dr inż. Maciej Kusy
Katedra Podstaw Elektroniki
Wydział Elektrotechniki i Informatyki
Politechnika Rzeszowska
al. Powstańców Warszawy 12
35–959 Rzeszów

Rzeszów, 17 września 2018 r.
Załącznik 2a

AUTOREFERAT

przedstawiający opis podstawowego osiągnięcia i pozostałego dorobku naukowego w związku z ubieganiem się o nadanie stopnia doktora habilitowanego

1 Posiadane dyplomy, stopnie naukowe

08.10.2008 Doktor nauk technicznych
Wydział Mechatroniki Politechniki Warszawskiej
Dyscyplina: **biocybernetyka i inżynieria biomedyczna**
Tytuł rozprawy: *System for Cancer Diagnosis Based on Support Vector Machines and Neural Networks* (praca w języku angielskim)

28.09.2000 Magister inżynier
Wydział Elektryczny Politechniki Rzeszowskiej
Specjalność: **aparatura elektroniczna**
Tytuł pracy: *Support Vector Machines for Two Data Sets Classification* (praca w języku angielskim)

2 Dotychczasowe zatrudnienie w jednostkach naukowych

01.11.2008 – Wydział Elektrotechniki i Informatyki Politechniki Rzeszowskiej
Katedra Podstaw Elektroniki
adiunkt

20.11.2000 – – 31.10.2008 Wydział Elektrotechniki i Informatyki Politechniki Rzeszowskiej
Katedra Podstaw Elektroniki
asystent

3 Podstawowe osiągnięcie naukowe

Jako osiągnięcie naukowe, o którym mowa w art. 16 ust. 2 ustawy z dnia 14 marca 2003 r. o stopniach naukowych i tytule naukowym oraz o stopniach i tytule w zakresie sztuki (Dz. U. 2016 r. poz. 882 ze zm. w Dz. U. z 2016 r. poz. 1311), wskazuję

jednotematyczny cykl publikacji pod wspólnym tytułem:

Zastosowanie technik uczenia maszynowego w modelowaniu probabilistycznej sieci neuronowej.

3.1 Publikacje wchodzące w skład osiągnięcia naukowego

Sumaryczny wskaźnik 5–year Impact Factor (w skrócie 5–letni IF) siedmiu publikacji, przedstawionych w ramach cyklu, według bazy Journal Citation Raport (JCR) wynosi **30,524**. Sumaryczna liczba punktów według MNiSW, zgodnie z obowiązującym w roku wydania publikacji wykazem czasopism naukowych, jest równa **215**.

Na jednotematyczny cykl publikacji składa się 7 artykułów opublikowanych w czasopismach z listy JCR:

[C1] **M. Kusy** i R. Zajdel: "Application of Reinforcement Learning Algorithms for the Adaptive Computation of the Smoothing Parameter for Probabilistic Neural Network", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 26, nr 9, s. 2163–2175 (2015). DOI: 10.1109/TNNLS.2014.2376703
JCR, 5–letni IF: 7,658; MNiSW: 45 pkt.
Mój wkład w powstanie tej publikacji szacuję na **50%**; obejmował on wszystkie etapy przygotowywania pracy, w szczególności: a) opracowywanie jej koncepcji i układu oraz sformułowanie celów badań - 50%; b) projekt i implementację algorytmów oraz wyprowadzenie wyników - 50%; c) analizę i interpretację wyników - 50%; d) przygotowanie publikacji do druku - 50%.

[C2] **M. Kusy** i R. Zajdel: "Probabilistic neural network training procedure based on Q(0)–learning algorithm in medical data classification", *Applied Intelligence*, vol. 41, nr 3, s. 837–854 (2014). DOI: 10.1007/ s10489-014-0562-9
JCR, 5–letni IF: 1,856; MNiSW: 10 pkt.
Mój wkład w powstanie tej publikacji szacuję na **50%**; obejmował on wszystkie etapy przygotowywania pracy, w szczególności: a) opracowywanie jej koncepcji i układu oraz sformułowanie celów badań - 50%; b) projekt i implementację algorytmów oraz wyprowadzenie wyników - 50%; c) analizę i interpretację wyników - 50%; d) przygotowanie publikacji do druku - 50%.
Wyjaśnienie: Artykuł wyszczególniony w poz. [C2] został wysłany do czasopisma *Applied Intelligence* (*APIN*) 28 listopada 2013 r. Czasopismo *APIN* obowiązywał wówczas wskaźnik IF z 2012 r, który wynosił 1,853. W 2013 r. i 2014 r. (rok publikacji artykułu) czasopismo *APIN* nie posiadało wskaźnika IF. Według informacji w bazie JCR z dnia 17 września 2018 r., wskaźnik IF dla czasopisma *APIN* za 2015 r., 2016 r. i 2017 r. wynosi odpowiednio: 1,215; 1,904; 1,983. Czasopismo *APIN* nie zostało zamieszczone na liście A wykazu czasopism w 2014 r. Ze względu na to, iż liczbę punktów przyznawaną za publikację wyszczególniono zgodnie z obowiązującym w roku wydania wykazem czasopism naukowych, liczba punktów MNiSW za artykuł [C2] równa jest 10.

[C3] P.A. Kowalski i **M. Kusy**: "Sensitivity Analysis for Probabilistic Neural Network Structure Reduction", *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, nr 5, s. 1919–1932 (2018). DOI: 10.1109/TNNLS.2017.2688482
JCR, 5–letni IF: 7,658; MNiSW: 45 pkt.
Mój wkład w powstanie tej publikacji szacuję na **50%**; obejmował on wszystkie etapy przygotowywania pracy, w szczególności: a) opracowywanie jej koncepcji i układu oraz sformułowanie celów badań - 50%; b) projekt i implementację algorytmów oraz wyprowadzenie wyników - 50%; c) analizę i interpretację wyników - 50%; d) przygotowanie publikacji do druku - 50%.

[C4] P.A. Kowalski i **M. Kusy**: "Determining significance of input neurons for probabilistic neural network by sensitivity analysis procedure", *Computational Intelligence*, vol. 34, nr 3, s. 895–916 (2018). DOI: 10.1111/coin.12149
JCR, 5–letni IF: 1,545; MNiSW: 20 pkt.

Mój wkład w powstanie tej publikacji szacuję na **50%**; obejmował on wszystkie etapy przygotowywania pracy, w szczególności: a) opracowywanie jej koncepcji i układu oraz sformułowanie celów badań - 50%; b) projekt i implementację algorytmów oraz wyprowadzenie wyników - 50%; c) analizę i interpretację wyników - 50%; d) przygotowanie publikacji do druku - 50%.

[C5] **M. Kusy** i P.A. Kowalski: "Weighted probabilistic neural network", *Information Sciences*, vol. 430–431, s. 65–76 (2018). DOI: 10.1016/j.ins.2017.11.036
JCR, 5–letni IF: 4,378; MNiSW: 45 pkt.
Mój wkład w powstanie tej publikacji szacuję na **50%**; obejmował on wszystkie etapy przygotowywania pracy, w szczególności: a) opracowywanie jej koncepcji i układu oraz sformułowanie celów badań - 50%; b) projekt i implementację algorytmów oraz wyprowadzenie wyników - 50%; c) analizę i interpretację wyników - 50%; d) przygotowanie publikacji do druku - 50%.

[C6] **M. Kusy** i J. Kluska: "Assessment of prediction ability for reduced probabilistic neural network in data classification problems", *Soft Computing*, vol. 21, nr 1, s. 199–212 (2017). DOI: 10.1007/s00500-016- 2382-9
JCR, 5–letni IF: 2,204; MNiSW: 25 pkt.
Mój wkład w powstanie tej publikacji szacuję na **50%**; obejmował on wszystkie etapy przygotowywania pracy, w szczególności: a) opracowywanie jej koncepcji i układu oraz sformułowanie celów badań - 50%; b) projekt i implementację algorytmów oraz wyprowadzenie wyników - 50%; c) analizę i interpretację wyników - 50%; d) przygotowanie publikacji do druku - 50%.

[C7] **M. Kusy**: "Fuzzy c–means-based architecture reduction of a probabilistic neural network", *Neural Networks*, vol. 108, s. 20–32 (2018). DOI: 10.1016/j.neunet.2018.07.012
JCR, 5–letni IF: 5,225; MNiSW: 30 pkt.

3.2 Opis osiągnięcia naukowego

Stosując aparat matematyczny można tworzyć algorytmy, które potrafią automatycznie udoskonalać swoją wydajność w danym zadaniu, wykorzystując do tego celu dane wejściowe. Zdolność ta określana jest jako uczenie maszynowe (ang. *machine learning, ML*). Uczenie maszynowe stanowi obecnie gałąź sztucznej inteligencji, która to jest odrębnym działem szeroko pojętej informatyki. Wyróżnia się trzy kategorie ML: uczenie nadzorowane, uczenie nienadzorowane oraz uczenie się ze wzmocnieniem [L1]. Algorytmy ML znajdują zastosowanie w klasyfikacji, regresji, klasteryzacji i redukcji wymiarowości, co jest nazywane mianem eksploracji danych.

Mając na uwadze możliwości aplikacyjne algorytmów ML, jako cel swoich badań obrałem wykorzystanie wybranych, ale odrębnych kategorii, technik uczenia maszynowego do modelowania probabilistycznej sieci neuronowej (ang. *probabilistic neural network, PNN*). PNN jest siecią jednokierunkową, wielowarstwową, stosowaną w klasyfikacji danych. Mimo że jest ona w literaturze znana od wielu lat – pierwsza praca opublikowana przez D. Spechta w 1990 r. [L2] – jest nadal powszechnie używana m.in. w: diagnostyce medycznej [L3, L4], rozpoznawaniu obrazów [L5, L6], przetwarzaniu informacji niedokładnej typu przedziałowego [L7], klasyfikacji w zmieniającym się środowisku [L8], zabezpieczaniu wiadomości elektronicznych [L9], a nawet została zaimplementowana sprzętowo [L10].

Niniejszy rozdział przedstawia omówienie wkładu w rozwój dyscypliny, opis tematyki oraz wyników w ramach mojego jednotematycznego cyklu publikacji.

Wkład w rozwój dyscypliny

Przedmiotem prowadzonych przeze mnie badań, których wyniki zaprezentowane zostały w postaci cyklu publikacji powiązanych tematycznie, jest zastosowanie wybranych technik uczenia maszynowego w procesie modelowania probabilistycznej sieci neuronowej w klasyfikacji danych. Jako modelowanie rozumiane tu są: (i) wpływ na przebieg uczenia, (ii) uproszczenie architektury oraz (iii) wprowadzenie współczynników strukturalnych w rozważanej sieci neuronowej.

Do moich najważniejszych osiągnięć, przedstawionych w publikacjach [C1]–[C7], i wnoszących istotny wkład w rozwój dyscypliny informatyka, zaliczam:

1) zaproponowanie i implementację nowych algorytmów uczenia dla PNN w oparciu o wybrane metody uczenia się ze wzmocnieniem [C1, C2];

2) opracowanie poprzez opis formalny oraz implementację nowego podejścia redukcji liczby neuronów wejściowych i wzorcowych PNN w oparciu o procedurę analizy czułości [C3];

3) zaprojektowanie algorytmu podającego istotność poszczególnych neuronów wejściowych PNN na podstawie procedury lokalnej analizy czułości [C4];

4) przedstawienie sposobu modyfikacji PNN poprzez wprowadzenie współczynników wagowych do jej struktury [C5];

5) zastosowanie klasycznej i rozmytej klasteryzacji danych do selekcji neuronów wzorcowych PNN [C6, C7];

6) wykorzystanie wektorów wspierających w budowie PNN [C6].

W poniższych sekcjach zaprezentowano wstęp do tematyki oraz całościowy opis prac w ramach jednotematycznego cyklu publikacji.

3.2.1 Ogólne przedstawienie tematyki

Jak wiadomo, probabilistyczna sieć neuronowa łączy w sobie prostotę i dużą skuteczność w rozwiązywaniu zadań klasyfikacji danych. Po dokonaniu przeglądu literaturowego zaobserwowałem, że istnieje sporo alternatyw w stosunku do opracowanych algorytmów uczenia tej sieci oraz luki w zakresie optymalizacji jej architektury.

