

Politechnika Wrocławска  
Wydział Elektroniki  
Rok akad. 2015/2016  
Kierunek Informatyka

---

## ZASTOSOWANIA INFORMATYKI W MEDYCYNIE

*Projekt - "Badanie różnych typów komitetów klasyfikatorów"*

---

***Grupa projektowa:***

Grzegorz Zając, 200664  
Sebastian Bujwid, 200736  
Jakub Wężykowski, 200813

***Prowadzący:***

dr inż. Bartosz Krawczyk

***Termin:***

Czwartek 11:15

Wrocław, 1.06.2016

# Spis treści

<b>Spis rysunków</b>	<b>3</b>
<b>1 Wstęp</b>	<b>3</b>
1.1 Cel projektu . . . . .	3
1.2 Zakres projektu . . . . .	3
<b>2 Opis zaimplementowanego systemu</b>	<b>5</b>
2.1 Architektura systemu . . . . .	5
2.2 Klasyfikatory . . . . .	5
2.2.1 Sieć neuronowa ze wsteczną propagacją . . . . .	5
2.2.2 Extreme Learning Machine . . . . .	6
2.2.3 Support Vector Machine . . . . .	6
2.3 Komitety . . . . .	6
2.3.1 Random Networks . . . . .	6
2.3.2 Bagging . . . . .	6
2.3.3 Random Subspace . . . . .	6
2.4 Systemy głosowania . . . . .	7
2.4.1 Głosowanie większościowe . . . . .	7
2.4.2 Ciągłe odpowiedzi . . . . .	7
2.5 Zestawy danych . . . . .	7
2.5.1 Wykorzystane zestawy danych . . . . .	7
2.5.2 Problem wartości brakujących . . . . .	8
2.6 Metodologia wykonywania badań . . . . .	9
<b>3 Analiza wyników przeprowadzonych badań</b>	<b>10</b>
3.1 Klasyfikatory bazowe . . . . .	10
3.1.1 Skuteczność . . . . .	10
3.1.2 Czasy uczenia . . . . .	10
3.2 Wyniki komitetów . . . . .	11
3.2.1 Random Subspace . . . . .	14
3.2.2 Bagging . . . . .	15
3.2.3 Random Networks . . . . .	16
3.3 Skuteczność komitetów w zależności od zastosowanego modułu decyzyjnego . . . . .	18
3.4 Wyniki badanych modeli zagregowane ze wszystkich zestawów danych . . . . .	18

<b>4 Podsumowanie</b>	<b>21</b>
<b>A Graficzna reprezentacja wyników</b>	<b>22</b>
<b>Bibliografia</b>	<b>42</b>

# Rozdział 1

## Wstęp

### 1.1 Cel projektu

Celem projektu było przetestowanie następujących algorytmów:

- pojedyncza sieć neuronowa ze wsteczną propagacją błędu (BPNN - *Back Propagation Neural Network*)
- pojedyncza sieć neuronowa typu Extreme Learning Machine (ELM) [1]
- komitet ELM wykorzystujący losowe generowanie sieci i głosowanie większościowe [2]
- komitet ELM bazujący na *Bagging* (różne obiekty uczące)
- komitet ELM bazujący na *Random Subspace* (różne cechy)

Zbadanie działania algorytmów należało przeprowadzić na co najmniej 15 różnorodnych bazach danych z *UCI ML Repository*[5].

Dodatkowo, w ramach projektu trzy typy komitetów klasyfikatorów (*Bagging*, *Random Subspace*, *Randomizacja*) zostały zbadane z użyciem dwóch dodatkowych klasyfikatorów (Sieć neuronowa z propagacją wsteczną oraz Support Vector Machine).

Co więcej, wszystkie kombinacje typów komitetów oraz ich członków zostały zasymulowane przy użyciu dwóch różnych wersji modułu decyzyjnego komitetu - kombinacja wyników częściowych poszczególnych członków komitetów została przeprowadzona za pomocą głosowania większościowego i przy użyciu średniej arytmetycznej (wyliczanej ze wsparć dla każdej z klas).

### 1.2 Zakres projektu

**W zakres projektu wchodziło:**

1. Zaprojektowanie systemu pozwalającego na przeprowadzenie kompleksowych badań.
2. Wykorzystanie gotowych implementacji dwóch typów klasyfikatorów bazowych:
  - sieć neuronowa ze wsteczną propagacją błędów
  - Extreme Learning Machine

3. Implementacja trzech typów komitetów:

- *Random Networks*
- *Bagging*
- *Random Subspace*

4. Przeprowadzenie badań wyznaczających skuteczność zaimplementowanych rozwiązań
5. Zbadanie wpływu liczby członków komitetu na jego działanie
6. Interpretacja wyników i wywnioskowanie zależności pomiędzy skutecznością, a badanym zestawem klasyfikującym
7. Porównanie czasów uczenia Sieci Neuronowej z Wsteczną Propagacją oraz Extreme Learning Machine

**Zakres podstawowy został rozszerzony o:**

1. Implementacja dwóch typów modułu decyzyjnego wewnątrz każdego komitetu:
  - głosowanie większościowe
  - głosowanie wykorzystujące średnią arytmetyczną
2. Zbadanie wpływu metody kombinowania rozwiązań częściowych w komitecie
3. Zbudowanie komitetów homogenicznych składających się Sieci Neuronowych z Wsteczną Propagacją oraz Support Vector Machine
4. Zbadanie wpływu typu członków na działanie komitetu
5. Wykorzystanie dwóch różnych implementacji sieci neuronowych ze wsteczną propagacją (PyBrain, NeuroLab) oraz dwóch różnych implementacji ELM (HPELM, Python Extreme Learning Machine) w celu wyboru odpowiednieszych.

## Rozdział 2

# Opis zaimplementowanego systemu

### 2.1 Architektura systemu

System został zaprojektowany oraz zaimplementowany w taki sposób, aby mógł być w łatwy sposób rozszerzalny. Dzięki zastosowaniu obiektowego paradygmatu programowania system ma strukturę niezależnych modułów. Możliwe jest zastosowanie dowolnego typu klasyfikatora niezależnie od użytego typu komitetu. Dodatkowo, tak samo można użyć dowolny moduł decyzyjny w każdym typie komitetu. Konieczne jest jedynie implementowanie określonego interfejsu klasyfikatora, komitetu oraz modułu decyzyjnego - odbywa się to poprzez dziedziczenie po utworzonych klasach. Powstało także wiele narzędzi, dzięki którym zaimplementowany system stanowi środowisko

Infrastruktura systemu pozwala na:

- wczytywanie wszystkich zestawów danych z zadanych katalogów
- odpowiednie przetwarzanie zestawów w automatyczny sposób - skalowanie, normalizacja lub kodowanie w systemie OHE atrybutów
- zautomatyzowane rysowanie wykresów za pomocą zaimplementowanych skryptów
- wyznaczenie skuteczności modeli
- uruchomienie automatycznych testów uwzględniających wszystkie dostępne zbiorы danych, modele, komitety i systemy głosowania.

Zaimplementowany system stanowi wygodne środowisko do badania skuteczności komitetów w zależności od typu i liczby członków, modułu decyzyjnego oraz zbioru danych ze względu na fakt, że wszystkie te elementy mogą być łatwo wymieniane oraz rozszerzane.

### 2.2 Klasifikatory

#### 2.2.1 Sieć neuronowa ze wsteczną propagacją

Podstawowym klasyfikatorem bazowym użyтыm w projekcie była sieć neuronowa uczona przy użyciu wstecznej propagacji błędu. Wykorzystano implementację tego typu sieci z biblioteki PyBrain. Biblioteka ta posiada także algorytm trenujący wykorzystujący wsteczną propagację.

Użyta biblioteka pozwala na zbudowanie sieci neuronowej z gotowych modułów, jednak istnieją również metody pozwalające skrócić proces konfiguracji sieci. Najpierw określa się liczbę i rozmiar warstw sieci. W opisywanym rozwiążaniu użyto trzech warstw: wejściowa o rozmiarze równym liczbie cech, wyjściowa o rozmiarze równym liczbie klas oraz ukryta warstwa o rozmiarze równym średniej arytmetycznej rozmiarów pozostałych warstw. Jako funkcja aktywacji wykorzystano Sigmoid. W warstwie wejściowej oraz ukrytej dodane zostało obciążenie w postaci neuronów o stałej wadze równej 1 (ang. *bias*). Po utworzeniu sieci następuje jej trenowanie przy użyciu propagacji wstecznej błędu.

### 2.2.2 Extreme Learning Machine

ELM jako klasyfikator bazowy był oparty na gotowej implementacji z biblioteki *HPELM - High-Performance implementation of an Extreme Learning Machine ??*. Nie wymagała ona od nas ręcznej konstrukcji sieci, ani implementacji procesu uczenia. Twórcy biblioteki mówią o niej, że ma ona wysoką wydajność.

### 2.2.3 Support Vector Machine

Jako punkt odniesienia do klasyfikatorów opartych na sieciach neuronowych, został użyty również SVM. On również opierał się na gotowej implementacji z biblioteki *scikit-learn ??*. Problem klasyfikacji wieloklasowej został zrealizowany poprzez rozwiązywanie wielu problemów klasyfikacji binarnej oraz agregacje wyników zgodnie z ideą *One vs all*.

## 2.3 Komitety

### 2.3.1 Random Networks

Pierwszym typem komitetu klasyfikatorów utworzonym na potrzeby projektu jest komitet składający się z losowych sieci. Klasyfikatory w tym komitecie uczą się na tych samych danych, jednak różnią się początkowym stanem (losowe wagi początkowe w neuronach sieci). Pozwala to wyeliminować efekt osiągnięcia przez pojedynczy klasyfikator minimum lokalnego.

### 2.3.2 Bagging

Kolejny typ komitetu bazuje na idei uczenia pojedynczych klasyfikatorów na różnych danych. Najpierw dla każdego klasyfikatora losowana jest próbka danych ze zwracaniem (ang. *bootstrap sample*) i dany klasyfikator uczony jest na tej próbce. Następnie podczas rozpoznawania wyniki poszczególnych klasyfikatorów są agregowane, stąd nazwa *bagging* (ang. *bootstrap aggregating*).

### 2.3.3 Random Subspace

Ostatnim zaimplementowanym typem komitetu jest komitet klasyfikatorów uczonym na różnych podprzestrzeniach cech. Metoda podobnie jak *Bagging* dzieli zbiór danych na różne podzbiory, jednak w przypadku *Random Subspace* klasyfikatory uczą się tu na wybranych losowo podzbiorach cech, wykorzystując do tego wszystkie obiekty należące do zestawu uczącego.

Podczas treningu klasyfikatorów, dla każdego z nich generowany jest oddzielnny zestaw uczący zawierający losowo wybrane 50% wszystkich cech. Przy implementacji metody uczącej należało zwrócić uwagę na dwie kwestie. Po pierwsze, należy zapisać indeksy cech wykorzystanych do nauki każdego z klasyfikatorów, ponieważ są one potrzebne przy rozpoznawaniu obiektów ze zbioru testującego. Po drugie, niektóre cechy, określające kategorię a nie wartość rzeczywistą, zostały przekonwertowane do kodu 1 z n, przez co zostały zapisane na kilku kolumnach macierzy uczącej. Wymusiło to wprowadzenie metadanych określających, które kolumny macierzy odpowiadają której cesze, a następnie podczas losowania podzbioru wybranie wszystkich kolumn danej cechy. Dzięki temu klasyfikatory otrzymywały pełną informację o danej cesze określającej kategorię.

## 2.4 Systemy głosowania

### 2.4.1 Głosowanie większościowe

Jednym z wykorzystanych typów głosowania wewnątrz komitetu jest głosowanie większościowe. Najpierw dla każdego klasyfikatora bazowego określana jest klasa o najwyższym wsparciu w danym klasyfikatorze. Następnie sumowane są wyniki dla każdej z klas i klasa z największą liczbą wspierających klasyfikatorów jest wybierana.

### 2.4.2 Ciągłe odpowiedzi

Ze względu na to, że agregacja wyników cząstkowych poprzez *głosowanie większościowe* w żaden sposób nie uwzględnia pewności podejmowanych decyzji, zaimplementowany został inny sposób agregacji wyników poprzez łączenie ciągłych odpowiedzi w postaci wektorów wsparć od poszczególnych klasyfikatorów.

## 2.5 Zestawy danych

### 2.5.1 Wykorzystane zestawy danych

Do przeprowadzenia badań wykorzystane zostało 15 różnorodnych baz danych z UCI ML Repository [5].

Zestawy danych zostały dobrane w taki sposób, aby przeprowadzić pomiary skuteczności działania zastosowanych modeli w zróżnicowanych środowiskach.

Wybrane zostały zestawy zarówno o dużej jak i małej liczbie obiektów uczących - od kilkudziesięciu do kilkuset. Podobnie w przypadku liczby cech - dobrane zostały zestawy od kilku cech do prawie stu pięćdziesięciu. Została także zwrócona uwaga na typ cech - uwzględniono zestawy o wyłącznie danych o wartościach rzeczywistych, wyłącznie danych o wartościach dyskretnych oraz zestawy mieszane. Bazy danych, które zostały wykorzystane, dotyczą różnych dziedzin techniki oraz nauki.

Ponadto zadbano o wyszukanie zbiorów danych o stosunkowo równomiernym rozkładzie klas, co okazało się istotne z powodu wykorzystania do walidacji metody k-fold.

Zbiory danych zostały poddane dodatkowemu przetworzeniu przed wykorzystaniem do badań. Cechy o charakterze przynależności do kategorii były poddawane systemowi kodowania OHE.

Tablica 2.1: Wykorzystane zestawy danych

Nazwa zbioru	liczba cech	liczba obiektów
<i>breast_cancer</i>	89	286
<i>crx</i>	15	653
<i>dermatology</i>	34	358
<i>house_votes</i>	16	435
<i>inosphere</i>	34	352
<i>iris</i>	4	150
<i>soybean_small</i>	35	47
<i>spect</i>	22	267
<i>thyroid</i>	5	215
<i>tic_tac_toe</i>	9	958
<i>urban_land_cover</i>	147	675
<i>wine</i>	13	178
<i>yeast</i>	8	1147

Natomiast cechy o charakterze ciągłym były poddane skalowaniu do zakresu od 0.0 do 1.0.

Podstawowe informacje o użytych zestawach danych zostały przedstawione w tabeli 2.1.

### 2.5.2 Problem wartości brakujących

W niektórych zestawach danych wykorzystanych do przeprowadzenia badań pojawiał się problem wartości brakujących (np. w zbiorach dotyczących danych medycznych wynikający z niemożności przeprowadzenia wszystkich badań u szczególnych pacjentów). Niestety, ciężko nam było przewidzieć zachowanie wykorzystywanych metod klasyfikacji w przypadku niepełnego zbioru danych.

W przypadku, gdy liczba brakujących danych jest stosunkowo niewielka możliwe jest zastosowanie najprostszej metody polegającej na usunięciu niewielkich grup obiektów posiadających niepełny zestaw wartości atrybutów.

Jednakże w innych sytuacjach nie warto usuwać obiektów lub atrybutów z takimi wartościami, lecz zastosować metody ich uzupełniania. Jedną z metod jest uzupełnianie niewiadomych średnimi pojedynczej cechy. Kolejna metoda zakłada wykorzystanie mediany pojedynczej cechy. W przypadku dużej liczby brakujących danych oraz założeniu całkowitej losowości braków danych (MCAR ang. missing completely at random) warto wykonywać uzupełnienia specjalnymi metodami takimi jak: metody regresyjne, metoda estymacji i maksymalizacji oraz wielokrotna imputacja.

Ostatecznie założyliśmy, że brakujące wartości nie były związane ze strukturą informacji o różnicowaniu grup i usuneliśmy niewielkie grupy obiektów dotknięte zjawiskiem brakujących wartości atrybutów. Zgodnie z uzyskanymi informacjami, w przypadku dużych zbiorów danych i stosunkowo niewielkich strat taka procedura nie stwarza problemu.

## 2.6 Metodologia wykonywania badań

W przeprowadzonych badaniach komitetów mierzoną metryką było skuteczność klasyfikacji (ang. *accuracy*), czyli iloraz liczby poprawnie rozpoznanych obiektów przez całkowitą ich liczbę. Przedstawione wartości są średnimi wynikami z 10-krotnej warstwowej walidacji krzyżowej z zachowaniem rozkładu klasy (ang. *stratified 10-fold cross validation*). Zbiory danych zostały indywidualnie podzielone na 10 podzbiorów (ang. *fold*). Następnie wykonano iteracje w których modele były uczone na każdej parze 9 podzbiorów, a testowane na 1 pozostałym.

Wykonane zostały również badania skuteczności pojedynczych klasyfikatorów oraz ich czasu uczenia liczonego dla poszczególnych zbiorów danych jako suma 10 czasów uczenia z poszczególnych iteracji walidacji krzyżowej.

