

**Grzegorz PIECUCH**

---

*Mgr inż., Politechnika Rzeszowska, Wydział Elektrotechniki i Elektroniki, Katedra Informatyki i Automatyki, ul. W. Pola 2, 35-959 Rzeszów; [gpiecuch@kia.prz.edu.pl](mailto:gpiecuch@kia.prz.edu.pl)*

---

## **WRZECIONO CNC MAŁEJ MOCY JAKO OBIEKT BADAWCZY DLA METOD CI\* NA PRZYKŁADZIE DETEKCJI PRĘDKOŚCI**

## **LOW-POWER CNC SPINDLE AS A RESEARCH FACILITY FOR CI METHODS ON THE EXAMPLE OF SPEED DETECTION**

**Słowa kluczowe:** metody inteligencji obliczeniowej, sieć neuronowa, wrzeciono CNC.

**Keywords:** methods of computational intelligence, neural network, spindle CNC.

### **Streszczenie**

Artykuł omawia zastosowanie stanowiska dydaktycznego wyposażonego we wrzeciono CNC małej mocy do nauki metod inteligencji obliczeniowej na kierunkach technicznych wyższych uczelni np. automatyka i robotyka. Przedstawiono praktyczny przykład ćwiczenia, które zaznajamia studentów z rzeczywistym sprzętem sprzężonym z oprogramowaniem inżynierskim takim jak Matlab/Simulink, umożliwiającym stosowanie i analizę działania poszczególnych metod CI, także w czasie rzeczywistym.

### **Summary**

The article discusses the use of didactic station equipped with a low-power CNC spindle to teach computational intelligence methods to technical majors at universities, e.g. Automatic control and Robotics. This paper presents a practical example of exercises, which acquaint students with actual hardware integrated with engineering software such as Matlab/Simulink, which allows implementation and analysis of the operation of various CI methods, also in real time.

### **Wstęp**

Współcześni inżynierowie kształceni na kierunkach technicznych coraz częściej muszą stawać przed nowymi wyzwaniami. Zakłady produkcyjne zatrudniające automatyków w charakterze pracownika utrzymania ruchu czy też inżyniera

---

\* CI (ang. *computational intelligence*) – inteligencja obliczeniowa.

programisty PLC, projektanta, wymagają nierzadko, aby działania przez nich podejmowane polegały na modernizacjach obecnych linii produkcyjnych czy też projektowaniu nowych w taki sposób, aby wydajność produkcyjna była możliwie jak największa, przy zachowaniu bezawaryjności i uniknięciu jakichkolwiek przestojów w produkcji, które często przynoszą straty liczone w milionach złotych. Pojawiający się w branży trend wykorzystujący metody inteligencji obliczeniowej do przewidywania awarii maszyny zanim ona wystąpi jest coraz częstszą praktyką. Cenna diagnoza wykonana przez system nadzorujący stan pracy maszyny może w ten sposób uchronić zakład przed dłuższym i nieprzewidzianym przestojem produkcyjnym, a zużyty element grożący awarią można wówczas wymienić podczas zaplanowanej przerwy. Oprócz tak oczywistej zalety jak zdolność predykcji awarii maszyn, systemy monitorowania produkcji posiadają jeszcze wiele innych zalet m.in.: analiza czasu pracy maszyn, przyczyn przestojów, raportowanie produkcji, rozliczenia ilościowe czy też analiza wielkości odpadu technologicznego. Opracowanie profesjonalnego systemu TCM<sup>1</sup> (ang. *Tool Condition Monitoring*) w głównej mierze opiera się na zastosowaniu metod uczenia maszynowego, polegających na polepszaniu działania algorytmów komputerowych wraz z kolejnymi iteracjami analizy danych uczących<sup>2</sup>. Do obsługi takiego systemu nie jest niezbędna wiedza ekspercka, ponieważ dobrze przygotowany system będzie w stanie obsłużyć przeciętny operator danej maszyny. Prawdziwym wyzwaniem dla inżyniera staje się jednak opracowanie takiego systemu. Absolwenci studiów takich jak automatyka i robotyka, mających styczność z tą tematyką podczas procesu kształcenia, powinni być dobrze przygotowani do rozwiązywania tego typu problemów w swojej codziennej pracy w branży.