Ze strukturalnego punktu widzenia, PNN jest siecią składającą się z czterech warstw: wejściowej, wzorców, sumującej i wyjściowej. Warstwa wejściowa zbudowana jest z elementów, które są atrybutami danych; warstwa wzorców zawiera tyle neuronów ile wzorców uczących; w warstwie sumującej znajduje się jeden neuron dla każdej z klas; warstwa wyjściowa to pojedynczy neuron odpowiedzialny za finalny wynik klasyfikacji. Struktura PNN jest więc bardzo złożona, gdyż zależy od ilości rekordów oraz liczby klas w rozpatrywanym zbiorze danych. W związku z tym wykorzystanie tej sieci w klasyfikacji dużych zbiorów danych wiąże się z długim czasem obliczeń. Od kilkudziesięciu lat wielu autorów proponuje w swoich publikacjach różne podejścia w celu zredukowania architektury PNN. Prace te skupiają się przede wszystkim na zmniejszeniu liczby neuronów w warstwie wzorców. Należy tutaj wymienić następujące rozwiązania: kwantyzacja wektorowa [L11], klasteryzacja hierarchiczna [L12], ortogonalizacja liniowa [L13], algorytm k–średnich [L14], ekstrakcja cech połączona z klasteryzacją rekordów [L15], konstrukcja przez skalowanie [L16].

Oprócz problematyki związanej ze strukturą PNN, ważnym aspektem tej sieci jest proces uczenia. Jest on ściśle związany ze sposobem aktywacji neuronów w warstwie sumowania. Z tego powodu w literaturze rozpatruje się najczęściej dwa typy PNN: sieć o funkcji jądra

w postaci krzywej Gaussa oraz sieć o jądrze produktowym, wykorzystującym funkcję Cauchy'ego. W obydwu przypadkach parametrem w procesie uczenia jest współczynnik wygładzania. Do tej pory powstało wiele procedur umożliwiających wyznaczanie tego współczynnika w celu zminimalizowania błędu uczenia PNN. Do najpopularniejszych rozwiązań należą: metoda gradientów sprzężonych [L17], algorytm genetyczny [L13], procedura złotego podziału [L18], optymalizacja rojem cząstek [L19] oraz metoda pluginów [L20].

W przypadku PNN o funkcji jądra w postaci krzywej Gaussa, po podaniu wektora $\mathbf{x} = [x_1, \ldots, x_n]$ na wejście sieci, przetworzeniu go przez warstwę wzorców, j-ty neuron w warstwie sumowania generuje następujący sygnał:

$$f_j(\mathbf{x}) = \frac{1}{L_j (2\pi)^{n/2} \det \mathbf{H}_j} \sum_{l=1}^{L_j} \exp\left(-\frac{1}{2} \left(\mathbf{x} - \mathbf{x}_j^{(l)} \right)^T \mathbf{H}_j^{-2} \left(\mathbf{x} - \mathbf{x}_j^{(l)} \right) \right), \tag{1}$$

gdzie L_j jest liczbą danych uczących klasy o etykiecie j, $\mathbf{H}_j = \mathrm{diag}\,(h_{j1}, h_{j2}, \ldots, h_{jn})$, natomiast $\mathbf{x}_j^{(l)} = \left[x_{j1}^{(l)}, \ldots, x_{jn}^{(l)} \right]$ oznacza l-ty wzorzec uczący j-tej klasy. Współczynniki wygładzania w macierzy $\mathbf{H}_j$ mogą mieć następującą reprezentację:

- wartość skalarna skojarzona z całą siecią;
- składowa wektora skojarzona z pojedynczą cechą danych wejściowych;
- element macierzy skojarzony z pojedynczą cechą danych wejściowych oraz każdą j-tą klasą.

Dla PNN wzbogaconej o koncepcję jądra produktowego, w którym wykorzystywana jest funkcja Cauchy'ego, j-ty sumator wyznacza się w następujący sposób:

$$f_j(\mathbf{x}) = \frac{1}{L_j \det \mathbf{H}_j} \sum_{l=1}^{L_j} \frac{1}{s_l^n} K\left(\frac{1}{s_l} \left(\mathbf{x} - \mathbf{x}_j^{(l)} \right)^T \mathbf{H}_j^{-1} \right), \tag{2}$$

gdzie $K(\mathbf{x}) = (2/\pi)^n \prod_{i=1}^{n} \frac{1}{(x_i^2+1)^2}$. We wzorze (2) s_l jest parametrem modyfikacji, który jest niezależnie wyznaczany dla wszystkich wzorców uczących [L20].

Dla uproszczenia w późniejszych odwołaniach przyjmuje się h jako oznaczenie parametru wygładzania bez względu na reprezentację.

W tym miejscu należy również podkreślić, iż PNN nie posiada współczynników wagowych w poszczególnych warstwach. Przemawia to na pewno na jej korzyść, gdyż znacznie skraca się proces uczenia. Z drugiej jednak strony, brak wag pozbawia sieć możliwości dostrojenia i zminimalizowania błędu zarówno na etapie uczenia, jak i predykcji.

3.2.2 Szczegółowa charakterystyka badań i uzyskanych wyników

Uczenie się ze wzmocnieniem jako metoda nauki probabilistycznej sieci neuronowej
Zaproponowane w pracach [C1] oraz [C2] algorytmy uczenia dotyczą PNN, dla której w warstwie sumowania wyjście j-tego neuronu wyznacza się w oparciu o formułę (1). Algorytmy te polegają na dostosowaniu uczenia się ze wzmocnieniem (ang. *reinforcement learning, RL*) [L21] do aktualizacji wartości współczynnika h. Wykorzystywane są w tym celu różne warianty RL.

Celem algorytmu RL jest nauczenie się strategii, która doprowadzi agenta do optymalnego rozwiązania zadanego problemu. Proces uczenia sprowadza się do interakcji agenta ze

środowiskiem i może być realizowany za pomocą różnych podejść. Najbardziej popularnym podejściem jest uaktualnianie w sposób iteracyjny funkcji wartości akcji $Q_t(s_t, a_t)$ [L22]:

$$Q_t(s_t, a_t) = Q_{t-1}(s_t, a_t) + \alpha \left(r_t + \gamma \max_a Q_{t-1}(s_{t+1}, a) - Q_{t-1}(s_t, a_t) \right), \tag{3}$$

gdzie t jest dyskretnym krokiem czasu; $s_t \in S$ to stan, w jakim znajduje się agent; $a_t \in A$ oznacza akcję wybraną zgodnie ze strategią działania agenta, natomiast r_t to sygnał wzmocnienia. Algorytm wykorzystujący taką aktualizację funkcji wartości akcji nosi nazwę $Q(0)$–learning. Istnieje również udoskonalona wersja tego algorytmu, $Q(\lambda)$–learning, w której wyznaczanie funkcji wartości akcji wzbogacone jest o tzw. historię aktywności par stan-akcja. W obydwu algorytmach funkcja Q reprezentowana jest w postaci tablicowej.

Do najważniejszych zadań w algorytmie RL zalicza się ustalenie postaci środowiska, a co za tym idzie dobranie zbioru stanów S, zbioru akcji A i sygnału r_t oraz dostosowanie procedury aktualizacji funkcji wartości akcji do zadanego problemu. Aby zrealizować procedurę uczenia probabilistycznej sieci neuronowej, środowisko musiało zostać dostosowane do problemu klasyfikacji danych za pomocą PNN. W pracach [C1] oraz [C2] przyjęto, że środowisko składa się ze zbioru danych wejściowych w formie par $\langle \mathbf{x}^{(l)}, t^{(l)} \rangle$ dla $l = 1, \ldots, L$, gdzie L jest licznością danych, klasyfikatora PNN oraz wskaźnika jakości klasyfikacji.

W publikacji [C1] zbiór stanów S został zdefiniowany za pomocą miary błędu klasyfikacji:

$$\mathcal{E} = \frac{1}{L} \sum_{l=1}^{L} \delta \left[o(\mathbf{x}^{(l)}) \neq t^{(l)} \right], \tag{4}$$

gdzie $o(\mathbf{x}^{(l)})$ jest wartością na wyjściu PNN, wyznaczoną dla $\mathbf{x}^{(l)}$, natomiast $\delta[\cdot] = 1$, jeżeli $o(\mathbf{x}^{(l)}) \neq t^{(l)}$ i 0 w przeciwnym przypadku. Stąd $S = \{0, 1/L, 2/L, \ldots, (L-1)/L, 1\}$. Takie rozwiązanie ma dość naturalną interpretację: aktualny błąd sieci w procesie uczenia jest stanem, w którym w danej chwili znajduje się agent.

Akcje zdefiniowano jako wartości z symetrycznego zbioru $A = \{-a^{(1)}, -a^{(2)}, \ldots, -a^{(P)}, a^{(P)}, \ldots, a^{(2)}, a^{(1)}\}$. Elementy $a^{(p)} \in A$ zostały bezpośrednio wykorzystane do modyfikacji parametru h według wzoru:

$$h_t = h_{t-1} + a_t, \tag{5}$$

przez co współczynnik wygładzania zwiększa lub zmniejsza wartości w każdym kroku uczenia PNN. W algorytmie zaproponowano sześcioelementowy zbiór akcji $A = \{$–1 –0,1 –0,01 0,01 0,1 1$\}$.

Podstawą aktualizacji funkcji wartości akcji jest błąd różnic czasowych, którego integralnym elementem jest sygnał wzmocnienia r. Zadaniem sygnału r jest przekazanie uczącemu się agentowi informacji o pożytku z podjętej w stanie s_t akcji a_t. Ponieważ w omawianej implementacji istotą modyfikacji parametru PNN jest minimalizacja błędu klasyfikacji (4), naturalnym jest zaproponowanie takiej postaci sygnału wzmocnienia, która będzie nagradzać zmniejszanie błędu i karać jego zwiększanie. Sygnał wzmocnienia dany zależnością:

$$r_t = \mathcal{E}_{t-1}(h_{t-1}) - \mathcal{E}_t(h_t), \tag{6}$$

gdzie $\mathcal{E}_{t-1}(h_{t-1})$ oraz $\mathcal{E}_t(h_t)$ są błędami w poprzednim i obecnym kroku uczenia, nie tylko spełnia zadość tej idei, ale dodatkowo uwzględnia dynamikę zmian błędu klasyfikacji. Taka forma sygnału wzmocnienia, połączona z aktualizacją funkcji wartości akcji, utrwala agenta w przekonaniu, że wykonana przez niego akcja jest korzystna lub nie.

W omawianej pracy, do procesu aktualizacji wartości parametru wygładzania według formuły (5), zastosowano algorytmy: $Q(0)$–learning, $Q(\lambda)$–learning oraz Q–learning bezstanowy.

Do weryfikacji zdolności uogólniania PNN wykorzystano 10–częściowy sprawdzian krzyżowy. Cały algorytm uczenia PNN można podsumować w sposób następujący. Dla każdego kroku t należy:

1. wybrać akcję a_t wykorzystując aktualną strategię na podstawie funkcji wartości akcji;
2. uaktualnić wartość współczynnika wygładzania na podstawie formuły (5);
3. wyznaczyć błąd uczenia i testu klasyfikatora PNN;
4. obliczyć sygnał wzmocnienia r_t;
5. uaktualnić funkcję wartości akcji Q.

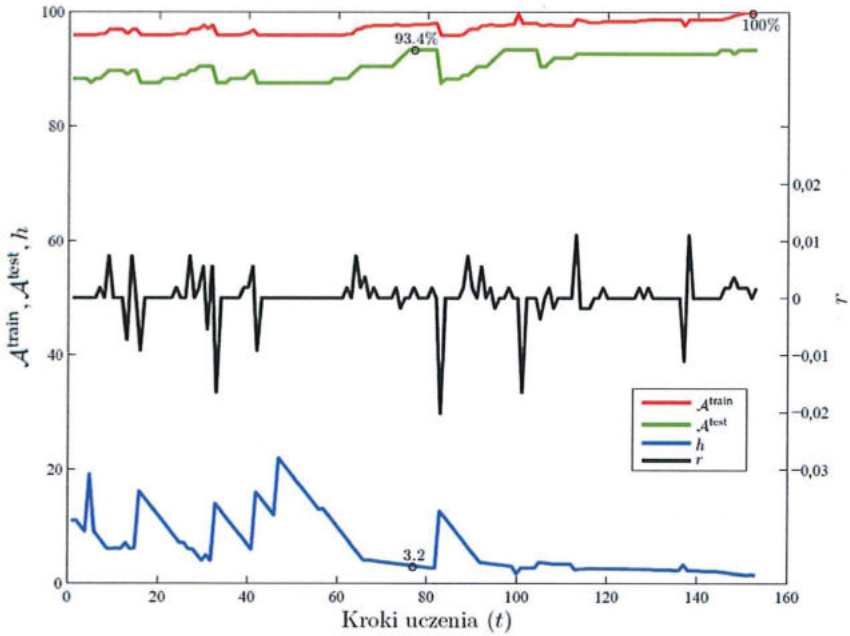
Szczegółowy pseudokod algorytmu uczenia PNN przy użyciu metod $Q(0)$–learning, $Q(\lambda)$–learning oraz Q–learning bezstanowy można znaleźć w artykule [C1].