## Rozdział 3

# Analiza wyników przeprowadzonych badań

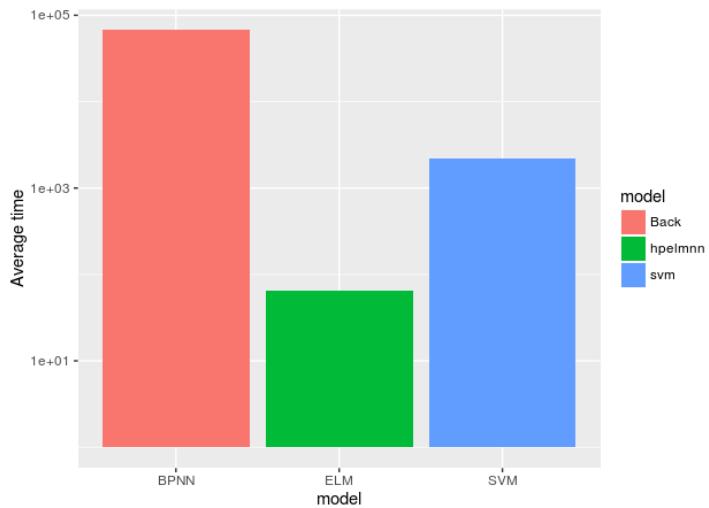
### 3.1 Klasifikatory bazowe

#### 3.1.1 Skuteczność

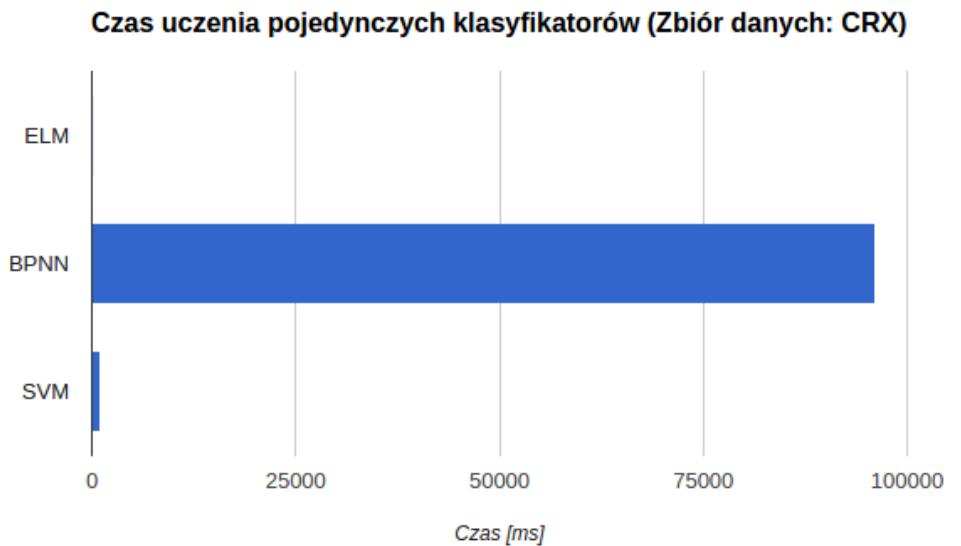
Wartość skuteczności pojedynczego klasifikatora bazowego waha się w przedziale od 55% do 98% w zależności od rozpatrywanego zbioru danych. Parametry sieci neuronowych zostały dobrane w taki sposób, aby osiągnąć wysoką skuteczność w jak największej liczbie zbiorów danych z zachowaniem rozsądnego czasu uczenia sieci. Z tego powodu w niektórych zbiorach skuteczność pojedynczego klasifikatora jest niska. Pozwala to jednak dostrzec przewagę niektórych typów komitetu nad pojedynczą siecią neuronową.

#### 3.1.2 Czasy uczenia

Jednym z aspektów projektu było porównanie sieci ELM ze zwykłymi sieciami neuronowymi. W szczególności należało zwrócić uwagę na czas nauki każdego klasifikatora. Przeprowadzenie badań dla klasifikatora ELM zajęło około 2 godzin na jednym komputerze posiadającym 4-rdzeniowy procesor. W przypadku zwykłych sieci neuronowych do przeprowadzenia badań użyto dwóch 36-rdzeniowych maszyn, aby jak najlepiej zrównoleglić obliczenia. Mimo maszyn o tak dużej mocy obliczeniowej, przeprowadzenie pełnych badań dla sieci neuronowych zajęło około 20 godzin. Wynika z tego, iż klasifikatory ELM potrzebowali o dwa rzędy wielkości mniej czasu niż zwykłe sieci neuronowe, osiągając tym samym bardzo zbliżoną skuteczność.



Rysunek 3.1: Czasy uczenia pojedynczych klasyfikatorów - skala logarytmiczna



Rysunek 3.2: Porównanie czasów uczenia pojedynczych klasyfikatorów - skala liniowa

### 3.2 Wyniki komitetów

W ramach badań sprawdzona została skuteczność działania komitetów typu:

- *Bagging*
- *Random Subspace*
- *Random Networks*

składających się z 3 typów modeli:

- Sieć neuronowa z propagacją wsteczną

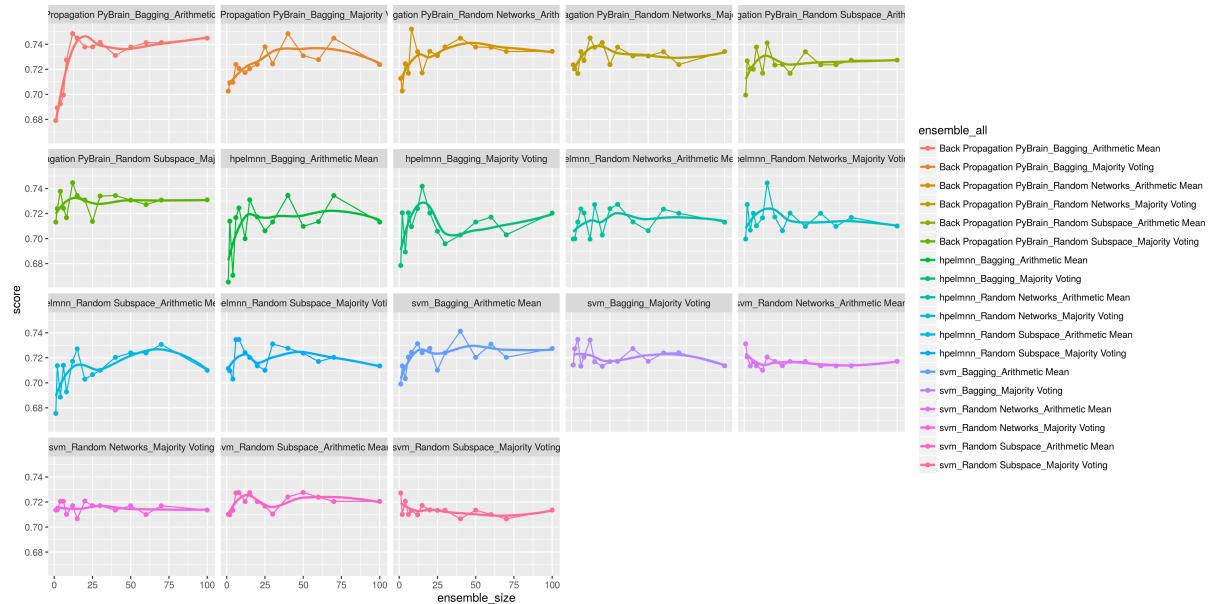
- Extreme Learning Machine
- Support Vector Machine

dla dwóch typów modułów decyzyjnych:

- głosowanie większościowe
- średnia arytmetyczna wsparć dla poszczególnych klas.

Wyniki wszystkich przeprowadzonych symulacji zostały przedstawione na zbiorczym wykresie 3.3.

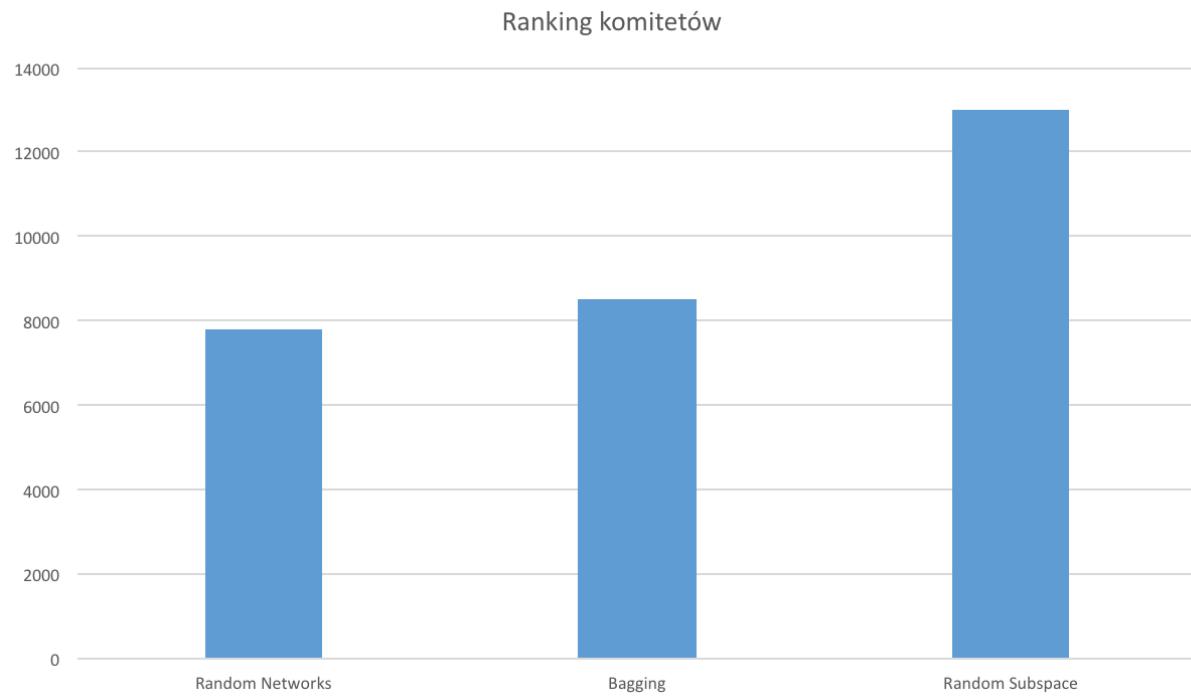
Na rysunku 3.4 przedstawione zostały przykładowe wyniki badań - w tym wypadku dla zbioru *breast\_cancer*, natomiast na rysunku 3.5 ze zbioru *spect*.



Rysunek 3.4: Skuteczność komitetów, *breast\_cancer* - 89 cech, 286 obiektów



Rysunek 3.5: Skuteczność komitetów, *spect* - 22 cechy, 267 obiektów

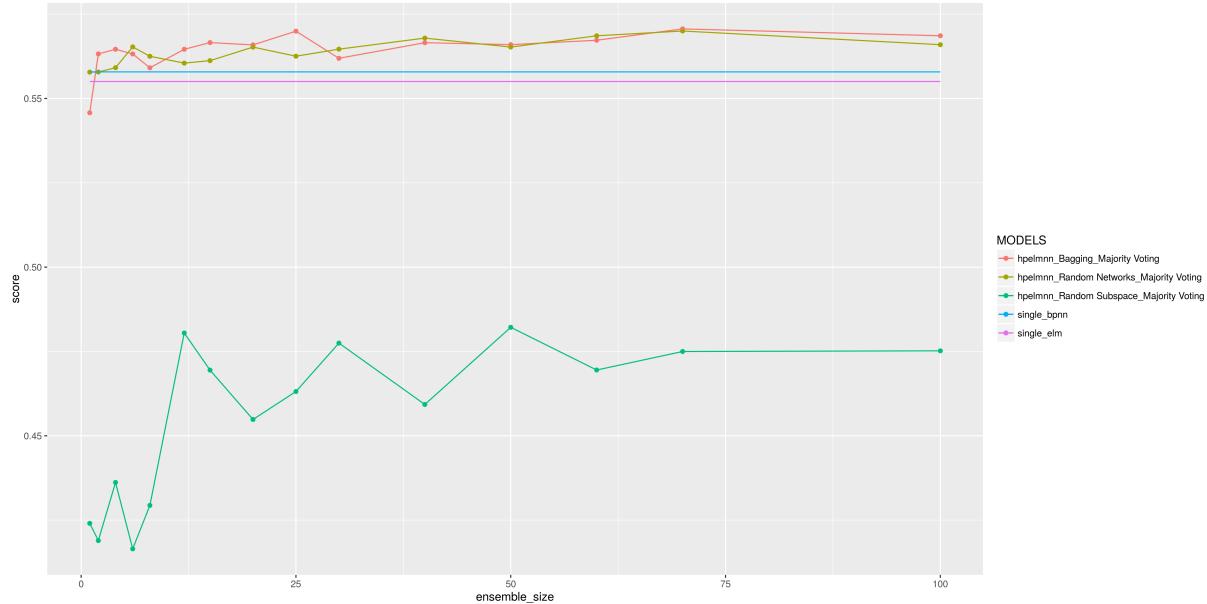


Rysunek 3.6: Wyniki według typu komitetu

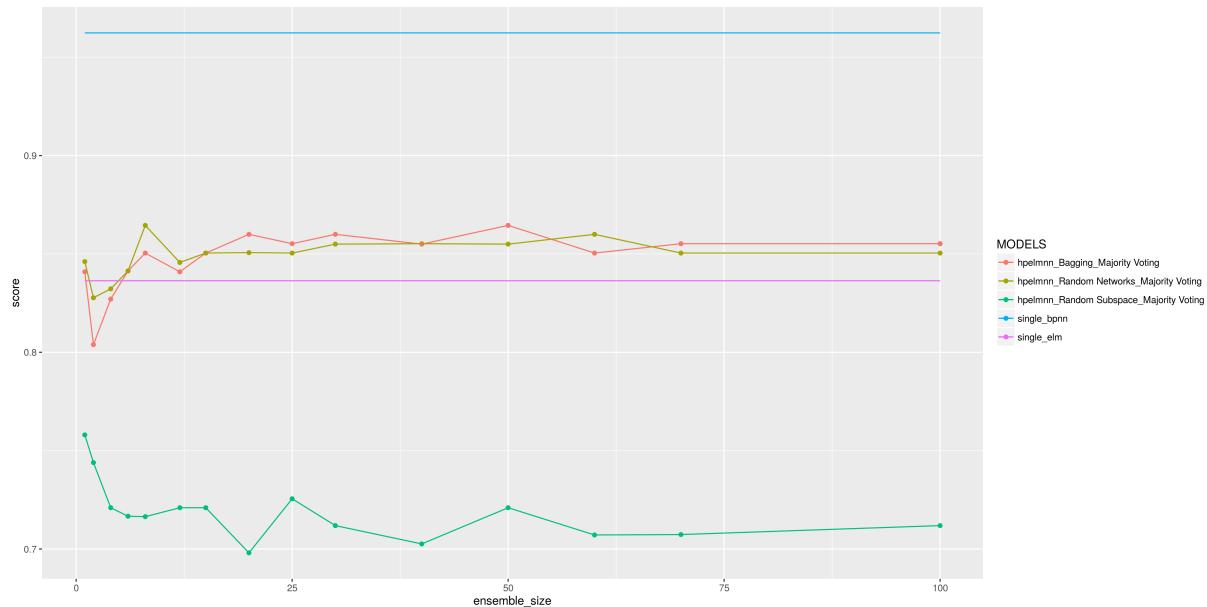
Na rysunku 3.6 przedstawione zostały zagregowane wyniki uzyskiwane przez poszczególne typy komitetów. Łatwo zauważać, że w ogólności najlepsze okazało się *Random Networks*, *Bagging* wypadł niewiele gorzej, natomiast *Random Subspace* wypadło najgorzej.

### 3.2.1 Random Subspace

Po przeanalizowaniu wyników komitetu typu *Random Subspace* można stwierdzić, że typ ten nie daje przewagi nad pojedynczą siecią neuronową dla zbiorów obiektów opisanych małą liczbą cech (poniżej 10). Przykłady takich wyników przedstawiono na rysunkach 3.7 oraz 3.8.



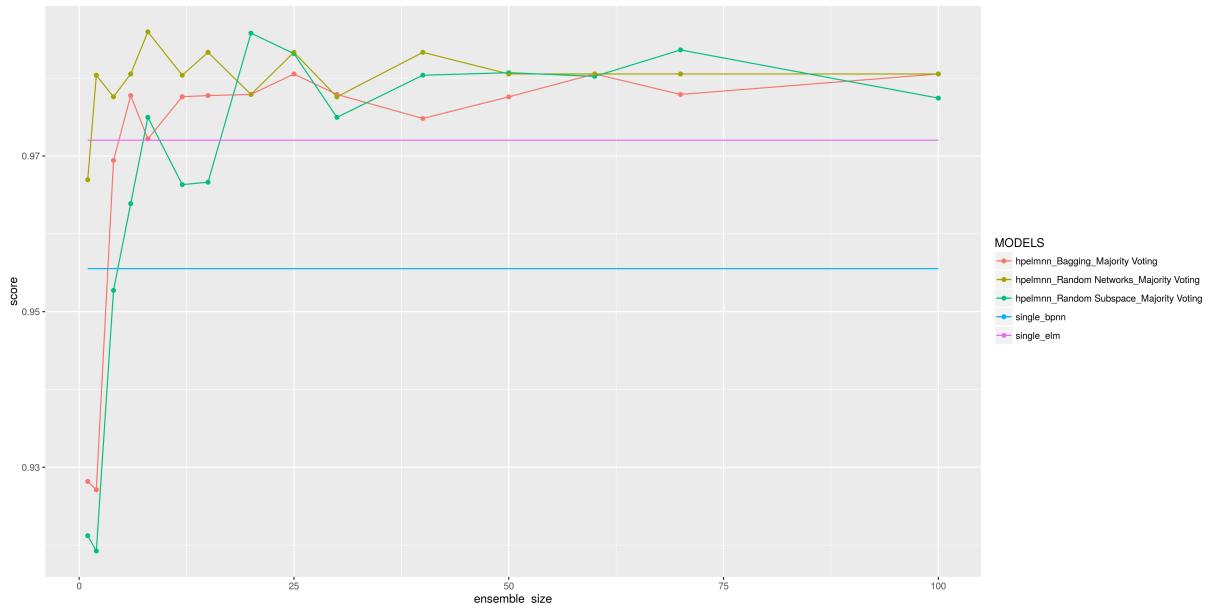
Rysunek 3.7: *yeast* - 8 cech, 1479 obiektów



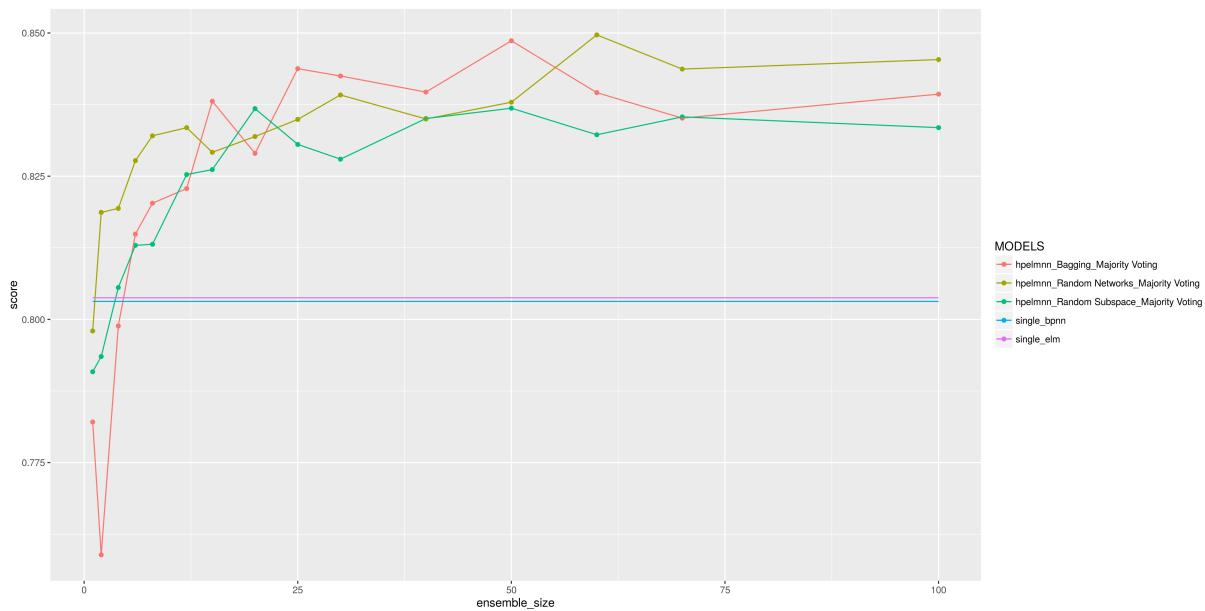
Rysunek 3.8: *thyroid* - 5 cech, 215 obiektów

W przypadku zbiorów danych opisanych większą liczbą cech (w badanych zbiorach wystarczyło już kilkanaście cech) oraz rozmiaru komitetu większego niż 10, komitet ten był porównywalny z innymi typami komitetów, a także okazał się lepszy od pojedynczego klasyfikatora. Przykłady wyników dla takich zbiorów przedstawiają rysunki 3.9 i 3.10.

