

# Deep forest

Stanisław Kaźmierczak

## **Deep Forest**

**Zhi-Hua Zhou** and **Ji Feng**

National Key Lab for Novel Software Technology, Nanjing University, Nanjing 210023, China  
{zhouzh, fengj}@lamda.nju.edu.cn

# Sukces głębokiego uczenia

Główne przyczyny sukcesu, zdaniem autorów, głębokich sieci neuronowych:

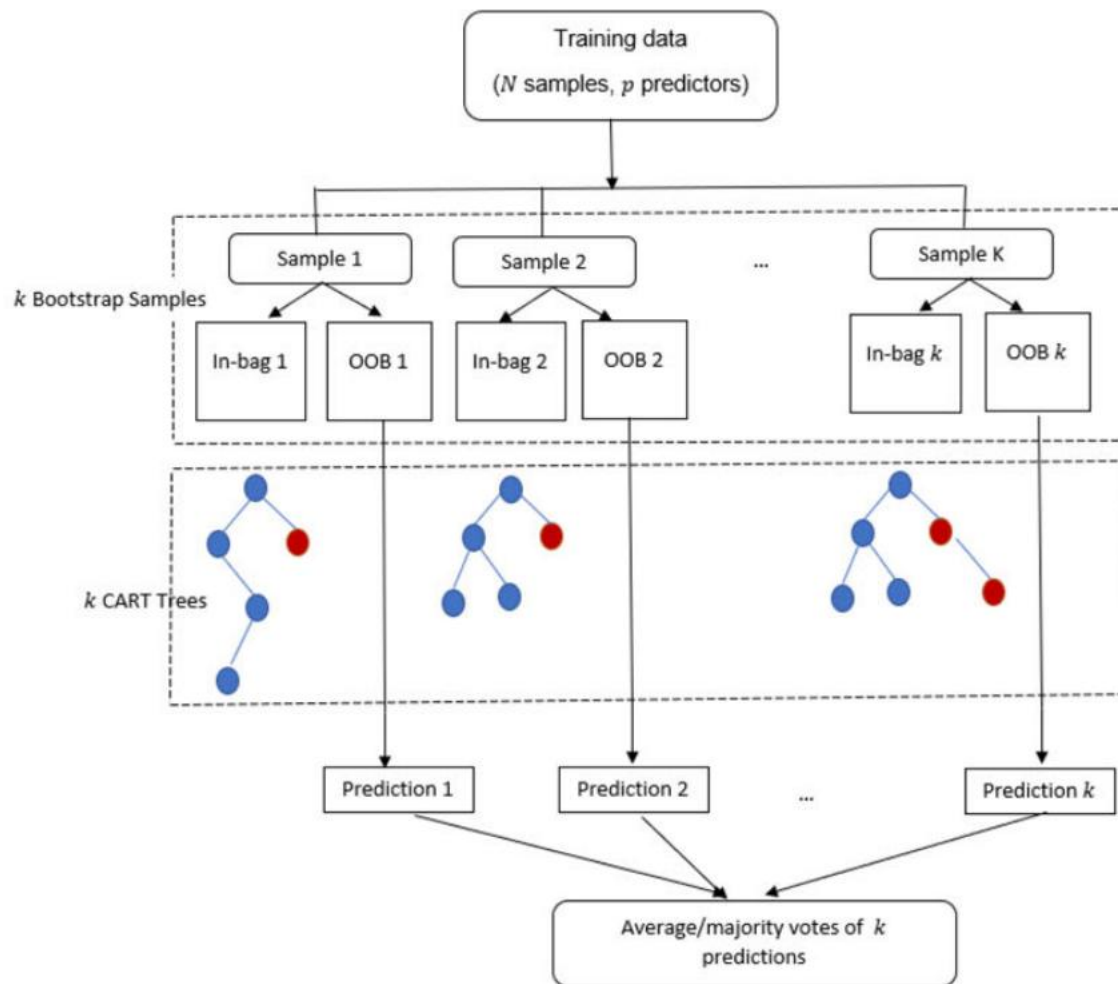
- Procesowanie warstwa po warstwie.
  - Ekstrakcja coraz bardziej złożonych właściwości.
  - Lepsze wyniki niż model o tej samej liczbie parametrów z jedną warstwą.
- Wbudowane w model mechanizmy transformacji cech.
- Wystarczająca złożoność modelu.

# Problemy/niedoskonałości głębokiego uczenia

- Dużo hiperparametrów uczenia.
  - Nieskończona liczba konfiguracji.
  - Trudna analiza teoretyczna.
  - Nie jest oczywiste jak stroić hiperparametry.
  - Jakość nauczonego modelu mocno od nich zależy.
- Potrzeba, generalnie, dużej ilości danych.
  - W wielu domenach danych, zwłaszcza zaetykietowanych, dalej jest mało.
- Architektura, i tym samym złożoność modelu, jest definiowana z góry, przed procesem uczenia.
- Modele są słabo interpretowalne.
- W wielu zadaniach, bazujących głównie na danych tabelarycznych, inne metody, np. las losowy czy gradient boosting, osiągają lepsze rezultaty.