### **Laboratoryjne stanowisko badawczo-dydaktyczne**

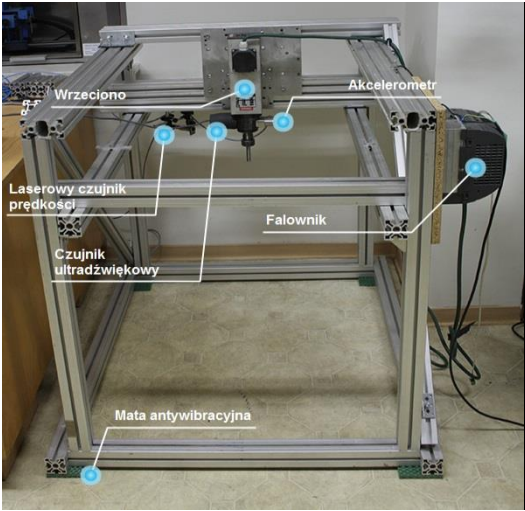
Stanowisko wyposażone jest w wrzeciono CNC małej mocy Teknomotor C41/47-C-3822-400 o maksymalnej prędkości obrotowej rzędu 24 000 obrotów/min. Pracą wrzeciona (załącz/wyłącz, szybciej/wolniej) steruje falownik Siemens Sinamics G110 o wydajności 3 kW. Jako sensorów użyto akcelerometru Hansford Sensors HS-100ST1000706 oraz czujnika ultradźwiękowego UE Systems UE Ultra-Trak 750, które zamontowane zostały na uchwycie narzędzia. Do akwizycji danych niezbędne jest podłączenie sensorów do modułów wejść sprzężonych z komputerem przemysłowym. Do tego celu użyto komputera IPC

---

<sup>1</sup> K. Jemielniak, J. Kosmol, *Diagnostyka narzędzia i procesu skrawania – stan aktualny i kierunki rozwoju*, „Mechanik” 1996, nr 10.

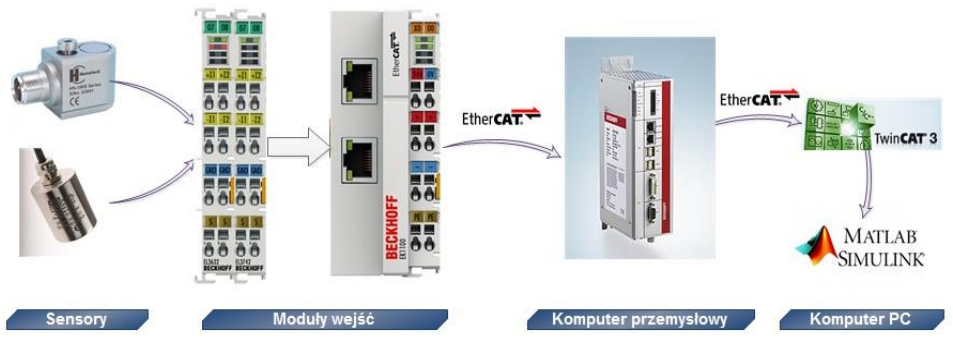
<sup>2</sup> K. Krawiec, J. Stefanowski, *Uczenie maszynowe i sieci neuronowe*. Wydawnictwo Politechniki Poznańskiej, Poznań 2003.

