

MODELOWANIE TESTÓW DIAGNOSTYCZNYCH ZA POMOCĄ METOD UCZENIA MASZYNOWEGO

Mateusz Kalisch

*Institut Podstaw Konstrukcji Maszyn, Politechnika Śląska
mateusz.kalisch@polsl.pl*

Streszczenie

W pracy zaprezentowano metodykę tworzenia testów diagnostycznych służących do detekcji i izolacji uszkodzeń za pomocą algorytmów uczenia maszynowego z wykorzystaniem darmowego oprogramowania RapidMiner. Porównano różne metody łączenia klasyfikatorów na przykładzie danych symulacyjnych wygenerowanych za pomocą modelu numerycznego zaworu elektro-pneumatycznego opracowanego w ramach projektu DAMADICS. Przedstawione wyniki badań potwierdzają poprawność proponowanego podejścia.

Słowa kluczowe: uczenie maszynowe, diagnostyka techniczna, detekcji i izolacja uszkodzeń

MODELLING OF DIAGNOSTIC TESTS USING MACHINE LEARNING METHODS

Summary

The papers deals with the methodology of designing diagnostics tests that can be used for fault detection and isolation using machine learning algorithms implemented in open source RapidMiner application. In the paper there were compared different methods of combining classifiers using the benchmark data generated by means of the simulator of electro-pneumatic valve that has been developed within the DAMADICS project. The results of the research study confirm the effectiveness of the proposed approach.

Keywords: machine learning, technical diagnostics, fault detection and isolation

1. WSTĘP

Uczenie maszynowe wspomaga inżynierów w procesie rozwiązywania zadań klasyfikacji, predykcji czy podejmowania decyzji. W dziedzinie tej można wyróżnić szeroką gamę metod opartych na wielu często znacznie różniących się od siebie podejściach. Zakres wykorzystania tych metod również jest bardzo rozległy, co pokazują aplikacje w medycynie [7, 17], przemyśle maszynowym [5], w zagadnieniach pozyskiwania wiedzy [6, 20] oraz przemyśle rozrywkowym. Zasadniczą zaletą tych metod jest możliwość automatycznego tworzenia klasyfikatorów za pomocą danych uczących, proces ten nazywany jest trenowaniem i pozwala na skuteczne opracowanie klasyfikatora. W wielu przypadkach jest to jedyna możliwa metoda utworzenia klasyfikatora, ponieważ sposób

reprezentacji wiedzy w nim zawartej uniemożliwia jej interpretację czy modyfikację przez eksperta. Do tego typu metod można zaliczyć np. sztuczne sieci neuronowe [18]. Jednym z obszarów, w którym proces klasyfikacji znajduje zastosowanie, jest diagnostyka techniczna [10]. Diagnostyka techniczna to zorganizowany zbiór metod i środków do oceny stanu technicznego obiektów technicznych. Istotne jest, aby ocena ta była podawana z jak najmniejszym opóźnieniem, aby zminimalizować szkody wynikłe z zaistniałej awarii. Jednocześnie należy zwrócić uwagę na ewentualne fałszywe alarmy informujące o zaistnieniu awarii, która w rzeczywistości nie wystąpiła. Jednak w tym wypadku wskazuje się na mniejszą szkodliwość tego zdarzenia niż w sytuacji, w której po wy-

stąpieniu uszkodzenia nikt nie zostanie o tym poinformowany [21]. Możliwe jest zastosowanie różnych podejść w procesie wykorzystania klasyfikatora w diagnostyce technicznej, gdzie sygnały określają wartość pomiarową zmieniającą się w czasie. Jako sygnały wejściowe mogą zostać użyte sygnały pochodzące bezpośrednio z aparatury pomiarowej oraz ich cechy, do których obliczenia można wykorzystać np. transformację falkową [2, 9, 19]. Problematyka zadania polegająca na przyporządkowaniu stanu maszyny do jednej ze zdefiniowanych klas mówiących o pełnej zdatności urządzenia lub wystąpieniu jednego ze znanych uszkodzeń jednoznacznie wskazuje na możliwość wykorzystania w tym celu technik uczenia maszynowego. Zaleta uczenia maszynowego polegająca na automatycznym wytrenowaniu klasyfikatora może okazać się w tym przypadku złudna. Klasyfikatory do poprawnego działania wymagają wytrenowania na danych, w których znajduje się zbliżona liczba próbek dotycząca poszczególnych klas (stanów). Charakter danych pomiarowych pochodzących z rzeczywistego układu pomiarowego odbiega od tego założenia z powodu unikania sytuacji, w których urządzenie pracuje w stanie niesprawności [4]. Takie podejście w znacznym stopniu ogranicza ilość informacji o stanach powiązanych z niesprawnością. Wynikiem takiego działania mogą być klasyfikatory niedostatecznie wytrenowane, których skuteczność klasyfikacji może być niewiele lepsza od decyzji losowej. W celu rozwiązania tego typu problemów i poprawy ogólnego wyniku klasyfikacji [8] opracowano różne metody łączenia klasyfikatorów. W zależności od zastosowanego klasyfikatora różni się sposób reprezentacji modelu wiedzy. Część modeli pozwala na jego interpretację przez eksperta, inne natomiast nie pozwalają (np. model czarnej skrzynki).

Głównym celem niniejszych badań jest przedstawienie korzyści płynących z zastosowania różnych metod łączenia klasyfikatorów oraz pokazanie metodyki tworzenia testów diagnostycznych z wykorzystaniem dostępnych narzędzi.

