przypadku odkrywania modeli nieliniowego systemu **S3** (7.20) pokazuje, że nie wszystkie z wyznaczonych modeli zawierają zmienne, jakich użyto do zbudowania systemu **S3**.

Z kolei w przypadku odkrywania modeli dla systemu **S4** (7.21) najbardziej adekwatne rezultaty uzyskano dla modeli klasy **M3** i funkcji heurystycznej. Zbiór zmiennych wejściowych dla tych modeli odpowiada dokładnie zmiennym jakich użyto do zbudowania systemu **S4**. Potwierdzają to wartości statystycznych wskaźników oceny modeli.

7.4. Weryfikacja metody dla rzeczywistej bazy danych

7.4.1. Plan weryfikacji

Plan weryfikacji dla danych dotyczących rzeczywistego obiektu technicznego obejmuje realizację następujących etapów:

1. Pozyskanie bazy danych zawierającej dane (pomiary) dotyczące obiektu technicznego istniejącego w rzeczywistym świecie.
2. Pozyskanie dodatkowych informacji o bazie danych oraz o systemie charakteryzowanym przez dane zgromadzone w bazie danych.
3. Przeprowadzenie ilościowej i jakościowej analizy bazy danych w celu określenia obszarów przydatnych dla przeprowadzenia weryfikacji metody.
4. Przygotowanie bazy danych do eksploracji obejmujące poprawienie błędnych wartości atrybutów oraz uzupełnienie wartości brakujących.
5. Przygotowanie zbiorów przykładów uczących i testowych.
6. Określenie planu identyfikacji modeli dynamicznych:
   - ustalenie wartości parametrów przekształcenia wejściowej przestrzeni atrybutów,
   - zdefiniowanie klas identyfikowanych modeli.
7. Selekcję i przekształcenie zbioru atrybutów.
8. Odkrywanie dynamicznego modelu systemu charakteryzowanego przez zgromadzone w bazie danych dane.
9. Ocenę modeli i analizę reszt.
10. Weryfikację możliwości zastosowania odkrytych modeli dla potrzeb diagnostyki.

7.4.2. Charakterystyka bazy danych

Aktualna sytuacja związana z restrukturyzacją górnictwa wymusiła potrzebę likwidacji nierentownych kopalń znajdujących się na Śląsku. Pozostałością po tych kopalniach są liczne szyby, które z różnych względów (m.in. ekonomicznych, technicznych) nie mogą zostać zlikwidowane. Część z tych szybów jest zalewana wodami podziemnymi.Stwarza to zagrożenie zalania innych (będących „*w ruchu*") kopalń. Dodatkowo zmiany poziomu wody w szybach wraz ze zmianami ciśnienia atmosferycznego powodują zmiany składu atmosfery tych szybów a także wydostawanie się na zewnątrz szkodliwych gazów (CO_2, CO). W związku z tym w szybach tych zainstalowano pompownie głębinowe, w których eksploatowane są 4 pompy o wydajnościach rzędu ok. 500 m^3/h każda. Każda z zainstalowanych pompowni jest wyposażona w system sterowania i monitorowania (klasy

SCADA) SMPG (System Sterowania i Monitorowania Pompowni Głębinowych) oraz system SMB (System Monitorowania Parametrów Bezpieczeństwa). Za pomocą tych dwóch systemów kontrolowane i mierzone są zmiany m.in. następujących parametrów:

- temperatura, wydajność, pobór mocy przez każdą z pomp wchodzących w skład pompowni,
- poziom lustra wody w szybie,
- skład atmosfery w szybie na poziomie pomostu operacyjnego,
- temperatura wewnątrz szybu jak i temperatura zewnętrzna,
- wilgotność względna powietrza na pomoście operacyjnym.

Pomiary wymienionych wielkości są prowadzone co 1 sekundę i archiwizowane w postaci plików tekstowych na twardym dysku lokalnego komputera. Równocześnie dokonywane jest przetwarzanie napływających danych przez komputer, który steruje pracą pompowni w zakresie procesu odwadniania.

Do badań udostępniono dane, które zostały zgromadzone w okresie od 16 października 2002 do 31 stycznia 2003. Dane udostępniono w celu weryfikacji możliwości pozyskania dodatkowej wiedzy o procesach zachodzących w odwadnianych szybach, jak również pozyskania wiedzy dotyczącej funkcjonowania pomp głębinowych w warunkach szybów pokopalnianych. W tablicy 7.8 zestawiono atrybuty występujące w udostępnionej bazie danych wraz z ich krótką charakterystyką. Atrybutom występującym w bazie danych przypisano nazwy odpowiadające charakterowi obserwowanych zmiennych procesowych.

Na rysunkach 7.11 i 7.12 przedstawiono zmiany wartości poszczególnych atrybutów w dniu 28 października 2002.

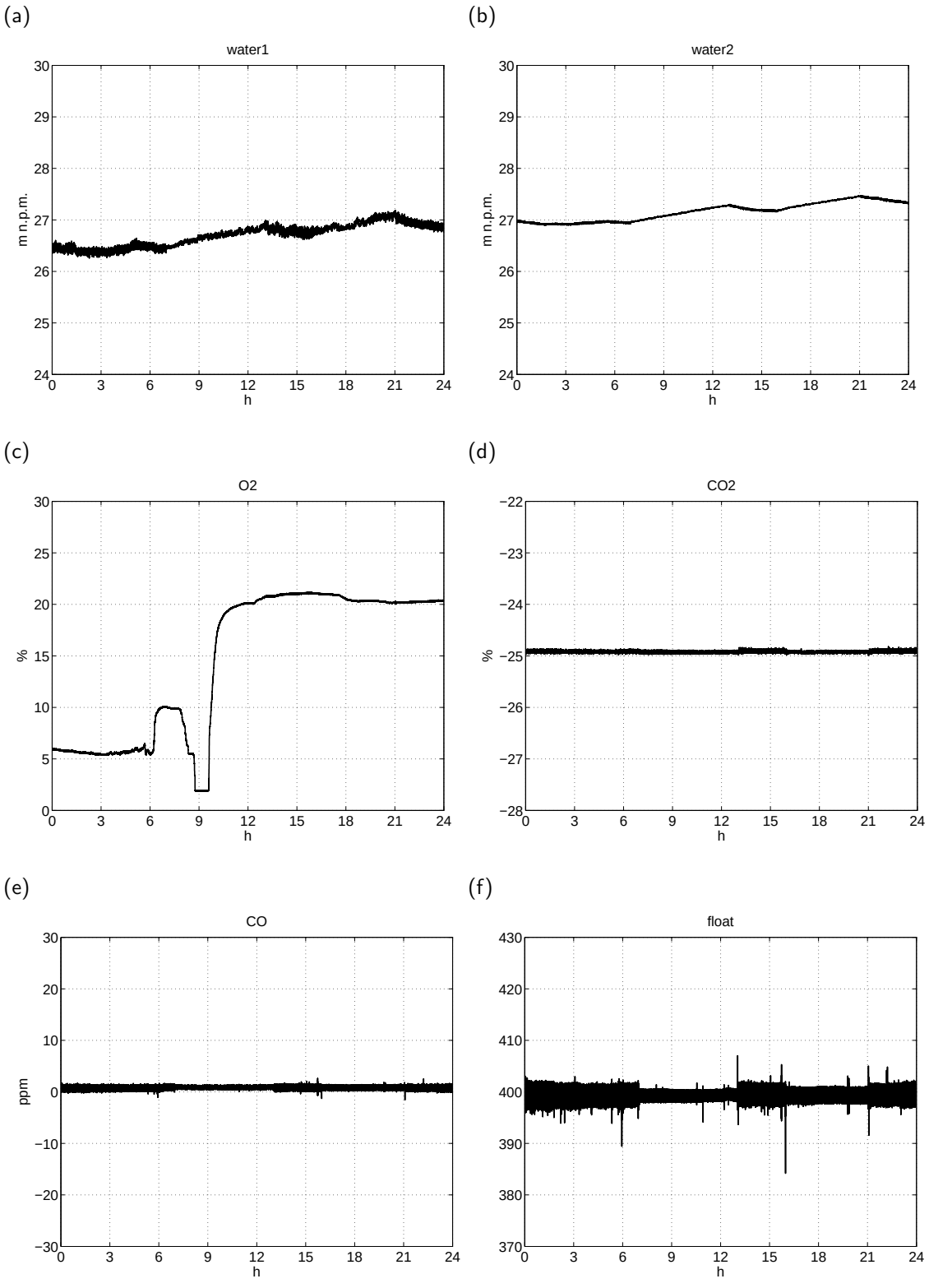
Tab. 7.8: Zestawienie atrybutów bazy danych *Pompownia Głębinowa*

Nazwa	Jednostka	Typ	Dziedzina	Opis
`date`	—	date	—	Data pomiaru.
`time`	—	time	—	Czas pomiaru.
`CO`	‰	real	$\mathbb{R}$	Stężenie tlenku węgla w szybie.
`CO2`	%	real	$\mathbb{R}$	Stężenie dwutlenku węgla w szybie.
`O2`	%	real	$\mathbb{R}$	Stężenie tlenu w szybie.
`water1`	m n.p.m.	real	$\mathbb{R}^+$	Poziom lustra wody w szybie. Mierzony jest za pomocą hydrostatycznej sondy głębokości o zakresie pomiarowym 100 m i dokładności pomiaru 3 m.
`water2`	m n.p.m.	real	$\mathbb{R}^+$	Poziom lustra wody w szybie. Mierzony jest za pomocą hydrostatycznej sondy głębokości o zakresie pomiarowym 20 m. Pomiar ma na celu kontrolę poprawności wskazań pierwszej sondy.

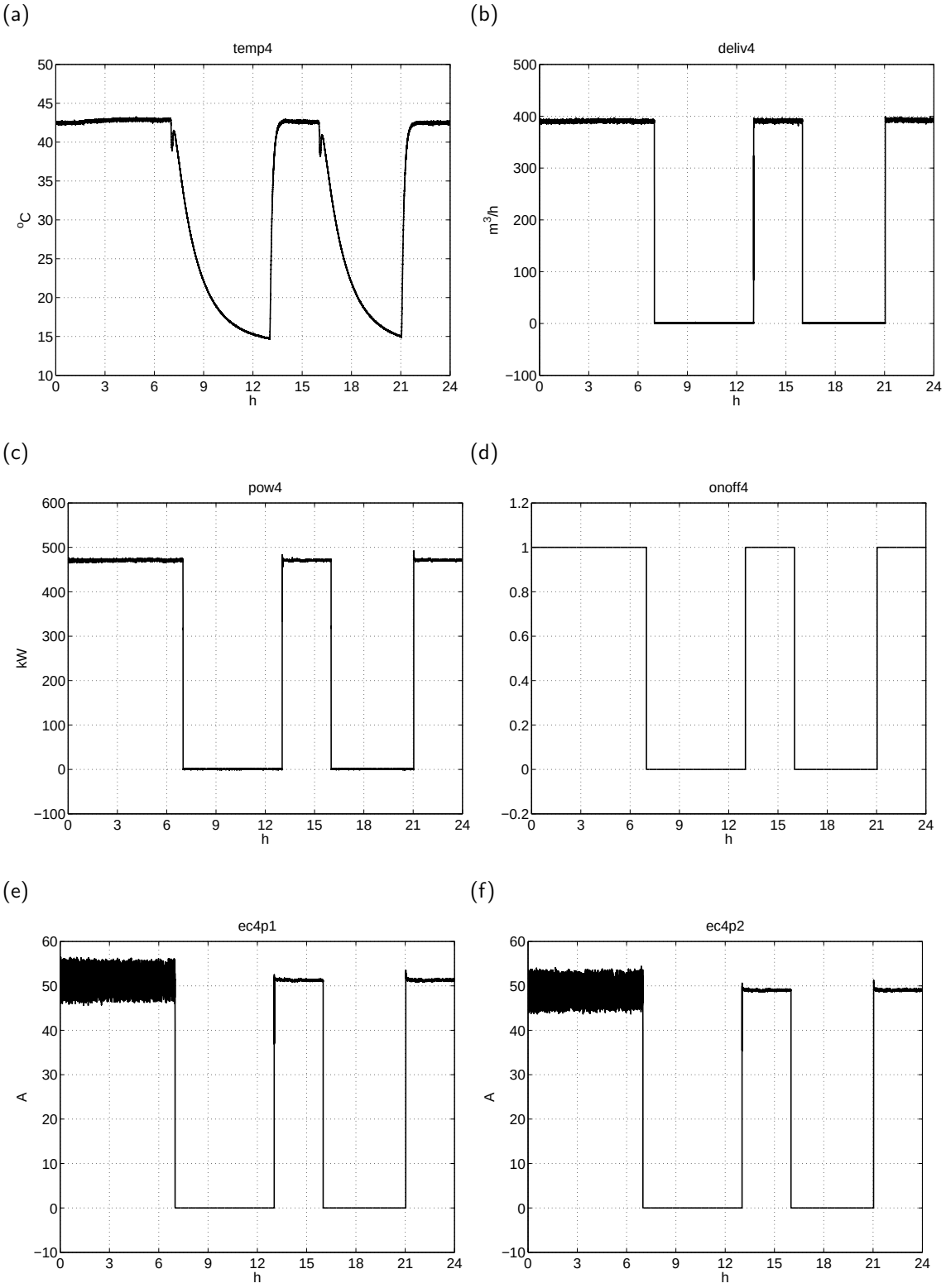
Tab. 7.8 – cd.

Nazwa	Jednostka	Typ	Dziedzina	Opis
`float`	—	real	$\mathbb{R}^+$	Atrybut (dwustanowy), za pomocą którego kontrolowane jest przekroczenie dopuszczalnego poziomu wody w szybie.
`temp1`	$^\circ\mathrm{C}$	real	$\mathbb{R}^+$	Temperatura łożyska silnika pompy nr 1.
`temp2`	$^\circ\mathrm{C}$	real	$\mathbb{R}^+$	Temperatura łożyska silnika pompy nr 2.
`temp3`	$^\circ\mathrm{C}$	real	$\mathbb{R}^+$	Temperatura łożyska silnika pompy nr 3.
`temp4`	$^\circ\mathrm{C}$	real	$\mathbb{R}^+$	Temperatura łożyska silnika pompy nr 4.
`deliv1`	m^3/h	real	$\mathbb{R}^+$	Wydajność pompy nr 1.
`deliv2`	m^3/h	real	$\mathbb{R}^+$	Wydajność pompy nr 2.
`deliv3`	m^3/h	real	$\mathbb{R}^+$	Wydajność pompy nr 3.
`deliv4`	m^3/h	real	$\mathbb{R}^+$	Wydajność pompy nr 4.
`pow1`	kW	int	$\mathbb{N}$	Moc czynna pobierana przez silnik pompy nr 1.
`pow2`	kW	int	$\mathbb{N}$	Moc czynna pobierana przez silnik pompy nr 2.
`pow3`	kW	int	$\mathbb{N}$	Moc czynna pobierana przez silnik pompy nr 3.
`pow4`	kW	int	$\mathbb{N}$	Moc czynna pobierana przez silnik pompy nr 4.
`ec1p1`	A	real	$\mathbb{R}$	Prąd pierwszej fazy na silniku pompy nr 1.
`ec2p1`	A	real	$\mathbb{R}$	Prąd pierwszej fazy na silniku pompy nr 2.
`ec3p1`	A	real	$\mathbb{R}$	Prąd pierwszej fazy na silniku pompy nr 3.
`ec4p1`	A	real	$\mathbb{R}$	Prąd pierwszej fazy na silniku pompy nr 4.
`ec1p2`	A	real	$\mathbb{R}$	Prąd drugiej fazy na silniku pompy nr 1.
`ec2p2`	A	real	$\mathbb{R}$	Prąd drugiej fazy na silniku pompy nr 2.
`ec3p2`	A	real	$\mathbb{R}$	Prąd drugiej fazy na silniku pompy nr 3.
`ec4p2`	A	real	$\mathbb{R}$	Prąd drugiej fazy na silniku pompy nr 4.
`ec1p3`	A	real	$\mathbb{R}$	Prąd trzeciej fazy na silniku pompy nr 1.
`ec2p3`	A	real	$\mathbb{R}$	Prąd trzeciej fazy na silniku pompy nr 2.
`ec3p3`	A	real	$\mathbb{R}$	Prąd trzeciej fazy na silniku pompy nr 3.
`ec4p3`	A	real	$\mathbb{R}$	Prąd trzeciej fazy na silniku pompy nr 4.
`onoff1`	—	int	$\{0;1\}$	Stan zał./wył. pompy nr 1.
`onoff2`	—	int	$\{0;1\}$	Stan zał./wył. pompy nr 2.
`onoff3`	—	int	$\{0;1\}$	Stan zał./wył. pompy nr 3.
`onoff4`	—	int	$\{0;1\}$	Stan zał./wył. pompy nr 4.