Beckhoff C6920 wraz z couplerem interfejsu EtherCAT EK1100 oraz modułów EL3742 (podłączony czujnik ultradźwiękowy) i EL3632 (podłączony akcelero-  
metr). Istotne jest, że oba moduły wejściowe dysponują funkcją nadpróbkowania  
(ang. *oversampling*), co w konsekwencji oznacza, że do komputera przemys-  
łowego trafia jednorazowo pakiet danych złożony z 50 próbek<sup>3</sup>. Stanowisko  
badawczo-dydaktyczne pokazano na rys. 1, a tor transmisji danych z czujni-  
ków na rys. 2.



Rys. 1. Stanowisko badawczo-dydaktyczne

Źródło: opracowanie własne.



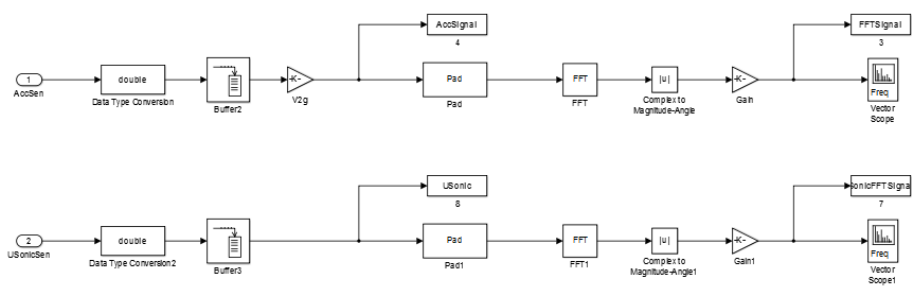
Rys. 2. Schemat transmisji sygnałów pomiarowych

Źródło: opracowanie własne.

<sup>3</sup> G. Piecuch, *Zastosowanie metod inteligencji obliczeniowej w diagnozowaniu wrzeciona*, praca magisterska, Politechnika Rzeszowska, Rzeszów 2015.

Na warstwę programową po stronie komputera PC składają się pakiet Matlab/Simulink oraz oprogramowanie przemysłowe TwinCAT3 służące do obsługi komputera przemysłowego oraz modułów I/O. Oba środowiska są ze sobą zintegrowane za pomocą specjalnego modułu TE1400 (TC3 Target for Matlab/Simulink). Dzięki temu możliwe jest zaprojektowanie programu sterującego w pakiecie Matlab/Simulink oraz uruchomienie go w trybie *External* na komputerze przemysłowym, wraz z podglądem zmiennych i zapisem danych w czasie rzeczywistym.

Do akwizycji danych pomiarowych w Simulinku należy przygotować schematy jak przedstawione na rys. 3 oraz odpowiednio skonfigurować parametry pracy, m.in. takie jak cykl programu, częstotliwość zapisu danych, przypisanie zmiennych do rzeczywistych wejść modułów.



**Rys. 3. Schemat przygotowany w Simulinku służący do rejestracji danych pomiarowych: z akcelerometru (górny), z czujnika ultradźwiękowego (dolny)**

Źródło: opracowanie własne.