2. DOSTĘPNE NARZĘDZIA WSPOMAGAJĄCE KLASYFIKACJĘ

Dostępna jest szeroka baza aplikacji wspomagających pracę z klasyfikatorami poprzez zaimplementowanie w nich powszechnie znanych algorytmów służących do klasyfikacji. Wśród tych aplikacji wymienić można na przykład: Weka [16], Język R [13], RapidMiner [14], Orange [12], Statistica [15], Angoss Knowledge Studio [11]. Duża część z dostępnych rozwiązań jest darmowa lub oparta na licencji Open Source. Większość aplikacji została wyposażona w przejrzysty interfejs graficzny wspomagający użytkownika w procesie drążenia danych. Powszechnie stosowany jest również język R, który pozbawiony jest zaawansowanych narzędzi graficznych i

pozwała na pracę z klasyfikatorami za pomocą edytora tekstowego i skryptu. Komercyjne rozwiązania stosowane są przede wszystkim przez firmy jako rozwiązania pewne, sprawdzone i stabilne w działaniu.

3. ŚRODOWISKO RAPIDMINER

RapidMiner jest darmową aplikacją przeznaczoną do realizacji zadań z dziedziny drążenia danych [1]. Program ten został wyposażony w przejrzysty interfejs graficzny działający na podstawie metodyki „przeciągnij i upuść”. Dostępne wtyczki, obszerna dokumentacja oraz dostępny kod źródłowy aplikacji pozwala na rozbudowę jej funkcjonalności w bardzo szerokim zakresie. Zdecydowano się na wykorzystanie tej aplikacji ze względu na szeroki zakres domyślnie dostępnych klasyfikatorów oraz szereg operatorów wspomagających proces łączenia klasyfikatorów.

3.1 METAKLASYFIKACJA

Aplikacja została wyposażona w narzędzie pozwalające na tworzenie metaklasyfikatorów. Metaklasyfikacja polega na równoległej pracy wielu często różnych klasyfikatorów (czasami również na podstawie innych danych wejściowych) i następnie na połączeniu wyników wnioskowania początkowych klasyfikatorów w końcowym klasyfikatorze. Głównym celem takiego podejścia jest uzyskanie lepszej dokładności klasyfikacji w porównaniu do pojedynczych klasyfikatorów wejściowych. Narzędzie dostępne w oprogramowaniu RapidMiner pozwala na dodanie dowolnej liczby klasyfikatorów wejściowych i pojedynczego metaklasyfikatora. Dostępny jest parametr pozwalający na określenie, czy meta-klasyfikator powinien brać pod uwagę tylko etykiety wyjściowe klasyfikatorów (wyniki ich klasyfikacji), czy również dane źródłowe podawane na ich wejściu.

3.2 FUZJA KLASYFIKATORÓW

Nieco innym podejściem w kontekście łączenia wielu klasyfikatorów jest fuzja polegająca na użyciu szeregu klasyfikatorów, z których końcowy rezultat zostaje wyodrębniony na podstawie np. głosowania. Ta metoda ponownie pozwala na zastosowanie wielu różnych klasyfikatorów, tak jak to było w przypadku metaklasyfikacji. Wyróżnić można różne metody głosowania, jak np. jednogłośnie, ważone czy większościowe [8]. Głosowanie jednogłośnie wymaga zgodnej decyzji spośród wszystkich klasyfikatorów wchodzących w skład komitetu, co może w znacznym stopniu utrudnić klasyfikację, zwłaszcza w przypadku większej liczby dostępnych klas. Głosowanie większościowe polega na wybraniu klasy o największej liczbie głosów spośród wszystkich biorących udział w głosowaniu klasyfikatorów. W przypadku głosowania ważonego każdy z klasyfikatorów ma przypisaną wagę, gdzie waga klasyfikatorów o większej skuteczności klasyfikacji jest wyższa w stosunku do klasyfi-

katorów o niższej skuteczności klasyfikacji. Wagi są najczęściej stosowane w metodach fuzji klasyfikatorów takich jak „Bagging” i „Adaboost” (opisanych w dalszej części artykułu), gdzie przypisywane są w sposób automatyczny w zależności od skuteczności klasyfikacji poszczególnych klasyfikatorów. W razie „remisu” wynik jest uzyskiwany na podstawie losowania spośród konfliktowych etykiet.