Z punktu widzenia eksploatacji i diagnostyki maszyn, w udostępnionej bazie danych najbardziej interesujący jest podzbiór atrybutów obejmujący parametry procesowe pomp głębinowych. Przyjęto, że wielkości reprezentowane za pomocą pozostałych atrybutów nie wpływają w sposób istotny na pracę pomp. Pomimo tego, grupa tych atrybutów może stanowić bazę do odkrywania np. modelu zmian składu atmosfery szybu na poziomie platformy pomiarowej w zależności od aktualnego poziomu wody w szybie i ciśnienia atmosferycznego. Tego typu model pozwoli przewidywać sytuacje, w których zmiana



Rys. 7.11: Przebiegi zmian wartości parametrów mierzonych w jednym z wyłączonych z eksploatacji szybów pokopalnianych w dniu 28.10.2002: (a) główny pomiar poziomu wody [m n.p.m.], (b) pomocniczy pomiar poziomu wody [m n.p.m.], (c) stężenie tlenu [%], (d) stężenie dwutlenku węgla [%], (e) stężenie tlenku węgla [‰], (f) dopuszczalny poziom wody w szybie



Rys. 7.12: Przebieg zmian wartości mierzonych parametrów pracy pompy głębinowej nr 4 w dniu 28.10.2002 : (a) temperatura [$^{\circ}$C], (b) wydajność [m^3/h], (c) moc [kW], (d) zał./wył., (e) prąd – I faza [A], (f) prąd – II faza [A]

składu atmosfery szybu będzie powodować zagrożenie życia obsługi przebywającej na platformie operacyjnej lub w pobliżu zrębu szybu.

Z uwagi na przyjęty zakres badań, który dotyczy pozyskania wiedzy o eksploatowanych pompach głębinowych, grupa uwzględnianych w dalszych analizach atrybutów obejmuje atrybuty charakteryzujące (Tab. 7.8):

- temperaturę pompy,
- moc pobieraną przez pompę,
- wydajność,
- prąd elektryczny zasilający pompę,
- stan pracy pompy.

W dalszej kolejności pozyskano dodatkowe informacje o pompowni jak również o sposobie jej eksploatacji i sposobie dokonywania pomiarów. Informacje te uzyskano z dwóch źródeł: wywiadu z osobami bezpośrednio związanymi z nadzorem pompowni oraz na podstawie udostępnionej dokumentacji i publikacji [32].

Ustalono, że najważniejszym parametrem eksploatacyjnym agregatu pompowego jest temperatura jego silnika. Czujnik temperatury jest fabrycznie montowany przez producenta w jednym z łożysk silnika pompy. Dokładność pomiaru temperatury wynosi $0.1^o\mathrm{C}$. Ze względu na prawidłową eksploatację, temperatura silnika pompy w stanie ustalonym nie może przekraczać temperatury granicznej równej $75^o\mathrm{C}$ (wartość ta została ustalona przez producenta). Z kolei rozruch pompy może nastąpić gdy jej temperatura będzie poniżej $25^o\mathrm{C}$. Z praktyki eksploatacyjnej wynika, że wartości temperatury poniżej $65^o\mathrm{C}$ jest typowa dla agregatu w stanie „*dobrym*", w przedziale $65^o\mathrm{C}$–$75^o\mathrm{C}$ pompa wymaga przeglądu, a powyżej $75^o\mathrm{C}$ agregat powinien zostać wymieniony. Tak więc podane zakresy temperatur stanowią podstawę do oceny ogólnego stanu technicznego pompy.

Pracujące w pompowni pompy głębinowe charakteryzują się stałą wydajnością. Regulacja ich wydajności odbywa się poprzez dławienie przepływu cieczy na rurociągu pompowym. Tak prowadzona regulacja wydajności powoduje zmianę obciążenia pompy. Należy przewidywać, że taki sposób eksploatacji będzie wpływał na trwałość pompy.

Pomiar wydajności został zrealizowany przez zastosowanie przepływomierza elektromagnetycznego, który zainstalowano na rurociągu pompowym na powierzchni. Czujnik przepływomierza został zamontowany na prostym odcinku rurociągu a przetwornik umieszczono w skrzynce przyłączeniowej, która osłania przetwornik przed działaniem środowiska. Niedokładność pomiaru przepływu wody jest rzędu 0.5 % zakresu pomiarowego.

W przypadku parametrów elektrycznych (moc, prąd elektryczny) za wyjątkiem danych zestawionych w tablicy 7.8 nie pozyskano żadnych dodatkowych informacji.

7.4.3. Przygotowanie danych do eksploracji

Przedstawiona w poprzednim punkcie pracy baza danych została udostępniona do badań w postaci zbioru plików tekstowych. Każdy z plików tekstowych zawierał wyniki pomiarów obserwowanych zmiennych procesowych w czasie jednej doby. Częstotliwość

dokonywania pomiarów przez stosowany w pompowni system akwizycji danych wynosiła 1 [Hz]. W związku z tym każdy z plików powinien zawierać dokładnie 86400 pomiarów. Jednakże wiele plików zawierało mniejszą liczbę pomiarów. Prawdopodobną przyczynę takiej sytuacji należy upatrywać w niewłaściwym funkcjonowaniu układu akwizycji danych lub w czynnikach losowych, które nie pozwoliły na zapis pomiarów.

W dalszej kolejności z uwagi na nieefektywność przetwarzania danych przechowywanych w plikach tekstowych, udostępniony zbiór plików z danymi przekształcono do formatu przechowywania danych w systemie *MATLAB*. Dodatkowo zastosowanie pakietu *MATLAB* ułatwia zarządzanie danymi.

Kolejny etap przygotowania danych obejmował zidentyfikowanie dni w rozpatrywanym okresie eksploatacji pompowni, w których liczba brakujących pomiarów była zbyt duża na to aby brakujące dane można było uzupełnić za pomocą sztucznie wygenerowanych danych. W przypadku pozostałych dni, dla których stwierdzono występowanie brakujących danych zastosowano w celu ich uzupełnienia techniki interpolacyjne i ekstrapolacyjne. Potrzeba uzupełnienia brakujących danych wynika z wymogu zachowania ciągłości szeregów czasowych.

Następna faza czyszczenia danych dotyczyła wartości znacząco różniących się od wartości obserwowanych (ang. *outliers*). Wartości te zastępowano wartościami średnimi wyznaczanymi na podstawie zbioru wartości sąsiadujących. Zbiór wartości sąsiadujących był wyznaczany na podstawie przyjętej szerokości okna czasowego.

Biorąc pod uwagę zmienność parametrów procesowych charakteryzujących pompy oraz częstotliwość z jaką dokonywano pomiarów, przetwarzanie tak dużej liczby danych nie jest uzasadnione. Dla tego przypadku przeprowadzono proces resamplingu danych z zastosowaniem techniki decymacji [57]. Przyjęto, iż w wyniku przeprowadzenia decymacji zostanie pozostawiona co sześćdziesiąta próbka zbioru *oryginalnych* zmiennych procesowych, czyli odstęp czasowy pomiędzy kolejnymi pomiarami przetworzonych zmiennych procesowych wyniesie 1 min. Dodatkowo w celu eliminacji zjawiska aliasingu [20, 57] został zastosowany dolnopasmowy filtr cyfrowy typu FIR (ang. *Finite Impulse Response*) [57] 30. rzędu. Oprócz minimalizacji zjawiska aliasingu zastosowanie filtru pozwoliło zwiększyć stosunek SNR dzięki odseparowaniu składowych wysokoczęstotliwościowych.

7.4.4. Analiza bazy danych

Na podstawie informacji zestawionych w poprzednim punkcie pracy ustalono, że jedynym istotnym z punktu widzenia diagnostyki i eksploatacji maszyn atrybutem w udostępnionej bazie danych jest temperatura agregatu pompowego. Pod kątem tego atrybutu dokonano przeglądu udostępnionych danych, które można by było zastosować do pozyskania wiedzy diagnostycznej.

Na podstawie dokonanego przeglądu stwierdzono, że stan techniczny pomp 1–3 odpowiada różnym i nie do końca zidentyfikowanym etapom okresu eksploatacyjnego. Jedynie w przypadku pompy oznaczonej numerem 4 można na podstawie przebiegu zmian tem-

peratury zaobserwować pełen cykl eksploatacyjny. Daje to możliwość zidentyfikowania modelu procesu zmian temperatury pompy w stanie „dobrym", a następnie zastosowania go do budowy modelu diagnozującego stan pompy, wykorzystującego residua. Tak więc do dalszych analiz przyjęto zbiór zmiennych procesowych dotyczących pompy nr 4, w szczególności:

- temperaturę,
- wydajność,
- moc czynną,
- prąd I fazy silnika,
- prąd II fazy silnika,
- prąd III fazy silnika,
- stan pracy pompy.

Dokonując analizy wykresów przebiegów wszystkich mierzonych parametrów (Rys. 7.12) można zauważyć, że oprócz temperatury pozostałe parametry niosą taką samą informację i są ze sobą skorelowane. Te spostrzeżenia potwierdza zamieszczona w tablicy 7.9 macierz korelacji parametrów pompy głębinowej.

Tab. 7.9: Macierz korelacji parametrów pompy głębinowej

	Temp.	Wydaj.	Moc	Prąd I	Prąd II	Prąd III	Zał./Wył.
Temp.	1.00	0.86	0.86	0.86	0.86	0.86	0.86
Wydaj.	0.86	1.00	1.00	1.00	1.00	1.00	1.00
Moc	0.86	1.00	1.00	1.00	1.00	1.00	1.00
Prąd I	0.86	1.00	1.00	1.00	1.00	1.00	1.00
Prąd II	0.86	1.00	1.00	1.00	1.00	1.00	1.00
Prąd III	0.86	1.00	1.00	1.00	1.00	1.00	1.00
Zał./Wył.	0.86	1.00	1.00	1.00	1.00	1.00	1.00

Uwzględniając przebiegi parametrów procesowych i przedstawione powyżej rozważania, zbiór rozpatrywanych zmiennych zredukowano do dwóch tj. *temperatury* i *wydajności* pompy. Zmienne te zostały zastosowane w procesie odkrywania dynamicznego systemu reprezentowanego przez udostępniony zbiór danych.

7.4.5. Plan eksploracji danych

Podobnie jak w przypadku weryfikacji z zastosowaniem danych symulacyjnych, na potrzebę odkrywania modelu zmian temperatury pompy nr 4 wybrano ten sam algorytm metody SVM (ν-SVR) oraz ustalono pięć klas modeli (Tab. 7.10). Wartości metaparametrów algorytmu ν-SVR oraz parametry funkcji jądra dobrano na podstawie zaleceń literaturowych [13, 14].

W dalszej kolejności przyjęto plan odkrywania modelu zmian temperatury pompy nr 4, oraz kryterium selekcji modeli. W szczególności dla kryterium HF ustalono postać funkcji $acc(\,)$ oceniającej dokładność predykcji modelu oraz wartości parametru ρ. Przyjęty plan odkrywania modeli został przedstawiony w tablicy 7.11.

Tab. 7.10: Definicje klas modeli stosowanych do odkrywania zależności funkcyjnych opisujących dynamikę zmian temperatury pompy głębinowej

Identyfikator klasy modelu	Algorytm SVM	Funkcja jądra	ν	C	γ
MP1	ν-SVR	LIN	0.54	1	—
MP2	ν-SVR	RBF	0.54	1	0.001
MP3	ν-SVR	RBF	0.54	1	0.01
MP4	ν-SVR	RBF	0.54	1	0.1
MP5	ν-SVR	RBF	0.54	1	1

Tab. 7.11: Plan identyfikacji modelu zmian temperatury pompy nr 4

Kryterium	$acc(\,)$	ρ	
AIC	MSE	–	**MP1**, **MP2**, **MP3**, **MP4**, **MP5**
HF	MAPE	0.1	**MP1**, **MP2**, **MP3**, **MP4**, **MP5**
HF	MAPE	0.5	**MP1**, **MP2**, **MP3**, **MP4**, **MP5**
HF	MAPE	1.0	**MP1**, **MP2**, **MP3**, **MP4**, **MP5**

W kolejnym kroku dobrano wartości następujących parametrów opracowanej metody:

1. parametry transformacji **TSI** w **SI**:
   - $\Delta t = 1$,
   - $n = 10$,
2. parametry algorytmu genetycznego wg [3, 34]:
   - prawdopodobieństwo krzyżowania $p_c = 0.7$,
   - prawdopodobieństwo mutacji $p_m = 0.01$,
   - wielkość populacji: 100,
   - liczba iteracji algorytmu (liczba generacji): 1000,
   - sukcesja z całkowitym zastępowaniem (tzw. sukcesja trywialna).

Dla wyznaczenia wartości przyjętej (Tab. 7.11) funkcji oceny błędu predykcji $acc(\,)$ będącej częścią danego kryterium selekcji, użyto metody *Hold-out*, dla której $\mathrm{card}(L) = \mathrm{card}(T) = 0.5\,\mathrm{card}(U)$. Zbiór przykładów trenujących L został przygotowany z zastosowaniem danych pomiarowych pompy pozyskanych w dniu 28.10.2002, natomiast zbiór przykładów testowych T z zastosowaniem danych z następnego dnia pomiarów. Przyjęty sposób podziału danych na dane trenujące i dane testowe wynika z kilku przesłanek.

Pierwsza przesłanka jest związana z wymogiem zachowania ciągłości i kolejności danych na pierwszym etapie eksploracji danych procesowych wg proponowanej metody. Drugi aspekt jest związany z tym, że dynamika procesu zmian temperatury pompy wynikająca z jej funkcjonowania uwidacznia się w fazach nagrzewania i studzenia, które są istotne ze względu na późniejsze zastosowanie identyfikowanego modelu. Każdą z tych faz można zaobserwować w ciągu doby dwukrotnie. Jest to efekt przyjętego sposobu eksploatacji pomp oraz czynników ekonomicznych.