Taki wynik badań był łatwy do przewidzenia. W przypadku małej liczby cech, podzbiór wylosowany do trenowania klasyfikatorów bazowych jest niewystarczający do poprawnego klasyfikowania obiektów testowych. W przypadku dużej liczby cech sytuacja zmienia się i zróżnicowanie znanych przez modele bazowe cech zwiększa generalizację sieci poprawiając tym samym jej skuteczność.



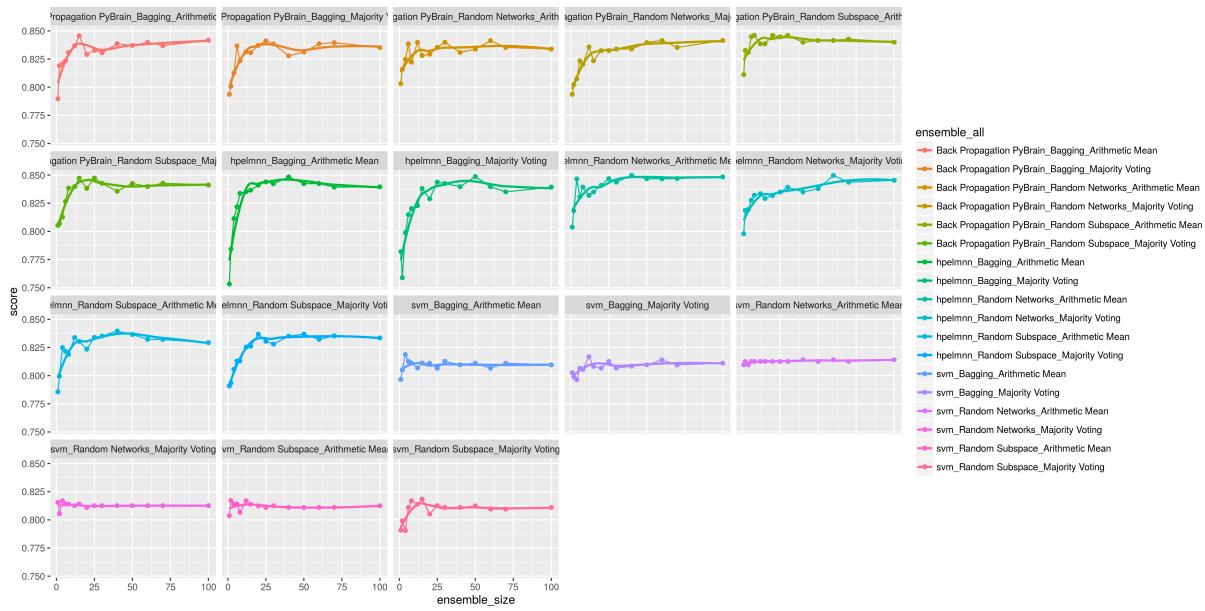
Rysunek 3.9: *dermatology* - 34 cechy, 358 obiektów



Rysunek 3.10: *urban\_land* - 147 cech, 675 obiektów

### 3.2.2 Bagging

Skuteczności uzyskiwane przez *Bagging* (rysunek 3.6) są stosunkowo dobre. Nie został zaobserwowany żaden szczególny wpływ liczby cech lub obiektów uczących na rezultaty.



Rysunek 3.11: *urban\_land* - 147 cech, 675 obiektów

Na rysunku 3.11 można zaobserwować, jak zmieniała się skuteczność wraz ze zmianą liczbą członków. Na początku jakość klasyfikacji zwiększa się szybko, osiąga ekstremum najczęściej w okolicach rozmiaru 15–25. Następnie stabilizuje się i dalsze dodawanie klasyfikatorów nie wpływa już znacząco na wyniki.

### 3.2.3 Random Networks

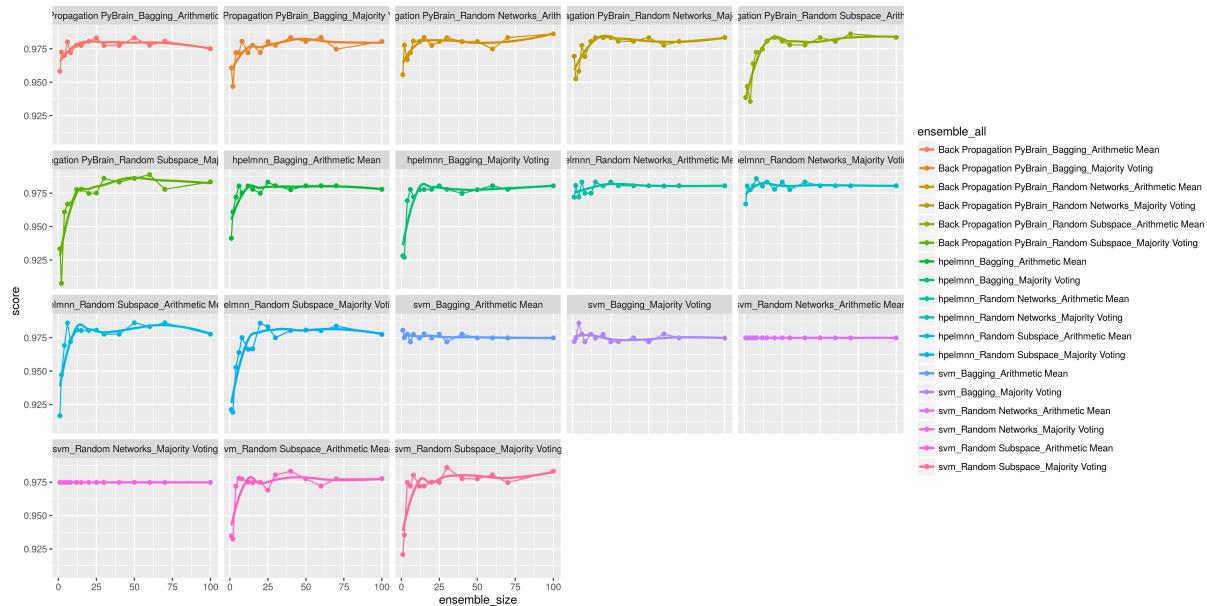
W naszych badaniach komitety typu *Random Networks* okazały się najskuteczniejsze. Zjawisko to można próbować tłumaczyć tym, że ten sposób budowy komitetów w żaden sposób nie zmniejsza skuteczności działania indywidualnych klasyfikatorów. W przypadku podziału zbioru danych tak jak w *Random Subspace* czy *Baggingu*, część danych jest faktycznie przed pojedynczym klasyfikatorem ukrywana. Można więc stwierdzić, że w zastosowanych przez nas zbiorach danych i komitetach czynnikiem istotniejszym od różnorodności pojedynczego klasyfikatora okazała się skuteczność.

Warto także zwrócić uwagę na fakt, że skuteczność komitetów typu *Random Networks* w mniejszym stopniu zależy od rozmiaru badanych komitetów. Sytuacja została zobrazowana na wykresie 3.12. Wpływ na takie zjawisko ma fakt, że poszczególne elementy komitetu *Random Networks* są uczone na całym zbiorze uczącym, przez co pojedynczy klasyfikator powinien być bardziej skuteczny niż w metodach *Random Subspace* i *Bagging*.



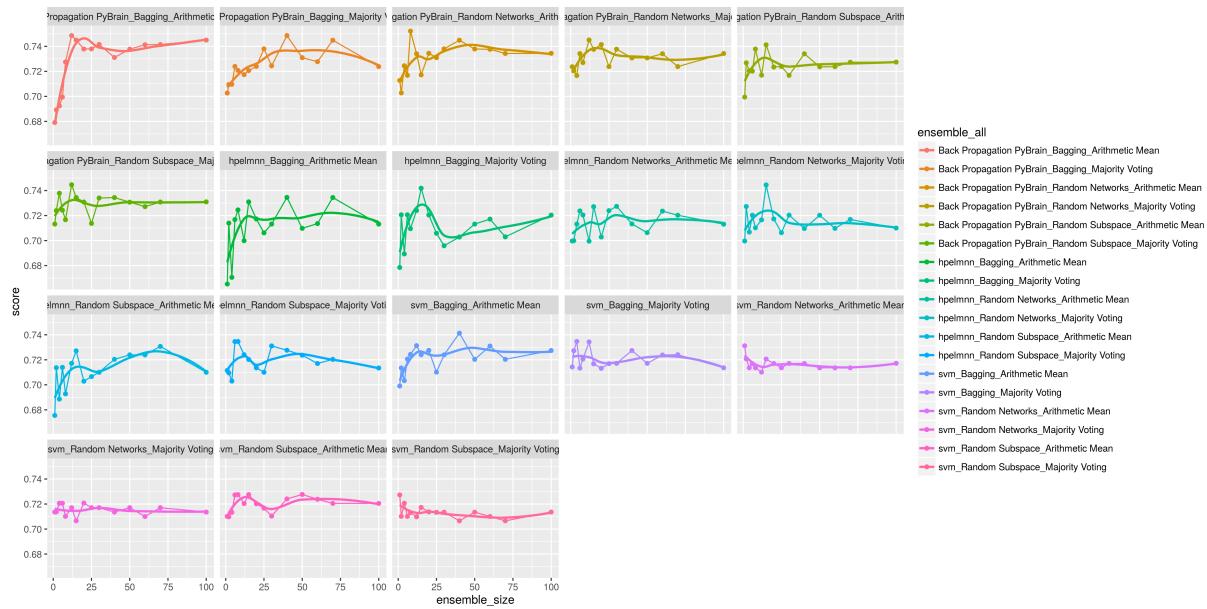
Rysunek 3.12: Wyniki skuteczności wszystkich typów komitetów na zbiorze *yeast*

W większości przeanalizowanych zestawów danych maksymalna skuteczność komitetu Random Networks jest osiągana przy rozmiarze 8-12 modeli. Przy zwiększeniu liczby członków powyżej 12 nie ma jużauważalnego wzrostu skuteczności, co zostało zobrazowane na wykresie 3.13.



Rysunek 3.13: Wyniki skuteczności wszystkich typów komitetów na zbiorze *dermatology*

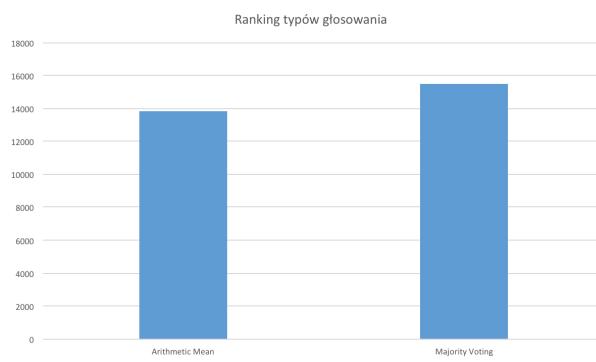
Na wynikach badań przeprowadzonych na części zestawów danych zwiększenie liczby członków komitetu powyżej dwudziestu skutkowało ustabilizowaniem otrzymywanych skuteczności. Zjawisko to przedstawia wykres 3.14.



Rysunek 3.14: Wyniki skuteczności wszystkich typów komitetów na zbiorze *breast\_cancer*

### 3.3 Skuteczność komitetów w zależności od zastosowanego modułu decyzyjnego

Na rysunku 3.15 przedstawiono ranking komitetów według typów głosowania. Jak widać *Arithmetic Mean* uzyskał nieco lepszy wynik. Prawdopodobnie jest to związane z tym, że metoda ta bierze pod uwagę pełną informację o cząstkowych decyzjach podejmowanych przez klasyfikatory bazowe, w tym wypadku są to wektory wsparć dla poszczególnych klas.

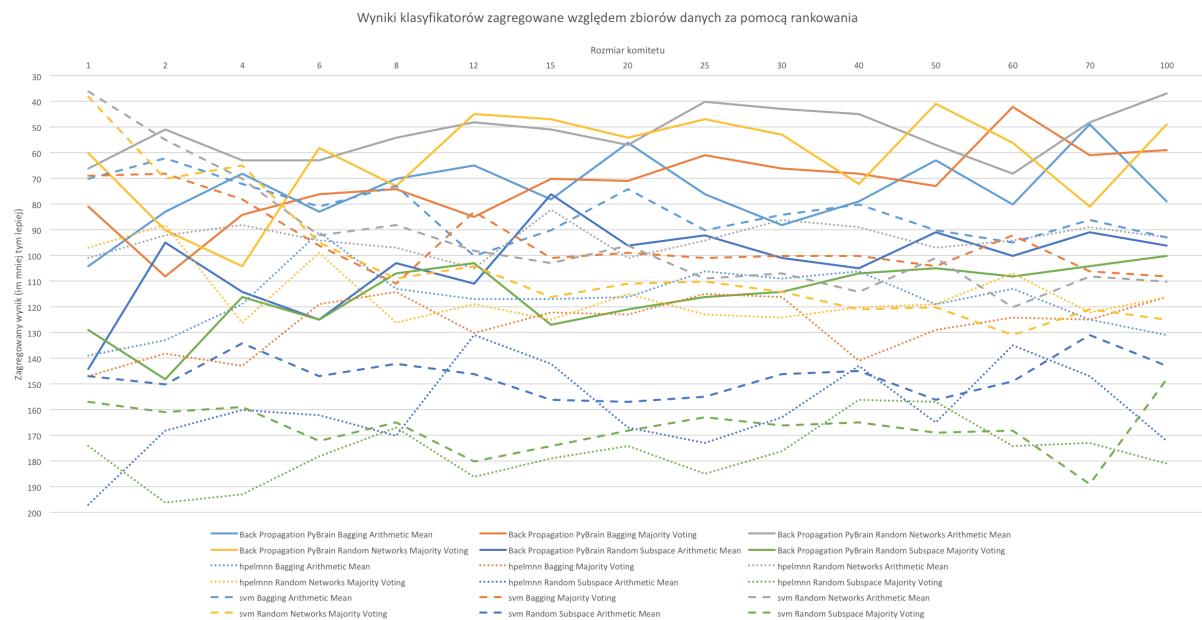


Rysunek 3.15: Zagregowany ranking systemów głosowania

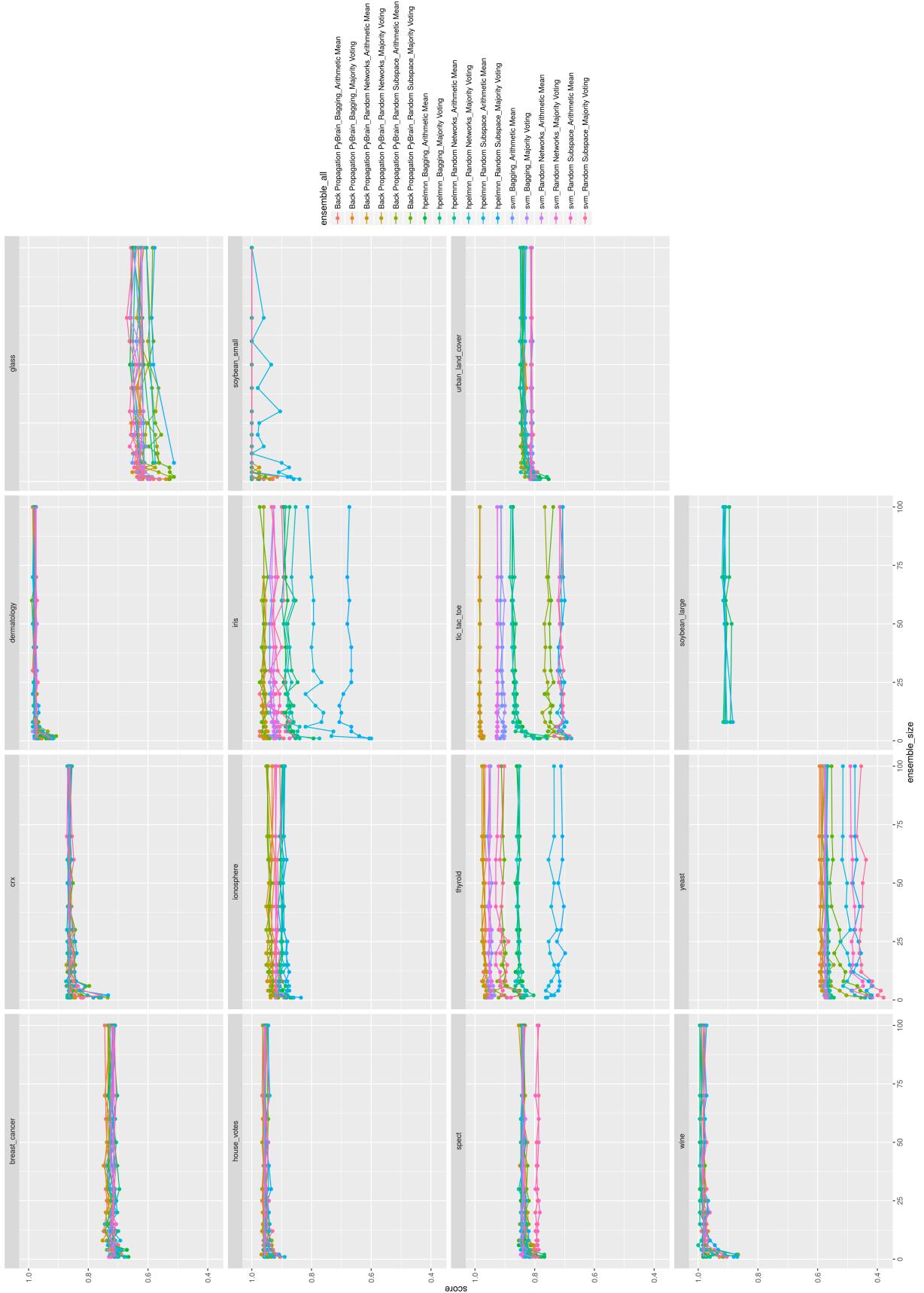
### 3.4 Wyniki badanych modeli zagregowane ze wszystkich zestawów danych