Powyższy algorytm zastosowano w klasyfikacji ośmiu zbiorów danych, w tym siedmiu z repozytorium UCI–MLR [L23]. Został on zaimplementowany dla sieci, w której parametr h przyjmuje postać skalarną, wektorową oraz macierzową i jest wyznaczany za pomocą metody $Q(0)$–learning, $Q(\lambda)$–learning oraz Q–learning bezstanowy. Otrzymane rezultaty porównane zostały z wynikami, jakie otrzymano dla klasyfikatora PNN przy użyciu komercyjnego oprogramowania DTREG, w którym procedurą uczenia jest metoda gradientów sprzężonych. W artykule zaprezentowano również wyniki, które opublikowano w literaturze światowej. Na podstawie szczegółowej analizy porównawczej można wyciągnąć kilka głównych wniosków: (a) w czterech na osiem zbiorów danych błąd klasyfikacji zaproponowanych algorytmów jest niższy niż błąd PNN otrzymany za pomocą oprogramowania DTREG; (b) spośród wszystkich algorytmów RL wykorzystanych w procesie uczenia sieci, metody $Q(0)$–learning oraz $Q(\lambda)$–learning generują najniższy błąd klasyfikacji; (c) biorąc wyłącznie pod uwagę sposób doboru współczynnika wygładzania, najniższe błędy klasyfikacji daje sieć, w której przyjmuje on postać wektora, a jego elementy skojarzone są z każdym atrybutem wejściowym; (d) w porównaniu z wynikami jakie otrzymano dla PNN w literaturze światowej w klasyfikacji siedmiu zbiorów benchmarkowych, w pięciu przypadkach dzięki zaproponowanym algorytmom uzyskano niższe błędy klasyfikacji.

Artykuł [C2] przedstawia szczególny przypadek podejścia zaprezentowanego w pracy [C1]. Wprawdzie probabilistyczna sieć neuronowa jest także poddawana uczeniu algorytmem wykorzystującym RL, ale w tym wypadku jest to wyłącznie metoda $Q(0)$–learning. Sam sposób wyznaczania współczynników wygładzania jest również zmodyfikowany według nowej procedury. Dodatkowo współczynnik ten, oprócz trzech rozpatrywanych wcześniej form, przyjmuje postać wektora, w którym każdy element skojarzony jest z inną j-tą klasą, co powoduje, że neurony wzorcowe należące do danej klasy wykorzystują do aktywacji tę samą wartość h.

Tak jak to przedstawiono w formule (5), aktualizacja h jest zależna od wartości akcji w danym kroku t algorytmu. Podejście zaprezentowane w [C1] narzuca jeden możliwy zbiór akcji, jakie mogą być wykonane. W niniejszej pracy zbiór akcji A podzielony jest na M podzbiorów, gdzie $A^{(m)} = \{-a_1^{(m)}, -a_2^{(m)}, \ldots, -a_{p^{(m)}}^{(m)}, a_{p^{(m)}}^{(m)}, \ldots, a_2^{(m)}, a_1^{(m)}\}$ dla $m = 1, \ldots, M$; $p^{(m)}$ oznacza połowę liczności zbioru $A^{(m)}$ w m-tym kroku zaproponowanej procedury. Zbiór akcji powinien być wybierany tak, aby zapewnić $\max(A^{(1)}) > \ldots > \max(A^{(m)}) > \ldots > \max(A^{(M)})$. W rozważanych badaniach przyjęto $M = 3$ oraz zbiory $A^{(m)}$ w postaci:

$$\begin{aligned} A^{(1)} &= \{-10\ -1\ -0{,}1\ 0{,}1\ 1\ 10\}, \\ A^{(2)} &= \{-1\ -0{,}1\ -0{,}01\ 0{,}01\ 0{,}1\ 1\}, \\ A^{(3)} &= \{-0{,}1\ -0{,}01\ -0{,}001\ 0{,}001\ 0{,}01\ 0{,}1\}. \end{aligned} \tag{7}$$



Rysunek 1: Zmiany $\mathcal{A}^{\text{train}}$, $\mathcal{A}^{\text{test}}$ (wyrażone w %), h oraz r w trakcie procesu uczenia PNN przy współczynniku wygładzania w formie skalarnej w klasyfikacji danych raka piersi (partycja 80%–20%).

W eksperymentach zaprezentowanych w [C2] wyznaczono zdolność predykcji PNN za pomocą miary dokładności sieci, ogólnie mierzonej jako $\mathcal{A} = 1 - \mathcal{E}$, przy odpowiednim podziale danych wejściowych na zbiory uczące i testujące. Zbiory uczące powstały poprzez wylosowanie 90%, 80%, 70% oraz 60% rekordów z całego zbioru danych, natomiast zbiory testujące stworzono z pozostałych przypadków. Powstały zatem cztery partycje uczenie–test: 90%–10%, 80%–20%, 70%–30% oraz 60%–40%. Za takim podejściem przemawia fakt, iż liczba możliwych podziałów l rekordów wejściowych na v podzbiorów uczenie–test (każdy o liczności k) jest bardzo duża, gdyż wynosi $l!/(v! \cdot (k!)^v)$ [L24]. Wyniki dokładności przedstawiono w artykule dla każdego podziału z osobna, a także wartość maksymalną i średnią po wszystkich partycjach.

Wyznaczoną dokładność PNN (uwzględniając cztery formy współczynnika h) porównano z rezultatami otrzymanymi przez sieć zrealizowaną w komercyjnym oprogramowaniu DTREG oraz znanymi metodami sztucznej inteligencji (SI): algorytm wektorów wspierających (ang. *support vector machine, SVM*) [L25], algorytm wyrażeń genetycznych (ang. *gene expression programming, GEP*) [L26], metoda k–średnich (ang. *k–means*) [L27], perceptron wielowarstwowy (ang. *multilayer perceptron, MLP*) [L28], sieć neuronowa o radialnej funkcji aktywacji (ang. *radial basis function network, RBFN*) [L29] oraz sieć neuronowa o kwantyzacji wektorowej (ang. *learning vector quantization network, LVQN*) [L30]. Proponowany algorytm oraz wymienione powyżej referencyjne metody SI zastosowano do rozwiązania sześciu problemów klasyfikacji danych dostępnych w repozytorium UCI–MLR, wykorzystując takie same partycje danych. Otrzymane wyniki nasunęły dwie najważniejsze konkluzje: (i) na sześć możliwych problemów klasyfikacji, PNN uczona za pomocą zaproponowanego algorytmu trzykrotnie osiągnęła najwyższą średnią dokładność, podczas gdy algorytm SVM uzyskał najwyższą dokładność dwukrotnie, natomiast algorytm GEP – tylko raz; (ii) wybór parametru wygładzania w formie skalarnej, w porównaniu z pozostałymi wariantami doboru h, okazał się najgorszym rozwiązaniem w kontekście dokładności PNN na zbiorze testującym.

W pracy [C2] analizowano również zmiany dokładności, które wyznaczano na zbiorze

uczącym i testującym (odpowiednio $\mathcal{A}^{\text{train}}$ i $\mathcal{A}^{\text{test}}$), wartości h, oraz sygnału wzmocnienia r w funkcji kroków uczenia t dla każdego zbioru danych z osobna w przypadku, gdy współczynnik wygładzania przyjmuje wartość skalarną. Rysunek 1 przedstawia przebieg zmian tych parametrów w klasyfikacji danych raka piersi [L23]. Duże zmiany współczynnika h, obserwowane do 100-nej iteracji, o maksymalnie ± 10 powodowane są wyborem akcji a_t ze zbioru $A^{(1)}$. W późniejszych krokach uczenia ($t > 100$) modyfikacje współczynnika wygładzania nie są już takie duże, ponieważ akcje wybierane są ze zbioru $A^{(2)}$. Wartym odnotowania jest również fakt, iż sygnał wzmocnienia podąża za zmianami dokładności wyznaczanej na zbiorze uczącym. Jeżeli wartość A^{train} rośnie, r ma wartość dodatnią; natomiast jeżeli dokładność maleje, sygnał wzmocnienia od razu przyjmuje wartość ujemną. Podobna analiza została przeprowadzona w [C2] w przypadku pozostałych zbiorów danych dla przykładowych partycji uczenie–test.

Procedura analizy czułości w modyfikacji struktury probabilistycznej sieci neuronowej Jak już wspomniano w sekcji 3.2.1, ze względu na liczbę neuronów w warstwie wzorców równą liczności zbioru danych, PNN jest siecią neuronową o złożonej budowie. Dlatego też, jako jeden z celów w mojej pracy naukowej obrałem redukcję i modyfikację struktury tej sieci. Jednym z kilku rozważanych przeze mnie podejść było zastosowanie procedury lokalnej analizy czułości (ang. *local sensitivity analysis, LSA*) zaproponowanej w publikacji [L31] do zmniejszenia liczby neuronów w: (i) warstwie wejściowej, (ii) warstwie wzorców oraz (iii) warstwie wejściowej i wzorców jednocześnie. Do tego celu w artykule [C3] zostały zaproponowane trzy algorytmy. W tym wypadku rozpatrywana jest sieć, w której sygnał wyjściowy j-tego neuronu w warstwie sumowania oblicza się w oparciu o formułę (2).

Lokalna analiza czułości polega na wyznaczaniu po procesie uczenia sieci neuronowej wpływu atrybutów wejściowych x_i, $i = 1, \ldots, n$ na wyjście sieci y_j, $j = 1, \ldots, J$. Wpływ ten można określić za pomocą następującego wzoru:

$$S_{j,i}^{(l)} = \frac{\partial y_j(x_1^{(l)}, x_2^{(l)}, \ldots, x_n^{(l)})}{\partial x_i}. \tag{8}$$

Równanie (8) opisuje czułość j-tego wyjścia sieci na i-ty atrybut wektora wejściowego $\mathbf{x}$, wyznaczoną na podstawie l-tego wzorca uczącego, $l = 1, \ldots, L$; czułość ta jest elementem macierzy $\mathbf{S}^{(l)} = \{S_{j,i}^{(l)}\}_{J \times n}$. Po obliczeniu $\mathbf{S}^{(l)}$ dla wszystkich L wzorców wejściowych, elementy macierzy należy zagregować stosując normalizację wybranego typu [L31]. Ostatnie etapy procedury LSA sprowadzają się do wyznaczenia wektora $\mathbf{q}$, przechowującego maksymalne wartości z każdej i-tej kolumny zagregowanej macierzy czułości $\mathbf{S}$, posortowania ich w kolejności malejącej i obliczenia różnic $\Delta q_i = q_i - q_{i+1}$ dla $i = 1, \ldots, n-1$. Znacząca wartość różnicy Δq_i wskazuje, że wszystkie elementy q_i o małej wartości mają niewielki wpływ na wyjście generowane przez sieć neuronową.

Zaproponowane w artykule [C3] trzy algorytmy uproszczenia architektury PNN wykorzystują procedurę LSA opisaną powyżej.

Pierwszy algorytm redukuje liczbę neuronów w warstwie wejściowej. Redukcja ta polega na selekcji istotnych cech rekordów danych, przez co PNN wykorzystuje najważniejsze atrybuty w budowie tej warstwy. Realizowane jest to poprzez obliczenie czułości j-tego neuronu w warstwie sumowania na i-ty neuron wejściowy dla każdego wzorca l. Wszystkie tak wyznaczone czułości przechowywane są w macierzy $\mathbf{S}^{(l)}$. Następnie ma miejsce agregacja elementów macierzy po wszystkich L danych. Wykorzystywane są do tego celu trzy rodzaje normalizacji, zaproponowane przez Żuradę i in. w [L31]: średniokwadratowa (w skrócie mean), wartość bezwzględna (w skrócie abs) oraz wartość maksymalna (w skrócie max). Powstają zatem trzy macierze czułości $\mathbf{S}^{\text{mean}}$, $\mathbf{S}^{\text{abs}}$ i $\mathbf{S}^{\text{max}}$, a w konsekwencji trzy wektory $\mathbf{q}^{\text{mean}}$, $\mathbf{q}^{\text{abs}}$ oraz

$\mathbf{q}^{\max}$, których elementy definiują miary czułości PNN na wybraną cechę wejściową uwzględniając naturalnie typ normalizacji. Posortowanie elementów w każdym $\mathbf{q}$ w kolejności malejącej umożliwia wskazanie istotnych cech, a co za tym idzie neuronów wejściowych w sieci. W artykule zaproponowano dwie reguły usuwania zbędnych neuronów na wejściu PNN:
Reguła 1: *Jeśli ostatni element wektorów* $\mathbf{q}^{\text{mean}}$ *i* $\mathbf{q}^{\text{abs}}$ *i* $\mathbf{q}^{\max}$ *lub ostatni element w dowolnych dwóch wektorach* $\mathbf{q}$ *wskazuje ten sam neuron* c*, neuron* c *usuwany jest z warstwy wejściowej.*
Reguła 2: *Jeśli ostatni element wektorów* $\mathbf{q}^{\text{mean}}$ *i* $\mathbf{q}^{\text{abs}}$ *i* $\mathbf{q}^{\max}$ *wskazuje inny neuron, neuron* c*, dla którego wartość* q_i *jest po normalizacji najmniejsza we wszystkich tych wektorach, usuwany jest z warstwy wejściowej. Normalizację wprowadza się, aby porównać wartości* q_i *tych wektorów.*

Procedura eliminacji neuronów wykonuje się iteracyjnie do momentu aż przestrzeń wejściowa będzie składała się z pojedynczego atrybutu danych. Równolegle wyznaczana jest dokładność PNN przy użyciu 10–częściowego sprawdzianu krzyżowego. Różnica w dokładności otrzymanej w danej i poprzedniej iteracji algorytmu porównywana jest z przyjętym progiem ϵ. Jeżeli różnica ta spadnie poniżej założonego progu, iteracyjny proces usuwania poszczególnych neuronów zostaje przerwany. W algorytmie istnieje możliwość przyjęcia $\epsilon > 0$, co wprowadza pewien kompromis pomiędzy redukcją warstwy wejściowej PNN a spadkiem dokładności klasyfikacji sieci.