W trakcie realizacji badań wstępnych ustalono, że zastosowanie danych obejmujących jeden cykl pracy pompy z fazami nagrzewania i studzenia może nie być wystarczający dla poprawnej identyfikacji modelu zmian temperatury pompy. W związku z tym zakres danych trenujących poszerzono o kolejny cykl pracy pompy. Tym samym zakres tych danych obejmuje czas jednej doby co skutkuje koniecznością przyjęcia do etapu testowania danych z następnego dnia. Opisane postępowanie jest prowadzone przy założeniu, że w czasie tych dwóch dni w pompie, otoczeniu pompy identyfikowanym za pomocą wejść i wyjść tej pompy, jaki i w układzie pomiarowym nie nastąpiły istotne zmiany stanu technicznego. Potwierdza to wizualna ocena tych danych.

7.4.6. Otrzymane wyniki analiz w przypadku badań prowadzonych dla bazy danych zawierającej wyniki obserwacji pompowni głębinowej

W wyniku dążenia autora do ograniczenia objętości opracowania zrezygnowano z prezentacji wszystkich wyników w pełnej formie, na którą składają się parametry odkrytych modeli oraz ich oceny. Podobnie jak w przypadku weryfikacji z zastosowaniem danych symulacyjnych, wszystkie wyniki ilościowe zestawiono w postaci tablicy. Poszczególnym kolumnom tej tablicy przypisywane jest takie samo znaczenie jak to wynika z opisu przedstawionego w rozdz. 7.3.6. W omawianym zakresie weryfikacji metody zbiór danych testowych obejmował 1430 przykładów. W kolumnach „*Test* t" oraz „*Test* l" oprócz wartości statystyk t-Studenta i Lilliefors'a, za pomocą odpowiedniej liczby gwiazdek oznaczono przekroczenie określonej wartości krytycznej. Dla liczby stopni swobody równej 1429 oraz poziomu istotności $\alpha = 0.01$ przekroczenie wartości krytycznej jest oznaczane jedną gwiazdką; dla poziomu istotności $\alpha = 0.05$ dwiema gwiazdkami.

Prezentację wyników weryfikacji uzupełniono o wybrane przykłady odkrytych modeli, dla których uzyskane wyniki przedstawiono w pełnej formie. Prezentowane przykłady dotyczą tych modeli, które w opinii autora ukazują cechy zaproponowanej metody oraz pozwalają dokonać właściwej oceny pod względem przydatności metody do zastosowań praktycznych.

Na prezentację wybranych modeli w pełnej formie złożyły się parametry odkrytych modeli, m.in. parametry modelu SVM, zbiór zmiennych wejściowych oraz wyznaczone z zastosowaniem zbioru przykładów testowych takie oceny jak:

- wartość odchylenia średniokwadratowego MSE,
- wartość współczynnika determinacji r^2,

- wartość średniego bezwzględnego błędu prognozy MAPE,
- wartość statystyki t,
- wartość ststystyki l,
- wykres wartości wyjścia modelu
- wykres szeregu reszt będących różnicą pomiędzy wartościami generowanymi za pomocą modelu a wartościami obserwowanymi na wyjściu systemu,
- wykres zmian wartości statystyki t dla 20 stopni swobody,
- wykres zmian wartości statystyki t dla 120 stopni swobody.

Wykresy zmian wartości statystyki t są wyznaczane zgodnie z opisem podanym w rozdz. 6.3.3.

Tab. 7.12: Wyniki odkrywania modelu zmian temperatury pompy głębinowej

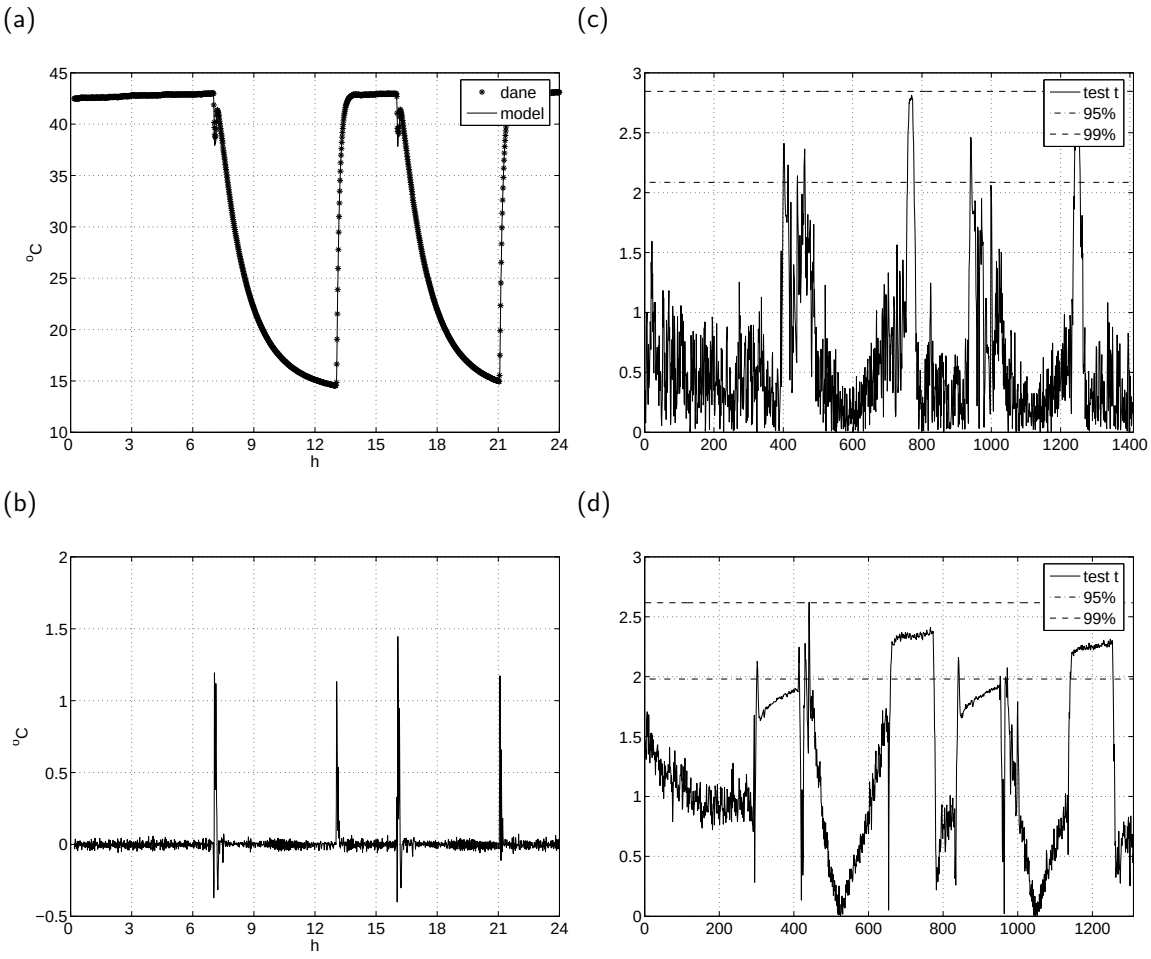
Kryterium	b	nIN	nSV	MSE	r^2	MAPE	Test t	Test l
Model SVM: funkcja jądra: *liniowa*, parametry funkcji jądra: —								
AIC	$-1.278\,\mathrm{E}{-5}$	10	787	$1.260\,\mathrm{E}{-2}$	0.9999	0.1143	3.67^{**}	0.351^{**}
$\mathrm{HF}^{\rho=0.1}$	$-1.137\,\mathrm{E}{-4}$	6	790	$1.332\,\mathrm{E}{-2}$	0.9999	0.1122	3.86^{**}	0.359^{**}
$\mathrm{HF}^{\rho=0.5}$	$7.723\,\mathrm{E}{-5}$	4	786	$3.142\,\mathrm{E}{-2}$	0.9998	0.1469	2.80^{**}	0.396^{**}
$\mathrm{HF}^{\rho=1.0}$	$3.423\,\mathrm{E}{-2}$	2	780	$1.636\,\mathrm{E}{-2}$	0.9999	0.1345	2.35^{*}	0.331^{**}
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 0.001$								
AIC	-0.6287	3	774	0.2699	0.9980	0.5697	2.48^{*}	0.395^{**}
$\mathrm{HF}^{\rho=0.1}$	-0.3494	17	777	0.9047	0.9933	0.7012	8.27^{**}	0.434^{**}
$\mathrm{HF}^{\rho=0.5}$	-0.3732	11	776	1.1669	0.9914	0.7908	7.88^{**}	0.432^{**}
$\mathrm{HF}^{\rho=1.0}$	-0.3916	9	775	0.9195	0.9932	0.7094	8.00^{**}	0.437^{**}
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 0.01$								
AIC	-0.3874	9	783	$0.556\,\mathrm{E}{-2}$	0.9999	0.1060	9.52^{**}	0.209^{**}
$\mathrm{HF}^{\rho=0.1}$	-0.3712	6	784	$1.559\,\mathrm{E}{-2}$	0.9999	0.1373	8.36^{**}	0.309^{**}
$\mathrm{HF}^{\rho=0.5}$	-0.3520	4	778	$2.571\,\mathrm{E}{-2}$	0.9998	0.1583	7.75^{**}	0.350^{**}
$\mathrm{HF}^{\rho=1.0}$	-0.3757	4	776	$5.417\,\mathrm{E}{-2}$	0.9996	0.2128	8.00^{**}	0.395^{**}
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 0.1$								
AIC	-0.5752	11	794	$1.213\,\mathrm{E}{-2}$	0.9999	0.1304	5.49^{**}	0.268^{**}
$\mathrm{HF}^{\rho=0.1}$	-0.5429	15	793	$2.200\,\mathrm{E}{-2}$	0.9998	0.1364	4.95^{**}	0.327^{**}
$\mathrm{HF}^{\rho=0.5}$	-0.5127	8	795	$2.357\,\mathrm{E}{-2}$	0.9998	0.1391	4.77^{**}	0.327^{**}
$\mathrm{HF}^{\rho=1.0}$	-0.4307	2	778	$1.218\,\mathrm{E}{-2}$	0.9999	0.1699	4.16^{**}	0.249^{**}
Model SVM: funkcja jądra: *radialna* (RBF), parametry funkcji jądra: $\gamma = 1.0$								
AIC	-0.5621	9	815	$0.480\,\mathrm{E}{-2}$	0.9999	0.0923	1.91	0.244^{**}
$\mathrm{HF}^{\rho=0.1}$	-0.5707	7	808	$0.758\,\mathrm{E}{-2}$	0.9999	0.1032	1.98^{*}	0.278^{**}
$\mathrm{HF}^{\rho=0.5}$	-0.5355	3	797	$1.768\,\mathrm{E}{-2}$	0.9999	0.1288	2.50^{*}	0.334^{**}
$\mathrm{HF}^{\rho=1.0}$	-0.5484	3	795	$2.520\,\mathrm{E}{-2}$	0.9998	0.1462	3.13^{**}	0.350^{**}

- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	$-1.2782\,\mathrm{E}{-5}$
liczba SV	:	787
zmienne wejściowe	:	$T(k-1)$, $T(k-5)$, $T(k-7)$, $T(k-8)$, $T(k-9)$, $D(k-1)$, $D(k-2)$, $D(k-5)$, $D(k-8)$, $D(k-10)$

- Oceny modelu

MSE	:	$1.2599\,\mathrm{E}{-2}$
r^2	:	0.9999
MAPE	:	0.1143
test t	:	3.67^{**}
test l	:	0.351^{**}



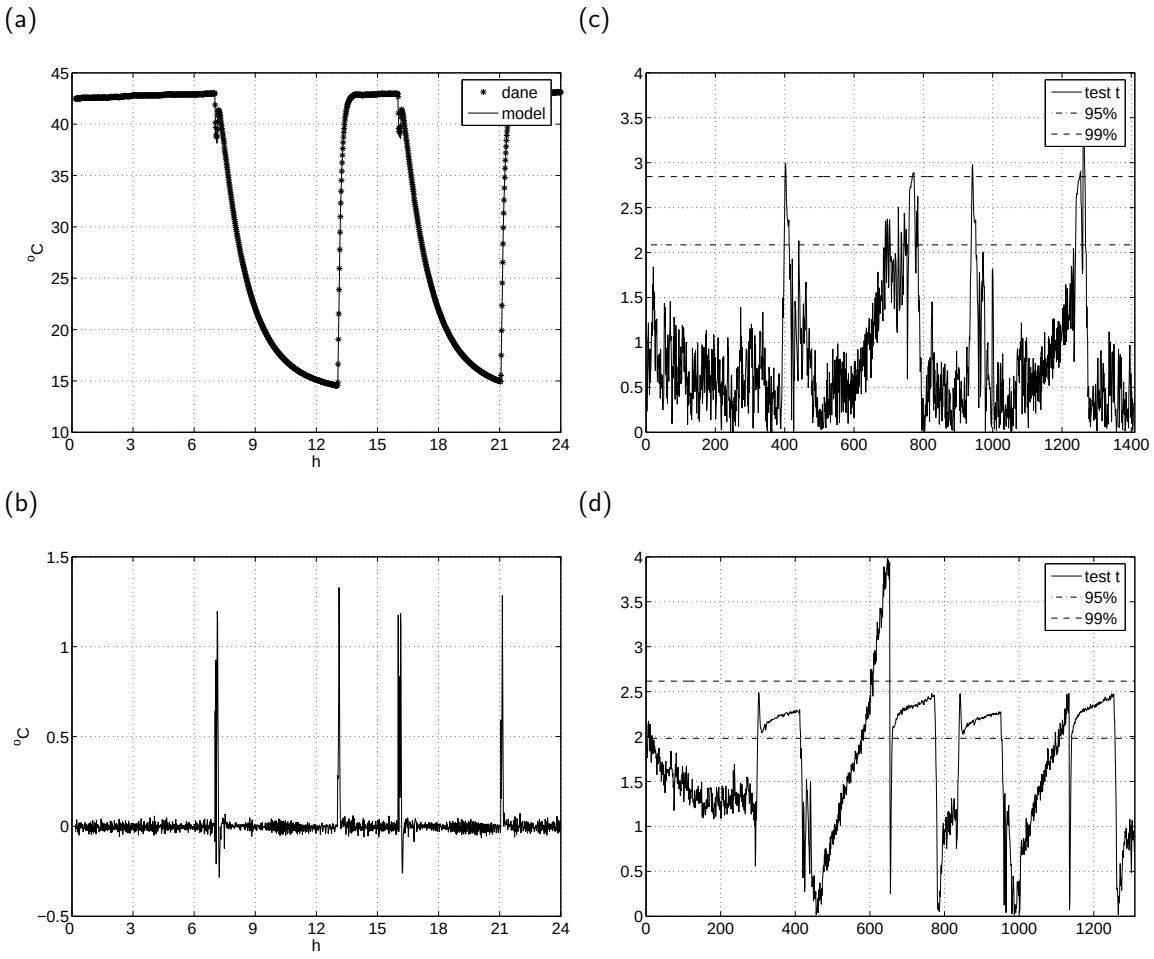
Rys. 7.13: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP1** oraz kryterium AIC: (a) model, (b) residum, (c) wartości statystyki t dla DOF=20, (d) wartości statystyki t dla DOF=120

- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	$-1.1370\,\mathrm{E}{-4}$
liczba SV	:	790
zmienne wejściowe	:	$T(k-1)$, $T(k-7)$, $T(k-10)$, $D(k-1)$, $D(k-3)$, $D(k-4)$

- Oceny modelu

MSE	:	$1.3325\,\mathrm{E}{-2}$
r^2	:	0.9999
MAPE	:	0.1143
test t	:	3.86^{**}
test l	:	0.359^{**}



Rys. 7.14: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP1** oraz kryterium $\mathrm{HF}^{\rho=0.1}$: (a) model, (b) residum, (c) wartości statystyki t dla DOF=20, (d) wartości statystyki t dla DOF=120