Kolejnym podejściem w fuzji klasyfikatorów są metody ingerujące w zbiór danych trenujących [8, 22]. W tej koncepcji uznano, że przyczyną zbyt niskiej skuteczności klasyfikacji nie jest rodzaj wybranego klasyfikatora, a dane znajdujące się w zbiorze uczącym, których rozkład w przestrzeni cech utrudnia poprawną klasyfikację. Podstawową metodą jest tu metoda „Bagging”. Metoda ta polega na utworzeniu szeregu klasyfikatorów opartych na tym samym sposobie działania, ale dla każdego z nich przygotowany jest inny zestaw danych o rozmiarze równym rozmiarowi danych źródłowych. Dane dla każdego z klasyfikatorów są losowane spośród danych wejściowych, przy czym każda z wartości w całym procesie losowania ma takie samo prawdopodobieństwo wylosowania. Skutkiem takiego działania jest możliwość wielokrotnego powtarzania się tych samych danych w pojedynczym zbiorze trenującym. Na podstawie tej metody opracowane zostały kolejne metody określane mianem „Boostingu”, do których zaliczany jest algorytm „AdaBoost”. Ta metoda fuzji klasyfikatorów dopasowuje się do błędów uzyskiwanych przez poprzednie klasyfikatory w komitecie. Każda z wartości w zbiorze trenującym ma przypisaną wagę, która określa „ważność” danej próbki. Po nauczeniu pierwszego klasyfikatora na losowym zbiorze danych weryfikowana jest skuteczność klasyfikacji na tym samym zbiorze. Dane trenujące, dla których wynik działania klasyfikatora był błędny, w następnej iteracji mają wyższą wagę, co przekłada się na większe prawdopodobieństwo ich wylosowania. Dzięki temu kolejne klasyfikatory są trenowane na zbiorze wypełnionym w większym stopniu danymi „trudnymi”. Dodatkowym elementem działania algorytmu jest nadanie wag dla poszczególnych klasyfikatorów, które są uwzględnione w procesie końcowego wnioskowania (głosownie ważne). Zasadniczym elementem determinującym stosowanie wymienionych wcześniej metod fuzji klasyfikatorów jest stosowanie jako ich elementów bazowych tzw. klasyfikatorów słabych, czyli takich, dla których wynik klasyfikacji jest niewiele lepszy od decyzji losowej. Innym przypadkiem stosowania fuzji jest stosowanie klasyfikatorów niestabilnych, dla których niewielka zmiana w zbiorze danych może w znacznym stopniu wpłynąć na skuteczność klasyfikacji [22]. Do klasyfikatorów niestabilnych można zaliczyć np. drzewa decyzyjne czy sieci neuronowe. Natomiast do klasyfikatorów mocnych można zaliczyć np. naiwny klasyfikator bayesowski. Zastosowanie w tych metodach klasyfikatorów mocnych, które uzyskują wysoki wynik skuteczności

klasyfikacji, może skutkować wygenerowaniem wielu bardzo podobnych klasyfikatorów, a końcowy wynik ich działania może być zbliżony do wyniku działania pojedynczego klasyfikatora o najwyższej skuteczności.

3.3 DOSTĘPNE KLASYFIKATORY

Aplikacja RapidMiner jako narzędzie do drażenia danych ma zaimplementowane algorytmy klasyfikacji, które są szeroko opisywane w licznych publikacjach i podręcznikach. Wśród nich można wymienić k – najbliższych sąsiadów, naiwny klasyfikator bayesowski (również jądrowej funkcji rozkładu), drzewa decyzyjne, sieć neuronową, regresję liniową, regresję logistyczną czy maszynę wektorów wspierających [22]. Wszystkie te metody dostępne są w aplikacji jako „bloczki”, które wchodzi w skład procesu w postaci grafu. Każdy z tych klasyfikatorów posiada parametry umożliwiające użytkownikowi modyfikowanie ich działania. Wynikiem działania tych klasyfikatorów jest przede wszystkim etykieta mówiąca o klasie, do której została przydzielona próbka danych. Dodatkowym parametrem jest wartość stopnia pewności przynależności danej próbki do każdej z klas. Jest to opisane za pomocą wartości od 0 do 1, gdzie 1 oznacza 100% pewność. Sam proces wyznaczania pewności różni się w zależności od klasyfikatora. Część z nich określa stałą wartość już na etapie trenowania (drzewo decyzyjne, reguły) a pozostałe w chwili klasyfikacji nowych danych (k – najbliższych sąsiadów, naiwny klasyfikator bayesowski, sztuczna sieć neuronowa).

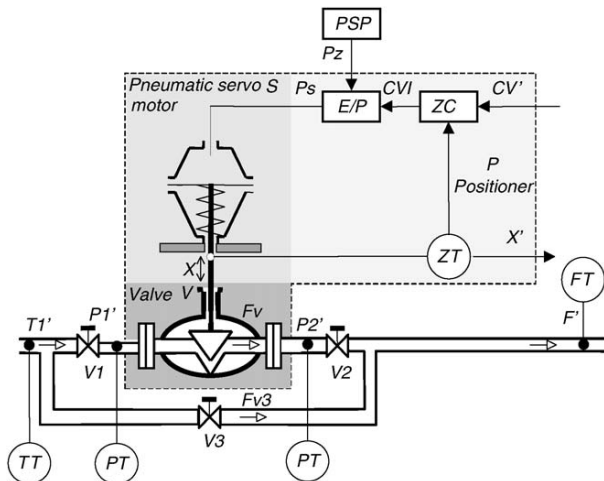
4. PRZYKŁAD ZASTOSOWANIA

W niniejszym rozdziale zaprezentowano przykład zastosowania klasyfikatorów, ich kombinacji oraz dostępnych metod fuzji klasyfikatorów w celu realizacji zadania detekcji i izolacji uszkodzeń za pomocą aplikacji RapidMiner.

4.1 BADANY OBIEKT

DAMADICS (Development and Application of Methods for Actuator Diagnosis in Industrial Control System [3]) jest projektem, w wyniku którego powstał symulator zaworu elektro-pneumatycznego. Schemat blokowy zaworu będącego przedmiotem rozważań znajduje się na rys. 1. Symulator zaworu opracowano w celu ułatwienia przeprowadzania badań nad nowymi metodami detekcji i izolacji uszkodzeń urządzeń mechanicznych. Został on zaimplementowany w środowisku MATLAB Simulink, w którym można bezpośrednio realizować zadania związane z detekcją i izolacją uszkodzeń, lub wygenerować dane, które można zastosować w innych aplikacjach. Symulator powstał na bazie dokładnych badań mechanizmu działania zaworu ze szczególnym uwzględnieniem zjawisk związanych z wystąpieniem niesprawności. Wzorem dla opracowanego

modelu był zawór stosowany na linii produkcyjnej cukrowni Lublin.