Drugi algorytm przeznaczony jest do zmniejszenia liczby neuronów w warstwie wzorców PNN. Procedura LSA przeprowadzana jest na wektorach wzorcowych a nie na cechach, jak to miało miejsce w przypadku redukcji warstwy wejściowej sieci. Macierz czułości liczona jest dla każdej klasy z osobna. Przyjmuje ona postać:

$$\mathbf{S}_j = \begin{bmatrix} \frac{\partial f_j\left(\mathbf{x}_j^{(1)}\right)}{\partial \mathbf{x}_j^{(1)}} & \cdots & \frac{\partial f_j\left(\mathbf{x}_j^{(1)}\right)}{\partial \mathbf{x}_j^{(k)}} & \cdots & \frac{\partial f_j\left(\mathbf{x}_j^{(1)}\right)}{\partial \mathbf{x}_j^{(L_j)}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial f_j\left(\mathbf{x}_j^{(l)}\right)}{\partial \mathbf{x}_j^{(1)}} & \cdots & \frac{\partial f_j\left(\mathbf{x}_j^{(l)}\right)}{\partial \mathbf{x}_j^{(k)}} & \cdots & \frac{\partial f_j\left(\mathbf{x}_j^{(l)}\right)}{\partial \mathbf{x}_j^{(L_j)}} \\ \vdots & & \vdots & & \vdots \\ \frac{\partial f_j\left(\mathbf{x}_j^{(L_j)}\right)}{\partial \mathbf{x}_j^{(1)}} & \cdots & \frac{\partial f_j\left(\mathbf{x}_j^{(L_j)}\right)}{\partial \mathbf{x}_j^{(k)}} & \cdots & \frac{\partial f_j\left(\mathbf{x}_j^{(L_j)}\right)}{\partial \mathbf{x}_j^{(L_j)}} \end{bmatrix}, \tag{9}$$

gdzie $\frac{\partial f_j\left(\mathbf{x}_j^{(l)}\right)}{\partial \mathbf{x}_j^{(k)}}$ jest czułością j-tego wyjścia (2) w warstwie sumowania sieci na k-ty neuron wzorcowy ($k = 1, \ldots, L_j$) wyznaczoną dla danego wektora $\mathbf{x}_j^{(l)} \in \{\mathbf{x}_j^{(1)}, \ldots, \mathbf{x}_j^{(L_j)}\}$. Ponieważ każdy element macierzy $\mathbf{S}_j$ jest gradientem:

$$\nabla f_j^{(l,k)} = \frac{\partial f_j\left(\mathbf{x}_j^{(l)}\right)}{\partial \mathbf{x}_j^{(k)}} = \left[\frac{\partial f_j\left(\mathbf{x}_j^{(l)}\right)}{\partial x_{j,1}^{(k)}}, \ldots, \frac{\partial f_j\left(\mathbf{x}_j^{(l)}\right)}{\partial x_{j,n}^{(k)}}\right], \tag{10}$$

aby obliczyć wartość współczynnika czułości j-tego sumatora na k-ty neuron wzorcowy, konieczne jest wyznaczenie normy z (10). W artykule [C3] wykorzystywana jest do tego celu norma euklidesowa. Kolejne kroki algorytmu polegają na: (i) zsumowaniu wszystkich $l = 1, \ldots, L_j$ elementów w każdej k-tej kolumnie, co daje zagregowany wektor czułości:

$$\mathbf{q}_j = \left[\sqrt{\frac{1}{L_j}\sum_{l=1}^{L_j}\left\|\nabla f_j^{(l,1)}\right\|_p^2}, \ldots, \sqrt{\frac{1}{L_j}\sum_{l=1}^{L_j}\left\|\nabla f_j^{(l,L_j)}\right\|_p^2}\right], \tag{11}$$

gdzie $p = 2$; (ii) posortowaniu elementów $\mathbf{q}_j$ w kolejności rosnącej; (iii) wyznaczeniu wektora różnic $\Delta\mathbf{q}_j = q_j^{(l+1)} - q_j^{(l)}$ dla $l = 1, \ldots, L_j - 1$. Wybór wzorców uczących, wykorzystanych do stworzenia neuronów wzorcowych j-tej klasy w strukturze PNN, dokonywany jest według następującej definicji:

Definicja 1. *Niech $\mathbf{k}_j$ będzie podzbiorem indeksów $\mathbf{k}_j \subset \{1, 2, \ldots, L_j\}$. $\mathbf{X}_j^{\text{red}} = \mathbf{X}_j\{\mathbf{k}_j\}$ definiuje się jako zredukowany zbiór danych j-tej klasy, jeżeli dla każdego indeksu $k \in \mathbf{k}_j$ zachodzą dwie poniższe nierówności:*

$$\begin{cases} \frac{\Delta q_j^{(k)}}{q_j^{(k)}} > \eta \\ q_j^{(k)} > \tau \cdot \overline{\Delta\mathbf{q}_j} \end{cases}, \tag{12}$$

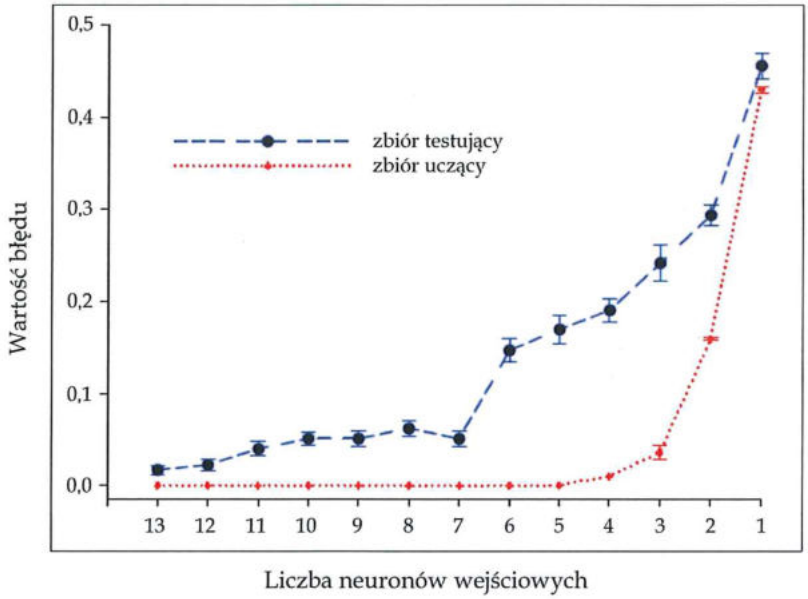
gdzie $\overline{\Delta\mathbf{q}_j}$ jest wartością średnią elementów $\Delta\mathbf{q}_j$, a η i τ to przyjęte parametry. $\mathbf{X}_j^{\text{red}}$ jest bezpośrednio wykorzystany do aktywacji neuronów wzorcowych j-tej klasy.

W Definicji 1, η odseparowuje redundantne rekordy ze zbioru wejściowego, natomiast τ zabezpiecza algorytm przed możliwością wyboru nieistotnych wektorów, które są reprezentowane przez początkowe elementy posortowanego wektora $\mathbf{q}_j$.

Trzeci algorytm służy do uproszczenia całej architektury probabilistycznej sieci neuronowej. Łączy on w swoim działaniu obydwa algorytmy opisane powyżej i w rezultacie tworzy PNN o zredukowanej liczbie neuronów w warstwie wejściowej i w warstwie wzorców. W pierwszym etapie stosowany jest algorytm do selekcji neuronów w warstwie wejściowej. Otrzymywany jest wektor, który przechowuje cechy od najmniej do najbardziej znaczących. Następnie, w sposób iteracyjny, usuwany jest ze struktury PNN jeden neuron wejściowy na podstawie najmniej znaczącej cechy danych i stosowany jest algorytm do redukcji neuronów w warstwie wzorców. Daje to możliwość zbadania, który wariant usuniętych neuronów wejściowych jest najbardziej odpowiedni w kontekście dalszej redukcji neuronów w warstwie wzorców. Cała procedura powtarzana jest do momentu aż na wejściu PNN pozostanie pojedynczy neuron.

W artykule [C3] do zmniejszenia liczby neuronów w warstwie wejściowej wykorzystano również globalną analizę czułości (ang. *global sensitivity analysis, GSA*), a w szczególności metody: Sobola [L32], FAST [L33] i EFAST [L34], oraz znane procedury selekcji i ekstrakcji cech (m.in. ReliefF i PCA). Ponadto zastosowano odpowiednie metody w celu zmniejszenia liczby neuronów wzorcowych (np. algorytm SVM czy kNN). Za pomocą 10–częściowego sprawdzianu krzyżowego wyznaczono dokładność oryginalniej PNN i zestawiono z wynikami uzyskanymi za pomocą trzech przedstawionych algorytmów oraz wszystkich pozostałych metod porównawczych w klasyfikacji ośmiu zbiorów danych dostępnych w repozytorium UCI–MLR. Użycie algorytmu redukcji całej architektury poprawiło zdolność predykcji PNN w odniesieniu do oryginalnej sieci w sześciu na osiem problemach klasyfikacji. Jest to lepszy wynik w porównaniu z algorytmem do redukcji neuronów w warstwie wzorców, gdzie odnotowano poprawę dokładności w porównaniu do sieci o pełnej strukturze dla czterech zbiorów danych. W przypadku algorytmu zmniejszającego liczbę neuronów wejściowych dokładność uproszczonej sieci jest wyższa w stosunku oryginalnej PNN również w sześciu problemach klasyfikacji ale otrzymane rezultaty są gorsze w stosunku do wyników dla algorytmu upraszczającego całą strukturę. Wykorzystanie procedur GSA do redukcji warstwy wejściowej okazało się dobrym rozwiązaniem, ale otrzymana liczba neuronów wejściowych była większa w porównaniu z zaproponowanymi algorytmami. Pozostałe metody zastosowane do uproszczenia struktury PNN nie dały satysfakcjonujących wyników.

Procedura lokalnej analizy czułości może być w łatwy sposób użyta do podania istotności neuronów wejściowych w probabilistycznej sieci neuronowej. W pracy [C4] zaproponowany



Rysunek 2: Wartości błędów dla PNN przy sukcesywnym usuwaniu najmniej znaczących neuronów wejściowych w klasyfikacji danych gatunków wina.

został algorytm, który tworzy wektor istotności neuronów wejściowych od najbardziej do najmniej znaczącego. Wektor ten przechowuje indeksy, które wskazują na odpowiednie cechy zbioru danych. Sposób wybierania poszczególnych neuronów w algorytmie polega na iteracyjnym zastosowaniu procedury LSA w celu eliminacji kolejnych nieistotnych atrybutów. Na początku wyznaczana jest macierz czułości dla pełnego zbioru danych. Następnie, przy pomocy Reguły 1 i 2, wskazywana i usuwana jest z przestrzeni wejściowej najmniej znacząca cecha. Cecha ta dodawana jest na koniec budowanego wektora – reprezentuje ona najmniej znaczący neuron wejściowy. Na tym etapie przeprowadzany jest proces uczenia i testu zredukowanej PNN. Powyższe kroki algorytmu powtarzane są do momentu aż w zbiorze wejściowym pozostanie wyłącznie jedna cecha. Identyfikuje ona najbardziej istotny neuron wejściowy. Algorytm zastosowano w klasyfikacji sześciu zbiorów danych z repozytorium UCI–MLR. W celach porównawczych wyznaczono również istotność neuronów wejściowych PNN przy użyciu metody ReliefF oraz procedury ważności zmiennych (ang. *variable importance procedure, VI*), dostępnych odpowienio w oprogramowaniu Matlab i DTREG. Skuteczność klasyfikacji PNN mierzono za pomocą błędu 10–częściowego sprawdzianu krzyżowego. Porównano błąd sieci zawierającej wszystkie neurony na wejściu z błędami, jakie otrzymano po sukcesywnym usuwaniu najmniej znaczącego neuronu.