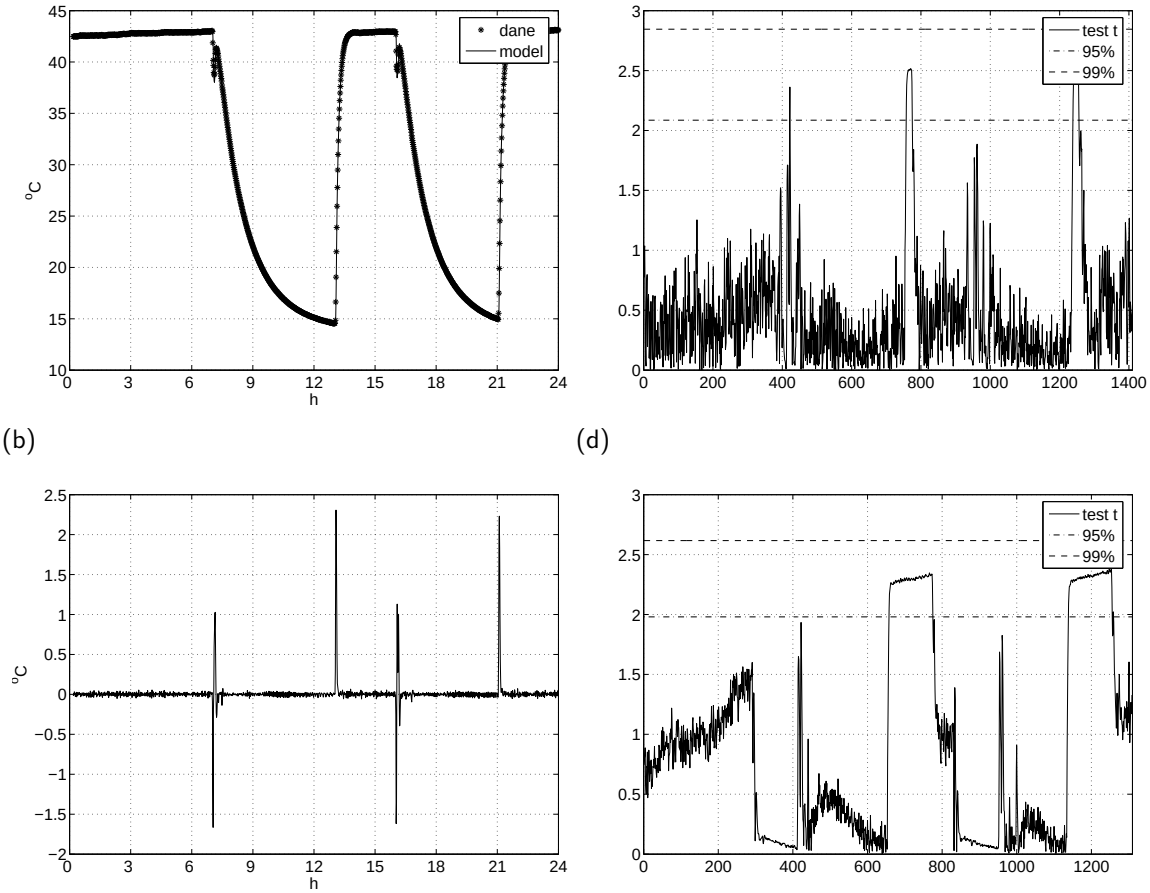
- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	$7.7230\,\mathrm{E}{-5}$
liczba SV	:	786
zmienne wejściowe	:	$T(k-1)$, $T(k-2)$, $T(k-5)$, $D(k-4)$

- Oceny modelu

MSE	:	$3.1416\,\mathrm{E}{-2}$
r^2	:	0.9998
MAPE	:	0.1469
test t	:	2.80^{**}
test l	:	0.396^{**}

(a) (c)



(b) (d)

Rys. 7.15: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP1** oraz kryterium $\mathrm{HF}^{\rho=0.5}$: (a) model, (b) residum, (c) wartości statystyki *t* dla DOF=20, (d) wartości statystyki *t* dla DOF=120

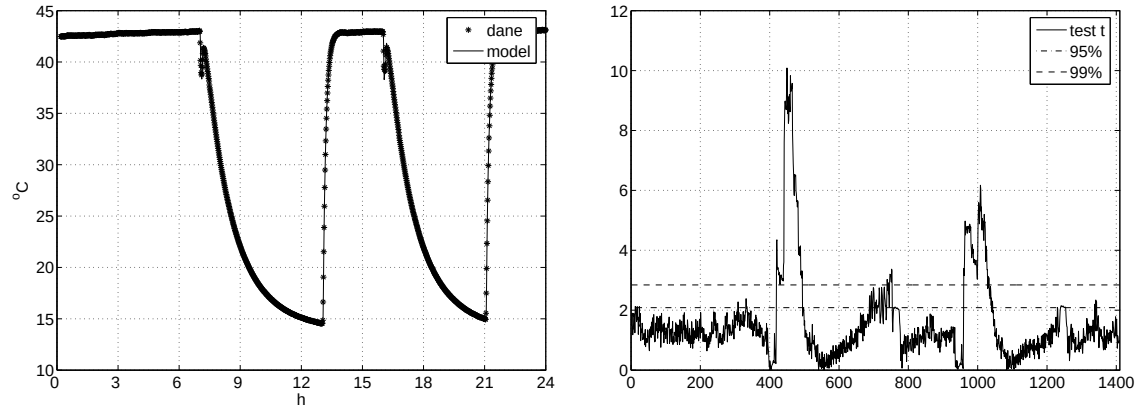
- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	$3.4231\,\mathrm{E}{-4}$
liczba SV	:	786
zmienne wejściowe	:	$T(k-1)$, $T(k-3)$

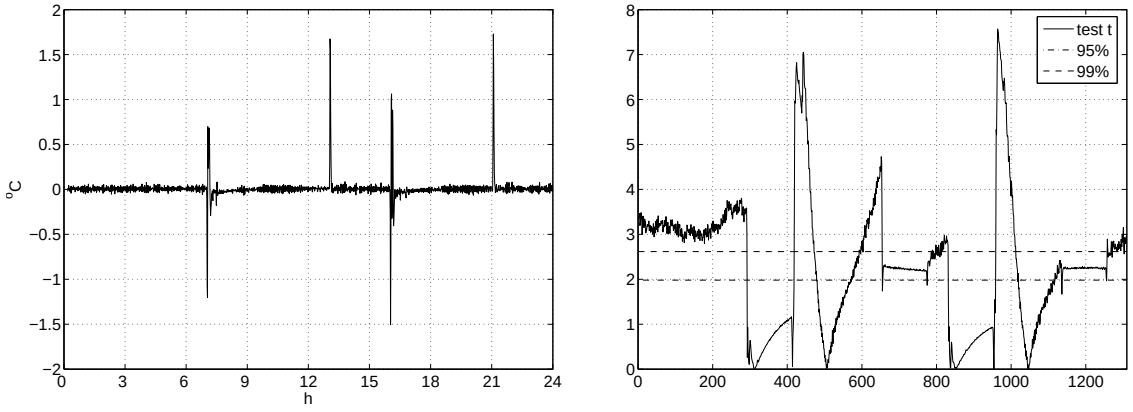
- Oceny modelu

MSE	:	$1.6362\,\mathrm{E}{-2}$
r^2	:	0.9999
MAPE	:	0.1345
test t	:	2.35^{*}
test l	:	0.331^{**}

(a) (c)



(b) (d)



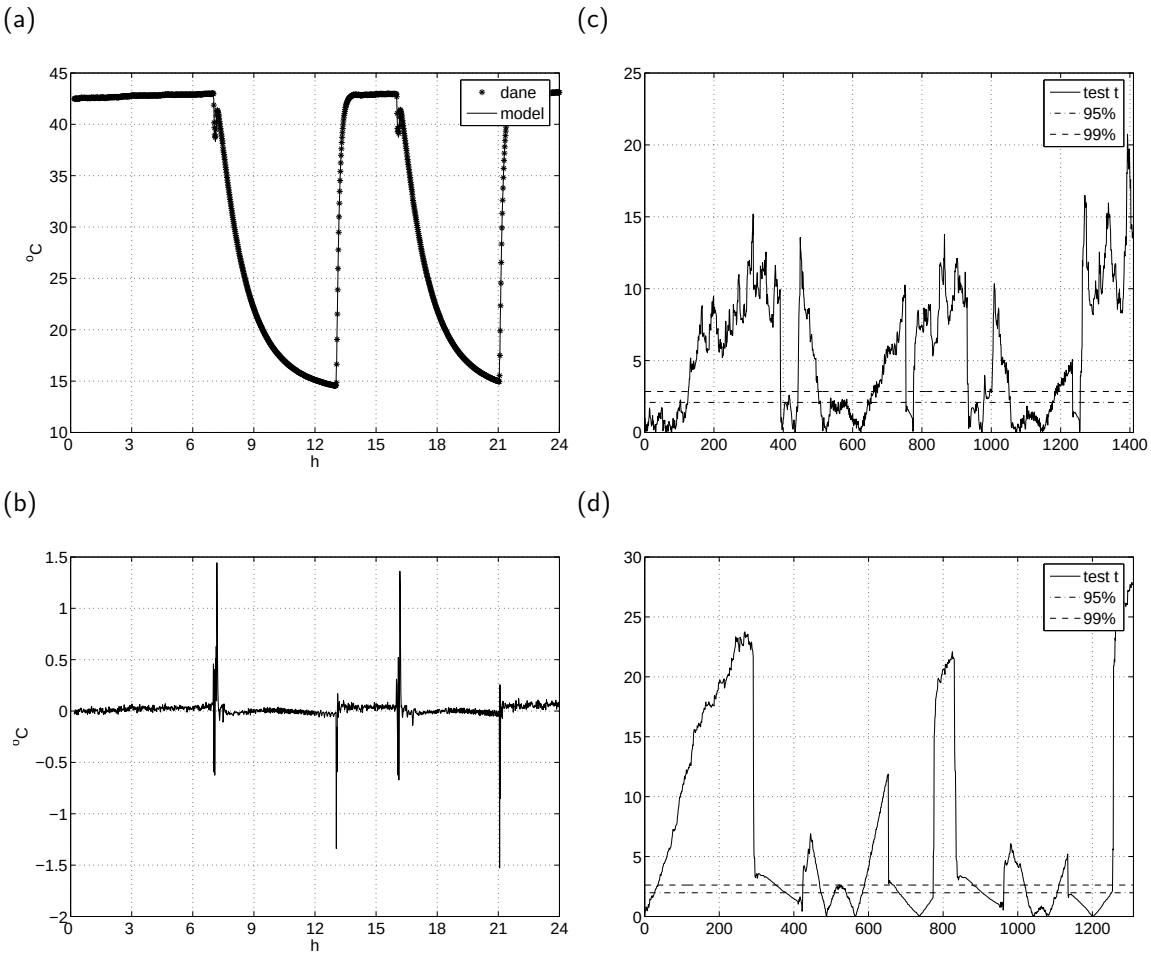
Rys. 7.16: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP1** oraz kryterium $\mathrm{HF}^{\rho=1.0}$: (a) model, (b) residum, (c) wartości statystyki t dla DOF=20, (d) wartości statystyki t dla DOF=120

- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 0.1$
stała b	:	-0.5752
liczba SV	:	794
zmienne wejściowe	:	$T(k-1)$, $T(k-2)$, $T(k-3)$, $T(k-7)$, $T(k-10)$, $D(k-0)$, $D(k-2)$, $D(k-4)$, $D(k-7)$, $D(k-9)$, $D(k-10)$

- Oceny modelu

MSE	:	$1.2132\,\mathrm{E}{-2}$
r^2	:	0.9999
MAPE	:	0.1304
test t	:	5.49^{*}
test l	:	0.209^{**}



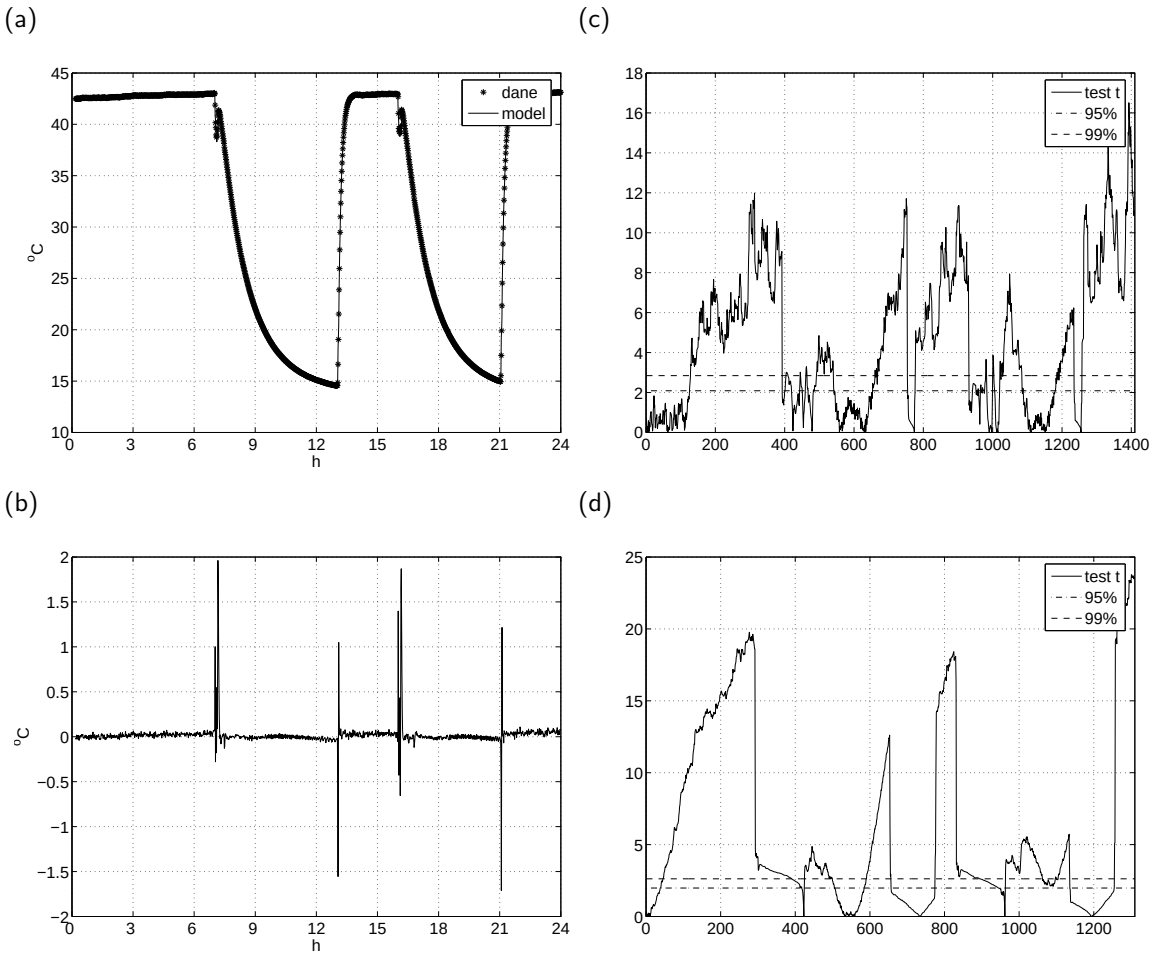
Rys. 7.17: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP4** oraz kryterium AIC: (a) model, (b) residum, (c) wartości statystyki *t* dla DOF=20, (d) wartości statystyki *t* dla DOF=120

- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 0.1$
stała b	:	-0.5429
liczba SV	:	793
zmienne wejściowe	:	$T(k-1)$, $T(k-2)$, $T(k-3)$, $T(k-4)$, $T(k-5)$, $T(k-6)$, $T(k-8)$, $T(k-9)$, $T(k-10)$, $D(k-0)$, $D(k-1)$, $D(k-4)$, $D(k-5)$, $D(k-7)$, $D(k-9)$

- Oceny modelu

MSE	:	$2.1999\,\mathrm{E}{-2}$
r^2	:	0.9998
MAPE	:	0.1364
test t	:	4.95^{*}
test l	:	0.309^{**}



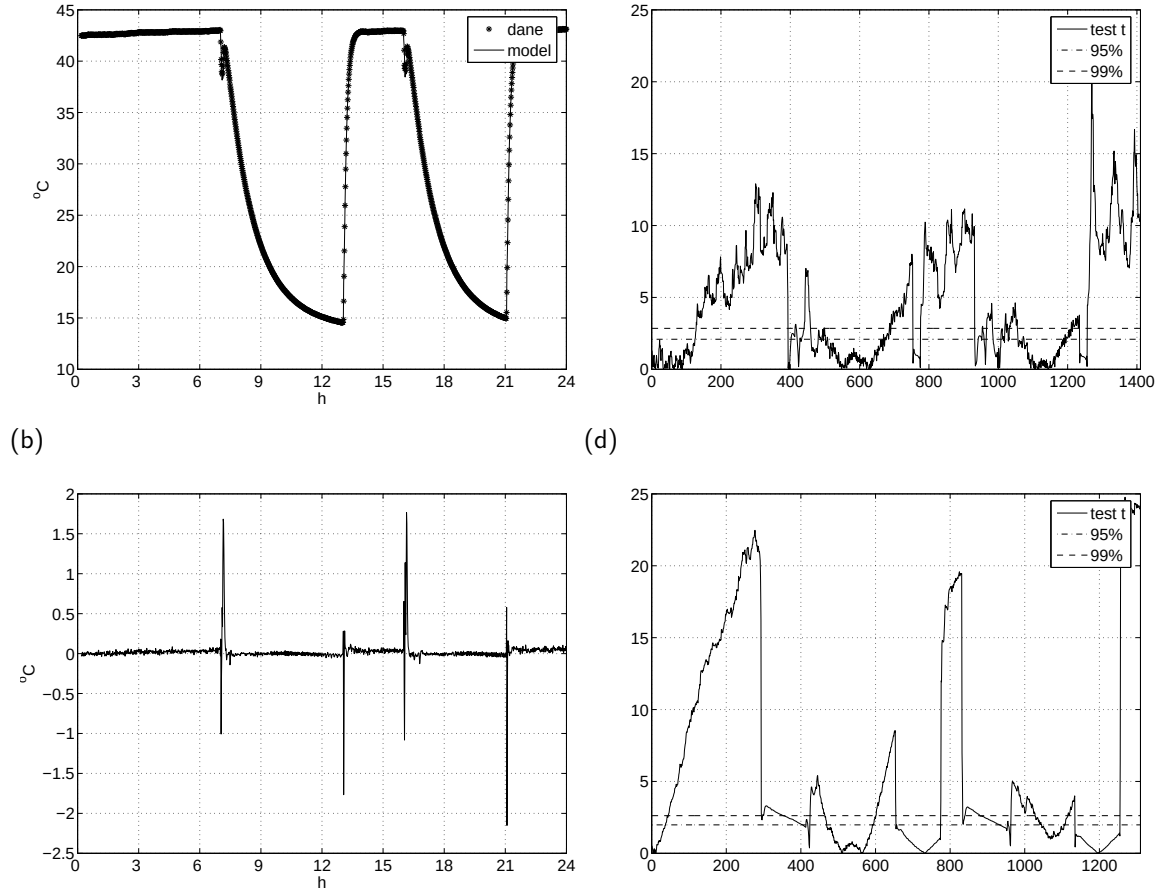
Rys. 7.18: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP4** oraz kryterium $\mathrm{HF}^{\rho=0.1}$: (a) model, (b) residum, (c) wartości statystyki *t* dla DOF=20, (d) wartości statystyki *t* dla DOF=120

- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 0.1$
stała b	:	-0.5127
liczba SV	:	795
zmienne wejściowe	:	$T(k-1)$, $T(k-2)$, $T(k-3)$, $T(k-6)$, $T(k-8)$, $T(k-9)$, $D(k-1)$, $D(k-6)$

- Oceny modelu

MSE	:	$2.3567\,\mathrm{E}{-2}$
r^2	:	0.9999
MAPE	:	0.1391
test t	:	4.77^{**}
test l	:	0.350^{**}



Rys. 7.19: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP4** oraz kryterium $\mathrm{HF}^{\rho=0.5}$: (a) model, (b) residum, (c) wartości statystyki *t* dla DOF=20, (d) wartości statystyki *t* dla DOF=120

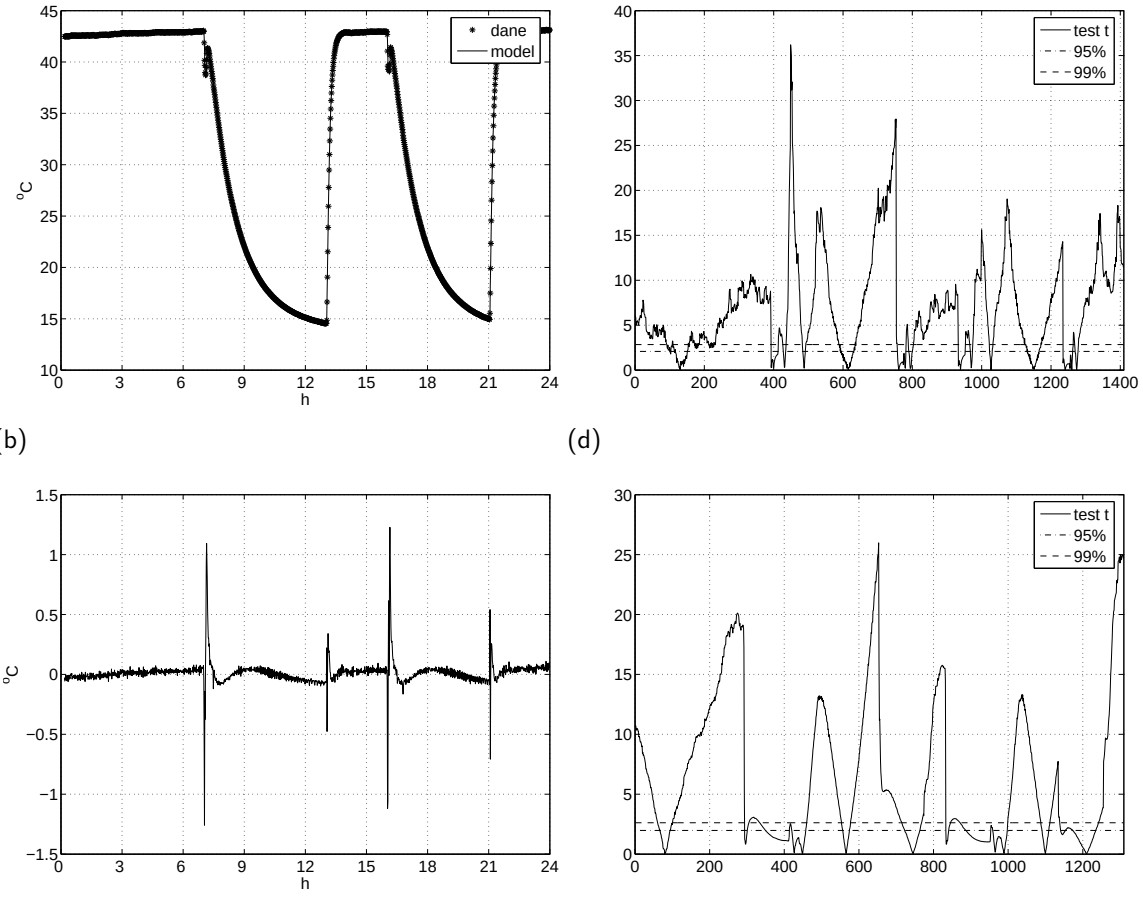
- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 0.1$
stała b	:	-0.4307
liczba SV	:	778
zmienne wejściowe	:	$T(k-1)$, $D(k-1)$

- Oceny modelu

MSE	:	$1.2184\,\mathrm{E}{-2}$
r^2	:	0.9999
MAPE	:	0.1699
test t	:	4.16^{**}
test l	:	0.395^{**}

(a) (c)



(b) (d)

Rys. 7.20: Wyniki odkrywania modelu zmian temperatury pompy nr 4 z zastosowaniem modeli klasy **MP4** oraz kryterium $\mathrm{HF}^{\rho=1.0}$: (a) model, (b) residum, (c) wartości statystyki t dla DOF=20, (d) wartości statystyki t dla DOF=120

7.4.7. Dyskusja wyników

Przedstawioną w rozdziale 6 metodę odkrywania modeli dynamicznych zastosowano do identyfikacji modelu zmian temperatury pompy głębinowej pracującej w pompowni. W celu identyfikacji modelu założono 5 klas identyfikowanych modeli **MP1**, **MP2**, **MP3**, **MP4** i **MP5**. Pierwszą z rozpatrywanych klas modeli **MP1** były modele liniowe. Na pozostałe klasy złożyły się modele oparte na radialnej funkcji jądra oraz następującym zbiorze wartości parametru $\gamma = \{0.001, 0.01, 0.1, 1\}$, który wpływa na *szerokość* funkcji jądra. Wyniki przeprowadzonych na tym etapie analiz zestawiono w rozdziale 7.3.6. Badania prowadzono stosując do selekcji modeli kryterium AIC oraz funkcję heurystyczną HF, dla której jako miarę $acc(\,)$ przyjęto wskaźnik MAPE oraz zbiór wartości parametru $\rho = \{0.1, 0.5, 1.0\}$.

Na podstawie przeprowadzonej analizy uzyskanych wyników stwierdzono, że najbardziej adekwatnymi modelami zmian temperatury pompy są modele należące do klasy modeli liniowych **MP1**. W szczególności są to modele liniowe wyznaczone z zastosowaniem kryterium AIC oraz funkcji hurystycznej z parametrem $\rho = 0.5$.

W przypadku modeli nieliniowych jedynie modele należące do klasy **MP5** cechują się podobnym jak modele liniowe stopniem redukcji liczby zmiennych wejściowych oraz wartościami wskaźników statystycznych, przy czym uzyskanie takiej dokładności powoduje zwiększenie liczby wektorów wspomagających. Powoduje to zwiększenie złożoności jak i dopasowania odkrytych modeli do danych trenujących co z kolei wpływa na zmniejszenie stopnia uogólnienia tych modeli.

W prowadzonych badaniach założono, że rozpatrywany będzie zbiór zmiennych wejściowych, dla którego maksymalne opóźnienie wynosi 10 jednostek. Zwiększenie wartości tego opóźnienia prowadzi do powiększenia przestrzeni regresorów. Wpływa to czas obliczeń, który rośnie wykładniczo.

Dla wszystkich stosowanych klas modeli przyjęcie większych wartości parametru ρ wpływa na zmniejszenie liczby zmiennych wejściowych w odkrywanych modelach.

7.4.8. Przykład zastosowania diagnostycznego

Prezentowany przykład dotyczy detekcji zmiany stanu technicznego pompy nr 4, która wchodzi w skład pompowni głębinowej. Na podstawie zebranych wcześniej informacji oraz dostępnych danych zidentyfikowano modele zmian temperatury pompy nr 4 w stanie „*dobrym*" . Pozyskane modele można zastosować do detekcji zmian stanu pompy poprzez podanie na ich wejścia danych z innych okresów cyklu eksploatacyjnego pompy, a następnie poprzez przeprowadzenie analizy uzyskanych residuów.

Formalną ocenę zmiany stanu technicznego można przeprowadzić stosując statystyczny test *t*-Studenta do weryfikacji hipotezy zerowej $H_0 : \mu = 0$, która pozwala sprawdzić czy wartość średnia szeregu reszt różni się w sposób istotny od wartości zerowej.

W celu weryfikacji hipotezy H_0 należy obliczyć wartość statystyki *t* dla szeregu residuum i porównać ją z wartościami granicznymi rozkładu *t* dla ustalonego poziomu istotności α oraz danej liczby stopni swobody. W przypadku gdy $t > t_\alpha$ oraz $t_M \leq t_\alpha$ (t_M – wartość statystyki t wyznaczona na etapie weryfikacji modelu) można uznać, że nastąpiła istotna zmiana stanu technicznego.

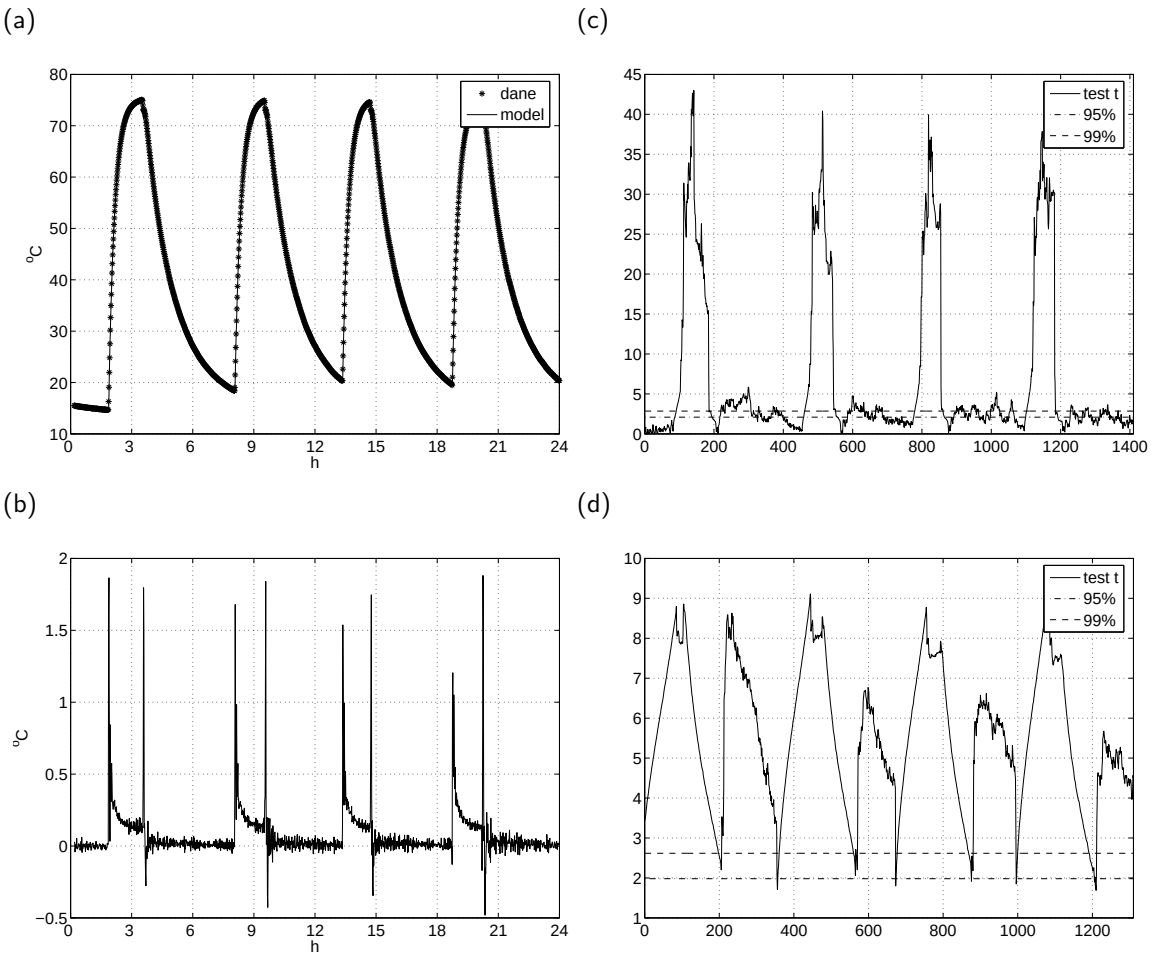
W przypadku modeli, które nie opisują dobrze pewnych obszarów modelowanego procesu hipotezę zerową H_0 należy weryfikować z zastosowaniem techniki przesuwanego *okna*. Polega ona na przesuwaniu wzdłuż szeregu residuów *okna* o szerokości odpowiadającej ustalonej liczbie stopni swobody. Dla każdego przesunięcia *okna* wyznaczana jest wartość t i porównywana z wartościami granicznymi rozkładu *t* dla ustalonego poziomu istotności α. Rysunki 7.20(c) i 7.20(d) przedstawiają wynik weryfikacji hipotezy H_0 z zastosowaniem techniki okna o szerokości odpowiadającej liczbie stopni swobody równej odpowiednio 20 i 120.

- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	$-1.2782\,\mathrm{E}{-5}$
liczba SV	:	787
zmienne wejściowe	:	$T(k-1)$, $T(k-5)$, $T(k-7)$, $T(k-8)$, $T(k-9)$, $D(k-1)$, $D(k-2)$, $D(k-5)$, $D(k-8)$, $D(k-10)$

- Oceny predykcji

MSE	:	$4.4310\,\mathrm{E}{-2}$
r^2	:	0.9999
MAPE	:	0.2171
test t	:	15.17^{***}



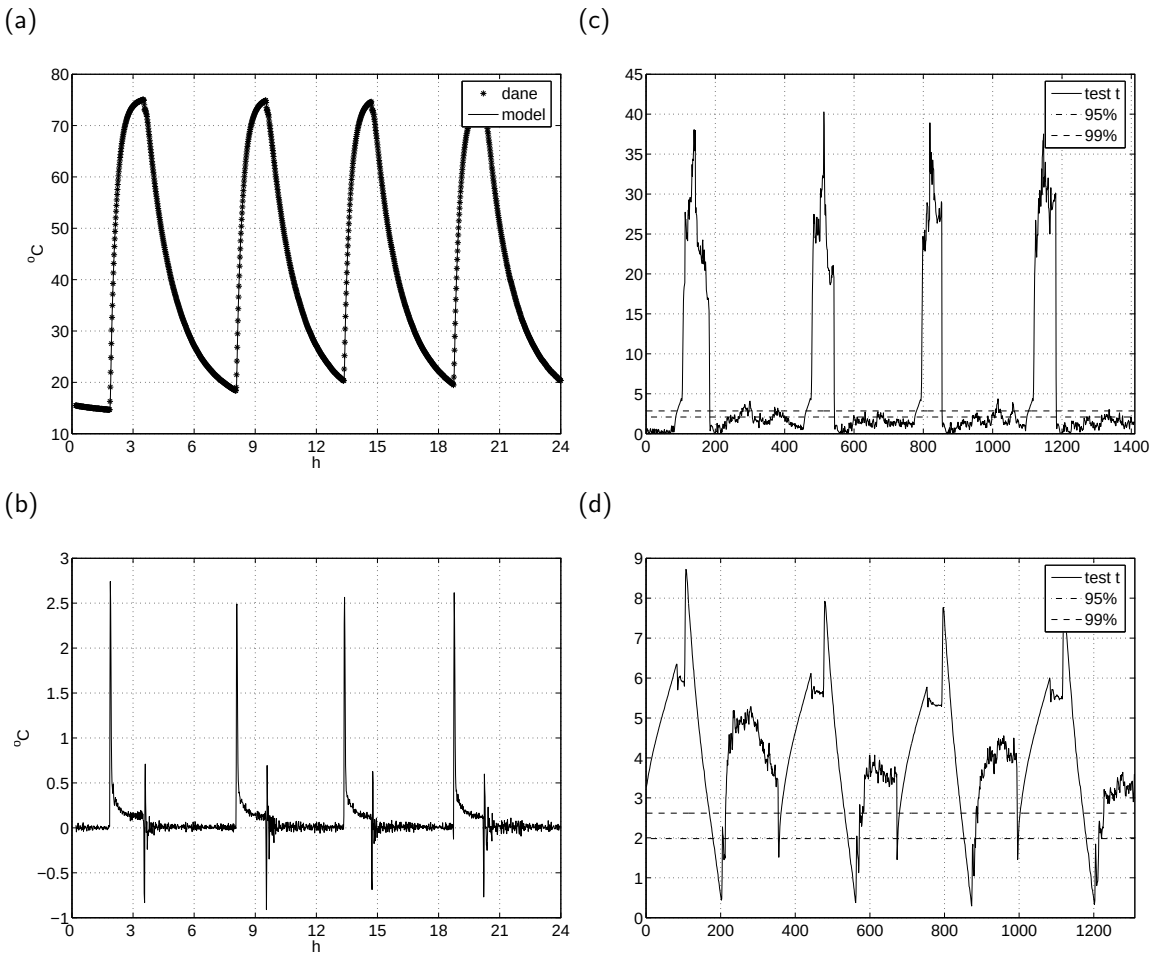
Rys. 7.21: Predykcja zmian temperatury pompy głębinowej nr 4 w końcowej fazie okresu eksploatacji z zastosowaniem liniowego modelu SVM wyznaczonego z zastosowaniem kryterium AIC: (a) model, (b) residum, (c) wartości statystyki t dla DOF=20, (d) wartości statystyki t dla DOF=120

- Model SVM

funkcja jądra	:	*liniowa*
parametry funkcji jądra	:	—
stała b	:	$7.7230\,\mathrm{E}{-5}$
liczba SV	:	786
zmienne wejściowe	:	$T(k-1)$, $T(k-2)$, $T(k-5)$, $D(k-4)$

- Oceny modelu

MSE	:	$6.9852\,\mathrm{E}{-2}$
r^2	:	0.9998
MAPE	:	0.2568
Test t	:	11.17^{***}



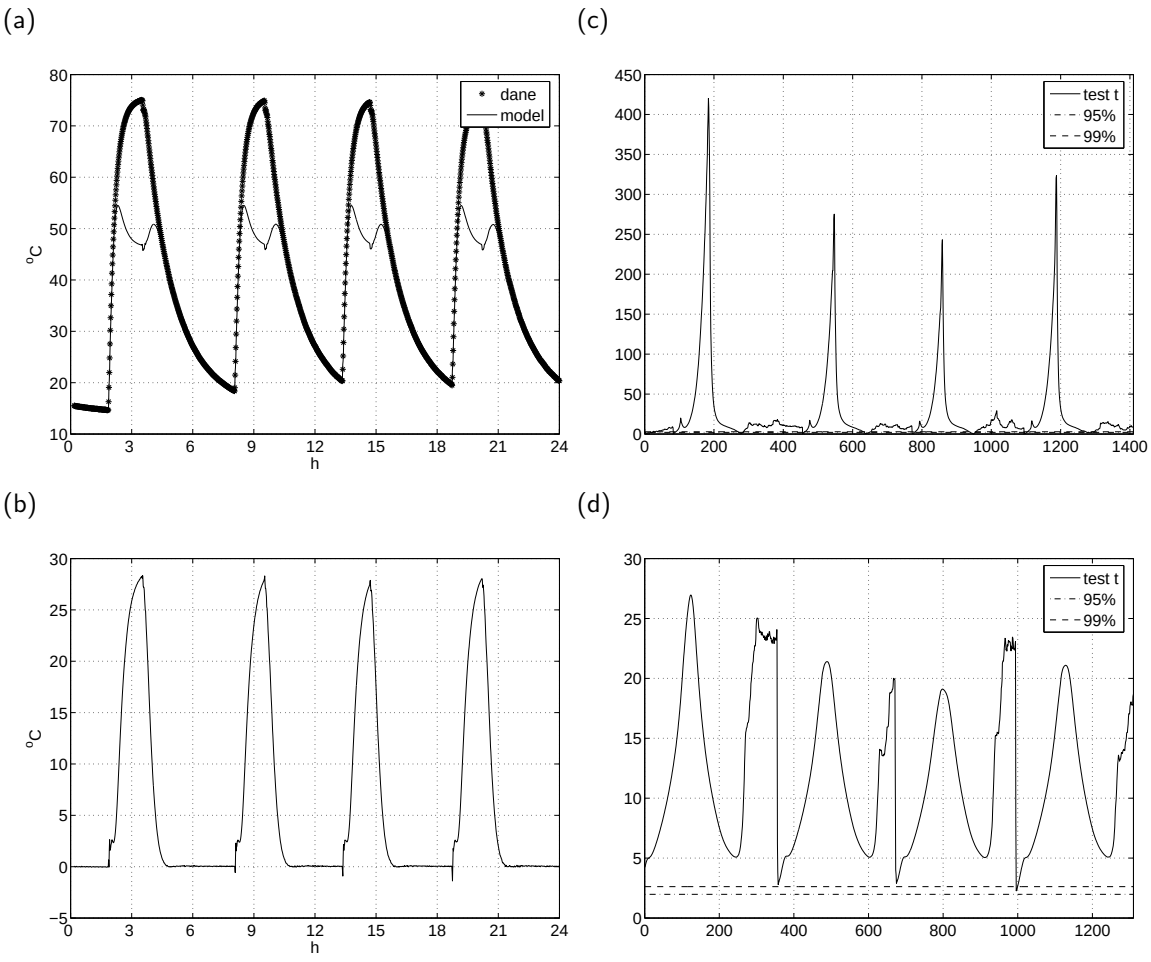
Rys. 7.22: Predykcja zmian temperatury pompy głębinowej nr 4 w końcowej fazie okresu eksploatacji z zastosowaniem liniowego modelu SVM wyznaczonego z zastosowaniem kryterium $\mathrm{HF}^{\rho=0.5}$: (a) model, (b) residum, (c) wartości statystyki t dla DOF=20, (d) wartości statystyki t dla DOF=120

- Model SVM

funkcja jądra	:	*radialna* (RBF)
parametry funkcji jądra	:	$\gamma = 0.1$
stała b	:	-0.5429
liczba SV	:	793
zmienne wejściowe	:	$T(k-1)$, $T(k-2)$, $T(k-3)$, $T(k-4)$, $T(k-5)$, $T(k-6)$, $T(k-8)$, $T(k-9)$, $T(k-10)$, $D(k-0)$, $D(k-1)$, $D(k-4)$, $D(k-5)$, $D(k-7)$, $D(k-9)$

- Oceny modelu

MSE	:	126.61
r^2	:	0.6881
MAPE	:	8.7548
Test *t*	:	23.74^{***}



Rys. 7.23: Predykcja zmian temperatury pompy głębinowej nr 4 w końcowej fazie okresu eksploatacji z zastosowaniem nielinowego modelu SVM wyznaczonego z zastosowaniem kryterium $HF^{\rho=0.1}$: (a) model, (b) residum, (c) wartości statystki *t* dla DOF=20, (d) wartości statystyki *t* dla DOF=120

7.5. Badania porównawcze

Zaproponowane przekształcenie temporalnego systemu informacyjnego w *zwykły* system informacyjny zostało użyte w trakcie badań realizowanych w ramach pracy magisterskiej mgra inż. Arkadiusza Trędy [88]. Badania te obejmowały identyfikację modeli neuronowo-rozmytych klasy TSK z zastosowaniem danych pomiarowych z pompowni głębinowej. Wśród identyfikowanych modeli znalazł się również model opisujący dynamikę zmian temperatury pompy. Wspomniane dane zostały także wykorzystane w kilku pracach zaliczeniowych realizowanych przez studentów, w ramach których identyfikowano różne typy modeli autoregresyjnych. We wszystkich przypadkach jakość uzyskanych modeli nie była zadawalająca.

7.6. Podsumowanie badań weryfikacyjnych

Weryfikację zaproponowanej metody odkrywania zależności dynamicznych przeprowadzono dla danych pozyskanych w wyniku eksperymentów numerycznych oraz pochodzących z obiektu rzeczywistego.

Celem weryfikacji z użyciem danych pozyskanych w wyniku eksperymentów numerycznych było wykazanie poprawności zaproponowanej metody w zakresie selekcji właściwego podzbioru zmiennych (atrybutów) jakie posłużyły do zbudowania testowych systemów dynamicznych. Drugi aspekt tej części badań weryfikacyjnych był związany z określeniem dokładności odkrywanych modeli.

Przeprowadzone w tym zakresie badania wykazały, że przy odpowiednim doborze parametrów proponowanej metody można uzyskać poprawne wyniki. W szczególności jest to widoczne w przypadku odkrywania modeli dla liniowych systemów testowych. W przypadku systemów nieliniowych (jak np. dla systemu S3) wyselekcjonowany zbiór zmiennych jest najczęściej nadzbiorem zawierającym wymagany zbiór zmiennych wejściowych. Taki stan rzeczy wynika z tego, że odkrywany model jest pewnym przybliżeniem systemu, co z kolei jest konsekwencją ograniczeń występujących w przyjętej metodzie indukcji zależności funkcyjnych. Autor uważa, że sytuacja w której w odkrytym modelu występuje nadzbiór zmiennych zawierający wymagane zmienne jest dopuszczalna pod warunkiem, że liczba dodatkowo uwzględnianych w modelu zmiennych nie jest zbyt duża, a ich występowanie nie wpływa w istotny sposób na dokładność przewidywanych za pomocą modelu wartości wyjściowych systemu.

W ramach drugiego etapu weryfikacji opracowanej metody badano jej przydatność w zastosowaniach praktycznych. W tym celu użyto danych, na które złożyły się wyniki pomiarów obiektu rzeczywistego jakim była jedna z pomp użytkowanych w pompowni głębinowej. Zaproponowanej metody użyto do odkrywania modelu zmian temperatury pompy.

Uzyskane wyniki pokazały, że w przedziałach czasu, w których następuje nagrzewanie pompy, występuje znaczna różnica pomiędzy wartościami wyjść wyliczonych za pomocą modeli, a wartościami zmierzonymi na obiekcie. Występujący błąd może być spowo-

dowany tym, że rozpatrzono jedynie modele SVM z liniowymi i radialnymi funkcjami jądra. Być może uwzględnienie jeszcze innych funkcji jądra w modelach SVM umożliwiłoby uzyskanie mniejszego błędu. Również brak dostępności specjalistów zajmujących się eksploatacją pomp uniemożliwił poprawę tego błędu.

Otrzymane wyniki potwierdzają możliwość użycia metod odkrywania zależności statycznych do odkrywania zależności dynamicznych w bazach danych, w których dokonano transformacji atrybutów polegającej na ich projekcji w wielowymiarową przestrzeń regresorów. Ponadto uzyskane wyniki potwierdzają drugą tezę, że selekcja atrybutów poprzez łączne zastosowanie algorytmu genetycznego i metody wektorów wspomagających pozwala odkrywać modele cechujące się mniejszą złożonością oraz dokładnością wystarczającą do zastosowań diagnostycznych.