Otrzymane w badaniach wyniki ukazują, który model klasyfikujący najlepiej radzi sobie z opisanyem przez określony zbiór danych problemem. Jednak aby uzyskać ranking modeli w ogólnym przypadku należy pośrednio wyniki ze wszystkich zbiorów danych zagregować. Użyto do

tego metody polegającej na przypisaniu każdemu modelowi rang oznaczających jego skuteczność względem innych modeli w każdym problemie osobno. Im mniejsza jest wartość rangi, tym model miał wyższą skuteczność dla danego zbioru obiektów. Następnie zsumowano wszystkie rangi każdego modelu. Model z najniższą sumą traktowany jest jako najlepszy. Powyższą procedurę powtórzono dla wszystkich zbadanych rozmiarów komitetów. Otrzymano więc zależność ogólnego wyniku modelu od rozmiaru komitetu. Wykres tych zależności przedstawia rysunek 3.16.



Rysunek 3.16: Zagregowany ranking badanych modeli klasyfikujących



Rysunek 3.3: Zbiorowe wyniki wszystkich przeprowadzonych symulacji na wszystkich zbiorach danych

## Rozdział 4

# Podsumowanie

Analiza otrzymanych wyników pozwoliła na wysunięcie kilku wniosków dotyczących badań przeprowadzonych podczas realizacji projektu.

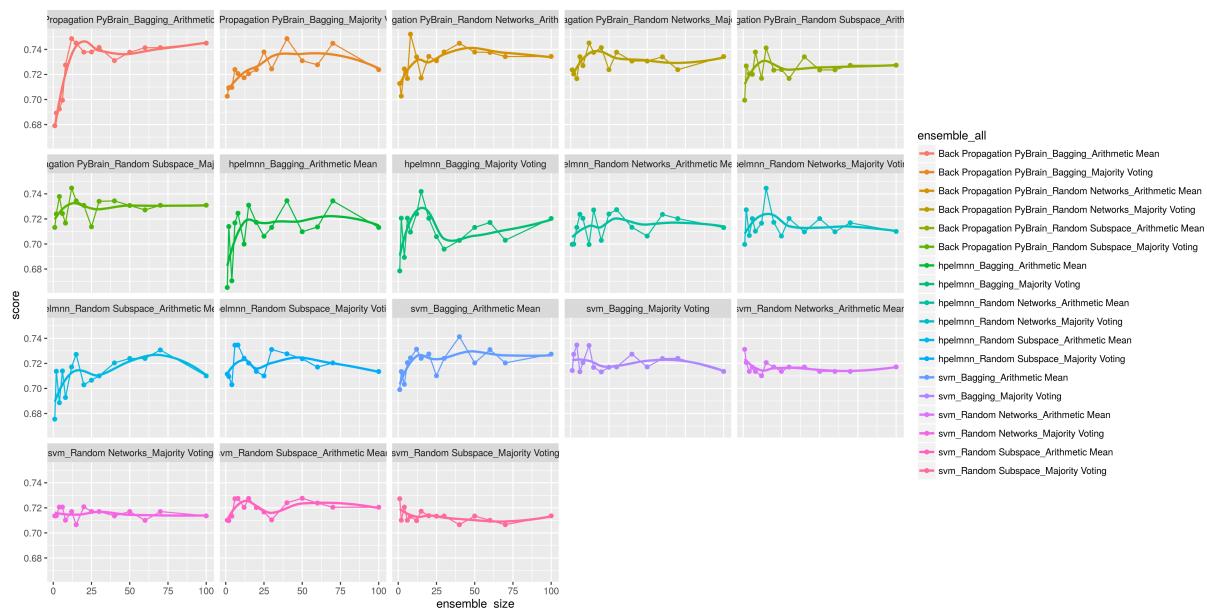
Ważnym aspektem badania skuteczności modeli jest wcześniejsze przygotowanie danych. Należy dobierać zróżnicowane zbiory tak, aby otrzymać jak najszerzy obraz skuteczności każdego z modeli. Istotne jest również przetworzenie danych, ponieważ implementacje sieci neuronowych zwykle oczekują sygnałów wejściowych w postaci liczb rzeczywistych z zakresu  $\{0, 1\}$ . Podczas realizacji projektu przygotowano narzędzia ułatwiające przystosowanie nowych zbiorów danych do istniejącego systemu.

W przypadku badania różnych typów klasyfikatorów nie jest kluczowe uzyskanie jak najlepszej skuteczności pojedynczego klasyfikatora, natomiast ważny jest krótki czas potrzebny na naukę sieci, z uwagi na duże rozmiary komitetów w badaniach. Znalezienie dobrego stosunku skuteczności pojedynczego klasyfikatora bazowego do czasu potrzebnego na przeprowadzanie dla niego pełnych badań może być trudnym zadaniem.

Jednym z celów projektu było wyznaczenie zależności skuteczności komitetu od jego rozmiaru. Wybranie odpowiednich wartości okazało się istotne z punktu widzenia czasu przeprowadzania badań. Różnice w skuteczności pomiędzy dużymi wartościami rozmiaru dla danego modelu są mniejsze niż dla małych rozmiarów, dlatego warto zmniejszać zagęszczenie próbek rozmiaru wraz z jego wzrostem. Pozwala to zmniejszyć czas badań przy zachowaniu ogólnej zależności skuteczności od rozmiaru.