Na podstawie bardzo dużej liczby testów wyciągnięto dwa główne wnioski. Zaproponowany algorytm podaje istotność poszczególnych neuronów wejściowych we wszystkich problemach klasyfikacji w przeciwieństwie do procedury VI, która w przypadku jednego zbioru danych wygenerowała taką samą istotność dla wszystkich cech. Wartości błędów sprawdzianu krzyżowego PNN, dla której neurony usuwano za pomocą zaproponowanego algorytmu, są zbliżone do błędów otrzymanych za pomocą pozostałych metod, a w dwóch przypadkach są niższe.

W pracy przedstawiona jest również krótka dyskusja na temat konsekwencji redukcji neuronów w warstwie wejściowej PNN. Omawiane są trzy scenariusze - ideę jednego z nich prezentuje Rysunek 2. Jak można zaobserwować, błędy wyznaczone na zbiorze uczącym i testującym mają różne wartości w trakcie całego procesu usuwania neuronów. Błąd na zbiorze

uczącym równy jest 0, do momentu aż 5 najbardziej znaczących neuronów tworzy warstwę wejściową. To może sugerować, że 8 neuronów reprezentowanych przez najmniej istotne cechy może być śmiało usuniętych z warstwy wejściowej. Niestety błąd testu PNN zaczyna gwałtownie rosnąć, w momencie gdy na wejściu pozostaje już tylko 6 elementów. Zatem decyzja, które nieistotne neurony usunąć ze struktury sieci musi być podjęta w sposób nadzorowany, uwzględniając w każdym kroku algorytmu wartość błędu na zbiorze testującym.

W artykule [C5] również podejmowana jest tematyka modyfikacji struktury probabilistycznej sieci neuronowej. Tym razem modyfikacja ta polega na wprowadzeniu współczynników wagowych do architektury PNN. Do tego celu wykorzystana została procedura LSA. Współczynniki te umieszczone są pomiędzy warstwą wzorców a warstwą sumowania. Wyznacza się je oddzielne dla każdej j-tej klasy na podstawie zagregowanego wektora czułości:

$$\mathbf{w}_j = \frac{1}{\|\mathbf{q}_j\|_p} \left[q_j^{(1)}, \ldots, q_j^{(L_j)} \right], \tag{13}$$

gdzie wektor $\mathbf{q}_j$ zdefiniowany jest w (11) a p to przyjęta norma. Uwzględniając współczynniki wagowe, j-ty neuron w warstwie sumowania PNN generuje następujący sygnał:

$$f_j(\mathbf{x}) = \frac{1}{L_j \det \mathbf{H}_j} \sum_{\{l,k\}=1}^{L_j} w_j^{(k)} \frac{1}{s_l^n} K\left(\frac{1}{s_l} \left(\mathbf{x} - \mathbf{x}_j^{(l)} \right)^T \mathbf{H}_j^{-1} \right), \tag{14}$$

w którym:

$$w_j^{(k)} = \frac{1}{\|\mathbf{q}_j\|_{p_1}} \sqrt{\frac{1}{L_j} \sum_{l=1}^{L_j} \left\| \nabla f_j^{(l,k)} \right\|_{p_2}^2}, \tag{15}$$

gdzie $\{p_1, p_2\} \geqslant 1$ są przyjętymi normami.

Tak zmodyfikowaną sieć nazwano w omawianym artykule ważoną PNN (WPNN). W pracy porównano ją z klasyczną siecią, z PNN wyposażoną w wagi, które są obliczane na podstawie podobieństwa klas [L35], oraz ze znanymi metodami SI: SVM, GEP, MLP, RBFN i kNN w klasyfikacji dziesięciu zbiorów danych z repozytorium UCI–MLR. Oszacowano zdolność predykcji wszystkich klasyfikatorów na podstawie miary dokładności wyznaczonej za pomocą 10–częściowego sprawdzianu krzyżowego. W odniesieniu do oryginalnej PNN oraz sieci o strukturze opisanej w [L35], WPNN osiąga wyższą dokładność w siedmiu na dziesięć problemów klasyfikacji. W artykule przedstawiono również statystykę rankingową opartą na teście Friedmana. WPNN okazuje się być najlepszym klasyfikatorem spośród wszystkich badanych. Należy w tym miejscu podkreślić, że analityczne wzory współczynników wagowych, których wyprowadzenie bazuje na procedurze LSA, są interpretowalne i łatwe w implementacji. Dodatkowo, do tej pory nie zastosowano procedury analizy czułości do wyznaczania wag probabilistycznej sieci neuronowej.

Klasyczna i rozmyta klasteryzacja w selekcji neuronów probabilistycznej sieci neuronowej
Prace [C6] i [C7] prezentują kolejne podejścia do redukcji warstwy wzorców probabilistycznej sieci neuronowej. Do tego celu wykorzystano algorytmy klasycznej i rozmytej klasteryzacji danych. W obydwu pracach proces klasteryzacji przeprowadzony jest na danych każdej klasy z osobna, przez co otrzymane środki są reprezentantami klas i mogą być łatwo wykorzystane w budowie warstwy wzorców PNN. Liczba środków w każdej klasie wyznaczana jest według formuły:

$$C_{sj} = \text{round}\left(\frac{s}{N} L_j \right), s = 1, \ldots, N-1, \tag{16}$$

gdzie round$(\cdot)$ jest funkcją zaokrąglającą argument do najbliższej dodatniej liczby całkowitej, a N jest liczbą naturalną ($N \geqslant 2$). Takie rozwiązanie daje możliwość wykorzystania równego procentu danych wejściowych w procesie klasteryzacji, co ma istotne znaczenie w przypadku klas o niezbalansowanej liczności. Wykorzystanie formuły (16) do budowy warstwy wzorców nie zostało do tej pory zaproponowane w literaturze.

W publikacji [C6] przedstawiono algorytm, w którym neurony w warstwie wzorców PNN reprezentowane są za pomocą k środków wyznaczanych przy użyciu algorytmu k–średnich. Warstwa ta nie jest już zatem zbudowana wykorzystując oryginalne wzorce wejściowe, lecz zmodyfikowane środki. Neurony wzorcowe tworzone są w sposób iteracyjny przy użyciu wskaźnika C_{sj}, osobno dla wszystkich klas i $s = 1, \ldots, N-1$ przy $N = 10$. Dla każdej kombinacji s i j obliczany jest parametr globalnej wydajności zdefiniowany jako:

$$J(s) = \begin{cases} \alpha\mathcal{A} + \beta Sen + \gamma Spe & \text{dla} \quad J = 2 \\ \sum_{j=1}^{J} (\alpha_j \mathcal{A}_j + \beta_j Sen_j + \gamma_j Spe_j) & \text{dla} \quad J > 2, \end{cases} \tag{17}$$

gdzie Sen oraz Spe to wskaźniki czułości i specyficzności [L36]; współczynniki α, α_j, β, β_j i γ, γ_j są nieujemnymi wagami dobieranymi dla parametrów $\mathcal{A}$, Sen i Spe, przy założeniu, że $\alpha + \beta + \gamma = \sum_{j=1}^{J}(\alpha_j + \beta_j + \gamma_j) = 1$. Optymalna liczba neuronów wzorcowych wyznaczana jest na podstawie następującego kryterium:

$$s^* = \arg \max_{s=1,\ldots,N-1} J(s). \tag{18}$$

W omawianym artykule zaproponowano również inny algorytm do redukcji warstwy wzorców PNN. Neurony w tej warstwie tworzone są na podstawie wektorów wspierających wyłuskiwanych spośród wszystkich rekordów wejściowych przez algorytm SVM. Na liczność neuronów wpływ ma wartość parametru C, gdyż reguluje liczbę wektorów wspierających oraz argument funkcji jądra. W pracy przyjęto funkcję jądra w postaci krzywej Gaussa, stąd rozmiar warstwy wzorców zależny jest również od stałej rozkładu sc. W algorytmie realizowane jest przeszukiwanie po wszystkich przyjętych wartościach C i sc w celu znalezienia optymalnej pary (C^*, sc^*) zdefiniowanej jako:

$$(C^*, sc^*) = \arg \max_{(C,sc)} Q(C, sc), \tag{19}$$

gdzie $Q(C, sc)$ jest wskaźnikiem globalnej wydajności:

$$Q(C, sc) = \begin{cases} \alpha\mathcal{A} + \beta Sen + \gamma Spe & \text{dla} \quad J = 2 \\ \sum_{j=1}^{J} (\alpha_j \mathcal{A}_j + \beta_j Sen_j + \gamma_j Spe_j) & \text{dla} \quad J > 2. \end{cases} \tag{20}$$

Na podstawie pary (C^*, sc^*) zwracany jest zbiór wektorów wspierających, który służy do budowy warstwy wzorców PNN.

Zredukowaną za pomocą dwóch omówionych algorytmów sieć przetestowano w klasyfikacji ośmiu zbiorów danych medycznych, siedmiu z internetowego repozytorium UCI–MLR oraz jednego zbioru rzeczywistego. Przy użyciu 10–częściowego sprawdzianu krzyżowego wyznaczono wskaźniki $\mathcal{A}$, Sen, Spe oraz J i Q w zależności od parametru s w przypadku algorytmu pierwszego, jak i pary (C, sc) dla drugiego podejścia. Dla współczynników α, α_j, β, β_j i γ, γ_j przyjęto wartości po konsultacji medycznej. Do celów porównawczych, te same parametry wyznaczono także dla sieci MLP, pojedynczego drzewa klasyfikacji (ang. *single decision*

tree, SDT) [L37], algorytmu SVM i k–średnich. Zaprezentowano również wyniki dokładności PNN opublikowane w literaturze światowej dla tych zbiorów danych. Główne konkluzje są następujące. Zastosowanie algorytmu k–średnich do redukcji warstwy wzorców PNN powoduje, że w siedmiu na osiem przypadków klasyfikacji wartość wskaźnika globalnej wydajności rośnie w odniesieniu do J dla sieci o pełnej strukturze. W przypadku gdy to wektory wspierające budują neurony wzorcowe PNN, wskaźnik Q osiąga wyższą wartość w porównaniu z oryginalną siecią dla pięciu zbiorów danych. Z kolei dla sześciu zbiorów danych, PNN zredukowana jednym z dwóch zaproponowanych algorytmów osiąga lepszą wydajność w porównaniu z klasyfikatorami MLP, SDT, SVM oraz k–średnich. Otrzymane wartości parametru $\mathcal{A}$ są zawsze wyższe w odniesieniu do wyników opublikowanych w literaturze.

Artykuł [C7] stanowi nowe ujęcie w redukcji warstwy wzorców probabilistycznej sieci neuronowej na podstawie klasteryzacji danych. Po pierwsze, w pracy stosowana jest rozmyta klasteryzacja wykorzystująca algorytm fuzzy c–means (FCM) [L38]. Po drugie, neurony w tej warstwie ulegają aktywacji przy użyciu wzorców uczących, dla których otrzymana jest najwyższa wartość stopnia przynależności do danego środka. Należy również podkreślić, że w pracy przeprowadzono dogłębną i rozbudowaną analizę efektów redukcji PNN, gdyż w badaniach testowana jest sieć, w której w warstwie sumowania j-ty neuron generuje sygnał w oparciu o funkcję jądra w postaci krzywej Gaussa (1), jak i sygnał bazujący na jądrze produktowym wykorzystującym funkcję Cauchy'ego (2). Dodatkowo użyte są trzy procedury uczenia PNN: gradienty sprzężone (ang. *conjugate gradients, CG*), metoda wykorzystująca algorytm RL ($Q(0)$–learning) oraz metoda pluginów (ang. *plugin method, PM*). Wykonano również analizę statystyczną wyników, podano czasy obliczeń numerycznych i przedstawiono wyniki opublikowane w powiązanych tematycznie pracach innych autorów.