Rozdział 8

Podsumowanie i wnioski

8.1. Podsumowanie

Przedstawiona praca jest wynikiem zainteresowań autora związanych z metodami pozyskiwania wiedzy w zakresie dziedziny jaką jest diagnostyka maszyn i procesów przemysłowych. W obszarze diagnostyki technicznej istnieje wiele baz danych, które mogą być źródłem użytecznej wiedzy diagnostycznej. Wydobycie tej wiedzy za pomocą klasycznych metod analizy danych sprawia wiele trudności, które wynikają m.in. z dużej ilości danych, ich niekompletności i niepoprawności. Dodatkowo występujące w tych danych regularności mogą zależeć od czasu. Powyższe trudności złożyły się na problem badawczy, który autor podjął się rozwiązać poprzez opracowanie metody odkrywania zależności dynamicznych bazującej na metodach odkrywania wiedzy w bazach danych. Przykłady skutecznego zastosowania tych metod w obszarze diagnostyki maszyn pozwoliły sądzić, że będą one przydatne w celu rozwiązania postawionego problemu.

Sformułowany cel rozprawy został osiągnięty poprzez:

1. szczegółowy przegląd metod stosowanych w ramach odkrywania wiedzy w bazach danych, który pozwolił określić niedostatki tych metod, jak również pozwolił wskazać metody adekwatne do rozwiązania postawionego problemu,
2. opracowanie algorytmu, za pomocą którego dokonywana jest projekcja atrybutów będących szeregami czasowymi w wielowymiarową przestrzeń regresorów,
3. dobór elementów metody selekcji atrybutów relewantnych, w szczególności przyjęcie kryterium selekcji,
4. weryfikację metody w zakresie danych pozyskanych za pomocą eksperymentów numerycznych, jak również w zakresie danych pozyskanych z rzeczywistego obiektu, która to weryfikacja pozwoliła zbadać poprawność metody oraz jej przydatność do zastosowań diagnostycznych.

Opracowana w ramach pracy metoda składa się z wielu elementów, wśród których kluczową rolę odgrywa algorytm projekcji atrybutów w wielowymiarową przestrzeń regresorów. Przekształcenie to usuwa relację określającą kolejność występowania poszczególnych wartości atrybutów, dzięki czemu możliwe jest prowadzenie analizy z zastosowaniem dowolnej metody odkrywania zależności statycznych. Z drugiej strony przekształcenie to

prowadzi do poszerzenia (czasami znacznego) przestrzeni rozpatrywanych atrybutów. Zatem konieczne stało się wprowadzenie metody selekcji atrybutów bazującej na zastosowaniu algorytmu indukcji wiedzy. Rozwiązanie to pozwala w dowolny sposób konfigurować elementy wchodzące w skład tej metody selekcji co stanowi jej główną zaletę. W opracowanej metodzie jako algorytmu indukcji użyto metody wektorów wspomagających a do przeszukiwania przestrzeni atrybutów algorytmu genetycznego.

Przeprowadzona weryfikacja potwierdziła przydatność zaproponowanej metody do odkrywania zależności dynamicznych. Należy zwrócić uwagę, że uzyskanie odpowiednich rezultatów wymaga właściwego doboru parametrów metody, jak również elementów wchodzących w jej skład. Jednocześnie należy poprawnie nakreślić cel procesu odkrywania zależności dynamicznych i stosownie do tego celu przygotować we właściwy sposób dane. Realizacja tych działań wymaga dodatkowej wiedzy dziedzinowej, jak również wiedzy dotyczącej sposobu funkcjonowania zastosowanej metody odkrywania wiedzy. Tylko wówczas uzyskane rezultaty będą zadawalające.

8.2. Wnioski

Na podstawie przeprowadzonych badań i analizy uzyskanych wyników można sformułować następujące wnioski:

1. Przeprowadzone badania pozwalają stwierdzić, że zastosowanie procesu ewolucyjnego umożliwia odkrycie modeli dynamicznych o znaczeniu diagnostycznym. Modele te mogą być stosowane do detekcji uszkodzeń w systemach diagnostyki procesów przemysłowych.
2. Spośród zastosowanych kryteriów selekcji atrybutów lepsze wyniki uzyskuje się w przypadku zaproponowanej funkcji heurystycznej. Pozwala ona uwzględnić jednocześnie dwa rozbieżne kryteria tj. kryterium prostoty modelu i minimum błędu.
3. Na podstawie przeprowadzonej analizy uzyskanych wyników stwierdzono, że najbardziej adekwatnymi modelami zmian temperatury pompy są modele należące do klasy modeli liniowych **MP1**. W szczególności są to modele liniowe wyznaczone z zastosowaniem kryterium AIC oraz funkcji hurystycznej z parametrem $\rho = 0.5$.
4. W przypadku modeli nieliniowych jedynie modele należące do klasy **MP5** cechują się podobnym jak modele liniowe stopniem redukcji liczby zmiennych wejściowych oraz wartościami wskaźników statystycznych, przy czym uzyskanie takich wyników powoduje zwiększenie liczby wektorów wspomagających.
5. W prowadzonych badaniach założono, że rozpatrywany będzie zbiór zmiennych wejściowych, dla którego maksymalne opóźnienie wynosi 10 jednostek. Zwiększenie wartości tego opóźnienia prowadzi do powiększenia przestrzeni regresorów. Wpływa to na czas obliczeń, który rośnie wykładniczo. Z drugiej strony może to powodować obniżenie dokładności modeli, zwłaszcza w takich przedziałach czasu, dla których następują nagłe (skokowe) zmiany wartości atrybutu zależnego.

8.3. Kierunki dalszych badań

Autor zamierza prowadzić dalsze badania nad rozwojem zaproponowanej metody oraz badania dotyczące odkrywania wiedzy występującej w udostępnionej bazie danych. W szczególności:

1. w zakresie metody:
    - przeprowadzić badania dotyczące możliwości zastosowania różnych metod przeszukiwania, metod odkrywania równań oraz różnych kryteriów selekcji atrybutów,
    - przeprowadzić badania w zakresie możliwości zastosowania dla metody SVM innych funkcji jądra niż funkcje liniowe i radialne, w tym podjąć próbę zbudowania systemu do automatycznego generowania funkcji jądra np. z zastosowaniem metod programowania genetycznego;
2. w zakresie analizy danych:
    - przeprowadzić badania dotyczące możliwości budowania modeli lokalnych (chodzi tu o identyfikację modelu nagrzewania się pompy),
    - podjąć badania dotyczące odkrycia modelu sterowania pompownią,
    - podjąć badania nad budową modeli diagnostycznych do określenia klasy stanu technicznego pompy z wykorzystaniem wiedzy personelu obsługującego pompownię.

Bibliografia

[1] R. Agrawal, R. Srikant. Fast algorithms for mining association rules. J. B. Bocca, M. Jarke, C. Zaniolo, redaktorzy, *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, ss. 487–499. Morgan Kaufmann, 12–15 1994.

[2] H. Akaike. A new look at the statistical model selection. *IEEE Transactions on Automatic Control*, 19:716–723, 1974.

[3] J. Arabas. *Wykłady z algorytmów ewolucyjnych*. WNT, Warszawa, 2001.

[4] J. S. Bendat, A. G. Piersol. *Metody analizy i pomiaru sygnałów losowych*. PWN, Warszawa, 1976.

[5] P. Beynon-Davies. *Systemy baz danych*. WNT, Warszawa, 1998.

[6] E. Bielińska. *Metody prognozowania*. Śląsk, Katowice, 2002.

[7] A. T. Bjorvand. Time series and rough sets. Praca magisterska, Norwegian University of Science and Technology, Trondheim, 1996.

[8] A. T. Bjorvand. Mining time series using rough sets – a case study. J. Komorowski, Żytkow J. M., redaktorzy, *Lecture Notes in Artificial Intelligence*, vol. 1263, ss. 351–358. Springer Verlag, 1997.

[9] L. Bolc, J. Cytowski. *Metody przeszukiwania heurystycznego*, vol. 1. PWN, Warszawa, 1989.

[10] L. Bolc, P. Zaremba. *Wprowadzenie do uczenia się maszyn*. Akademicka Oficyna Wydawnicza, Warszawa, 1992.

[11] G. E. P. Box, G. M. Jenkins. *Analiza szeregów czasowych. Prognozowanie i sterowanie*. PWN, Warszawa, 1983.

[12] C. Cempel. *Podstawy wibroakustycznej diagnostyki maszyn*. WNT, Warszawa, 1982.

[13] A. Chalimourda, B. Schölkopf, A. J. Smola. Experimentally optimal ν in support vector regression for different noise models and parameter settings. *IEEE Transactions on Neural Networks*, 17(1):127–141, 2004.

[14] V. Cherkassky, Y. Ma. Practical selection of svm parameters and noise estimation for svm regression. *Neural Netw.*, 17(1):113–126, 2004.

[15] A. Cholewa. *Reprezentacja sekwencji zdarzeń dla potrzeb wnioskowaniaw diagnostyce technicznej*. Praca doktorska, Politechnika Śląska : Wydział Mechaniczny Technologiczny, Gliwice, 2004.

[16] W. Cholewa. *Metoda diagnozowania maszyn z zastosowaniem zbiorów rozmytych. ZN Pol. Śląskiej nr 764*. nr 79 serii Mechanika. Politechnika Śląska, Gliwice, 1983.

[17] W. Cholewa, J. Kaźmierczak. *Diagnostyka Techniczna Maszyn. Przetwarzanie cech sygnałów. Skrypt nr 1693*. Wydawnictwo Politechniki Śląskiej, Gliwice, 1992.

[18] W. Cholewa, J. Kiciński, i in. *Diagnostyka techniczna. Odwrotne modele diagnostyczne*. Wydawnictwo Politechniki Śląskiej, Gliwice, 1997.

[19] W. Cholewa, J. Kiciński, i in. *Diagnostyka techniczna. Metody odwracania nieliniowych modeli obiektów*, vol. 120. Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska, Gliwice, 2001.

[20] W. Cholewa, W. Moczulski. *Diagnostyka Techniczna Maszyn. Pomiary i analiza sygnałów. Skrypt nr 1758*. Wydawnictwo Politechniki Śląskiej, Gliwice, 1993.

[21] P. Cichosz. *Systemy uczące się*. WNT, Warszawa, 2000.

[22] K. Ciupke. *Metoda selekcji i redukcji informacji w diagnostyce maszyn*, vol. 118. Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska, Gliwice, 2001.

[23] N. Cristianini, J. Shawe-Taylor. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000.

[24] W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz. *Sieci neuronowe*, vol. 6 serii *Biocybernetyka i inżynieria biomedyczna*. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2000.

[25] S. Džeroski, L. Todorovski. Discovering dynamics. *Proc. Tenth International Conference on Machine Learning*, ss. 97–103, San Mateo, CA, 1993. Morgan Kaufmann.

[26] S. Džeroski, L. Todorovski. Discovering dynamics. from inductive logic programming to machine discovery. *Journal of Intelligent Information Systems*, 3:1–20, 1994.

[27] S. Džeroski, L. Todorovski. Declarative bias in equation discovery. *Proc. 14th International Conference on Machine Learning*, ss. 376–384. Morgan Kaufmann, 1997.

[28] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. From data mining to knowledge discovery in databases. *AI Magazine*, 17:37–54, 1996.

[29] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy. From data mining to knowledge discovery: an overview. U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy, redaktorzy, *Advances in knowledge discovery and data mining*, ss. 1–34. AAAI Press / The MIT Press, 1996.

[30] K. D. Forbus. Qualitative physics: past, present, and future. ss. 239–296, 1988.

[31] E. Gantar. *Metody modelowania jakościowego*. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1994.

[32] M. Gibiec. Inteligentne prognozowanie stanu maszyn wykorzystywanych w przemyśle górniczym. *VI Krajowa Konferencja "Diagnostyka Procesów Przemysłowych" DPP'05, Rajgród*, ss. 243–245, Politechnika Warszawska, 2005.

[33] A. Giordana, L. Saitta, F. Bergadano, F. Brancadori, D. De Marchi. ENIGMA: A System that Learns Diagnostic Knowledge. *IEEE Trans. on Knowledge and Data Engineering*, 5(1):15–28, Feb. 1993.

[34] D. E. Goldberg. *Algorytmy genetyczne i ich zastosowania*. WNT, Warszawa, 1998.

[35] D. Hand, H. Mannila, P. Smyth. *Eksploracja danych*. WNT, Warszawa, 2005.

[36] R. Isermann, P. Ballé. Trends in the Application of Model-Based Fault-Detection and Diagnosis of Technical Processes. *Control Engineering Practice*, 5(5):709–719, 1997.

[37] N. Jankowski. *Ontogeniczne sieci neuronowe. O sieciach zmieniających swoją strukturę*. Problemy Współczesnej Nauki - Teoria i Zastosowania. Informatyka. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2003.

[38] T. Kaczorek. *Teoria sterowania i systemów*. PWN, Warszawa, 1999.

[39] A. Klimek. *Metody doskonalenia odwrotnych modeli diagnostycznych*, vol. 134 serii *Mechanika*. Politechnika Śląska, Gliwice, 1999.

[40] R. Kohavi. *Wrappers for Permormance Enhancement and Oblivious Decision Graphs*. Praca doktorska, Department of Computer Science, Stanford University, Stanford, 1995.

[41] R. Kohavi, G. H. John. Wrappers for feature subset selection. *Artificial Intelligence*, 97(1-2):273–324, 1997.

[42] J. Korbicz. *Sztuczne sieci neuronowe*. Akademicka Oficyna Wydawnicza, Warszawa, 1994.

[43] J. Korbicz, J. M. Kościelny, Z. Kowalczuk, W. Cholewa, i in. *Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania*, vol. 3 serii *Diagnostyka procesów*. WNT, Warszawa, 2002.

[44] J. Korbicz, A. Obuchowicz, D. Uciński. *Sztuczne sieci neuronowe. Podstawy i zastosowania*. Problemy Współczesnej Nauki - Teoria i Zastosowania. Informatyka. Akademicka Oficyna Wydawnicza PLJ, Warszawa, 1994.

[45] J. Koronacki, J. Mielniczuk. *Statystyka dla studentów kierunków technicznych i przyrodniczych*. WNT, Warszawa, 2001.

[46] J. M. Kościelny. *Diagnostyka zautomatyzowanych procesów przemysłowych*. Problemy Współczesnej Nauki - Teoria i Zastosowania. Automatyka. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 2001.

[47] K. Krawiec, J. Stefanowski. *Uczenie maszynowe i sieci neuronalne*. Wydawnictwo Politechniki Poznańskiej, Poznań, 2003.

[48] B. Kuipers. Qualitative simulation. *Artificial Intelligence*, 26:289–338, 1986. Reprinted in Qualitative Reasoning about Physical Systems, ed. Daniel Weld and J. De Kleer, Morgan Kaufmann, 1990, p.236-260.

[49] B. Kuipers. *Qualitative Reasoning: Modeling and Simulation with Incomplete Knowledge*. MIT Press, Cambridge, MA, 1994.

[50] P. Langley. Bacon 1. a general discovery system. *The Second National Conference of the Canadian Society for Computational Studies of Intelligence*, ss. 173–180, 1978.

[51] P. Langley. Rediscovering physics with bacon 3. *The Sixth International Joint Conference on Artificial Intelligence*, ss. 505–507, 1979.

[52] P. Langley, G. Bradshaw, H. A. Simon. Bacon 4. the discovery of intrinsic properties. *The Third National Conference of the Canadian Society for Computational Studies of Intelligence*, ss. 19–25, 1980.

[53] P. Langley, G. Bradshaw, H. A. Simon. Bacon 5. the discovery of conservation laws. *The Seventh International Joint Conference on Artificial Intelligence*, ss. 121–126, 1981.