## Dodatek A

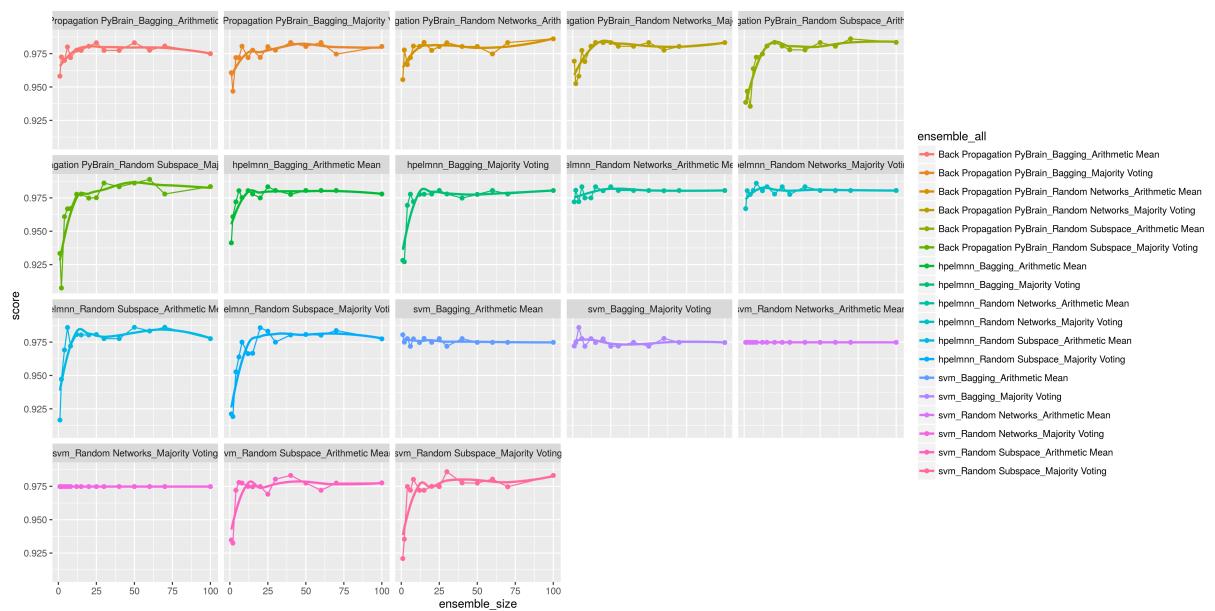
# Graficzna reprezentacja wyników



Rysunek A.1: breast\_cancer



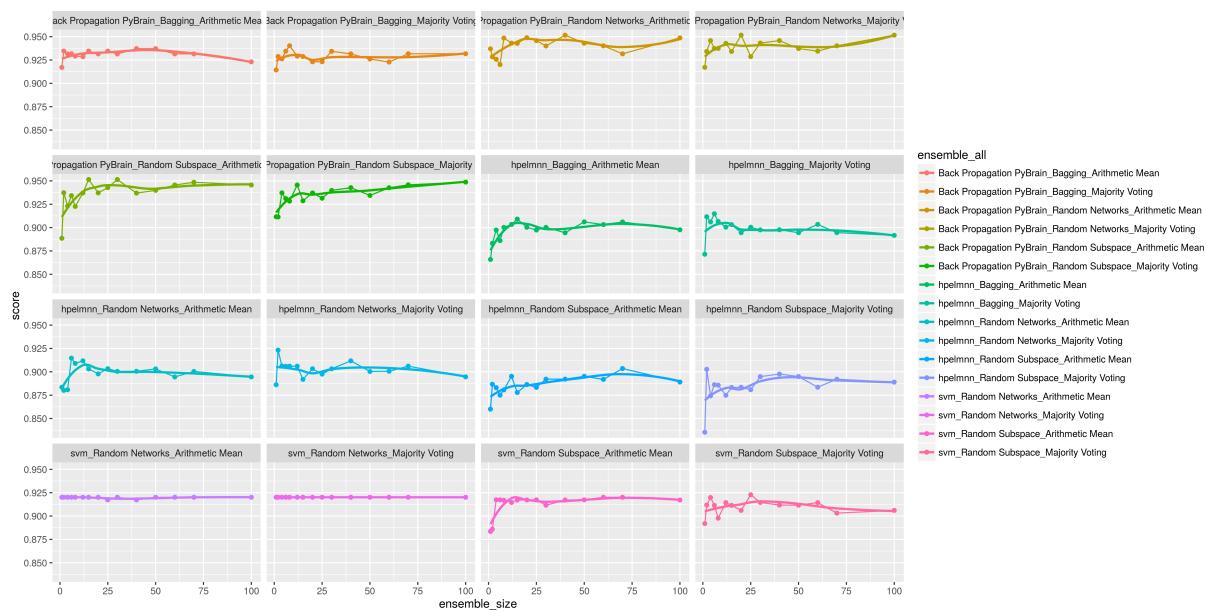
Rysunek A.2: crx



Rysunek A.3: dermatology



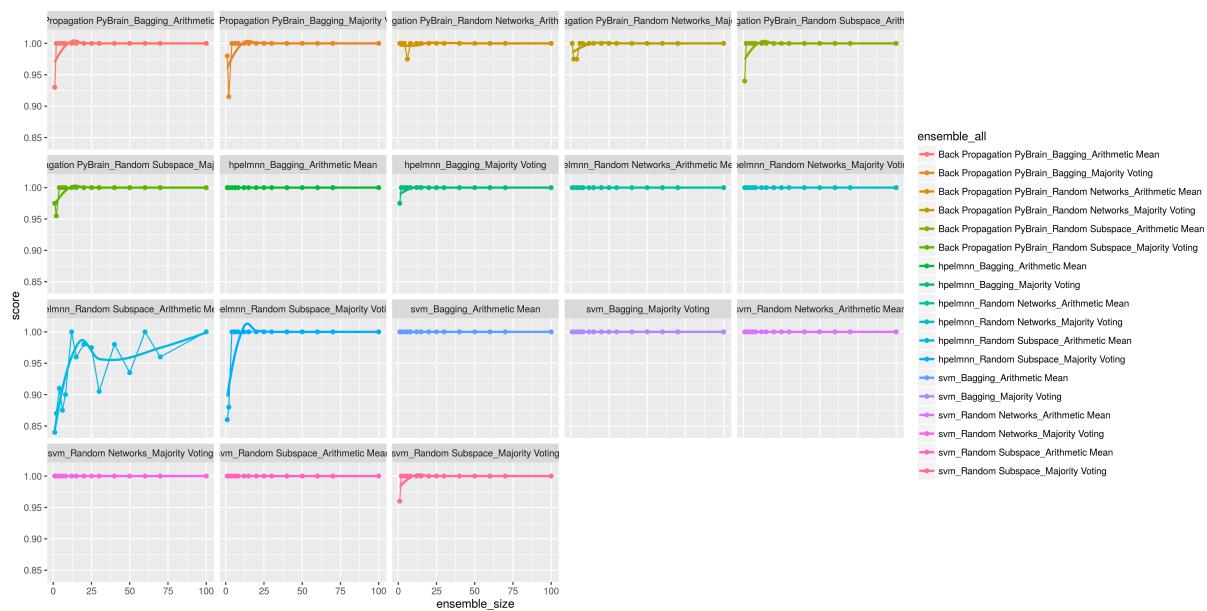
Rysunek A.4: house\_votes



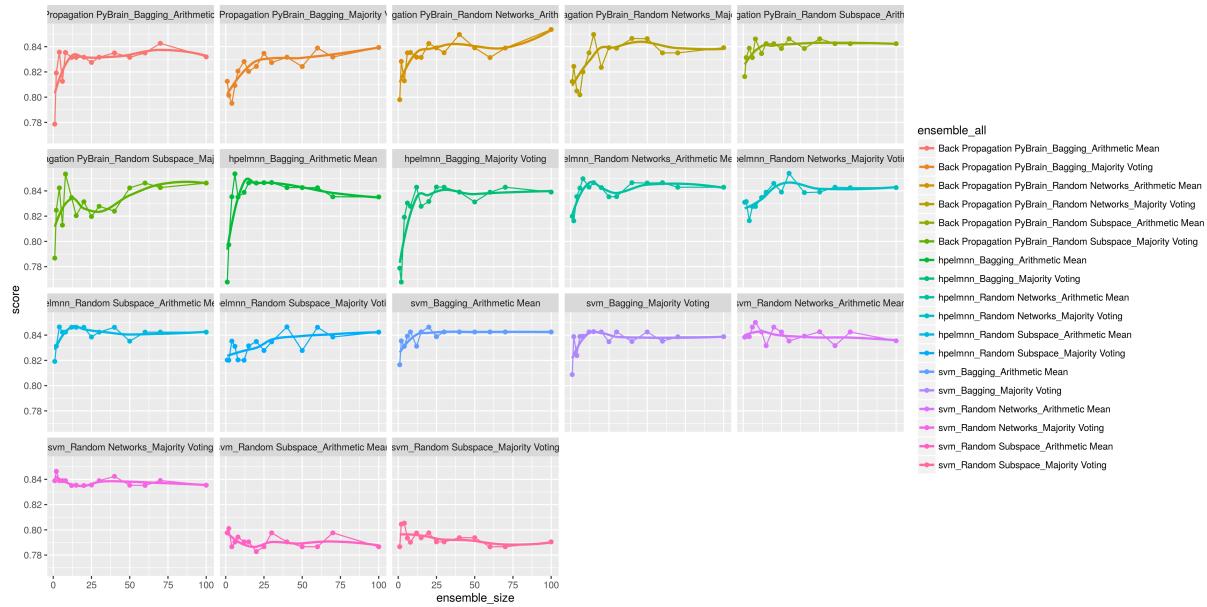
Rysunek A.5: ionosphere



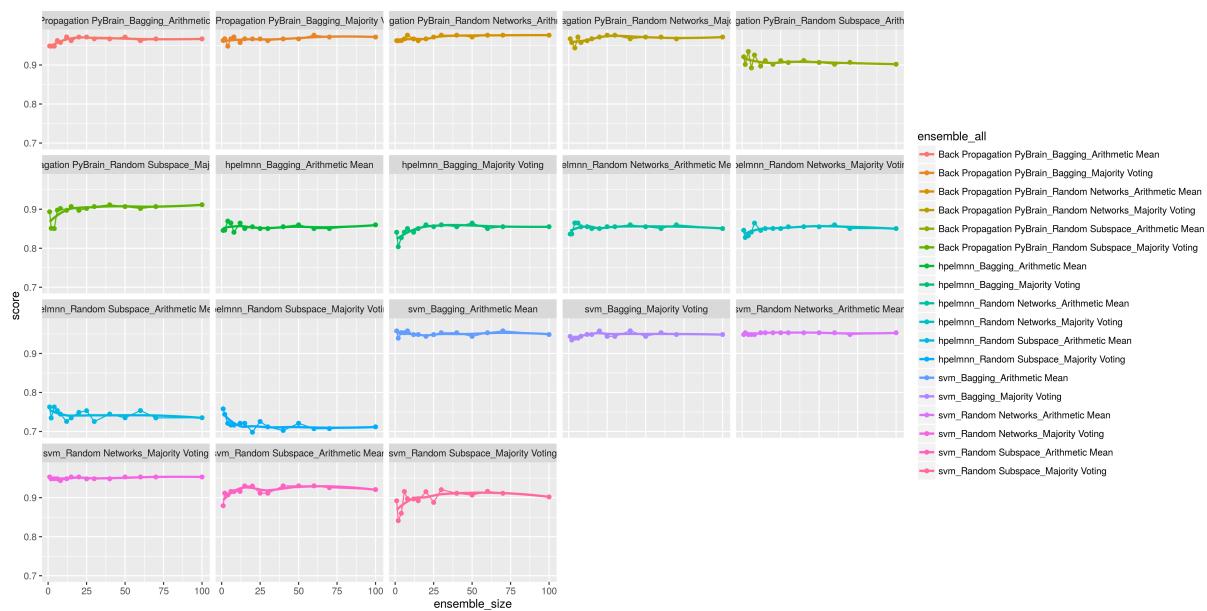
Rysunek A.6: iris



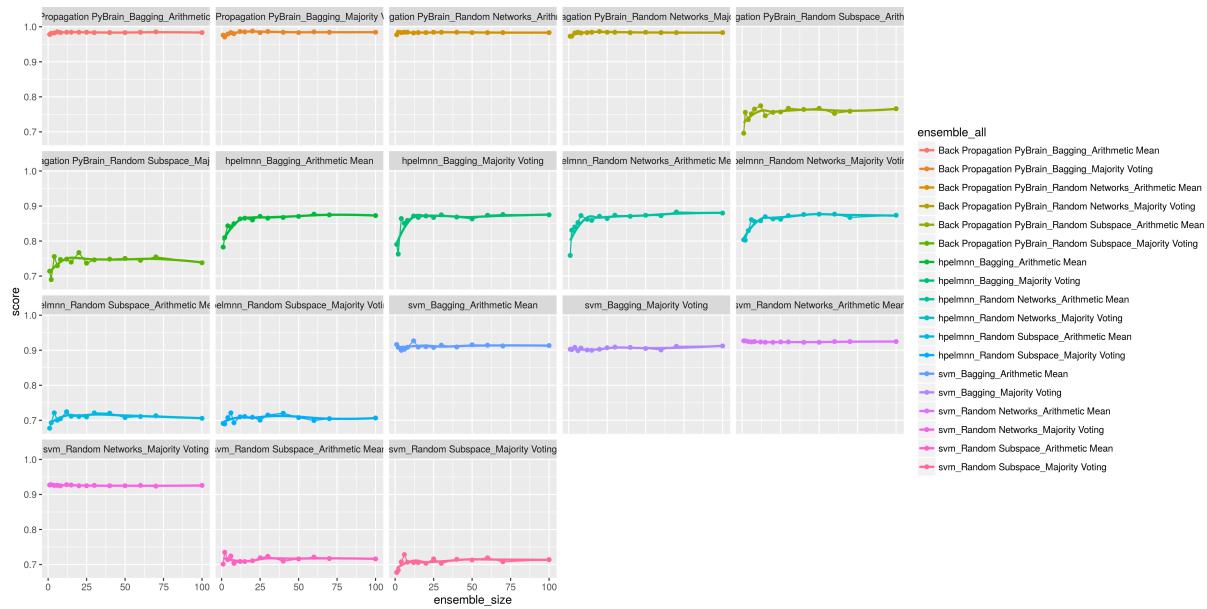
Rysunek A.7: soybean\_small



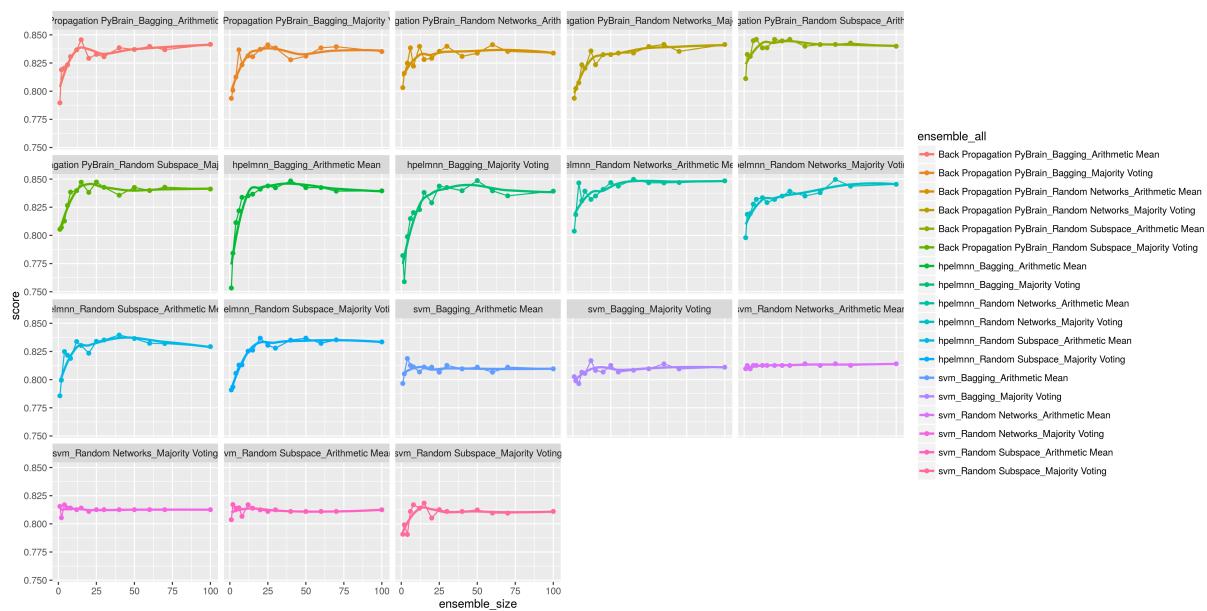
Rysunek A.8: spect



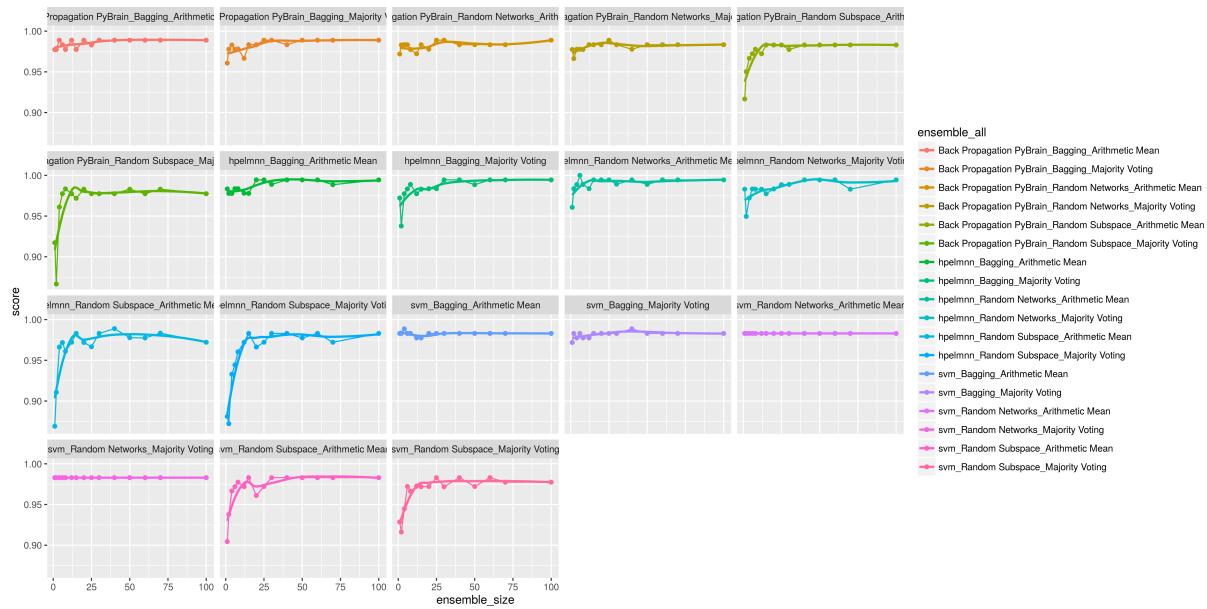
Rysunek A.9: thyroid



Rysunek A.10: tic\_tac\_toe



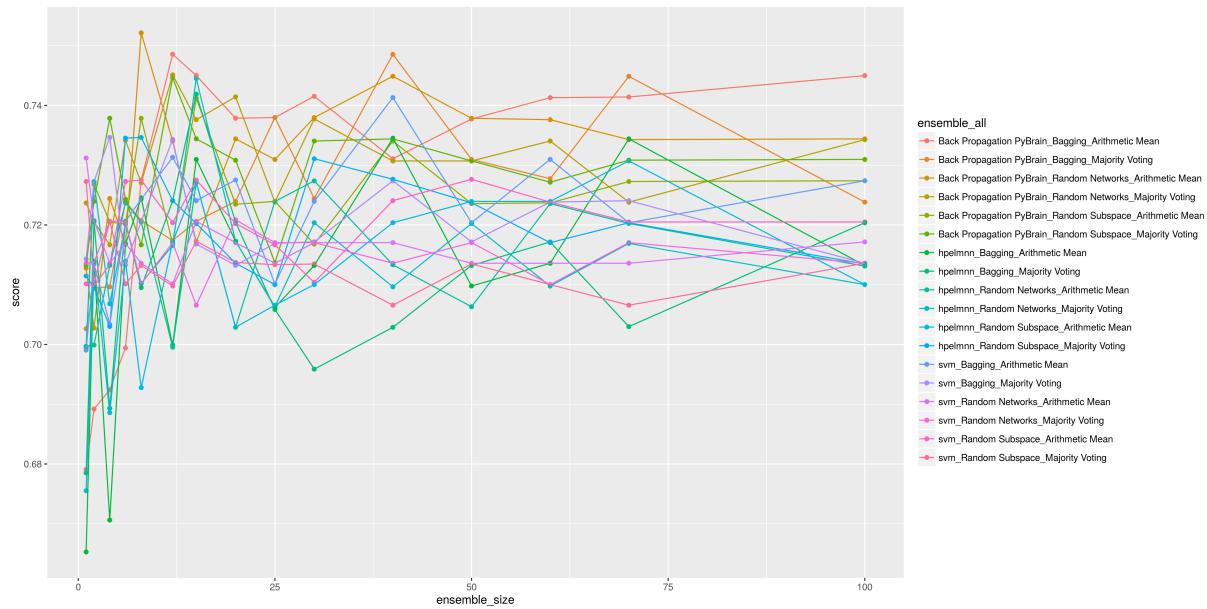
Rysunek A.11: urban\_land\_cover



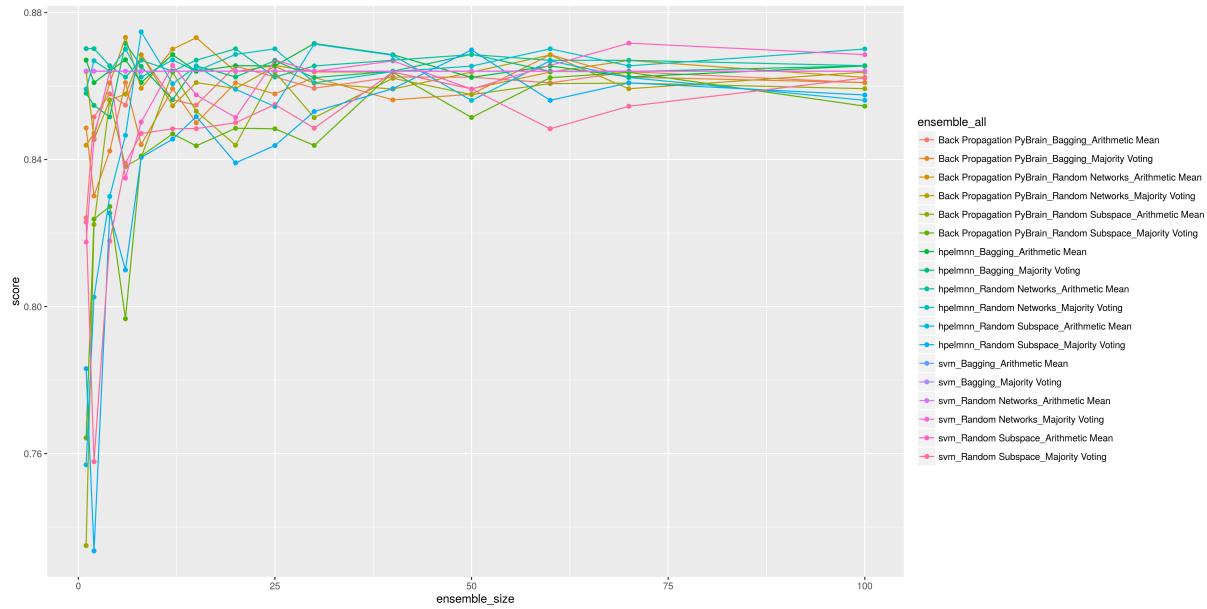
Rysunek A.12: wine



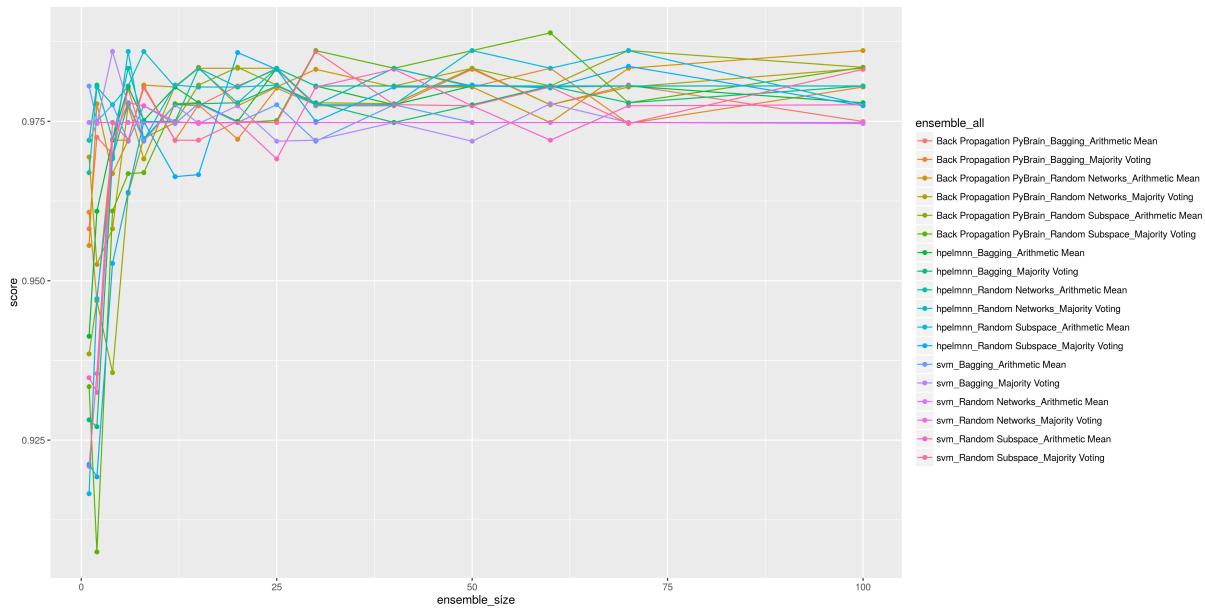