Rys. 1 Schemat obiektu badań [3]

Symulator składa się z wielu modułów pozwalających na realizację wielu różnych scenariuszy działania zaworu. Autor artykułu korzystał z modułu pozwalającego na symulację jednego z 19 predefiniowanych uszkodzeń oraz stanu pełnej zdatności:

- Uszkodzenia elementów kontrolujących zawór:
 - F1 – zablokowanie tłoka zaworu,
 - F2 – osadzanie się materiału na ścianach lub tłoku zaworu,
 - F3 – erozja zaworu lub gniazda zaworu,
 - F4 – wzrost tarcia pomiędzy ruchomymi elementami zaworu,
 - F5 – zewnętrzny wyciek,
 - F6 – wewnętrzny wyciek,
 - F7 – odparowywanie medium.
- Awarie serwowomatoru:
 - F8 – skrócony trzpień serwowomatoru,
 - F9 – rozszczelnienie obudowy zaworu lub połączeń,
 - F10 – perforacja membrany serwowomatoru,
 - F11 – uszkodzenie sprężyny serwowomatoru.
- Uszkodzenia pozycjonera:
 - F12 – uszkodzenie przekaźnika elektro-pneumatycznego,
 - F13 – uszkodzenie czujnika przemieszczenia rdzenia,
 - F14 – uszkodzenie czujnika ciśnienia,
 - F15 – uszkodzenie sprężyny pozycjonera.
- Ogólne / zewnętrzne uszkodzenia:
 - F16 – spadek ciśnienia medium zasilającego pozycjoner,
 - F17 – niespodziewana zmiana ciśnienia pomiędzy wlotem i wylotem zaworu,
 - F18 – całkowicie lub częściowo otwarty zawór obejścia,
 - F19 – Awaria czujnika przepływu.

W dalszej części artykułu stan pełnej zdadności został oznaczony symbolem F0. Wśród wymienionych uszkodzeń zostały podkreślone te, które autor wykorzystał w procesie weryfikacji metod klasyfikacji.

Praca zaworu w wybranym trybie ogranicza się do harmonicznego zadanego sygnału sterującego o okresie 100 sekund. Sygnał wartości zadanej (oznaczenie CV' na rys. 1) wpływa na poziom wychylenia trzpienia zaworu (X na rys. 1), co powoduje zmianę prześwitu w kanale przepływu cieczy. Po wybraniu uszkodzenia i rozpoczęciu symulacji przez pewien okres urządzenie pracuje prawidłowo. Po upływie 800 sekund następuje uruchomienie wybranego uszkodzenia, co najczęściej można zaobserwować poprzez zmianę charakterystyki przebiegu zmiennych procesowych. W wykorzystanym trybie pracy symulatora nie ma możliwości łatwej zmiany czasu, w którym ma wystąpić uszkodzenie, istnieje jednak możliwość uruchomienia symulacji w trybie „Rapid accelerator”, który w znacznym stopniu skraca czas trwania symulacji. Część spośród dostępnych uszkodzeń charakteryzuje się nagłym wystąpieniem, część narastającym, w innych możliwe jest wybranie zarówno wystąpienia nagłego jak i narastającego. W niektórych uszkodzeniach o charakterystyce nagłej możliwe jest określenie rozmiaru tego uszkodzenia poprzez wybranie jednej z opcji (małe, średnie, duże uszkodzenie). Wynikiem działania symulatora jest szereg zmiennych procesowych:

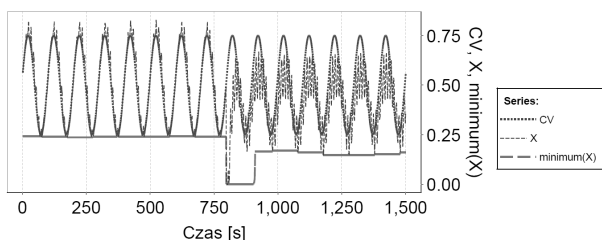
- CV – wartość zadana sygnału sterującego,
- P1 – ciśnienie na wejściu do zaworu,
- P2 – ciśnienie na wyjściu zaworu,
- X – przemieszczenie tłoka zaworu,
- F – prędkość przepływu cieczy przez zawór,
- T1 – temperatura cieczy,
- f – flaga wystąpienia uszkodzenia.

Wszystkie te sygnały są poddawane normalizacji do przedziału od 0 do 1, wyjątek stanowi parametr f, w przypadku którego istnieje możliwość wystąpienia wartości ujemnych. Znak dodatni lub ujemny niesie ze sobą dodatkową informację na temat określonego uszkodzenia, np. w przypadku wystąpienia uszkodzenia F13 wartość ujemna oznacza ustalenie stałej wartości wyjściowej sensora równej 0, wartość dodatnia natomiast oznacza, że czujnik zwraca stałą wartość równą 1.

4.2 PRZYGOTOWANIE DANYCH

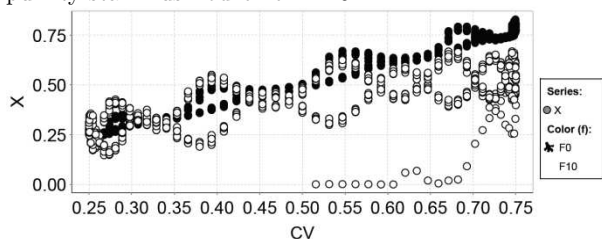
Charakterystyka sygnałów pomiarowych utrudnia stosowanie algorytmów klasyfikacji bezpośrednio na tych danych. Mają one charakter silnie oscylacyjny zależny od sygnału sterującego będącego sygnałem okresowym. Oprogramowanie RapidMiner pozwala na przeprowadzenie szeregu testów, jak np. korelacja czy informacja wzajemna, badających zależności pomiędzy wybranymi sygnałami. W wyniku tej analizy zdecydowano się na wybranie jedynie czterech sygnałów spośród wszystkich dostępnych (P2, X, F i T1). Odpowiednie narzędzie