[54] P. Langley, H. A. Simon, G. L. Bradshaw, J. M. Żytkow. *Scientific Discovery. Computational Explorations of the Creative Processes*. MA: MIT Press, Cambridge, 1987.

[55] H. Lilliefors. On the kolmogorov-smirnov test for normality with mean and variance unknown. *Journal of the American Statistical Association*, 1967.

[56] L. Ljung. *System Identification Toolbox - for use with MATLAB. User's Guide*. The Mathworks, Inc., 2000.

[57] R. G. Lyons. *Wprowadzenie do cyfrowego przetwarzania sygnałów*. WKŁ, Warszawa, 2000.

[58] A. Marciniak, J. Korbicz, J. Kuś. Wstępne przetwarzanie danych. W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz, redaktorzy, *Sieci neuronowe*, vol. 6 serii *Biocybernetyka i Inżynieria Biomedyczna 2000*, ss. 29–71. Wydawnictwo EXIT, Warszawa, 2000.

[59] Z. Michalewicz. *Algorytmy genetyczne + struktury danych = programy ewolucyjne*. WNT, Warszawa, 1999.

[60] W. Moczulski. *Metody pozyskiwania wiedzy dla potrzeb diagnostyki maszyn*, vol. Zeszyt 130 serii *Zeszyty Naukowe Politechniki Śląskiej*. Wydawnictwo Politechniki Śląskiej, Gliwice, 1997.

[61] W. Moczulski. *Diagnostyka techniczna. Metody pozyskiwania wiedzy*. Wydawnictwo Politechniki Śląskiej, Gliwice, 2001.

[62] W. Moczulski, P. Kostka. Spie discovery challenge data description. Raport, Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska, 2000.

[63] W. Moczulski, P. Tomasik, D. Wachla. Odkrywanie wiedzy w diagnostycznych bazach danych. *XXIX Og olnopolskie Sympozjum "Diagnostyka Maszyn"*, ss. 261–272, Węgierska Górka, 4-9.03. 2002.

[64] W. Moczulski, D. Wachla. Acquisition of Diagnostic Knowledge Using Discoveries in Databases. *V National Conference "Diagnostics of Industrial Processes" DPP'01, Łagów (Poland)*, ss. 219–224, Technical University of Zielona Góra, 2001.

[65] W. Moczulski, J. M. Żytkow. Discovery of diagnostic knowledge from multi-sensor data. *AeroSense - SPIE's 15th International Symposium on Aerospace/Defense Sensing, Simulation, and Controls, "Data Mining and Knowledge Discovery: Theory, Tools, and Technology III"*, vol. 4384, ss. 104–115, Orlando, FL, 16-20.04. 2001. Proceedings of SPIE.

[66] J. Mulawka. *Systemy ekspertowe*. WNT, Warszawa, 1996.

[67] K. Narendra, K. Parthasarathy. Identification and control of dynamical systems using neural networks. *IEEE Transactions on Neural Networks*, 1:4–27, 1990.

[68] K. Niijima, H. Uchida, E. Hirowatari, S. Arikawa. Discovery of differential equations from numerical data. *Discovery Science*, ss. 364–374, 1998.

[69] A. Obuchowicz, J. Korbicz. Metody ewolucyjne w projektowaniu systemów diagnostycznych. *Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania*, vol. 3 serii *Diagnostyka Procesów*, ss. 279–309. WNT, Warszawa, 2002.

[70] W. Oktaba. *Elementy statystyki matematycznej i metodyka doświadczalnictwa*. PWN, Warszawa, 1980.

[71] Z. Pawlak. Rough Sets. *International Journal of Information and Computer Sciences*, 11(5):341–356, 1982.

[72] Z. Pawlak. *Rough Sets. Theoretical Aspects of Reasoning About Data*. Kluwer Academic Publishers, 1991.

[73] A. Pieczyński. *Komputerowe systemy diagnostyczne procesów przemysłowych*. Politechnika Zielonogórska, Zielona Góra, 1999.

[74] A. Pieczyński. *Reprezentacja wiedzy w diagnostycznym systemie ekspertowym*. Lubuskie Towarzystwo Naukowe, Zielona Góra, 2003.

[75] A. Piegat. *Modelowanie i sterowanie rozmyte*. Problemy Współczesnej Nauki - Teoria i Zastosowania. Informatyka. Akademicka Oficyna Wydawnicza EXIT, Warszawa, 1998.

[76] W. H. Press, S. A. Teukolsky, W. T. Vetterling, B. P. Flannery. *Numerical Recipes in C*. Cambridge University Press, Cambridge, MA, 1992.

[77] B. Schölkopf, P. L. Bartlett, A. J. Smola, R. Williamson. Shrinking the tube: a new support vector regression algorithm. vol. 11, ss. 330 – 336, Cambridge, MA, 1999. MIT Press.

[78] B. Schölkopf, A. J. Smola. *Learning with Kernels: Support Vector Machines, Regularization, Optimization, and Beyond*. MIT Press, Cambridge, MA, USA, 2001.

[79] A. Siudak. Symulacja jakościowego modelu wybranego obiektu technicznego. Praca magisterska, Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska, Gliwice, 2005.

[80] A. J. Smola, B. Schölkopf. A tutorial on support vector regression. *Statistics and Computing*, 14(3):199–222, 2004.

[81] T. Söderström, P. Stoica. *Identyfikacja systemów*. PWN, Warszawa, 1997.

[82] R. Szulim. *Metoda pozyskiwania wiedzy do wspomagania prowadzenia złożonego procesu technologicznego*. Praca doktorska, Uniwersytet Zielonogórski: Wydział Elektroniki, Informatyki i Telekomunikacji, Zielona Góra, 2004.

[83] R. Tadeusiewicz. *Sieci neuronowe*. Problemy Współczesnej Nauki i Techniki. Informatyka. Akademicka Oficyna Wydawnicza RM, Warszawa, 1993.

[84] R. Tadeusiewicz, P. Lula. Neuronowe metody analizy szeregów czasowych i możliwości ich zastosowań w zagadnieniach biomedycznych. W. Duch, J. Korbicz, L. Rutkowski, R. Tadeusiewicz, redaktorzy, *Sieci neuronowe*, vol. 6 serii *Biocybernetyka i Inżynieria Biomedyczna 2000*, ss. 521–568. Wydawnictwo EXIT, Warszawa, 2000.

[85] P. N. Tan, M. Steinbach, V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.

[86] L. Todorovski. Declarative bias in equation discovery, 1993.

[87] L. Todorovski. *Using Domain Knowledge for Automated modeling of Dynamic Systems with Equation Discovery*. Praca doktorska, Faculty of Computer and Information Science, University of Ljubljana, Lubljana, Slovenia, 2003.

[88] A. Tręda. Model przybliżony procesu dynamicznego dla potrzeb diagnozowania. Praca magisterska, Katedra Podstaw Konstrukcji Maszyn, Politechnika Śląska, Gliwice, 2005.

[89] T. Uhl. *Komputerowo wspomagana identyfikacja modeli konstrukcji mechanicznych*. CAD/CAM. WNT, Warszawa, 1997.

[90] J. D. Ullman, W. J. *Podstawowy wykład z systemów baz danych*. WNT, Warszawa, 2001.

[91] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, New York, 1995.

[92] V. Verdult. *Nonlinear system identification. A State-Space Approach*. Praca doktorska, Univeristy of Twente, Twente, Netherlands, 2002.

[93] W. Volk. *Statystyka stosowana dla inżynierów*. WNT, Warszawa, 1973.

[94] D. Wachla. The idea of method of searching for global inverse models in machinery diagnostics. *AI-METH: Proceedings of the Symposium on Applications of Artificial Intelligence in Mechanics and Mechanical Engineering*, ss. 291–294, Gliwice, 14-16.11. 2001.

[95] D. Wachla. An example of genetic algorithm application in knowledge discovery in databases. *AI-METH: Proceedings of the Symposium on Methods of Artificial Intelligence*, ss. 429–432, Gliwice, 13-15.11. 2002.

[96] D. Wachla. Discovering dynamics of a simple mechanical system using context-free grammars. *AI-METH: Proceedings of the Symposium on Methods of Artificial Intelligence*, ss. 429–432, Gliwice, 5-7.11. 2003.

[97] T. Washio, H. Motoda, N. Yuji. Discovering admissible model equations from observed data based on scale-types and identity constrains. *IJCAI*, ss. 772–779, 1999.

[98] M. Witczak, J. Korbicz. Programowanie genetyczne w diagnostyce uszkodzeń i identyfikacji nieliniowych systemów dynamicznych. *Diagnostyka procesów. Modele. Metody sztucznej inteligencji. Zastosowania*, vol. 3 serii *Diagnostyka procesów*, ss. 427–464. WNT, Warszawa, 2002.

[99] R. Zembowicz, J. M. Żytkow. Discovery of Equations: Experimental Evaluation of Convergence. *Proceedings of the AAAI-92*, ss. 70–75. AAAI Press, Menlo Park, CA, 1992.

[100] R. Zembowicz, J. M. Żytkow. Database exploration in search of regularities. *Journal of Intelligent Information Systems*, 2:39–81, 1993.

[101] J. M. Żytkow, R. Zembowicz. Mining Patterns at Each Scale in Massive Data. Z. W. Raś, M. Michalewicz, redaktorzy, *Foundations of Intelligent Systems*, nr 1079 serii Lecture Notes in Artificial Intelligence, Proc. of 9th Int. Symposium ISMIS'96, Zakopane 1996, ss. 139–148. Springer-Verlag, Berlin-Heidelberg, 1996.

Identyfikacja dynamicznych modeli diagnostycznych metodami odkryć wiedzy w bazach danych

Streszczenie

W obszarze diagnostyki technicznej i eksploatacji maszyn istnieje wiele baz danych, które mogą być źródłem użytecznej wiedzy diagnostycznej. Pozyskanie tej wiedzy wiąże się z wieloma trudnościami. Źródłem tych trudności jest duża ilości dostępnych danych, ich niekompletność i niepoprawność, brak ekspertów potrafiących interpretować dane oraz ograniczenia klasycznych metod analizy danych. Ponadto w wielu przypadkach dane gromadzone są z uwzględnieniem czasu, co powoduje dodatkowe trudności w trakcie ich analizy. Niniejsza praca stanowi próbę rozwiązania tego problemu.

W wyniku przeglądu istniejącego stanu w zakresie metod analizy danych stwierdzono, że przydatne do rozwiązania przedstawionego problemu mogą być metody odkrywania wiedzy w bazach danych. Pewną niedogodnością tych metod jest to, że w większości przypadków są przeznaczone do analizy danych, w których nie uwzględniono zjawiska upływu czasu. W szczególności dotyczy to metod stosowanych na etapie *drążenia danych*. W związku z powyższym sformułowano tezę, że możliwe jest odkrywanie zależności dynamicznych z użyciem metod odkrywania zależności statycznych w bazie danych, w której dokonano transformacji wartości atrybutów polegającej na ich projekcji w wielowymiarową przestrzeń regresorów.

W celu wykazania słuszności postawionej tezy zaproponowano metodę pozwalającą na odkrywanie zależności dynamicznych z użyciem metod odkrywania zależności statycznych z jednoczesną selekcją atrybutów relewantnych. Głównym elementem metody jest algorytm, za pomocą którego dokonywana jest projekcja wartości atrybutów w wielowymiarową przestrzeń regresorów. Selekcja atrybutów relewantnych prowadzona jest w sposób pośredni z zastosowaniem algorytmu indukcji wiedzy. Sposób ten wymaga określenia trzech elementów: algorytmu indukcji wiedzy, metody przeszukiwania przestrzeni atrybutów oraz kryterium oceny. W szczególności w pracy do odkrywania modeli przyjęto metodę wektorów wspomagających, a do przeszukiwania przestrzeni atrybutów – algorytm genetyczny. Dobrane elementy metody selekcji atrybutów relewantnych stały się przyczynkiem do sformułowania drugiej tezy, że łączne zastosowanie algorytmów genetycznych i metody wektorów wspomagających pozwoli odkrywać modele dynamiczne cechujące się małą złożonością i dokładnością wystarczającą do zastosowań diagnostycznych.

Weryfikację obydwu tez pracy przeprowadzono na podstawie danych pozyskanych w wyniku eksperymentów numerycznych oraz udostępnionych danych, które gromadzone są w bazie danych systemu klasy SCADA kontrolującego pracę jednej z zainstalowanych na Śląsku pompowni głębinowych. Zakres weryfikacji obejmował przygotowanie danych łącznie z utworzeniem zbiorów przykładów trenujących i testowych oraz odkrywanie zależności dynamicznych wg przyjętego planu.

Wyniki otrzymane podczas procesu weryfikacji stanowią indukcyjne potwierdzenie sformułowanych w pracy tez. Odkrywanie zależności dynamicznych może odbywać się w sposób automatyczny z zastosowaniem metody wektorów wspomagających i z jednoczesną selekcją atrybutów relewantnych za pomocą algorytmu genetycznego.

Słowa kluczowe: sztuczna inteligencja, odkrywanie wiedzy w bazach danych, diagnostyka procesów przemysłowych, identyfikacja systemów

Identification of dynamic diagnostic models using methods of knowledge discovery in databases

Summary

In the area of technical diagnostics, many databases exist that can be sources of useful diagnostic knowledge. Acquiring such knowledge involves a number of difficulties. Too much data available, their deficiency and incorrectness, the lack of experts being able to interpret the data and constraints of classic methods of data analysis are te sources of these difficulties. Moreover, in many such cases, the data is obtained with the consideration of time, which causes additional difficulties during its analysis. This work constitutes an attempt to solve the problem.

Based on the finding of the existing state survey in the field of methods of data analysis, one could conclude that the methods of knowledge discovery in databases might be useful for the solution of the problem presented. A certain inconvenience of these methods is the fact that in most cases the phenomena of time passing were not taken into account. In particular, this concerns the methods used in the stage of data mining. For this reason, a hypothesis was formulated: that it was possible to discover dynamic dependencies with the use of methods of discovering static dependencies in databases, in which a transformation of attributes relying on a projection of their values into multidimensional space of the regressors was carried out.

With the aim of proving the rightness of the hypothesis stated, a method which allows to discover dynamic dependecies with the use of the method of discovering static dependecies with the selection simultaneous of relevant attributes, was proposed. The main element of the method is an algorithm, by means of which the projection of attributes values into the space of regressors is carried out. The selection of relevant attributes is conducted in an indirect manner with the application of a knowledge induction algorithm. This manner requires three elements: the knowledge induction algorithm, a method of searching the space of attributes and an evaluation criterion. In particular, the Support Vector Machines and the Genetic Algorithm for searching through the attributes space were adopted. The elements choosen for attributes selection became the cause of formulating the second hypothesis: that the co-application of Genetic Algorithm and Support Vector Machines would allow to discover dynamic models characterized by little complexity and, in the same time, the accuracy sufficient enough for diagnostic applications.

The verification of the hypotheses was conducted on the basis of the data obtained in a numerical experiments and the data which was gathered in the databases of SCADA system that controlled on of deep-well pumping stations installed in the Silesian district. The range of the verification included: preparation of the data along with a formation of sets of training and testing examples, and discovering dynamic dependencies according to the established plan.

The results, which were obtained during the verification constitute an inductive confirmation of the formulated hypotheses. Discovering the dynamic dependencies may take place in automatic manner with the application of the Support Vector Machines method and the selection of relevant attributes at the same time with the use of Genetic Algorithm.

Key words: artificial intelligence, knowledge discovery in databases, diagnostic of industrial pocesses, system identification