## Klasyfikacja stanu pracy maszyny

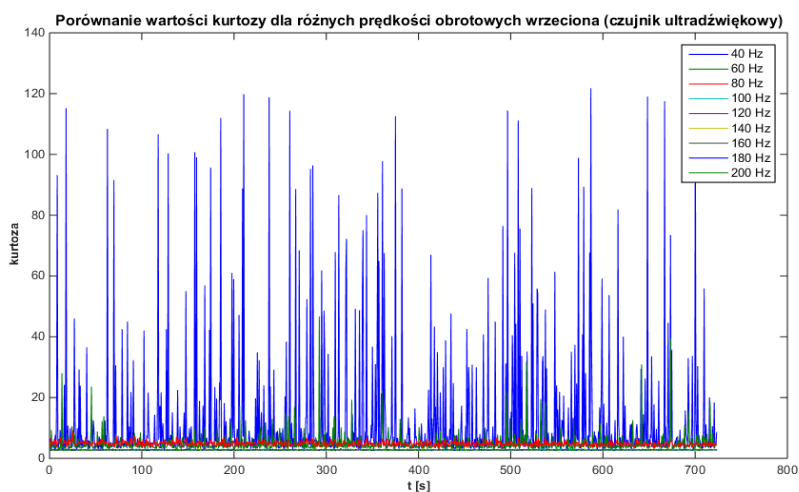
Metody klasyfikacji są konstruowane tak, aby umożliwiały automatyczne rozpoznawanie wzorców (ang. *Pattern Recognition*). Zjawiska bazujące m.in. na drganiach układu, ultradźwiękach, pomiarach prądu itp. są reprezentowane za pomocą zespołu cech, tzw. atrybutów. Zadaniem klasyfikatora jest rozpoznanie nieznanego obiektu (zjawiska) i przyporządkowanie go do jednej z określonych wcześniej kategorii, zwanej klasą. Niezbędne do opracowania i nauczania klasyfikatora jest zarejestrowanie danych, będących zbiorem obiektów uczących, których przyporządkowanie do poszczególnych klas jest z góry znane. W uproszczeniu, ideę klasyfikatora można określić jako próbę poszukiwania wśród obiektów znanych z danych uczących, obiektu najbardziej podobnego do tego aktualnie badanego<sup>4</sup>.

<sup>4</sup> M. Flasiński, *Wstęp do sztucznej inteligencji*, PWN, Warszawa 2011.

Zadaniem studentów jest opracowanie kilku klasyfikatorów, za pomocą których rozpoznawać będą prędkość obrotową wrzeciona, a następnie porównanie skuteczności użytych metod. W ćwiczeniach proponuje się zastosowanie np.: naiwnego klasyfikatora Bayesa, metody kNN, SVM, drzewa decyzyjnego oraz sieci neuronowych. Mając do czynienia z taką lub jeszcze większą liczbą klasyfikatorów niezbędne jest umiejętne porównanie ich skuteczności. Miarodajnymi wskaźnikami, dzięki którym można tego dokonać są m.in.: macierze rozbieżności (pomyłek), wskaźniki skuteczności takie jak: wrażliwość, specyficzność i dokładność oraz krzywe ROC i wskaźnik AUC (pole pod krzywą ROC)<sup>5</sup>.

## Ćwiczenie – detekcja prędkości

Operator (student) zadaje na falowniku sterującym pracą wrzeciona wartość prędkości obrotowej, z jaką ma ono wirować. Opracowany system każdorazowo powinien zwracać klasę odpowiadającą danej prędkości. Rozpoznawanie jej najlepiej bazować na czujniku ultradźwiękowym. W opisywanym przypadku jest to 9 prędkości z zakresu 40–200 Hz, z krokiem co 20 Hz. Dla każdego przypadku należy zarejestrować dane pomiarowe. Czas rejestracji dobiera się indywidualnie, wedle uznania, ale nie powinien on wynosić mniej niż 5 sekund, ze względu na to, aby zgromadzić możliwie dużo danych uczących.