aplikacji RapidMiner pozwala na wyznaczenie cech punktowych sygnałów obliczonych w oknie o dowolnej szerokości. Spośród wszystkich dostępnych wybrano cztery metody: wartość średnia, mediana, wartość maksymalna i minimalna. Wyniki uzyskane dla wybranych cech były zadowalające, a dodawanie kolejnych nie podniosło ogólnej skuteczności działania klasyfikatorów. Ze względu na charakter sygnału sterującego, który jest sygnałem harmonicznym o okresie równym 100 sekund, postanowiono zastosować okno o dokładnie takiej szerokości. Stosowanie okna o innych szerokościach nie wpłynęło na poprawę wyników klasyfikacji. Na rys. 2. został przedstawiony wykres pokazujący przebieg sygnału sterującego, wartość przemieszczenia tłoka zaworu oraz jedną z wykorzystywanych cech sygnału, a mianowicie minimum liczonego z sygnału za pomocą prostokątnego okna o szerokości 100 sekund.



Rys. 2 Porównanie przebiegów sygnału sterującego (CV), przemieszczenia trzpienia zaworu (X) oraz wartości minimalnej obliczanej dla sygnału X w oknie o szerokości 100 sekund

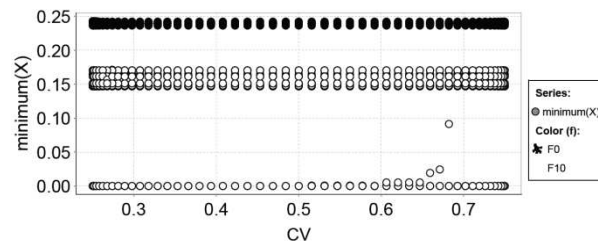
Na rys. 3. pokazano zależność przemieszczenia tłoka zaworu od sygnału sterującego z podziałem na klasy (stany urządzenia oznaczone parametrem f), gdzie czarne punkty oznaczają stan pełnej zdatności, a białe punkty stan z uszkodzeniem F10.



Rys. 3 Zależność przesunięcia trzpienia zaworu od sygnału sterującego z uwzględnieniem stanu pełnej zdatności (F0) i stanu z uszkodzeniem (F10)

Wyraźnie można zauważyć przenikanie się próbek z obu rozpatrywanych klas niemal w całej dziedzinie. Wyznaczenie granic pomiędzy dwoma występującymi w próbkach klasami jest praktycznie niemożliwe. W związku z tym, że sygnał sterujący jest sygnałem harmonicznym o okresie równym 100 sekund, postanowiono wyznaczyć szereg cech punktowych liczonych w oknie o dokładnie takiej szerokości. Zabieg ten pozwolił na zminimalizowanie oscylacji poszczególnych parametrów niemal do zera. Na rys. 4 przedstawiono zależność jednej z tych cech (minimum) obliczonej dla sygnału X od sygnału sterującego. Tym razem powstała wyraźna granica pomiędzy stanem pełnej zdatności a uszkodze-

niem. Można to też zaobserwować na wykresie zależnym od czasu (rys. 2). Spośród dostępnych uszkodzeń skupiono się jedynie na wybranych uszkodzeniach nagłych



Rys. 4 Zależność minimalnej wartości przesunięcia trzpienia zaworu a sygnałem sterującym z uwzględnieniem stanu pełnej zdatności (F0) i stanu z uszkodzeniem (F10)

4.3 TESTY DIAGNOSTYCZNE DO DETEKcji I IZOLACJI

Testy diagnostyczne reprezentowane są w oprogramowaniu RapidMiner jako procesy i mają charakter diagramów przepływu danych. Ścieżki mogą dowolnie się rozgałęziać, łączyć, a dane przekazywane pomiędzy blokami mogą być dowolnie modyfikowane. W jednym procesie może być zawartych wiele metod klasyfikacji, co sprzyja opracowywaniu własnych metaklasyfikatorów. Jednocześnie aplikacja została wyposażona w szereg narzędzi dostępnych jako dodatkowe bloki wspomagające opracowywanie różnych metod łączenia klasyfikatorów. W niniejszym artykule porównano skuteczność procesu izolacji uszkodzeń oraz stanu pełnej zdatności za pomocą różnych metod łączenia klasyfikatorów.

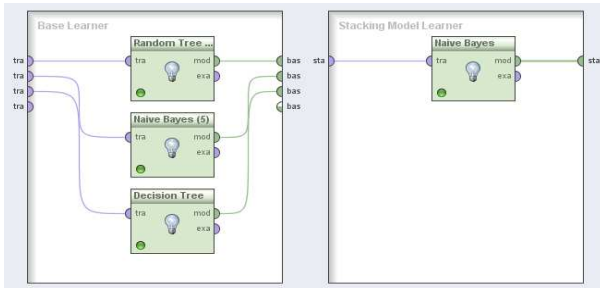
4.4 MIARA SKUTECZNOŚCI DZIAŁANIA KLASYFIKACJI