Rysunek A.13: yeast



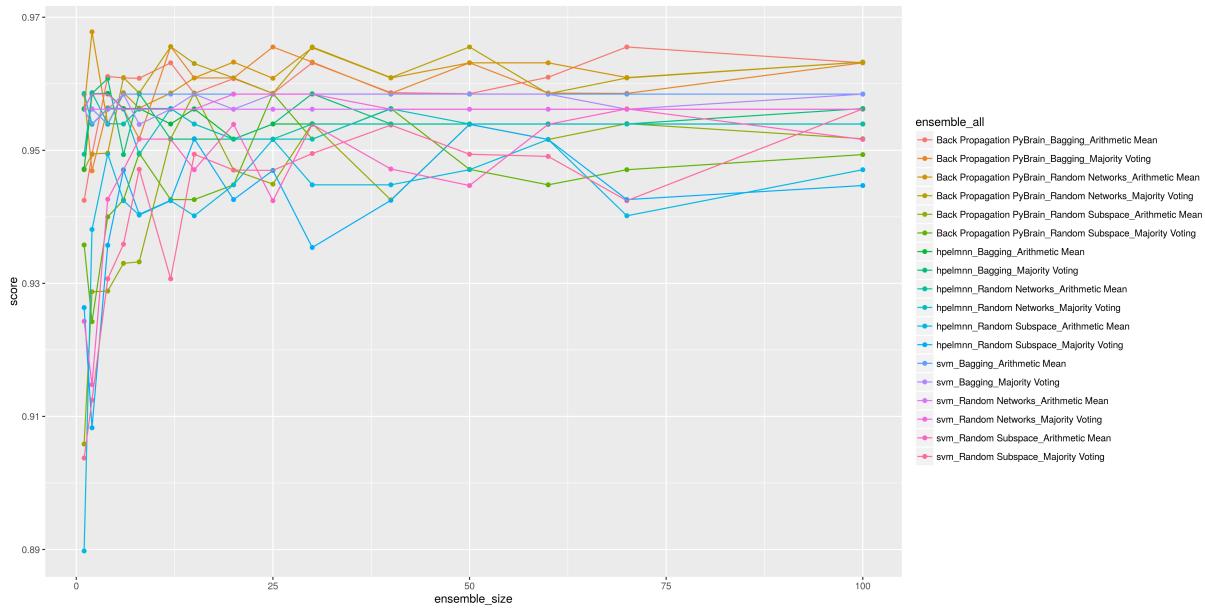
Rysunek A.14: breast\_cancer



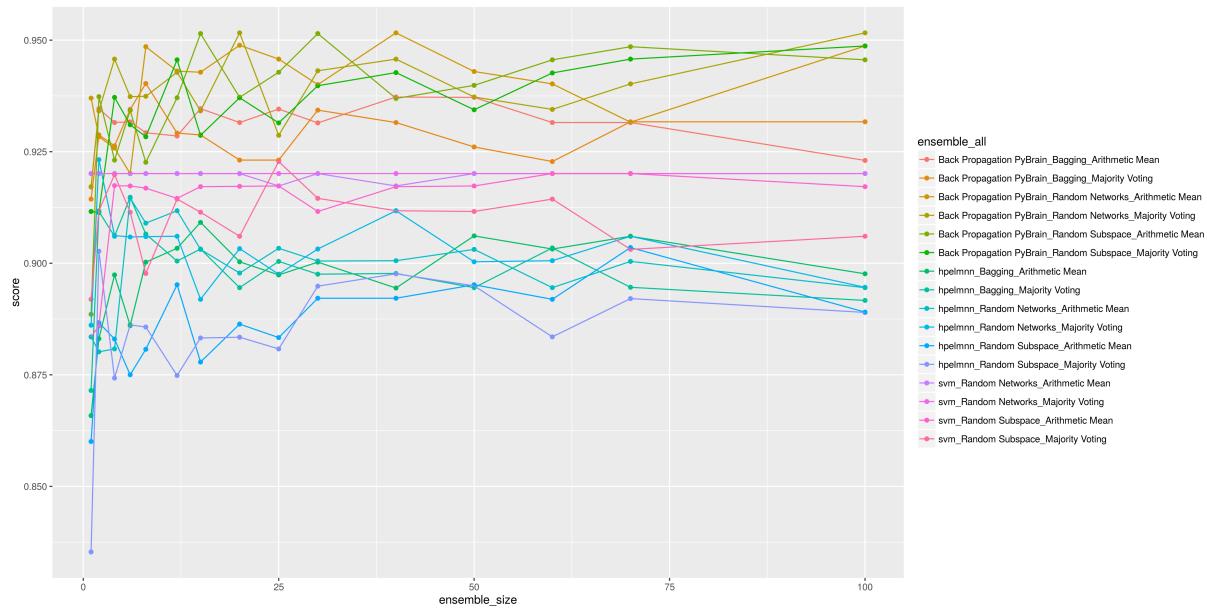
Rysunek A.15: crx



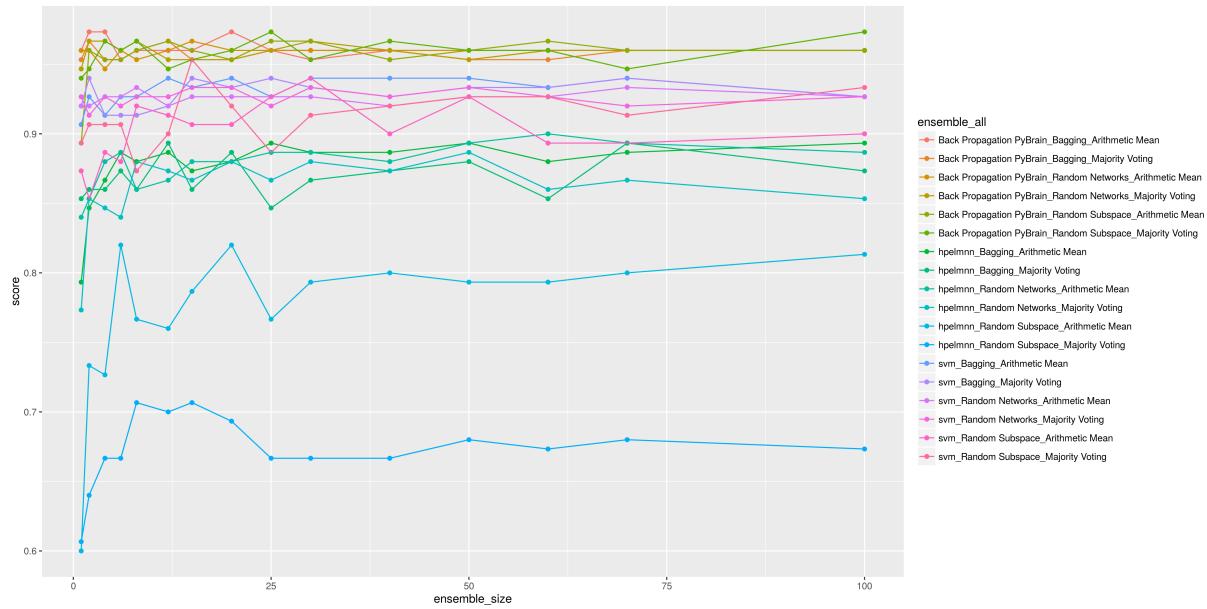
Rysunek A.16: dermatology



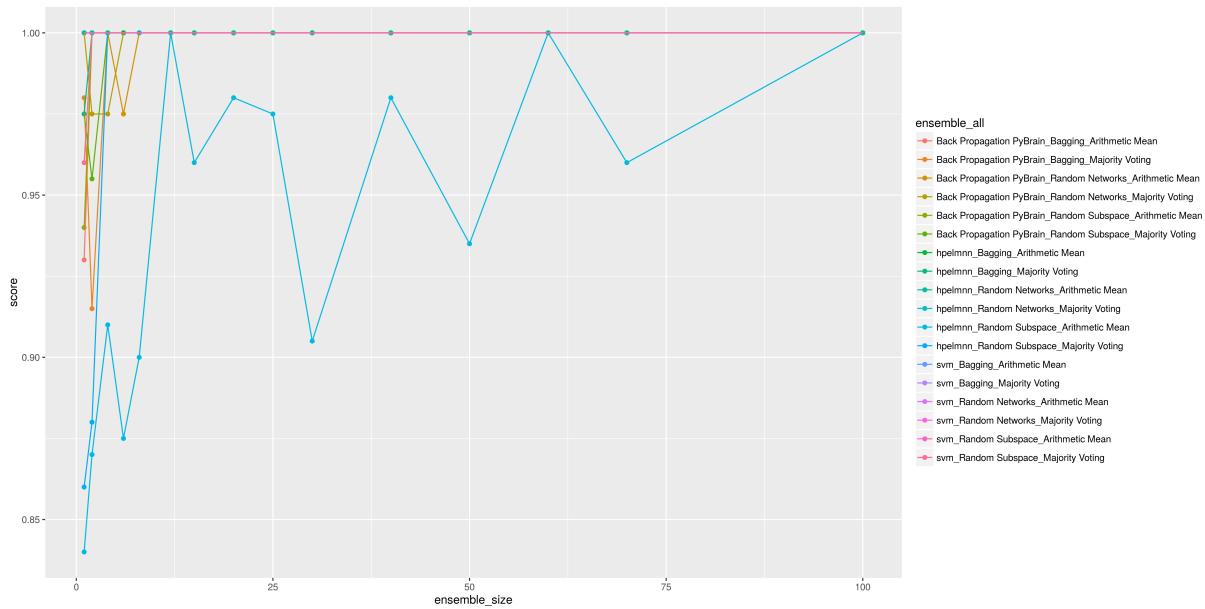
Rysunek A.17: house\_votes



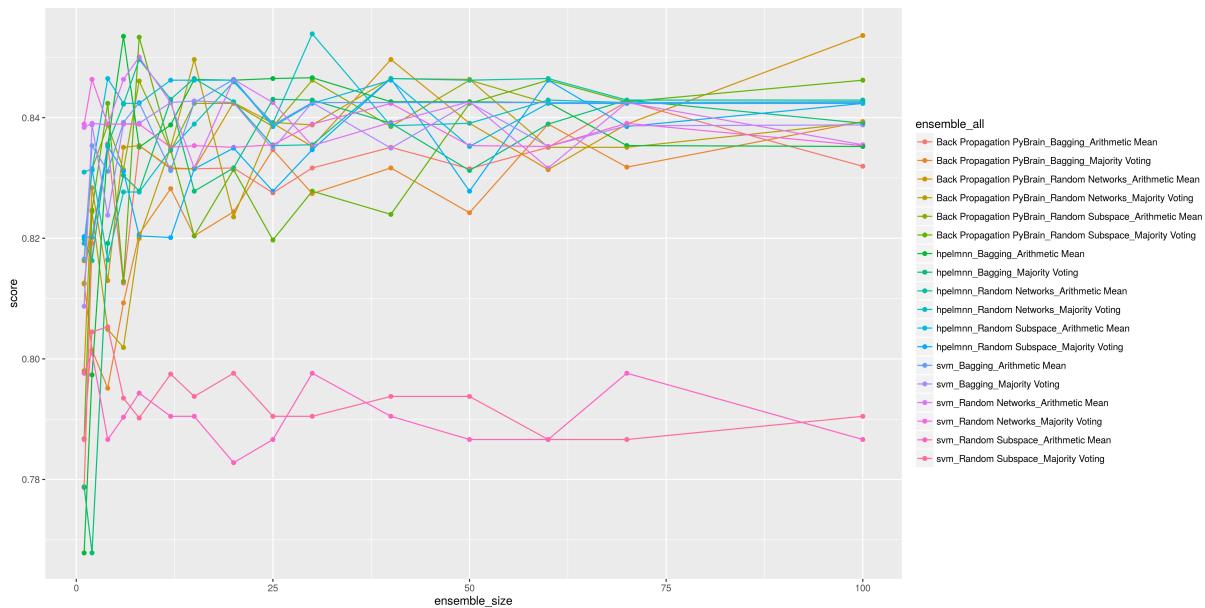
Rysunek A.18: ionosphere



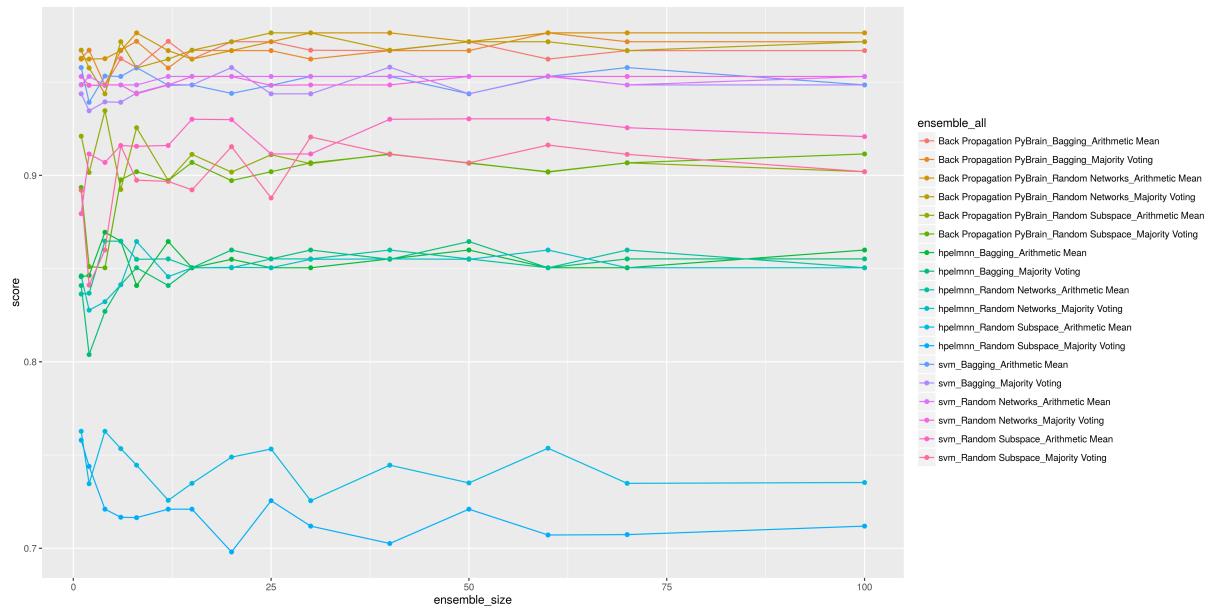
Rysunek A.19: iris



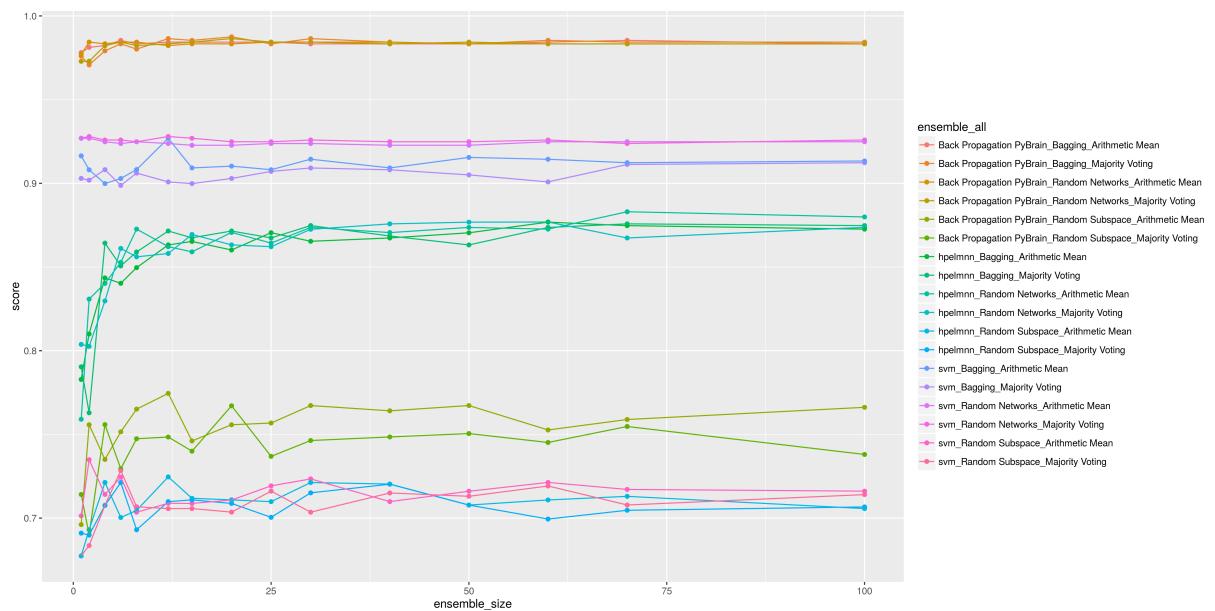
Rysunek A.20: soybean\_small



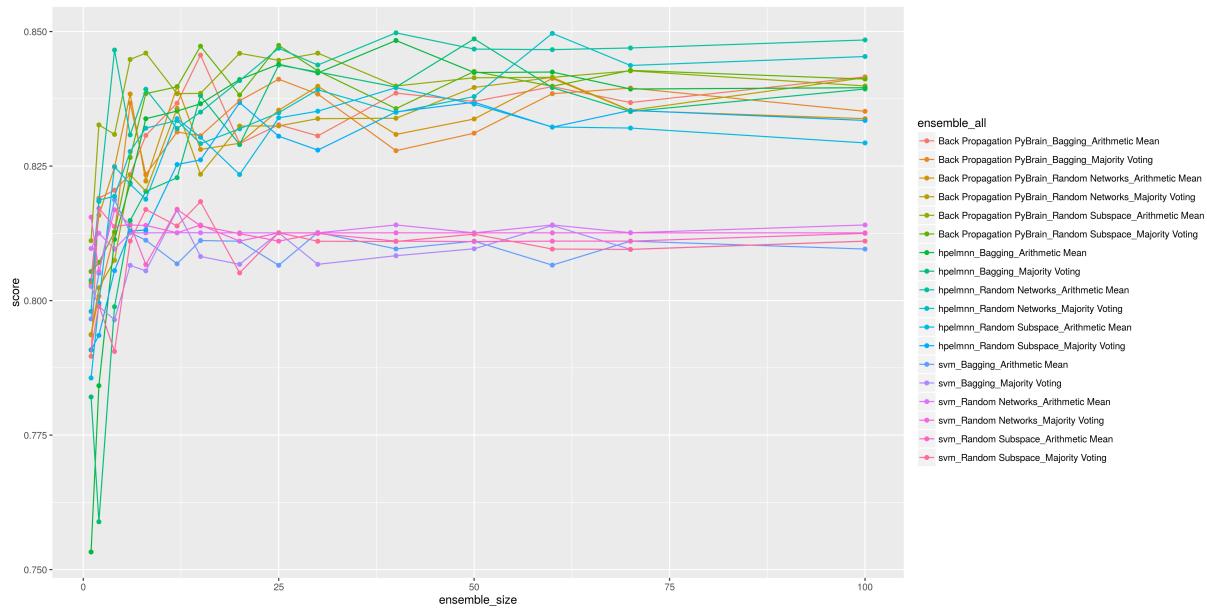
Rysunek A.21: spect



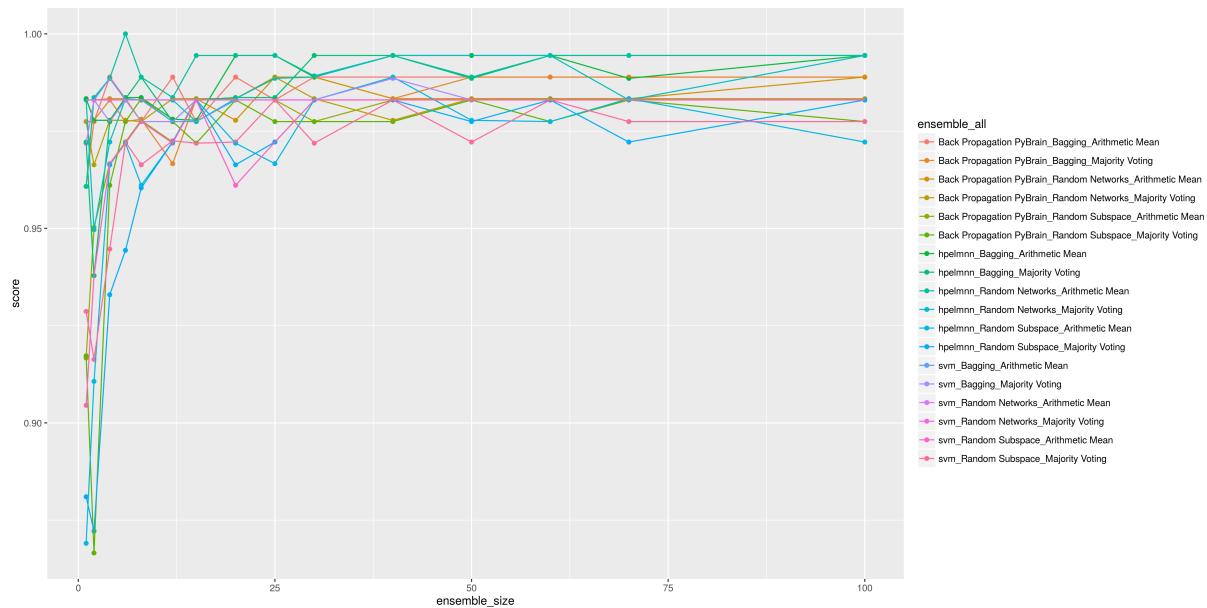
Rysunek A.22: thyroid



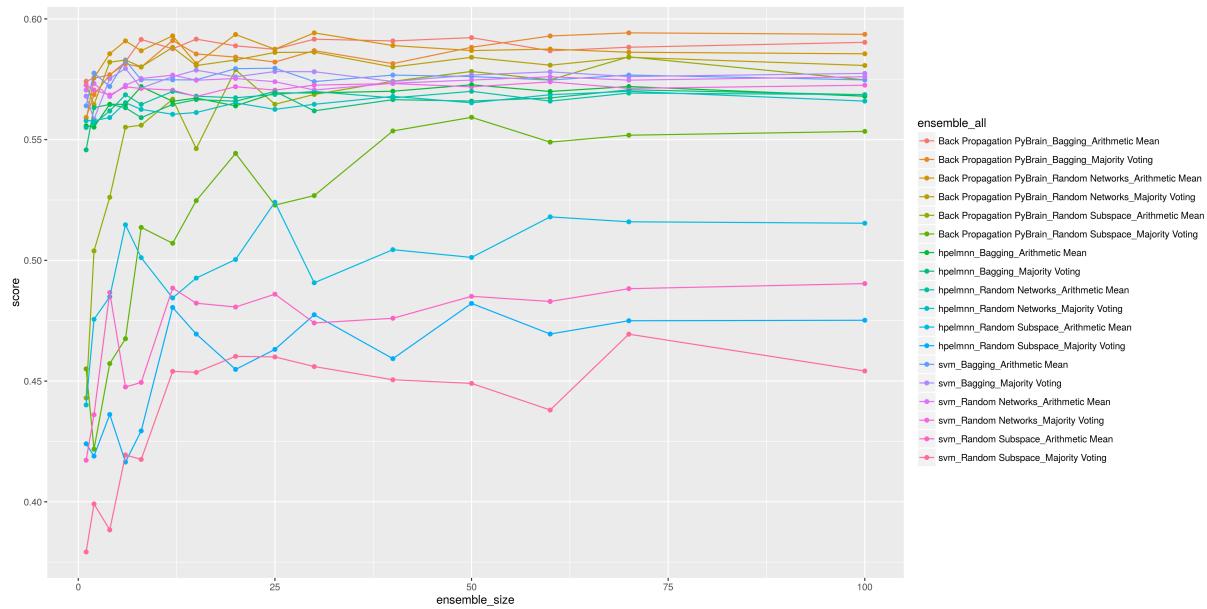
Rysunek A.23: tic\_tac\_toe



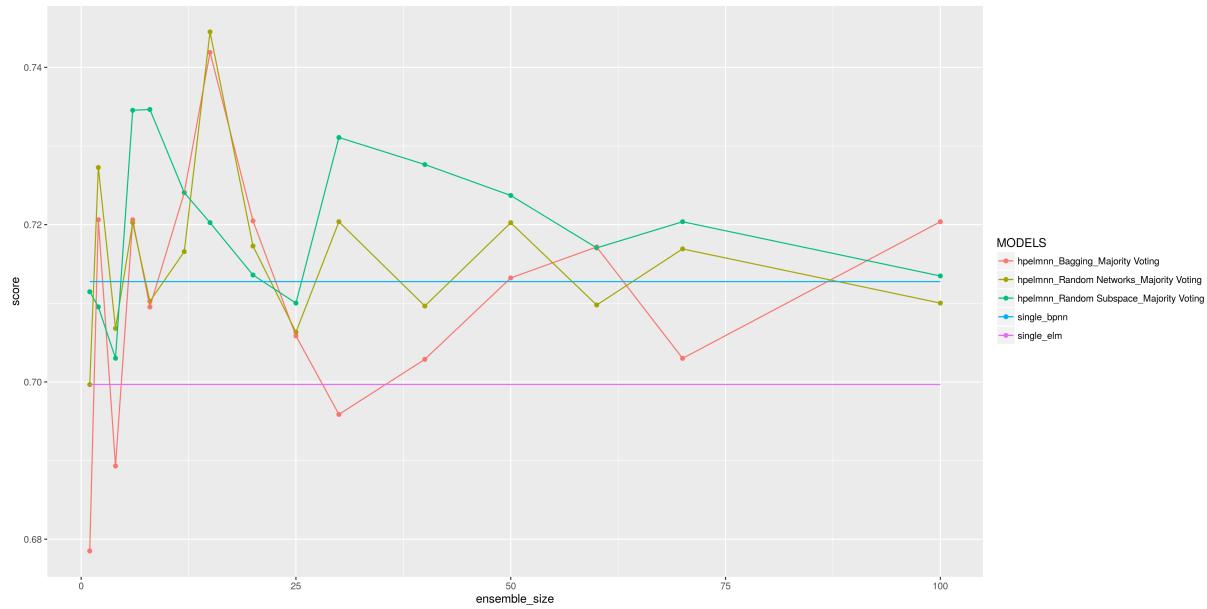
Rysunek A.24: urban\_land\_cover