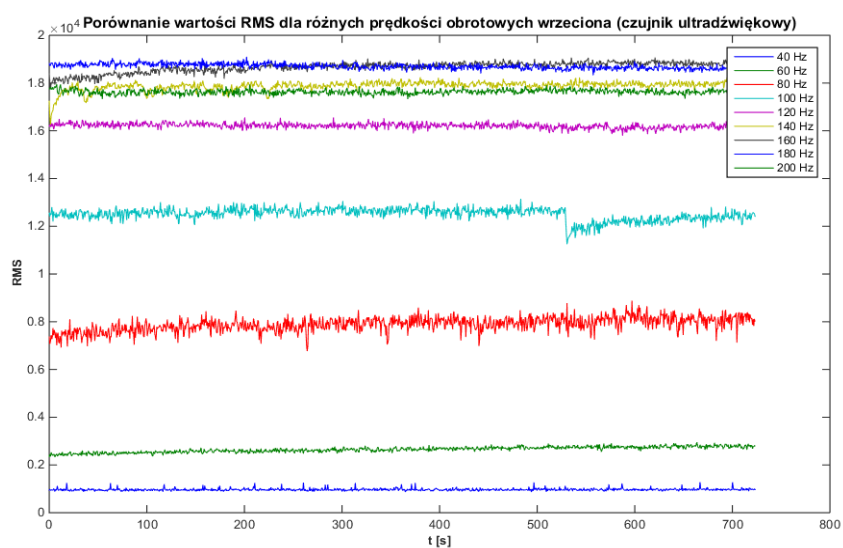


**Rys. 4. Wartość kurtozy dla różnych prędkości wrzeciona (niemożliwe rozróżnienie)**

Źródło: opracowanie własne.

<sup>5</sup> J. Stefanowski, *Zaawansowana eksploracja danych: Metody oceny wiedzy klasyfikacyjnej odkrytej z danych – materiały wykładowe*, Instytut Informatyki, Politechnika Poznańska, Poznań, 2010.

Do celów dydaktycznych i badawczych zaleca się, aby było to nawet kilka minut. Tak zgromadzone dane należy odpowiednio przetworzyć i wyznaczyć w dziedzinie czasu kilka atrybutów np.: RMS, kurtozę, skośność, średnią, odchylenie standardowe, a następnie wykreślić ich wykresy w ten sposób, aby zaobserwować dla jednej miary wszystkie rozpatrywane klasy. Powinno okazać się, że nie wszystkie z obliczonych miar będą przydatne do nauczania klasyfikatora. Ze względu na to, że przebiegi będą się na siebie nakładać, niemożliwe będzie rozróżnienie konkretnego przypadku. Przykład źle dobranego atrybutu (kurtoza) pokazano na rys. 4, a dobrego na rys. 5 (RMS).



**Rys. 5. Wartość RMS dla różnych prędkości wrzeciona (możliwe rozróżnienie)**

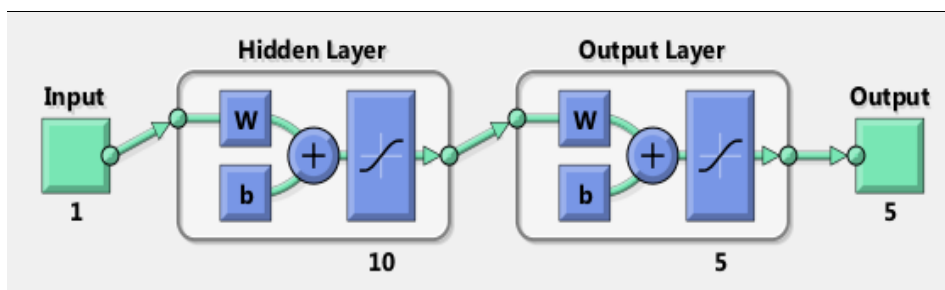
Źródło: opracowanie własne.

Jak można zaobserwować na rys. 5, prędkości z zakresu 140–200 Hz mogą być słabo rozróżnialne między sobą, stąd do przeprowadzenia klasyfikacji proponuje się:

- Klasa 1 – 40 Hz;
- Klasa 2 – 60 Hz;
- Klasa 3 – 80 Hz;
- Klasa 4 – 100 Hz;
- Klasa 5 – 120 Hz.

W dalszej części artykułu przykład opisywanego problemu rozwiązano za pomocą sieci neuronowej i pakietu *Neural Network Toolbox* dostępnego w środowisku Matlab/Simulink, ze względu na to, że tak przygotowany klasyfikator można uruchomić w czasie rzeczywistym. Za pomocą narzędzia *Neural*

*Network Pattern Recognition Tool (nprtool)* nauczono sieć oraz wygenerowano gotowy jej model w formie bloczka w Simulinku, który został w dalszej kolejności użyty do zaprojektowania schematu rozpoznawania prędkości online. Architektura sieci pokazana została na rys. 6. Jest to dwuwarstwowa sieć typu feed-forward z sigmoidalnymi neuronami ukrytymi i wyjściowymi, a także wsteczną propagacją błędów<sup>6</sup>. Wybrano ją ze względu na to, że jest uniwersalnym aproksymatorem nieliniowym<sup>7</sup>.