W skrócie: algorytm FCM dokonuje partycji rekordów wejściowych $\mathbf{X} = \{\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(L)}\}$ na kolekcję rozmytych klastrów poprzez iteracyjną minimalizację następującej funkcji celu:

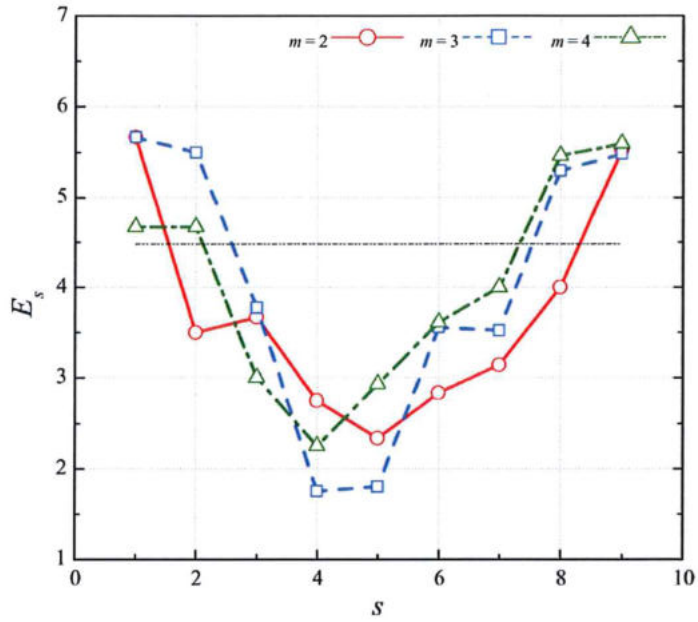
$$I_m(\mathbf{U}, \mathbf{C}) = \sum_{l=1}^{L} \sum_{c=1}^{C} \left(u_c^{(l)}\right)^m \|\mathbf{x}^{(l)} - \mathbf{c}^{(c)}\|_p^2, \tag{21}$$

gdzie $\mathbf{C} = \{\mathbf{c}^{(1)}, \ldots, \mathbf{c}^{(C)}\}$ oznacza zbiór środków tych klastrów, a $\mathbf{U} = \{u_c^{(l)}\}_{L \times C}$ to tak zwana macierz przynależności, w której każdy element $u_c^{(l)} \in [0, 1]$ określa stopień, w jakim $\mathbf{x}^{(l)}$ należy do $\mathbf{c}^{(c)}$; m jest parametrem fuzyfikacji ($1 < m < \infty$), natomiast $\|\cdot\|_p$ definiuje p–normę w przestrzeni $\mathbb{R}^n$. Optymalna, w sensie minimalizacji funkcji I_m, para $(\mathbf{U}, \mathbf{C})$ stanowi rozwiązanie równania (21).

W zaproponowanym algorytmie zarówno $\mathbf{U}$, jak i $\mathbf{C}$ są bezpośrednio użyte w selekcji neuronów wzorcowych PNN. Ogólnie algorytm składa się z dwóch faz.

W pierwszym etapie przeprowadzany jest proces rozmytej klasteryzacji na całym zbiorze wejściowym, dla każdej klasy z osobna. Liczba środków wyznaczana jest w sposób iteracyjny według wzoru (16). Otrzymywany jest zbiór środków $\mathbf{C}_{s1}, \ldots, \mathbf{C}_{sJ}$ oraz odpowiadające im macierze przynależności $\mathbf{U}_1, \ldots, \mathbf{U}_J$. Środki te użyte są do stworzenia neuronów w warstwie wzorców PNN, która podlega uczeniu – oszacowana jest wydajność sieci za pomocą tzw. błędu w oparciu o środki.

Definicja 2. *Niech* $\mathbf{C}_s = \bigcup_{j=1}^{J} \mathbf{C}_{sj}$ *będzie zbiorem środków wyznaczonym w każdej s-tej iteracji algorytmu, gdzie* $\mathbf{C}_{sj} = \{\mathbf{c}_{sj}^{(1)}, \ldots, \mathbf{c}_{sj}^{(C_{sj})}\}$ *jest podzbiorem środków obliczonych dla j-tej klasy. Błąd w oparciu o środki* E_s *definiuje się jako błąd klasyfikacji w sensie* $\mathcal{E}$ *w (4) dla PNN, w której neurony wzorcowe wykorzystują do aktywacji* C_s *środków ze zbioru* $\mathbf{C}_s$.



Rysunek 3: Wartości błędów E_s (w %) dla PNN zbudowanej z C_s środków w warstwie wzorców dla różnych wartości parametru m w klasyfikacji danych gatunków nasion (210 przypadków, 3 klasy o rozkładzie 70–70–70). Błąd sieci o pełnej strukturze przedstawiony jest za pomocą czarnej linii w formacie kreska-kropka-kropka. Jako procedurę uczenia zastosowano algorytm RL.

Proces ten powtarzany jest $s = 1, \ldots, N-1$ razy, uwzględniając każdą j-tą klasę. Równocześnie dla każdego s zapamiętywane są wektory wzorcowe $\mathbf{P}_s$, dla których otrzymano najwyższą wartość współczynnika przynależności w każdej macierzy $\mathbf{U}_j$.

W drugim etapie algorytmu znajdowany jest optymalny indeks $s^\star = \underset{s=1,\ldots,N-1}{\operatorname{argmin}} E_s$, który determinuje:

- minimalny błąd w oparciu o środki $E^\star = E_{s^\star}$,
- optymalną liczbę środków $C^\star = C_{s^\star}$,
- optymalny zbiór zredukowanych wektorów wzorcowych $\mathbf{P}^\star = \mathbf{P}_{s^\star}$.

Ostatni etap algorytmu sprowadza się do użycia wszystkich rekordów ze zbioru $\mathbf{P}^\star$ do budowy warstwy wzorców sieci. Wyznaczany jest ostateczny błąd zredukowanej PNN. Jest on zdefiniowany w następujący sposób:

Definicja 3. *Ostateczny błąd zredukowanej PNN $E_R^\star$ definiuje się jako błąd klasyfikacji w sensie $\mathcal{E}$ w (4) dla sieci, w której neurony wzorcowe wykorzystują do aktywacji rekordy ze zbioru $\mathbf{P}^\star$.*

Wszystkie etapy algorytmu powtarzane są dla każdej przyjętej wartości fuzyfikatora m.

Zaproponowany algorytm zastosowano w klasyfikacji sześciu zbiorów danych z repozytorium UCI–MLR. Najpierw wyznaczono wydajność PNN za pomocą błędu w oparciu o środki (E_s) dla $s = 1, \ldots, 9$ i $m = \{2, 3, 4\}$, ustalono wartość $s^\star$ i obliczono minimalny błąd $E^\star$. Następnie spośród wszystkich wektorów wzorcowych wybrano podzbiór $\mathbf{P}^\star$ do budowy warstwy wzorców i obliczono ostateczny błąd zredukowanej PNN ($E_R^\star$). Jako punkt odniesienia przyjęto błąd sieci wykorzystującej wszystkie rekordy do aktywacji neuronów wzorcowych. Do testów zastosowano 10–częściowy sprawdzian krzyżowy. Rysunek 3 przedstawia zmiany

E_s otrzymane dla poszczególnych wartości C_s dla jednego z rozpatrywanych zbiorów danych. W niniejszym przypadku widać, że dla $m = 3$, $s^\star = 4$, przez co – zgodnie z formułą (16) – dla $s = s^\star$, $C^\star = 84$ dla $j = \{1, 2, 3\}$, a to daje $E^\star = 1,75\%$. Na tej podstawie powstała sieć, która w warstwie wzorców wykorzystywała do aktywacji 84 wzorce uczące, znajdujące się najbliżej 84 środków w sensie wartości stopnia przynależności. Identyczna analiza została przeprowadzona dla wszystkich procedur uczenia PNN, uwzględniając odpowiednie funkcje aktywacji w warstwie sumowania oraz trzy rozpatrywane wartości fuzyfikatorów m w klasyfikacji sześciu zbiorów danych. Dla każdego przypadku wyznaczono ostateczny błąd zredukowanej PNN $E_R^\star$. W pracy porównano również czas obliczeń wymagany do zrealizowania procesu uczenia PNN o zredukowanej warstwie wzorców z czasem uczenia sieci kompletnej. Szczegółowa ocena otrzymanych wyników pozwala wyciągnąć kilka istotnych wniosków. Uwzględniając wszystkie trzy metody uczenia PNN, tylko dla jednego zbioru danych wartość błędu sieci o pełnej strukturze była mniejsza od błędu $E_R^\star$. W 43,75% problemów klasyfikacji $E_R^\star < E^\star$. Taki rezultat może sugerować, że wystarczy wykorzystać środki ze zbioru $\mathbf{C}_s$ do budowy warstwy wzorców PNN. Jednak głównym celem pracy było zastosowanie rekordów $\mathbf{P}_{s^\star}$ do aktywacji neuronów wzorcowych. Całkowity czas konieczny do nauczenia PNN, wykorzystującej podzbiór $\mathbf{P}_{s^\star}$ do budowy warstwy wzorców, jest mniejszy od czasu uczenia sieci oryginalnej we wszystkich problemach klasyfikacji danych. Na sam koniec warto również dodać, że procedury uczenia CG i RL dają dużo lepsze wyniki w porównaniu z metodą pluginów, zarówno pod kątem redukcji, jak i miar błędów $E_R^\star$ oraz $E^\star$.

Literatura

[L1] S.J. Russel i P. Norvig: *Artificial Intelligence: A Modern Approach*, Prentice Hall, New Jersey (1995)

[L2] D.F. Specht: "Probabilistic neural networks and the polynomial adaline as complementary techniques for classification", *IEEE Transactions on Neural Networks*, vol. 1, nr 1, s. 111–121 (1990)

[L3] D. Mantzaris, G. Anastassopoulos i A. Adamopoulos: "Genetic algorithm pruning of probabilistic neural networks in medical disease estimation", *Neural Networks*, vol. 24, nr 8, s. 831–835 (2011)

[L4] H. Temurtas, N. Yumusak i F. Temurtas: "A comparative study on diabetes disease diagnosis using neural networks", *Expert Systems with Applications*, vol. 36, nr 4, s. 8610–8615 (2009)

[L5] S. Ramakrishnan i S. Selvan: "Image texture classification using wavelet based curve fitting and probabilistic neural network", *International Journal of Imaging Systems and Technology*, vol. 17, nr 4, s. 266–275 (2007)

[L6] X.-B. Wen, H. Zhang, X.-Q. Xu i J.-J. Quan: "A new watermarking approach based on probabilistic neural network in wavelet domain", *Soft Computing*, vol. 13, nr 4, s. 355–360 (2009)

[L7] P.A. Kowalski i P. Kulczycki: "Interval probabilistic neural network", *Neural Computing & Applications*, vol. 28, nr 4, s. 817–834 (2017)

[L8] L. Rutkowski: "Adaptive probabilistic neural networks for pattern classification in time-varying environment", *IEEE Transactions on Neural Networks*, vol. 15, nr 4, s. 811–827 (2004)

[L9] T.P. Tran, T.T.S. Nguyen, P. Tsai i X. Kong: "BSPNN: boosted subspace probabilistic neural network for email security", *Artificial Intelligence Review*, vol. 35, nr 4, s. 369–382 (2011)

[L10] F. Zhou, J. Liu, Y. Yu, X. Tian, H. Liu, Y. Hao, S. Zhang, W. Chen, J. Dai i X. Zheng: "Field-programmable gate array implementation of a probabilistic neural network for motor cortical decoding in rats", *Journal of Neuroscience Methods*, vol. 185, nr 2, s. 299–306 (2010)

[L11] P. Burrascano: "Learning vector quantization for the probabilistic neural network", *IEEE Transactions on Neural Networks*, vol. 2, nr 4, s. 458–461 (1991)

[L12] Y. Chtioui, D. Bertrand i D. Barba: "Reduction of the size of the learning data in a probabilistic neural network by hierarchical clustering. Application to the discrimination of seeds by artificial vision", *Chemometrics and Intelligent Laboratory Systems*, vol. 35, nr 2, s. 175–186 (1996)

[L13] K.Z. Mao, K.C. Tan i W. Ser: "Probabilistic neural-network structure determination for pattern classification", *IEEE Transactions Neural Networks*, vol. 11, nr 4, s. 1009–1016 (2000)

[L14] R.K.Y. Chang, C.K. Loo i M.V.C. Rao: "A global k-means approach for autonomous cluster initialization of probabilistic neural network", *Informatica*, vol. 32, nr 2, s. 219–225 (2008)

[L15] O. Luzanin i M. Plancak: "Hand gesture recognition using low-budget data glove and cluster-trained probabilistic neural network", *Assembly Automation*, vol. 34, nr 1, s. 94–105 (2014)

[L16] Y. Kokkinos i K.G. Margaritis: "Simulating parallel scalable probabilistic neural networks via exemplar selection and em in a ring pipeline", *Journal of Computational Science*, vol. 25, s. 260–279 (2018)

[L17] Y. Chtioui, S. Panigrahi i R. Marsh: "Conjugate gradient and approximate Newton methods for an optimal probabilistic neural network for food color classification" *Optical Engineering*, vol. 37, nr 11, s. 3015–3023 (1998)

[L18] V.L. Georgiou, N.G. Pavlidis, K.E. Parsopoulos, P.D. Alevizos i M.N. Vrahatis: "New self-adaptive probabilistic neural networks in bioinformatic and medical tasks" *International Journal of Artificial Intelligence Tools*, vol. 15, nr 3, s. 371–396 (2006)

[L19] L.V. Georgiou, P.D. Alevizos i M.N. Vrahatis: "Novel approaches to probabilistic neural networks through bagging and evolutionary estimating of prior probabilities", *Neural Processing Letters*, vol. 27, nr 2, s. 153–162 (2008)