Dostępnych jest wiele metod umożliwiających przeprowadzenie weryfikacji działania klasyfikatora, jak np. podział zbioru danych na dwie odrębne części, gdzie jedna z nich służy do uczenia klasyfikatora, a druga do testowania. Działanie to ma na celu obiektywną ocenę skuteczności działania klasyfikatora poprzez zapobieganie zjawisku przeuczenia. Problem ten występuje w przypadku uczenia i testowania klasyfikatora za pomocą tych samych danych, w tym wypadku wynik może wskazywać na bardzo wysoką skuteczność klasyfikacji. Późniejsze zastosowanie nowych danych może skutkować działaniem klasyfikatora z o wiele niższą skutecznością niż wynikało to z badań weryfikacyjnych. Jednak uczenie i testowanie klasyfikatora za pomocą podzielonych danych może być utrudnione w niewielkim zbiorze, gdzie większość z próbek charakteryzuje się dość wysoką różnorodnością. Liczba wydzielonych próbek może okazać się niewystarczająca do skutecznego wytrenowania klasyfikatora. W takiej sytuacji pomocne mogą okazać się algorytmy walidacji krzyżowej oraz *Leave-k-out*. Działanie tych metod polega na kilkukrotnym procesie uczenia i testowania klasyfikatora za pomocą

różnych fragmentów danych w zależności od numeru iteracji metody. Wynikiem ich działania jest ogólna skuteczność klasyfikacji będąca średnią wyników wszystkich iteracji algorytmu oraz odchylenie standardowe. W dalszej części pracy w celu weryfikacji działania metod klasyfikacji autor korzystał z mechanizmu walidacji krzyżowej.

4.5 WERYFIKACJA DZIAŁANIA RÓŻNYCH MECHANIZMÓW FUZJI KLASYFIKATORÓW

Aplikacja RapidMiner została wyposażona w narzędzie pozwalające na budowanie komitetów klasyfikatorów. W celu zbudowania metaklasifikatora, spośród dostępnych w oprogramowaniu klasyfikatorów należy dobrać klasyfikatory bazowe trenowane i wnioskujące niezależnie oraz jeden klasyfikator pełniący rolę metaklasifikatora, którego wejściem są etykiety zwracane przez początkowy komitet klasyfikatorów tak jak to zostało pokazane na rys. 5.



Rys. 5 Fragment procesu wnioskowania opartego na meta-klasyfikacji

Analizując wyniki zawarte w tab. 1, można zauważyć, że wśród pojedynczych klasyfikatorów (wiersze 1, 2, 3, 4) drzewo decyzyjne charakteryzuje się najwyższą ogólną skutecznością klasyfikacji. Zastosowanie jednak metaklasifikatora w postaci naiwnego klasyfikatora bayesowskiego wraz z pozostałymi klasyfikatorami jako klasyfikatory bazowe (wiersz 5), pozwoliło na uzyskanie skuteczności o 4% wyższej w porównaniu do najlepszego pojedynczego klasyfikatora. Kolejnym etapem było zweryfikowanie skuteczności działania mechanizmu głosowania na zestawie tych samych klasyfikatorów, co w przypadku meta-klasyfikacji, jednak tym razem wszystkie klasyfikatory traktowane są równorzędnie (wiersz 6). Odrębną formą fuzji klasyfikatorów zaimplementowaną w oprogramowaniu RapidMiner jest rodzina metod ucząca szereg takich samych klasyfikatorów, ale na podstawie zróżnicowanych zbiorów danych trenujących. Przetestowano działanie metody AdaBoost zaimplementowanej w aplikacji na przykładzie drzewa decyzyjnego (wiersz 7 w tabeli 1).

Tab. 1 Porównanie wyników działania różnych metod klasyfikacji

	Klasyfikator	Skuteczność
1.	Drzewo losowe	50,07 %
2.	Naiwny klasyfikator bayesowski	84,19 %
3.	Drzewo decyzyjne	85,56 %
4.	Naiwny klasyfikator bayesowski (kernel)	84,20 %
5.	Metaklasifikator (Naiwny klasyfikator bayesowski)	89,55 %
6.	Głosowanie większościowe	85,00 %
7.	AdaBoost dla drzewa decyzyjnego	87,39 %
8.	Własny metaklasifikator	90,00 %

Uzyskany wynik jest nieco gorszy od metaklasifikatora, ale jednocześnie jest lepszy od pojedynczego klasyfikatora w postaci drzewa decyzyjnego. Interfejs aplikacji pozwala użytkownikowi na analizę każdego z drzew zbudowanego na podstawie innego zestawu danych uczących. W tab.2 porównano struktury drzew wchodzących w skład komitetu. Porównanie zostało wykonane za pomocą miary poziomu skomplikowania struktury drzewa, która wynika bezpośrednio z liczby gałęzi w danym drzewie. Wraz z kolejnymi iteracjami działania algorytmu generowane drzewo charakteryzuje się bardziej złożoną strukturą.

Tab. 2 Porównanie poziomów skomplikowania drzew decyzyjnych budowanych za pomocą metody AdaBoost

Iteracja	Poziom skomplikowania
1	22
2	22
3	36
4	44

Ostatnią metodą łączenia klasyfikatorów przedstawioną przez autora artykułu jest metaklasyfikacja niewykorzystująca gotowych narzędzi. Metody zaimplementowane w oprogramowaniu w dużym stopniu uproszczają proces budowania, uczenia czy testowania meta-klasyfikatora, ale jednocześnie ograniczająca w pewnym stopniu funkcjonalność tych metod. Gotowe narzędzie do budowania stosu klasyfikatorów pozwala na dobranie dowolnej liczby klasyfikatorów bazowych o dowolnym typie oraz na dodanie metaklasifikatora będącego jednym z dostępnych w oprogramowaniu klasyfikatorów. Rozbudowując proces o dodatkowe elementy, istnieje możliwość zdecydowania o rodzaju danych, z jakich mają korzystać poszczególne klasyfikatory. Wynikiem działania każdego z nich jest wyznaczona etykieta dla określonej próbki danych oraz szereg współczynników pewności przynależności danej próbki do każdej z klas. Pierwotnie wektor ten jest podawany jedynie do informacji użytkownika, nie ma możliwości