**Rys. 6. Architektura dwuwarstwowej sieci neuronowej wykorzystywanej w klasyfikacji prędkości obrotowej wrzeciona.**

Źródło: opracowanie własne.

Aby uniknąć nadmiernego dopasowania sieci (ang. *overfitting*), należy zadbać o to, aby nie wszystkie dostępne dane były używane w procesie uczenia. W tym celu stosuje się różne techniki, nazywane empirycznymi metodami estymacji: hold-out, krosvalidacja (ocena krzyżowa), leaving-one-out, bootstraping<sup>8</sup>. Wykorzystywane narzędzie ma wbudowany algorytm hold-out służący do podziału danych na uczące, walidacyjne i testowe, stąd łatwiej do celów dydaktycznych posłużyć się tą właśnie metodą. W kolejnym etapie należy nauczyć klasyfikator, co niejako dzieje się już automatycznie – wystarczy uprzednio wskazać zbiór danych. Jeśli proces nauki przebiegł pomyślnie i wstępny test wykazał bardzo dobrą skuteczność (w opisywanym przypadku 100%), można przejść do testów w czasie rzeczywistym.

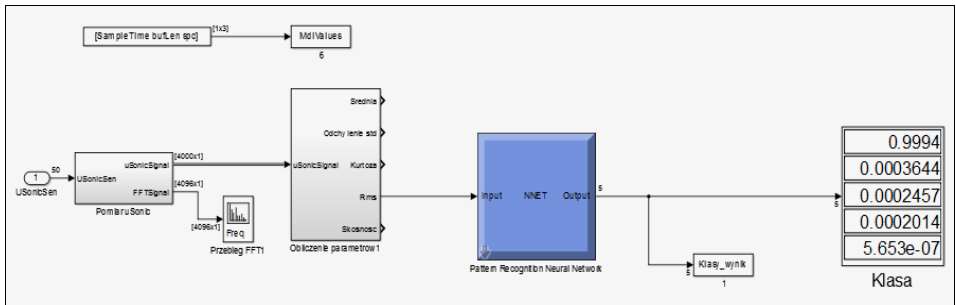
Utworzona sieć neuronowa (wygenerowany niebieski blok – widoczny na rys. 7) została włączona w zaprojektowany schemat służący testowaniu klasyfikatora w czasie rzeczywistym. Na rys. 7 pokazano rozpoznanie klasy 1 (40 Hz).

<sup>6</sup> Matlab&Simulink *Neural Network Pattern Recognition Tool* – instrukcja użytkownika.

<sup>7</sup> P. Wawrzyński, *Uczenie maszynowe – sztuczne sieci neuronowe, Materiały wykładowe*, PSZT.

<sup>8</sup> J. Stefanowski, *Zaawansowana eksploracja...*; C. Sammut, G. Webb, *Encyclopedia of Machine Learning*, Springer 2010, p. 600–601; P. Boniecki, W. Mueller, *Neuronowe techniki klasyfikacyjne w problemach identyfikacyjnych inżynierii rolniczej*, “Journal of Research and Applications in Agricultural Engineering” 2005, Vol. 50(3), p. 17.