[L20] M.P. Wand i M.C. Jones: *Kernel Smoothing*, CRC Press, Boca Raton (1994)

[L21] A.G. Barto, R.S. Sutton i C.W. Anderson: "Neuronlike adaptive elements that can solve difficult learning control problems", *IEEE Transactions on Systems, Man, Cybernetics*, vol. SMC-13, nr 5, s. 834–847 (1983)

[L22] C. Watkins: "Learning from delayed rewards", rozprawa doktorsksa, Department of Experimental psychology, Cambridge University, Cambridge, U.K. (1989)

[L23] M. Lichman: UCI Machine Learning Repository. http://archive.ics.uci.edu/ml (2013)

[L24] P. Jonathan, W.J. Krzanowski i W.V. McCarthy: "On the use of cross-validation to assess performance in multivariate prediction", *Statistics and Computing*, vol. 10, nr 3, s. 209–229 (2000)

[L25] V. Vapnik: *The nature of statistical learning theory*, Springer, New York (1995)

[L26] C. Ferreira: *Gene expression programming: mathematical modeling by an artificial intelligence*, Springer, Berlin (2006)

[L27] J.A. Hartigan i M.A. Wong: "A k–means clustering algorithm", *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, nr 1, s. 100–108 (1979)

[L28] D.E. Rumelhart i J.L. McClelland: *Parallel distributed processing: explorations in the microstructure of cognition. volume 1. foundations*, MIT Press, Cambridge (1986)

[L29] D.S. Broomhead i D. Lowe: "Multivariable functional interpolation and adaptive networks", *Complex Systems*, vol. 2, s. 321–355 (1988)

[L30] T. Kohonen: "Improved Versions of Learning Vector Quantization", *International Joint Conference on Neural Networks*, vol. 1, s. 545–550, San-Diego (1990)

[L31] J.M. Zurada, A. Malinowski i I. Cloete: "Sensitivity analysis for minimization of input data dimension for feedforward neural network", *Proceedings of IEEE International Symposium on Circuits and Systems*, vol. 6, s. 447–450 London (1994)

[L32] I.M. Sobol: "Sensitivity estimates for nonlinear mathematical models", *Mathematical modeling and computational experiments*, vol. 1, nr 4, s. 407–414 (1990)

[L33] R.I. Cukier, C.M. Fortuin, K.E. Shuler, A.G. Petschek i J.H. Schaibly: "Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. I theory", *The Journal of chemical physics*, vol. 59, nr 8, s. 3873–3878 (1973)

[L34] A. Saltelli, S. Tarantola i K.P.-S. Chan: "A quantitative modelindependent method for global sensitivity analysis of model output", *Technometrics*, vol. 41, nr 1, s. 39–56 (1999)

[L35] U. Krishnasamy i D. Nanjundappan: "Hybrid weighted probabilistic neural network and biogeography based optimization for dynamic economic dispatch of integrated multiple-fuel and wind power plants", *International Journal of Electrical Power & Energy Systems*, vol. 77, s. 385–394 (2016)

[L36] D.G. Altman i J.M. Bland: "Diagnostic tests 1: sensitivity and specificity", *British Medical Journal*, vol. 308, nr 6943, s. 1552 (1994)

[L37] L. Breiman: *Classification and Regression Trees*, Routledge, New York (1984)

[L38] J.C. Bezdek: *Pattern Recognition with Fuzzy Objective Function Algorithms*, Kluwer Academic Publishers, Norwell (1981)

4 Pozostałe osiągnięcia naukowe

Moje zainteresowania naukowe od początku związane były z problematyką zastosowań metod sztucznej inteligencji w klasyfikacji danych rzeczywistych. W poniższych sekcjach omawiam pozostałe osiągnięcia naukowo–badawcze po uzyskaniu stopnia doktora, które nie wchodzą w skład podstawowego osiągnięcia naukowego. Obejmują one następujące tematy:

1) predykcja komplikacji śródoperacyjnych oraz przeżyć 5–letnich u pacjentek po radykalnej histerektomii przy użyciu metod SI;

2) zastosowanie metod SI do identyfikacji struktury przepływu ciecz-gaz w rurociągu;

3) wpływ graficznej interpretacji wag oraz algorytmów filtracji obrazu na jakość klasyfikacji sieci neuronowych w zadaniu rozpoznawania cyfr.

4.1 Predykcja komplikacji śródoperacyjnych oraz przeżyć 5–letnich u pacjentek po radykalnej histerektomii przy użyciu metod SI

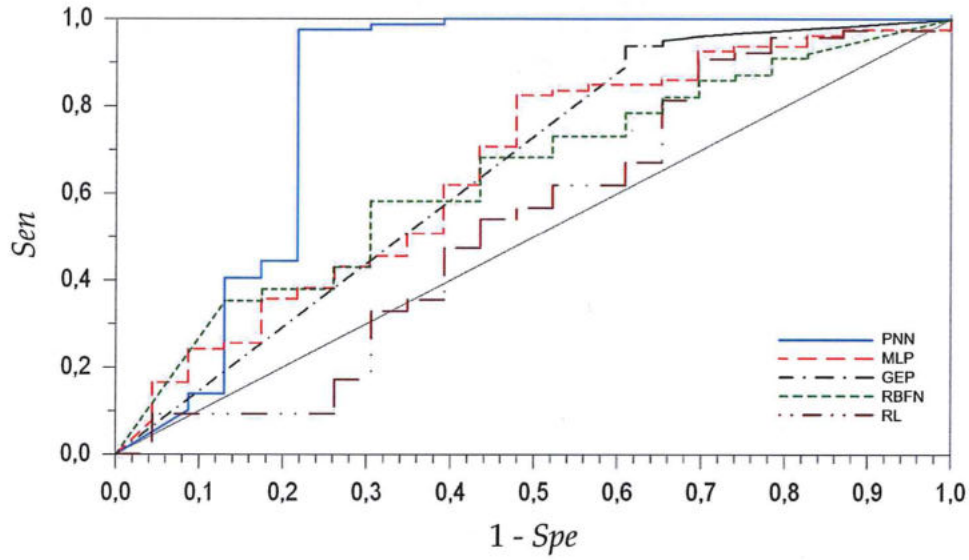
Pierwszą pracą związaną z tym aspektem badawczym jest publikacja:
M. Kusy, B. Obrzut i J. Kluska: "Application of gene expression programming and neural networks to predict adverse events of radical hysterectomy in cervical cancer patients", *Medical & Biological Engineering & Computing*, vol. 51, nr 12, s. 1357–1365 (2013). **JCR, 5–letni IF: 2,224; MNiSW: 25 pkt.**
Artykuł przedstawia porównanie algorytmu GEP z trzema typami sieci neuronowych (PNN, MLP i RBFN) w klasyfikacji danych pacjentek po zabiegu radykalnej histerektomii. Rozważana jest klasyfikacja binarna ze względu na podział chorych, u których wystąpiły powikłania śródoperacyjne lub nie. Pojedynczy rekord danych składa się z następujących zmiennych: wiek, stan hormonalny, wskaźnik masy ciała (BMI), choroby towarzyszące, przebyte operacje brzuszne, stopień zaawansowania klinicznego wg FIGO, typ histologiczny guza oraz skala złośliwości histologicznej nowotworu G. Badana grupa zawiera 107 pacjentek ze zdiagnozowanym rakiem szyjki macicy, które przebyły operację w Klinicznym Oddziale Ginekologii i Położnictwa Szpitala Wojewódzkiego nr 1 w Rzeszowie w latach 1998–2001. Jakość klasyfikacji algorytmu GEP i sieci neuronowych oceniono na podstawie parametrów dokładności, czułości, specyficzności oraz pola powierzchni pod krzywą ROC (ang. *area under the receiver operating characteristic curve, AUROC*). Parametry $\mathcal{A}$, Sen, Spe i $AUROC$ wyznaczono na niezależnych podzbiorach testujących zawierających 10%, 20% i 30% losowo wybranych danych z wszystkich przypadków. Wyniki uśredniono, podano wartości odchylenia standardowego oraz minimalne wartości parametrów, które są możliwe do przyjęcia w kontekście predykcji. Biorąc pod uwagę wszystkie rozpatrywane parametry jakości klasyfikacji, rezultaty otrzymane przez algorytm GEP były najlepsze.

W niniejszej pracy ewaluacji pod kątem predykcji poddawane jest również wyrażenie matematyczne, generowane przez algorytm GEP przy użyciu języka Karva. Podane ono zostało w przypadku, gdy zbiór uczący i testujący obejmują odpowiednio 70% i 30% danych, gdyż dla tego podziału dokładność jest najbliższa minimalnej dopuszczalnej wartości $\mathcal{A}$. Wyrażenie jest funkcją 5 zmiennych i przyjmuje następującą postać:

$$f = \frac{7,98}{1+\exp\left(7,98 \cdot x_1\right)} \cdot \frac{1}{1+\exp\left(-15,95 \cdot x_1\right)} + x_2 - 11,74 + \frac{1}{2} \cdot \frac{x_3}{1+\exp\left(x_3 \cdot x_4\right)} - x_5, \quad (22)$$

gdzie x_1 jest binarną reprezentacją zmiennej BMI (klasa "niedowaga"), x_2 jest binarną reprezentacją zmiennej FIGO (stopień "IB2"), $x_3 = \{29, \ldots, 73\}$ jest zmienną całkowitą oznaczającą wiek, x_4 jest binarną reprezentacją zmiennej BMI (klasa "otyłość II stopnia") oraz x_5 jest binarną reprezentacją zmiennej FIGO (stopień "IB1"). Wartość $f(\cdot) > 0$ determinuje wystąpienie powikłań śródoperacyjnych u pacjentki; w przypadku gdy $f(\cdot) \leqslant 0$ – powikłania nie mają miejsca. Funkcję przetestowano na dwóch przypadkach klinicznych. Wynik predykcji odpowiadał rzeczywistym wartościom wyjściowym. Wyrażenie (22) stanowi model częściowo interpretowalny, który można określić mianem szarej skrzynki.

Drugą pracą, która wiąże się z tym aspektem badawczym jest publikacja:
B. Obrzut, **M. Kusy**, A. Semczuk, M. Obrzut i J. Kluska: "Prediction of 5-year overall survival in cervical cancer patients treated with radical hysterectomy using computational intelligence



Rysunek 4: Krzywe ROC dla metod SI, w przypadku których $AUROC > 0{,}5$.

methods", *BMC Cancer*, vol. 17, nr 1, s. 840 (2017). **JCR, 5–letni IF: 3,679; MNiSW: 30 pkt.**
Artykuł prezentuje jak metody SI mogą być wykorzystane w badaniach naukowych i praktyce medycznej jako narzędzie podejmowania decyzji. Oceniana jest jakość klasyfikacji tych metod w predykcji przeżyć 5–letnich u pacjentek z rakiem szyjki macicy, które przebyły zabieg radykalnej histerektomii. Analizowana grupa obejmuje 102 przypadki raka szyjki macicy w stopniu zaawansowania FIGO IA2–IIB. Na pojedynczy rekord danych składają się 23 demograficzne i kliniczne parametry śródoperacyjne zarejestrowane w latach 1998–2001 w Klinicznym Oddziale Ginekologii i Położnictwa Rzeszowskiego Szpitala Wojewódzkiego Nr 1. W symulacjach numerycznych badano wskaźniki $\mathcal{A}$, Sen, Spe i $AUROC$ dla trzech sieci neuronowych (PNN, MLP, RBFN), algorytmów GEP, SVM i k–średnich oraz modelu regresji liniowej (LR). Rozpatrywane dane były niezbalansowane (79 przeżyć, 23 zgony powiązane z nowotworem), przez co wskaźnik specyficzności przyjmował dużo niższe wartości niż czułość. Z tego powodu za najbardziej obiektywny wskaźnik miary jakości przyjęto $AUROC$. Rysunek 4 przedstawia krzywe ROC otrzymane dla wybranych metod SI. Jak można zaobserwować, PNN osiąga największą wartość pola pod krzywą. Krzywa ROC dla algorytmu SVM nie została przedstawiona na rysunku, gdyż dla tego klasyfikatora $Spe = 0$.

4.2 Zastosowanie metod SI do identyfikacji struktury przepływu ciecz–gaz w rurociągu poziomym

Ten aspekt badawczy realizowany jest w ramach artykułu:
R. Hanus, M. Zych, **M. Kusy**, M. Jaszczur i L. Petryka: "Identification of liquid-gas flow regime in a pipeline using gamma-ray absorption technique and computational intelligence methods", *Flow Measurement and Instrumentation*, vol. 60, s. 17–23 (2018). **JCR, 5–letni IF: 1,522; MNiSW: 25 pkt.**
Artykuł opisuje zastosowanie metody absorpcji promieniowania gamma i wybranych metod SI w rozpoznawaniu struktury przepływu ciecz–gaz w rurociągu poziomym. Dane wejściowe otrzymano na podstawie analizy sygnałów uzyskanych za pomocą zestawu radiometrycznego, złożonego ze źródła promieniowania gamma Am–241 i sondy scyntylacyjnej NaI(Tl).