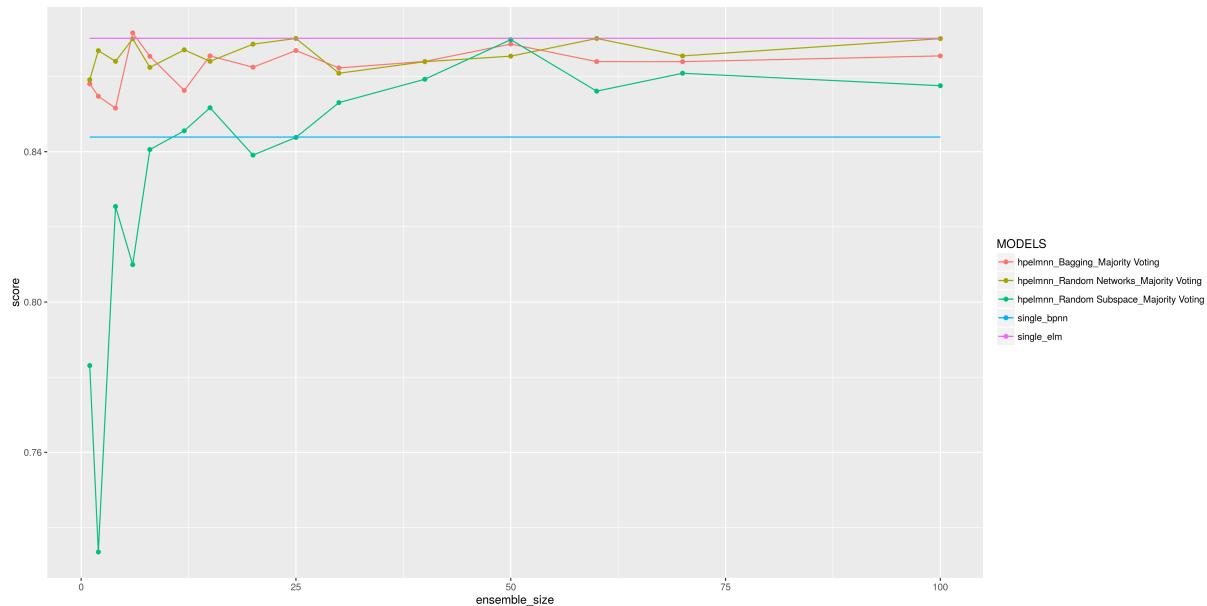
Rysunek A.25: wine



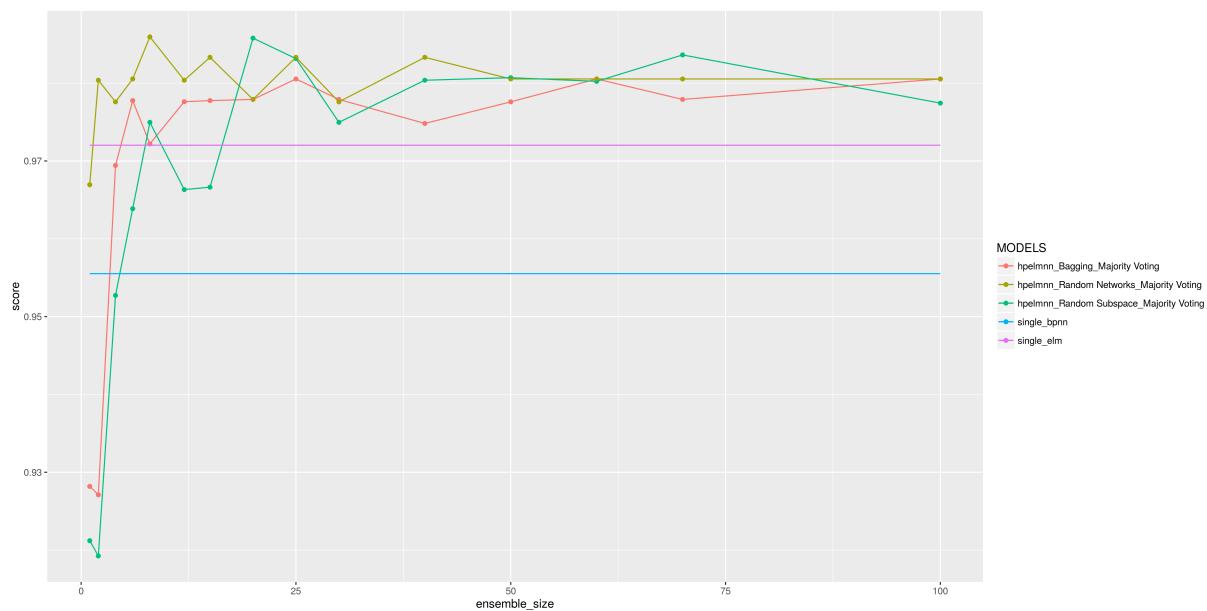
Rysunek A.26: yeast



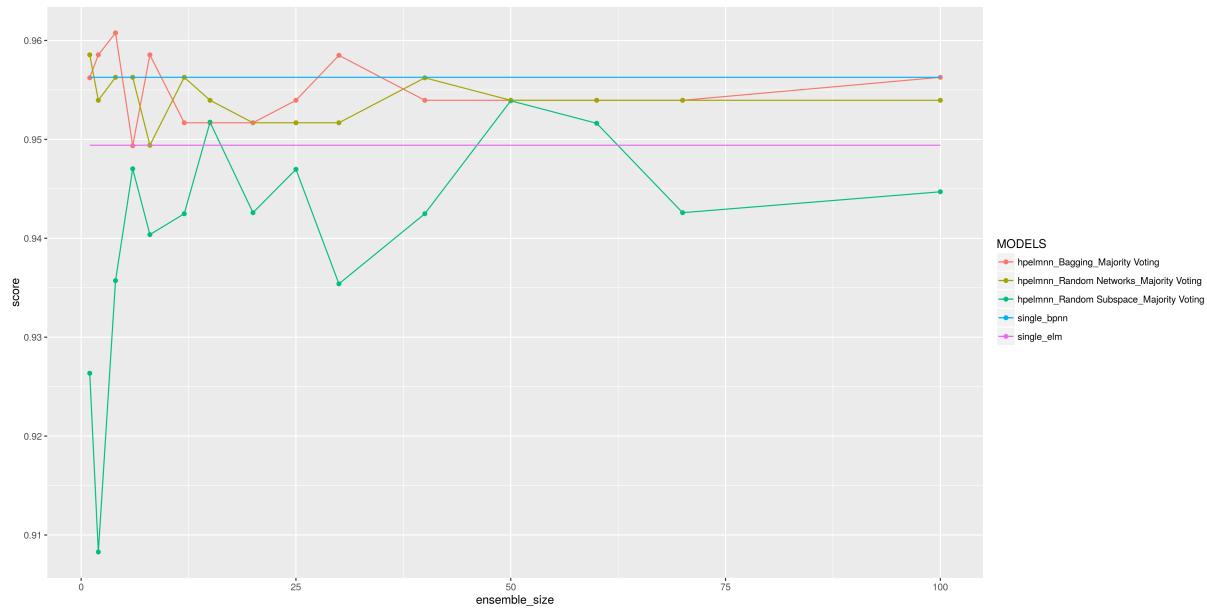
Rysunek A.27: breast\_cancer



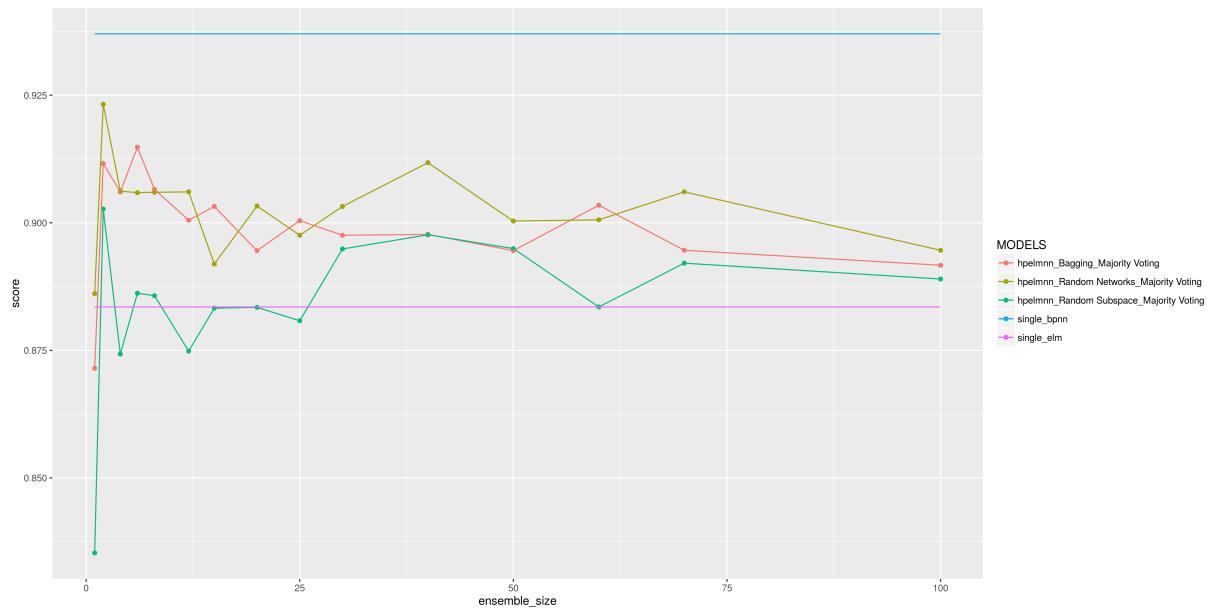
Rysunek A.28: crx



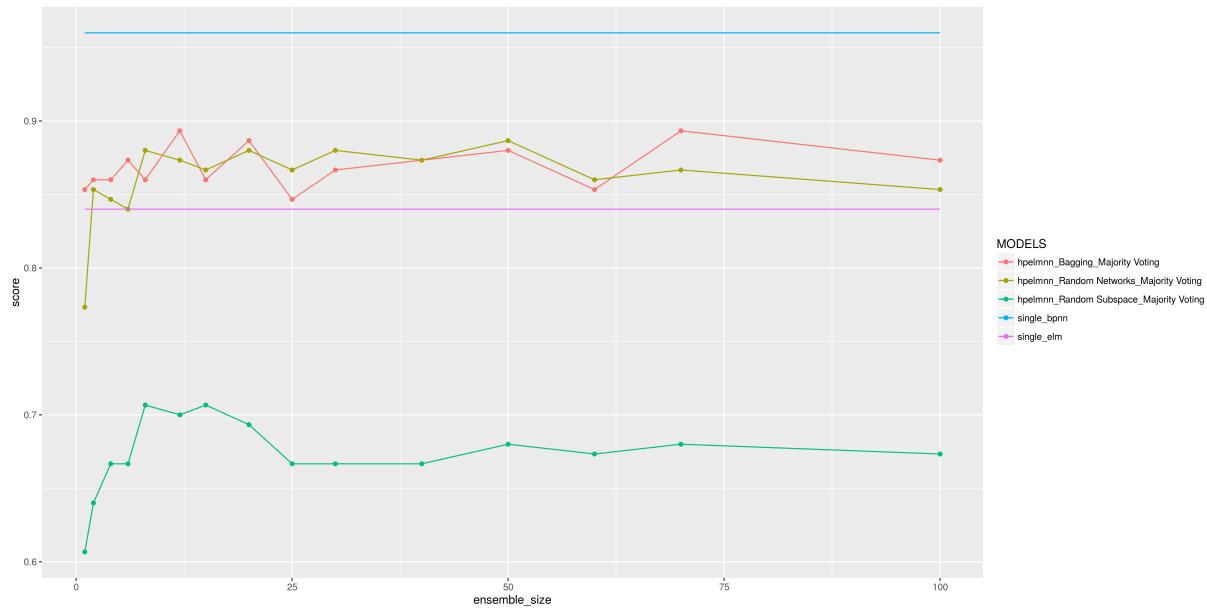
Rysunek A.29: dermatology



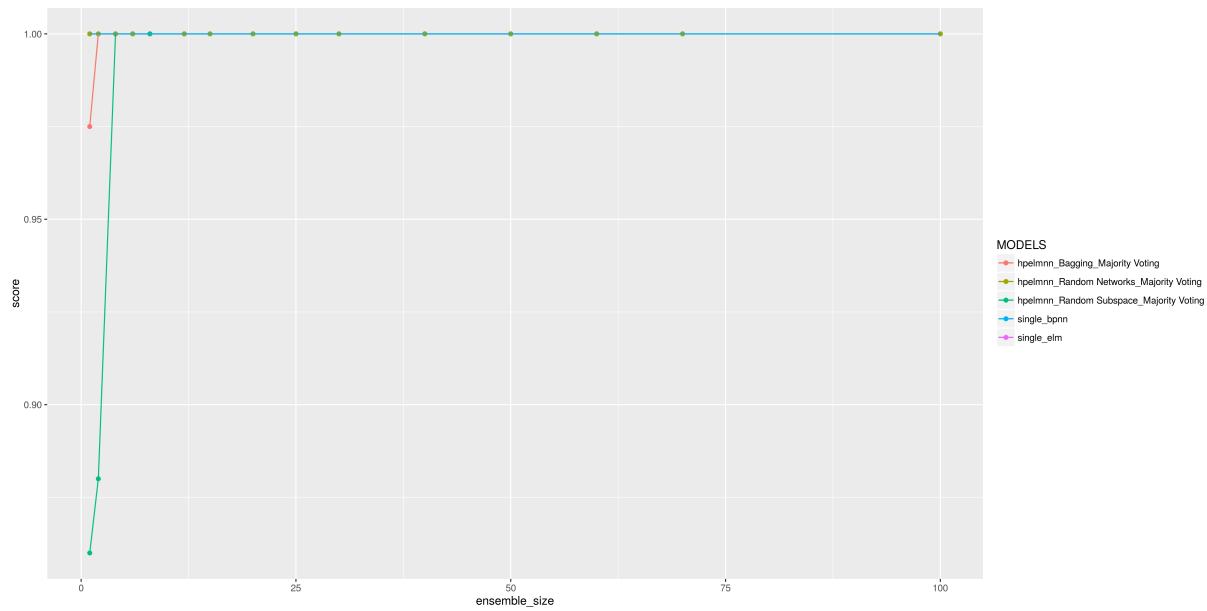
Rysunek A.30: house\_votes



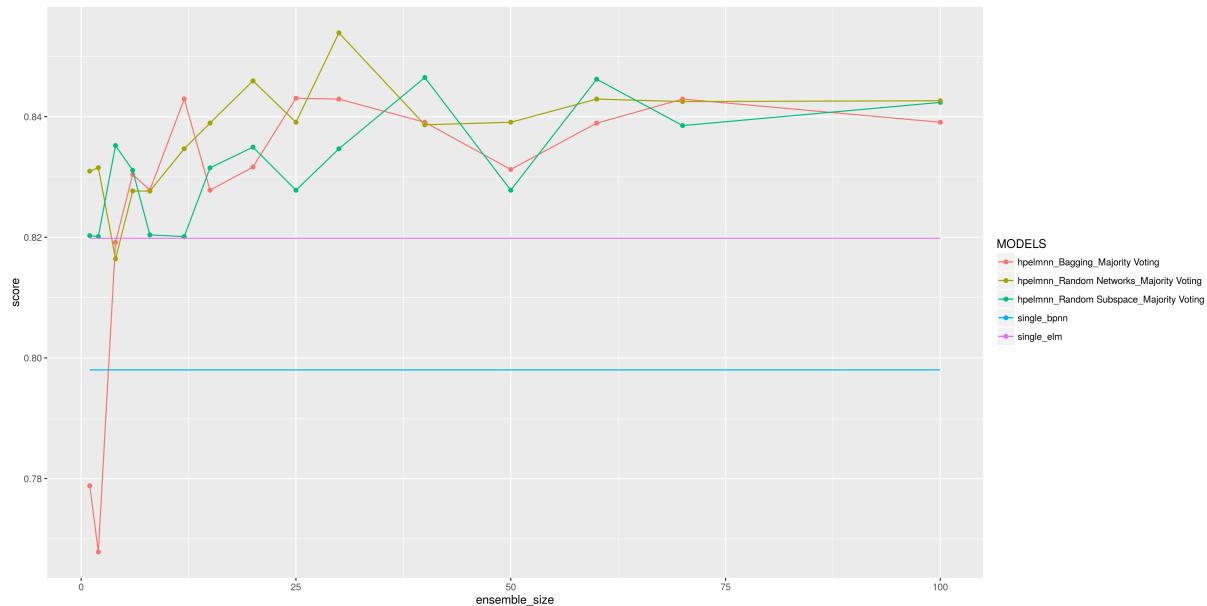
Rysunek A.31: ionosphere



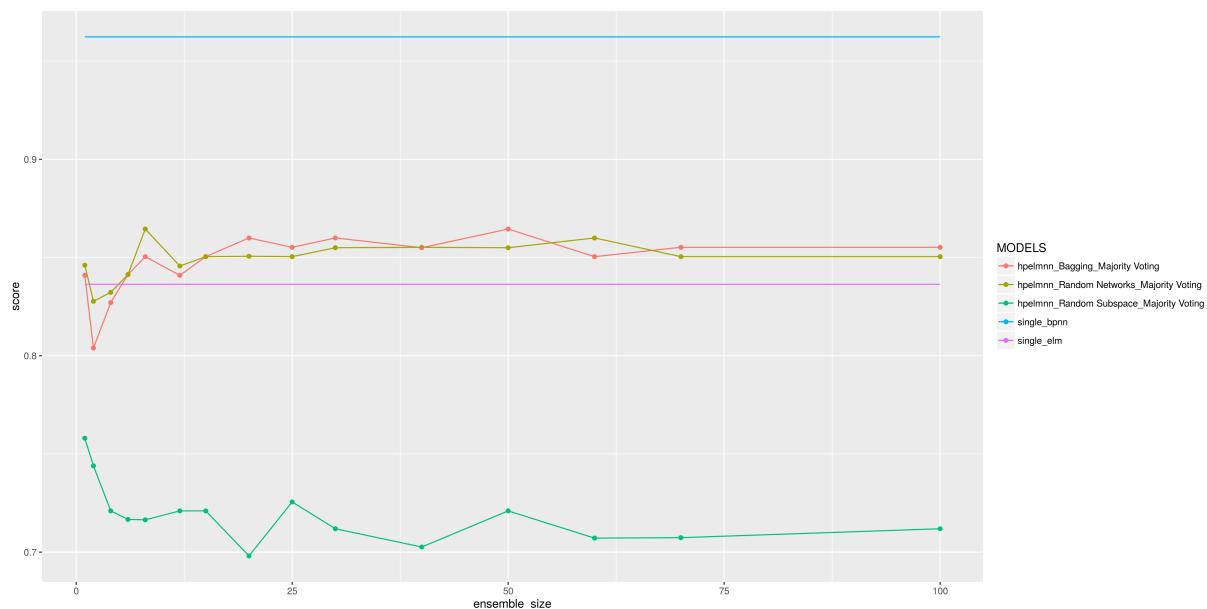
Rysunek A.32: iris



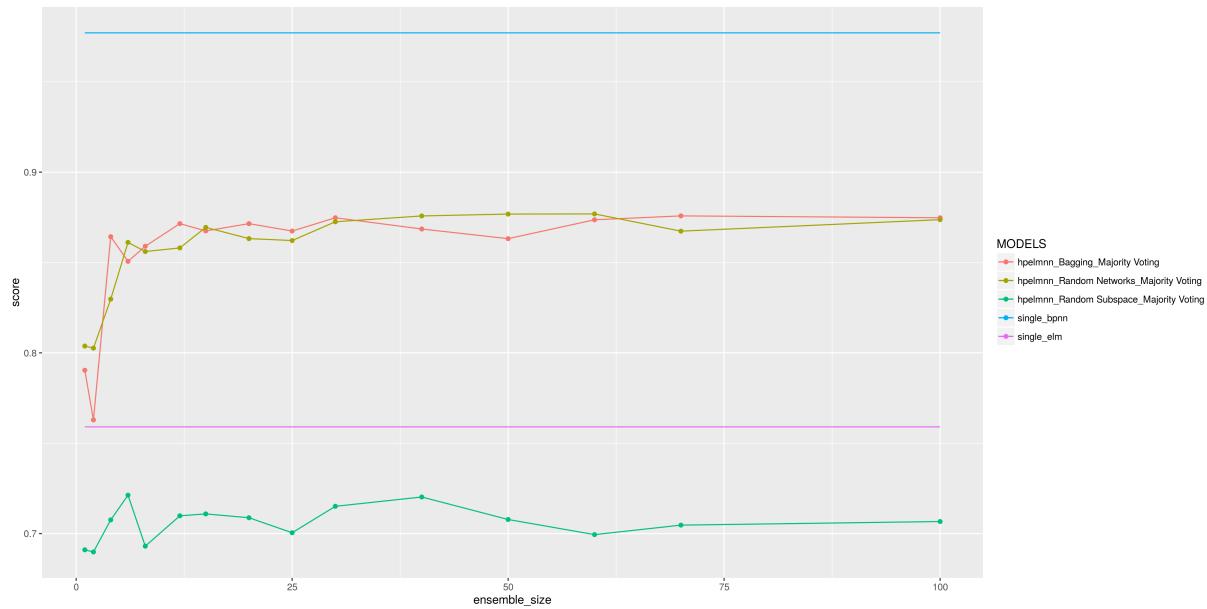
Rysunek A.33: soybean\_small



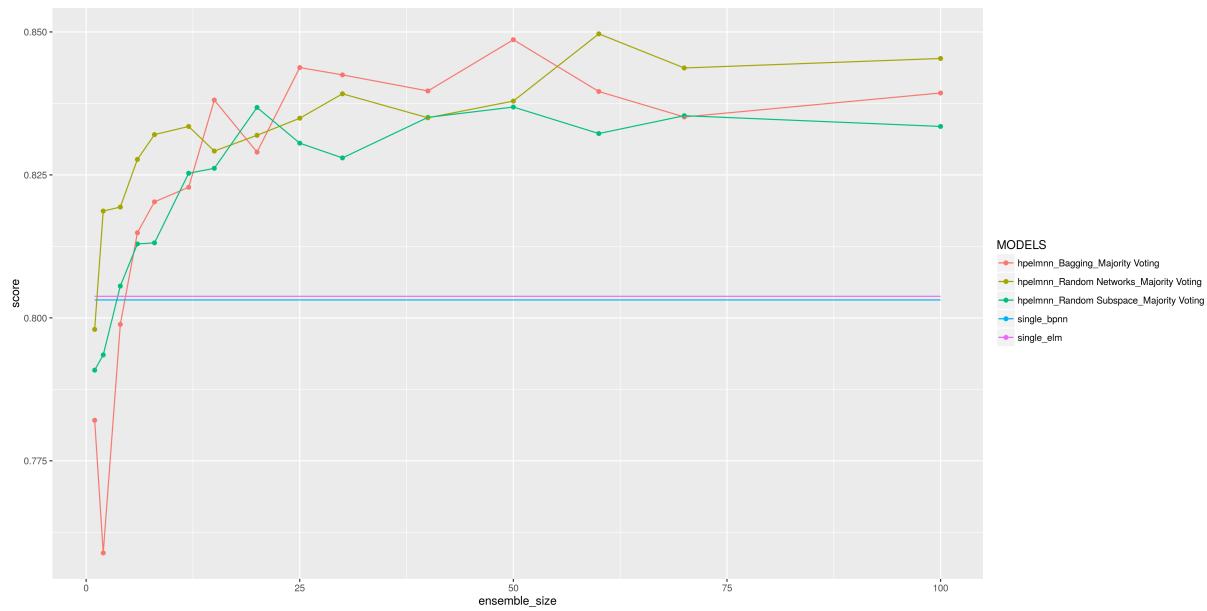
Rysunek A.34: spect



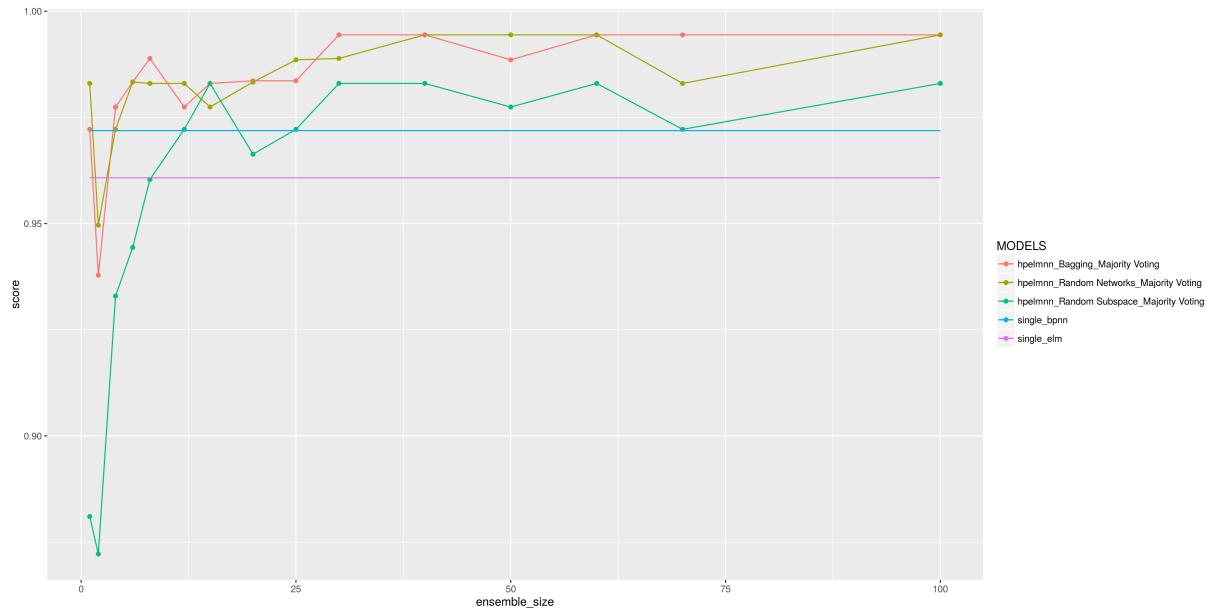
Rysunek A.35: thyroid



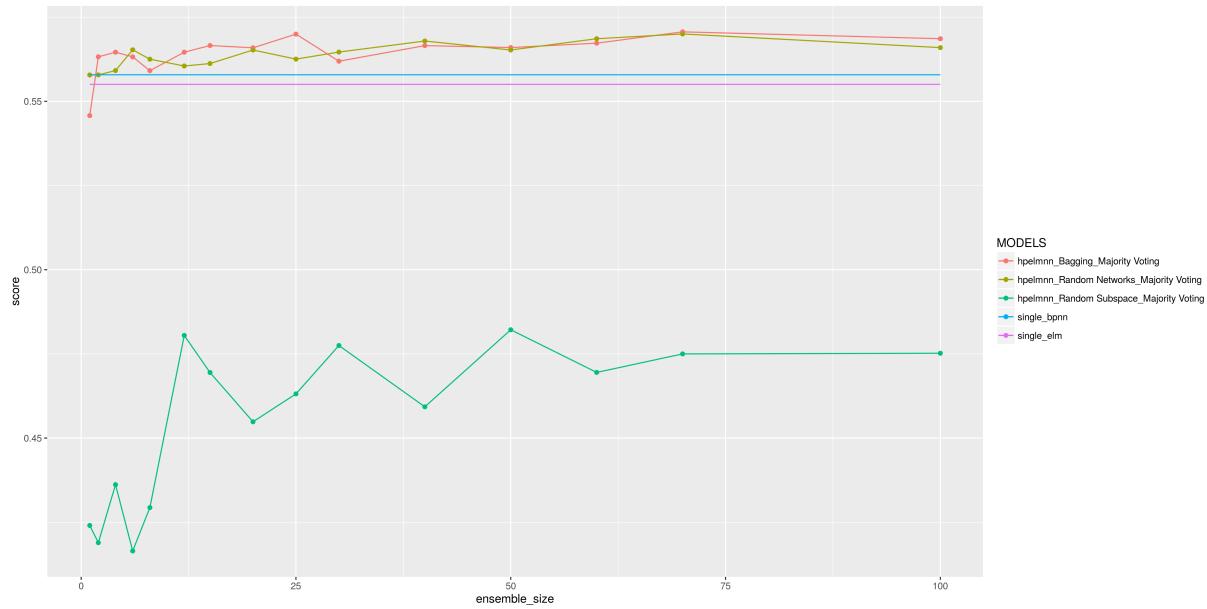
Rysunek A.36: tic\_tac\_toe



Rysunek A.37: urban\_land\_cover



Rysunek A.38: wine



Rysunek A.39: yeast

# Bibliografia

- [1] Huang, Guang-Bin and Zhu, Qin-Yu and Siew, Chee Kheong, *Extreme learning machine: Theory and applications.*, Neurocomputing, 2006
- [2] J. Cao, Z. Lin, G. Bin Huang and N. Liu, *Voting based extreme learning machine*, Inf. Sci. (Ny)., vol. 185, pp. 66-77, 2012
- [3] Akusok, A. and Björk, K.-M. and Miche, Y. and Lendasse, A., *High-Performance implementation of an Extreme Learning Machine* [<https://pypi.python.org/pypi/hpelm>]
- [4] scikit-learn developers, *Scikit-learn Machine Learning in Python* [<http://scikit-learn.org/stable/index.html>]
- [5] M. Lichman, *UCI Machine Learning Repository* [<http://archive.ics.uci.edu/ml>], Irvine, CA: University of California, School of Information and Computer Science, 2013