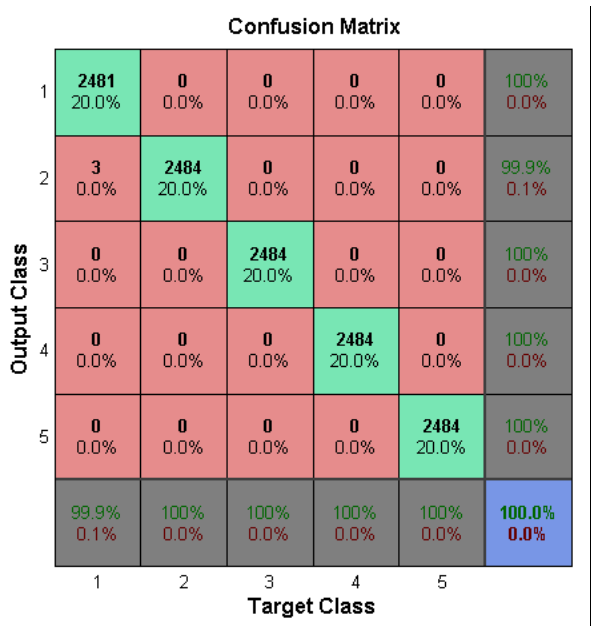
Wartość bliska 1, w danym wierszu okna *Klasa* informuje o rozpoznaniu próbek jako (odpowiednio od góry) klasy 1 do klasy 5.



**Rys. 7. Projekt klasyfikatora (Simulink) działający w czasie rzeczywistym w środowisku TwinCAT 3 na komputerze IPC**

Źródło: opracowanie własne.

Wyniki testów przeprowadzonych w czasie rzeczywistym potwierdziły praktycznie 100-procentową dokładność klasyfikatora uzyskaną w procesie uczenia. Na rys. 8 przedstawiono wynikową macierz rozbieżności.



**Rys. 8. Macierz rozbieżności testów przeprowadzonych w czasie rzeczywistym dla sieci neuronowej rozpoznającej prędkość obrotową wrzeczona**

Źródło: opracowanie własne.

## Zakończenie

W artykule przedstawiono propozycję ćwiczenia, które poprzez swój praktyczny charakter doskonale nadaje się do nauki metod inteligencji obliczeniowej w ramach kształcenia wyższego na kierunkach technicznych, zwłaszcza automatyce i robotyce. Omówiony projekt i jego rozwiązanie jest tylko przykładem i w zależności od wybranego sprzętu, jego konfiguracji, a nawet sposobu i miejsca umieszczenia czujników będzie w każdym przypadku się różnił, także otrzymywanymi wynikami. Własny przypadek należy przeanalizować od początku, pomocne mogą się jednak okazać wskazówki i sposób podejścia do tematu, które omówiono w niniejszym artykule.

## Bibliografia

- Boniecki P., Mueller W., *Neuronowe techniki klasyfikacyjne w problemach identyfikacyjnych inżynierii rolniczej*, „Journal of Research and Applications in Agricultural Engineering” 2005, Vol. 50(3), p. 17.
- Flasiński M., *Wstęp do sztucznej inteligencji*, PWN, Warszawa 2011.
- Jemielniak K., Kosmol J., *Diagnostyka narzędzia i procesu skrawania – stan aktualny i kierunki rozwoju*, „Mechanik” 1996, nr 10.
- Krawiec K., Stefanowski J., *Uczenie maszynowe i sieci neuronowe*. Wydawnictwo Politechniki Poznańskiej, Poznań 2003.
- Matlab&Simulink *Neural Network Pattern Recognition Tool – instrukcja użytkowania*.
- Piecuch G., *Zastosowanie metod inteligencji obliczeniowej w diagnozowaniu wrzeciona*, praca magisterska, Politechnika Rzeszowska, Rzeszów 2015.
- Sammut C., Webb G., *Encyclopedia of Machine Learning*, Springer 2010, p. 600–601.
- Stefanowski J., *Zaawansowana eksploracja danych: Metody oceny wiedzy klasyfikacyjnej odkrytej z danych – materiały wykładowe*, Instytut Informatyki, Politechnika Poznańska, Poznań 2010.
- Wawrzyński P., *Uczenie maszynowe – sztuczne sieci neuronowe, Materiały wykładowe*, PSZT.