Rysunek 5: Graficzna prezentacja wag otrzymana dla 10 neuronów wyjściowych jednowarstwowej sieci neuronowej uczonej przy użyciu 1000 danych reprezentujących cyfry o rozdzielczości 30×40.

Zarejestrowano je na stanowisku do badań przepływów, zbudowanym w Pracowni Radiometrii Przemysłowej Wydziału Fizyki i Informatyki Stosowanej AGH w Krakowie. Analizowano cztery typy przepływu: pęcherzykowy, pęcherzykowo-tłokowy, tłokowy oraz rzutowy i wyodrębniono 47 wektorów wejściowych dla poszczególnych sygnałów. Następnie, na podstawie analizy w dziedzinie czasu, wyznaczono 8 atrybutów: wartość średnia, odchylenie standardowe, RMS, wartość funkcji autokorelacji dla wybranego argumentu, skośność, kurtoza, momenty 3 i 4 rzędu. Tak wygenerowane dane wykorzystano do identyfikacji struktury przepływu. W tym celu zastosowano następujące metody SI: trzy sieci neuronowe (PNN, MLP i RBFN), algorytm SVM i k–średnich oraz SDT. Jakość klasyfikacji oszacowano za pomocą parametrów $\mathcal{A}$, Sen, Spe. Dla PNN, MLP, RBFN, SVM i k–średnich mierzone parametry osiągnęły wartości najwyższe z możliwych ($\mathcal{A} = Sen = Spe = 1$ dla każdej z klas); natomiast dla pojedynczego drzewa decyzyjnego otrzymano rezultaty nieznacznie gorsze.

4.3 Wpływ graficznej interpretacji wag oraz algorytmów filtracji obrazu na jakość klasyfikacji sieci neuronowych w zadaniu rozpoznawania cyfr

Praca, która odnosi się do tego aspektu badawczego to publikacja:
M. Kusy i D. Szczepański: "Influence of graphical weights' interpretation and filtration algorithms on generalization ability of neural networks applied to digit recognition", *Neural Computing & Applications*, vol. 21, nr 7, s. 1783–1790 (2012). **JCR, 5–letni IF: 2,696; MNiSW: 15 pkt.**
Artykuł łączy w całość dwa zagadnienia. Pierwsze z nich dotyczy graficznej interpretacji wyznaczonych po procesie uczenia wag jednowarstwowej sieci neuronowej, która w warstwie wyjściowej zbudowana jest z 10 neuronów reprezentujących rozpoznawane cyfry. Drugie zagadnienie, wykorzystujące fakt, że wagi mogą być postrzegane jako obrazy, przedstawia sposób ich filtracji. W ten sposób powstają nowe wartości współczynników wagowych, które wykorzystywane są w procesie rozróżniania nowych cyfr. Użyte w badaniach dane wejściowe zebrano za pomocą tabletu Wacom CTE–440/S, którego ekran roboczy ma wymiary $127,6 \times 92,8$ mm. Za pomocą specjalnego rysika zarejestrowano 1000 cyfr $(0, 1, \ldots, 9)$, które skonwertowano do rozmiaru 30×40. Wszystkie obrazy zostały ujednolicone poprzez zastosowanie operacji: przeskalowania, binaryzacji, szkieletyzacji i centrowania.

Przekształcenie wartości współczynników wagowych na piksel realizowane jest za pomocą następującej formuły:

$$p_{ji} = \left\lfloor 255 \cdot \frac{w_{ji} + |w_{j\min}|}{w_{j\max} - w_{j\min}} \right\rfloor, \tag{23}$$

gdzie $\lfloor x \rfloor$ to podłoga liczby rzeczywistej x, w_{ji} jest wagą łączącą i-te wejście sieci z j-tym neuronem wyjściowym, a $w_{j\min}$ i $w_{j\max}$ oznaczają odpowiednio minimalną i maksymalną wartość spośród wszystkich współczynników sieci. Na podstawie formuły (23) generowany jest obraz, składający się z pikseli o wartościach w skali szarości. Taka transformacja została zastosowana do zestawu wag rozważanej sieci neuronowej. Rysunek 5 ilustruje 10 obrazów przedstawiających wagi przekonwertowane na piksele. Jak widać, wszystkie obrazy w dużym stopniu

przypominają poszczególne cyfry. Można zatem stwierdzić, że parametry sieci nie muszą być po procesie uczenia traktowane wyłącznie jako liczby, które umożliwiają klasyfikację nieznanych przypadków. Wagi mogą być interpretowane jako obrazy, a to z kolei pokazuje, jak sieć neuronowa "rozumie" proces rozpoznawania wzorców.

Należy mieć jednak na uwadze, że zaprezentowane cyfry nie są tak idealne jak obrazy oryginalne. Jest to spowodowane przede wszystkim tym, że nie posiadają tzw. mocnych sygnałów (białych pikseli). Taką niedoskonałość można do pewnego stopnia poprawić poprzez zastosowanie filtracji otrzymanych obrazów. W omawianej pracy użyto do tego celu trzy rodzaje filtrów: filtr wysokopasmowy HP3 oraz dwa filtry niskopasmowe (LP3, LPG). Następnie, na podstawie danego filtra oraz pozycjonowania maski, na obrazie wyznaczono nowe wartości pikseli. Z tego względu, że dane obrazu są znormalizowanymi współczynnikami wagowymi sieci neuronowej, wyznaczono odwzorowanie odwrotne dla formuły (23), które wykorzystując nowe wartości pikseli p_{ji}, dało możliwość obliczenia nowych wag w_{ji}. Tak uaktualnione współczynniki wagowe użyto do budowy sieci, którą poddano testom na nowych cyfrach.

Tę samą ideę zastosowano w niniejszym artykule w rozpoznawaniu cyfr z bazy NIST, składającej się z 70000 obrazów o wymiarze 28×28 pikseli w skali szarości. W badaniach w celach porównawczych wykorzystano również sieć dwuwarstwową, uczoną algorytmem wstecznej propagacji błędu, i hybrydową, zbudowaną z warstwy samoorganizującej się mapy cech Kohonena i warstwy nieliniowej. Zarówno dla danych zgromadzonych przy użyciu tabletu, jak i tych z bazy NIST otrzymano poprawę jakości klasyfikacji na zbiorze testującym.

W trakcie pracy nad artykułem współautor Damian Szczepański był studentem kierunku Elektronika i telekomunikacja II stopnia na Wydziale Elektrotechniki i Informatyki Politechniki Rzeszowskiej.

5 Podsumowanie dorobku

5.1 Dorobek publikacyjny

Mój dorobek publikacyjny obejmuje 30 prac (w tym 24 po uzyskaniu stopnia doktora):

Kategoria	Przed dr	Po dr	Suma
Podręczniki akademickie	0	1	1
Artykuły w czasopismach znajdujących się na liście JCR	0	11	11
Artykuły w czasopismach z listy B MNiSW	0	2	2
Rozdziały w książkach	3	8	11
Materiały pokonferencyjne	3	2	5
Suma	6	24	30

Wskaźnik IF został podany według bazy JCR zgodnie z dostępem z dnia 17 września 2018 r. Sumaryczny 5–letni IF oraz sumaryczna liczba punktów MNiSW, podana zgodnie z obowiązującym w roku wydania wykazem czasopism naukowych, przedstawia się następująco:

Kategoria	Przed dr	Po dr	Suma
Sumaryczny 5–letni IF	0	40,645	40,645
Liczba punktów	27	464	491

Całkowita liczba cytowań moich prac według różnych baz publikacji:

Kategoria	Wartość
Liczba cytowań wg bazy Web of Science	68
Liczba cytowań wg bazy Scopus	87
Liczba cytowań wg bazy Google Scholar	123

Indeks Hirsha h publikacji jest równy:

Kategoria	Wartość
Indeks h wg bazy Web of Science	5
Indeks h wg bazy Scopus	5
Indeks h wg bazy Google Scholar	6

5.2 Dorobek naukowo-badawczy, dydaktyczny oraz organizacyjny

Pełny wykaz opublikowanych prac naukowych oraz szczegółowe informacje na temat osiągnięć naukowo-badawczych, dydaktycznych, współpracy naukowej i popularyzacji nauki (z uwzględnieniem wymagań określonych w Rozporządzeniu Ministra Nauki i Szkolnictwa wyższego z dnia 1 września 2011 r. w sprawie kryteriów oceny osiągnięć osoby ubiegającej się o nadanie stopnia doktora habilitowanego, Dz. U. 2011 nr 196 poz. 1165) przedstawiony jest w załączniku nr 5.
Do moich najważniejszych osiągnięć zaliczam:

1) kierowanie projektem badawczym własnym: *Zastosowanie metod sztucznej inteligencji w zagadnieniach rozpoznawania wzorców* (2010);

2) udział jako wykonawca w realizacji następujących projektów badawczych:

   - NCN: *Rozwój wielomianowych rozmytych systemów regułowych oraz metod inteligencji obliczeniowej do zastosowań w technice i medycynie* (2011–2013),
   - NCBR: *Optymalizacja i nadzór procesu obróbki skrawaniem cienkościennych zespołów silników lotniczych z zastosowaniem metod inteligencji obliczeniowej* (2012–2016);

3) otrzymanie nagrody *ICAISC 2018 Best Paper Award*, przyznanej przez wydawnictwo Springer za artykuł "Application of Reinforcement Learning to Stacked Autoencoder Deep Network Architecture Optimization", przedstawiony na konferencji ICAISC 2018 w Zakopanem, 2018;

4) otrzymanie trzech nagród zespołowych I stopnia za osiągnięcia naukowe, przyznanych przez JM Rektora Politechniki Rzeszowskiej im. Ignacego Łukasiewicza za cykl publikacji w latach 2012, 2016 i 2017;

5) wygłoszenie siedmiu referatów na międzynarodowych i krajowych konferencjach tematycznych;

6) sprawowanie opieki naukowej nad dwoma doktorantami (w jednym przypadku jako promotor pomocniczy);

7) sprawowanie opieki nad studentami z Hiszpanii, Portugalii i Turcji w ramach międzynarodowego programu Erasmus;

8) prowadzenie szkoleń dotyczących programowania w językach rodziny C w projekcie *Elektronika dla branży automotive* (2018) w ramach europejskiego programu Wiedza Edukacja Rozwój;

9) członkostwo w komitetach programowych/organizacyjnych następujących konferencji międzynarodowych:

   - Doctoral Symposium on Recent Advances in Information Technology, Federated Conference on Computer Science and Information Systems 2016 (Gdańsk), 2017 (Praga), 2018 (Poznań);
   - Soft Computing & Applications in Digital Ecosystems, 10th international ACM SIGAPP Conference on Management of Emergent Digital Ecosystems 2018 (Tokio);

10) współpracę z: Zespołem Technik Informacyjnych i Badań Systemowych Katedry Informatyki Stosowanej i Fizyki Komputerowej na Wydziale Fizyki i Informatyki Stosowanej Akademii Górniczo–Hutniczej w Krakowie, Katedrą Ginekologii i Położnictwa na Wydziale Medycznym Uniwersytetu Rzeszowskiego oraz z Wydziałem Ginekologii Uniwersytetu Medycznego w Lublinie;

11) udział w komitetach redakcyjnych dwóch czasopism na Wydziale Elektrotechniki i Informatyki Politechniki Rzeszowskiej: *Zeszyty Naukowe PRz – Elektrotechnika* oraz *Zeszyty Naukowe PRz – Informatyka*;

12) członkostwo w międzynarodowych i krajowych organizacjach i towarzystwach, w tym Polskim Stowarzyszeniu Sztucznej Inteligencji, Polskim Stowarzyszeniu Sieci Neuronowych i organizacji IEEE;

13) recenzowanie publikacji dla 12 międzynarodowych czasopism naukowych (w sumie 24 recenzje; m.in. IEEE Transactions on Cybernetics, Neural Computing & Applications, International Journal of Applied Mathematics and Computer Science, Neurocomputing, Expert Systems With Applications) oraz międzynarodowych konferencji naukowych (8 recenzji);

14) członkostwo w Komisji Dyplomującej na kierunkach Informatyka oraz Elektronika i telekomunikacja na Wydziale Elektrotechniki i Informatyki Politechniki Rzeszowskiej;

15) członkostwo w Komisji Rekrutacyjnej na Wydziale Elektrotechniki i Informatyki Politechniki Rzeszowskiej;

16) wypromowanie 61 dyplomantów (w tym 48 inżynierów i 13 magistrów) na kierunkach Informatyka oraz Elektronika i telekomunikacja na Wydziale Elektrotechniki i Informatyki Politechniki Rzeszowskiej.

Maciej Kusy