wykorzystania tych wartości w dalszym etapie wnioskowania. Wśród narzędzi dostępnych w oprogramowaniu można znaleźć blok, którego funkcjonalność można dowolnie modyfikować za pomocą języka skryptowego. W tym przypadku może on być wykorzystany do manipulacji typami danych, dzięki czemu istnieje możliwość wyodrębnienia poziomów przekonania o przynależności poszczególnych próbek do wszystkich możliwych klas i wykorzystanie ich jako np. wejście metaklasyfikatora. Takie działanie może w pewnym stopniu poprawić skuteczność działania metaklasyfikatora. Wykonano badanie polegające na zbudowaniu podstawowego metaklasyfikatora z trzema różnymi klasyfikatorami bazowymi, gdzie wejściem każdego z nich były te same cechy sygnałów diagnostycznych. Klasyfikatory zostały dobrane w ten sam sposób jak w przypadku pierwszej koncepcji metaklasyfikacji opisaną w ramach wykorzystania wbudowanych narzędzi oprogramowania RapidMiner. W danych wyjściowych klasyfikatorów bazowych zawarte zostały informacje o poziomach przekonania przynależności określonych danych do wszystkich możliwych klas istniejących w modelu. Tak przygotowane dane zostały podane na wejście metaklasyfikatora, którego celem było określenie ostatecznego stanu urządzenia. W tym przypadku uzyskano najwyższą skuteczność klasyfikacji spośród wszystkich testowanych metod równą 90 %.

Dla wbudowanego algorytmu metaklasyfikacji zostały przeprowadzone dodatkowe badania polegające na weryfikacji działania tej metody w zależności od konfiguracji klasyfikatorów i ich danych wejściowych. W każdym przypadku stosowano zbiór klasyfikatorów tego samego typu jako klasyfikatory bazowe. Metody fuzji lub łączenia klasyfikatorów wymaga zastosowania pewnego mechanizmu generującego różnicę pomiędzy nimi, ponieważ zastosowanie identycznych klasyfikatorów wraz z identycznymi danymi uczącymi wygeneruje prawdopodobnie zbiór takich samych klasyfikatorów zwracających tę samą konkluzję. Tab. 3 przedstawia wyniki działania tych badań, gdzie manipulowano trzema parametrami schematu: sposobem grupowania danych, typem klasyfikatorów wejściowych, typem metaklasyfikatora. Jednym z mechanizmów generujących pewną różnorodność w procesie fuzji klasyfikatorów może być zastosowanie różnych klasyfikatorów lub zróżnicowanie danych wejściowych. W przypadku aktualnie opisywanych dodatkowych koncepcji metaklasyfikacji zawsze stosowano zbiór identycznych klasyfikatorów bazowych, co wymusza przygotowanie odrębnych danych wejściowych dla każdego z nich. W tab. 3 zawarto etykiety o nazwach „Cechy” i „Sygnały”. Pierwsza oznacza, że dane wejściowe początkowych klasyfikatorów zostały pogrupowane zgodnie z zastosowanymi cechami sygnałów zmiennych procesowych, np. pierwszy klasyfikator korzystał jedynie z wartości średnich wszystkich dostępnych sygnałów, drugi tylko z

wartości maksymalnych itd. Etykieta „Sygnały” oznacza, że pojedynczy klasyfikator bierze pod uwagę wszystkie cechy określonego sygnału, np. pierwszy klasyfikator korzysta z wartości średniej, mediany, wartości maksymalnej i minimalnej z sygnału reprezentującego przemieszczenie trzpienia zaworu, kolejny wykorzystuje te same cechy, ale obliczone dla innego sygnału zmiennej procesowej. Oznaczenia „NB” i „DT” dotyczą rodzaju zastosowanych klasyfikatorów, gdzie NB oznacza naiwny klasyfikator bayesowski (Naive Bayes), a DT drzewo decyzyjne (Decision Tree). Etykieta znajdująca się przed strzałką oznacza typ klasyfikatorów wejściowych, zaś etykieta za strzałką określa typ metaklasyfikatora. Przetestowano wszystkie możliwe kombinacje tych parametrów i wyniki zawarto w tab. 3.

Tab. 3 Porównanie wyników działania różnych koncepcji metaklasyfikacji

	Metaklasyfikator	Skuteczność
1.	Cechy NB → NB	88,81 %
2.	Sygnały NB → NB	80,67 %
3.	Cechy DT → DT	89,69 %
4.	Sygnały DT → DT	68,02 %
5.	Cechy NB → DT	89,84 %
6.	Sygnały NB → DT	86,81 %
7.	Cechy DT → NB	86,18 %
8.	Sygnały DT → NB	81,33 %

Końcowy wynik klasyfikacji znacznie różni się od zastosowanego podejścia. Grupowanie danych dla klasyfikatorów bazowych ze względu na cechy charakteryzuje się wyższą skutecznością niż w przypadku grup zależnych od typu sygnału. Może to wynikać z podobieństwa sygnałów obliczonych na podstawie tej samej cechy.

5. PODSUMOWANIE

Dostępne aplikacje do drażenia i analizy danych pozwalają na tworzenie zróżnicowanych modeli zgromadzonej wiedzy w celu ich wykorzystania np. w procesie klasyfikacji. Duża elastyczność tych narzędzi wpływa pozytywnie na proces opracowywania i weryfikacji nowych metod łączenia klasyfikatorów lub przygotowywania dla nich danych w celu poprawy ich skuteczności działania. W artykule przedstawiono metodykę tworzenia testów diagnostycznych z użyciem metod uczenia maszynowego. Porównano różne modele wnioskowania oparte na dostępnych rozwiązaniach i wykazano poprawę skuteczności w przypadku zastosowania pewnych mechanizmów metaklasyfikacji oraz fuzji klasyfikatorów. Pokazano też różnicę w działaniu koncepcyjnie zbliżonych do siebie modeli, ale działających na podstawie różnych sposobów grupowania danych. Wśród opisanych metod najwyższą skutecznością charakteryzuje się koncepcja niezależna od wbudowanych mechanizmów aplikacji. Odpowiednie zarządzanie danymi i klasyfikatorami w sposób nieprzewidziany początkowo przez autorów

oprogramowania pozwala na uzyskanie dodatkowych korzyści w rozwiązywaniu określonego problemu.

Badania przedstawione w artykule zostały częściowo sfinansowane przez Narodowe Centrum Badań i Rozwoju w ramach projektu "Zintegrowany, szkieletowy system wspomagania decyzji dla systemów monitorowania procesów, urządzeń i zagrożeń" (grant nr. PBS2/B9/20/2013). Pozostała część badań została sfinansowana z funduszy statutowych Instytutu Podstaw Konstrukcji Maszyn

Literatura

1. Akthar F., Hahne C.: RapidMiner 5 Operator reference, 2012.
2. Anuradha B., Suresh Kumar K., Veera Reddy V.: Classification of cardiac signals using time domain methods. "Journal of Engineering and Applied Sciences" 2008, Vol. 3, No. 3, p. 7-12.
3. Bartyś M., Patton R., Syfert M., de las Haras S., Quevedo, J.: Introduction to the DAMADICS actuator FDI benchmark study. "Control Engineering Practice" 2006, Vol. 14, No. 6, p. 577-596.
4. Bekkar M., Alitouche T. A.: Imbalanced data learning approaches review. "International Journal of Data Mining & Knowledge Management Process" 2013, Vol. 3, No. 4, p. 15-33.
5. Broadhurts D., Taylor J., Kaderbhai N., Winson M. K., Small A. D., Rowland J. J., McGovern A. C., Kell D. B., Goodacre R.: Monitoring of complex industrial bioprocesses for metabolite concentrations using modern spectroscopies and machine learning: application to gibberellic acid product. "Biotechnology and Bioengineering" 2002, Vol. 78, No. 5, p. 527-538.
6. Grzymała-Busse D. M., Grzymała-Busse, J. W.: On the usefulness of machine learning approach to knowledge acquisition. "Computational Intelligence" 1995, Vol. 11, No. 2, p. 268-279.
7. Kononenko I.: Machine learning for medical diagnosis: history, state of the art and perspective. "Artificial Intelligence in Medicine" 2001, Vol. 23, No. 1, p. 89-109.
8. Kuncheva L. I.: Combining pattern classifiers: Methods and algorithms. New Jersey: Wiley-Interscience, 2004.
9. Mazarakis P. G., Avaritsiotis J. N.: Vehicle classification in sensor networks using time-domain signal processing and Neural Networks. "Microprocessors and Microsystems" 2007, Vol. 31, No. 6, p. 381-392.
10. Samanta B., Al-Balushi K. R., Al-Araimi S. A.: Artificial neural network and support vector machines with genetic algorithm for bearing fault detection. "Engineering Applications of Artificial Intelligence" 2013, Vol. 16, No. 7-8, p. 657-665.
11. Strona główna projektu Angoss Knowledge Studio. Pobrano Czerwiec 4, 2014 z lokalizacji <http://www.angoss.com/predictive-analytics-software/software/knowledgestudio/>
12. Strona główna projektu Orange. Pobrano Czerwiec 4, 2014 z lokalizacji <http://orange.biolab.si/>
13. Strona główna projektu R. Pobrano Czerwiec 4, 2014 z lokalizacji <http://www.r-project.org/>
14. Strona główna projektu RapidMiner. Pobrano Czerwiec 4, 2014 z lokalizacji <http://rapidminer.com/>
15. Strona główna projektu Statistica. Pobrano Czerwiec 4, 2014 z lokalizacji <http://www.statsoft.pl/>
16. Strona główna projektu WEKA. Pobrano Czerwiec 4, 2014 z lokalizacji <http://www.cs.waikato.ac.nz/ml/weka/>
17. Tomczak J. M., Gonczarek A.: Decision rules extraction from data stream in the presence of changing context for diabetes treatment. "Knowledge and Information Systems" 2013, Vol. 34, No. 3, p. 521-546.
18. Tu J. V.: Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes. "Journal of Clinical Epidemiology" 1996, Vol. 49, No. 11, p. 1225-1231.
19. Vergara A., Fonollosa J., Mahiques J., Trincavelli M., Rulkov N., Huerta R.: On the performance of gas sensor arrays in open sampling systems using Inhibitory Support Vector Machines. "Sensors and Actuators B: Chemical" 2013, Vol. 185, p. 462-477.
20. Webb G. I.: Integrating machine learning with knowledge acquisition through direct interaction with domain experts. "Knowledge-Based Systems" 1996, Vol. 9, No. 4, p. 253-266.
21. Wojtusik, J.: Wielomodele diagnostyczne maszyn wirnikowych. Gliwice: Pol. Śl., 2006. Zesz.Kat. PKM nr 128.
22. Woźniak M.: Metody fuzji informacji dla komputerowych systemów rozpoznawania. Wrocław: Ofic. Wyd. Pol. Wrocl., 2006.