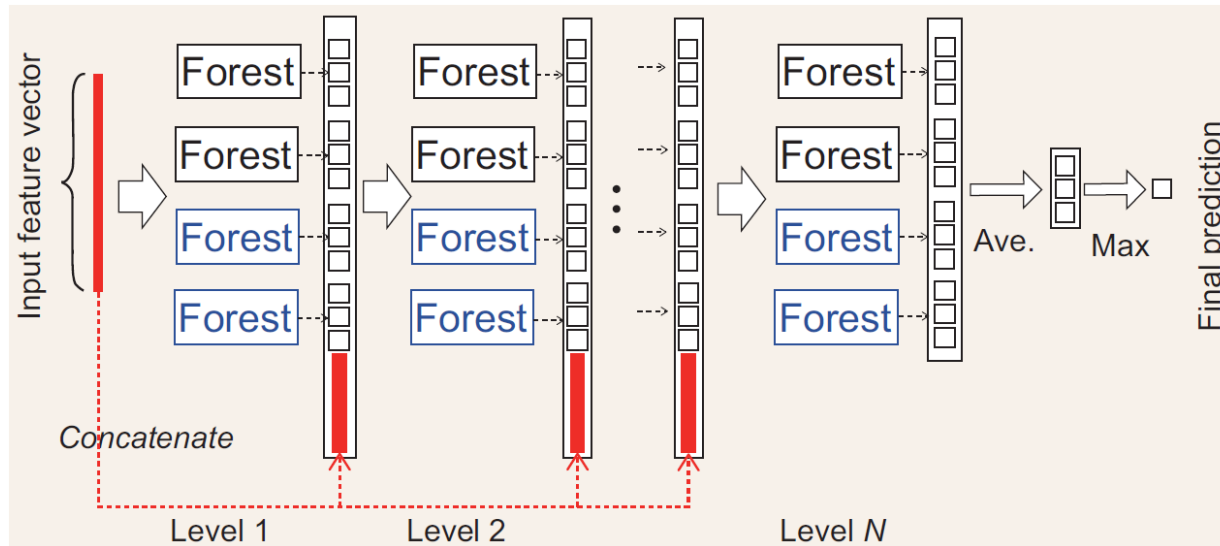
- Cel: zbudować głęboki las zachowujący właściwości głębokiego uczenia odpowiedzialne za jego sukces z jednoczesną mitygacją problemów z nim związanych.
- Głęboki model bazujący na nieróżniczkowalnych modułach.
  - Nie wszystkie „aspekty” otaczającego nas świata są różniczkowalne.
- Rolę nieróżniczkowalnych komponentów pełnią drzewa decyzyjne.
- Właściwości:
  - Mniej hiperparametrów niż głęboka sieć neuronowa.
  - Złożoność modelu automatycznie ustalana w zależności od danych.

# Klasyczny las losowy



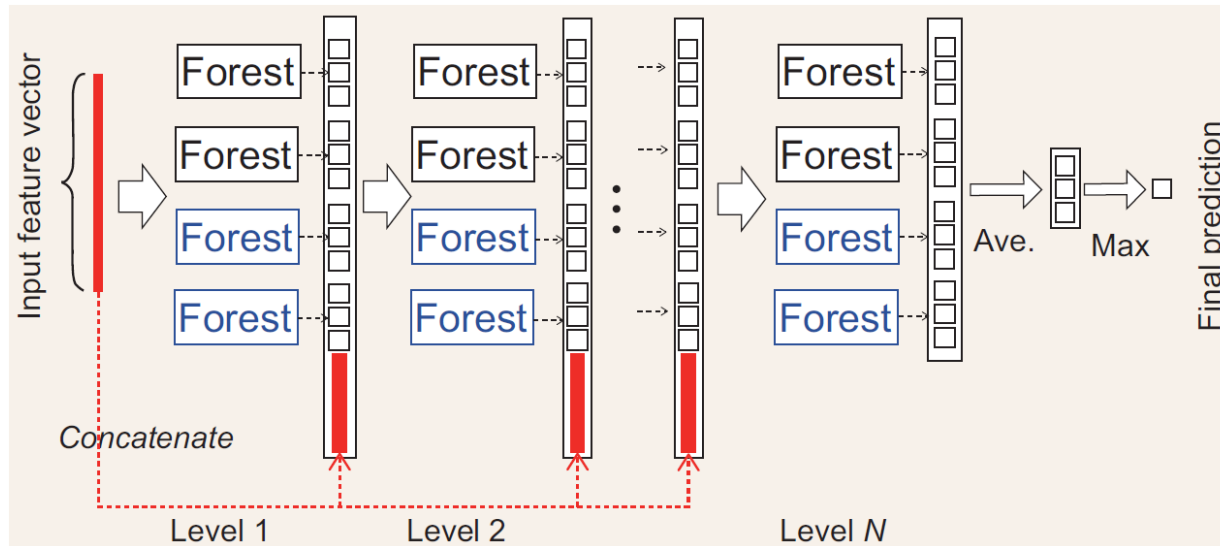
[1]

# Architektura (1)



- Struktura kaskadowa.
- Każdy poziom dostaje przetworzoną informację z wcześniejszej warstwy.
- Oryginalne wejście jest dodawane doklejanie do każdej warstwy.
- *Ensemble of ensembles.*

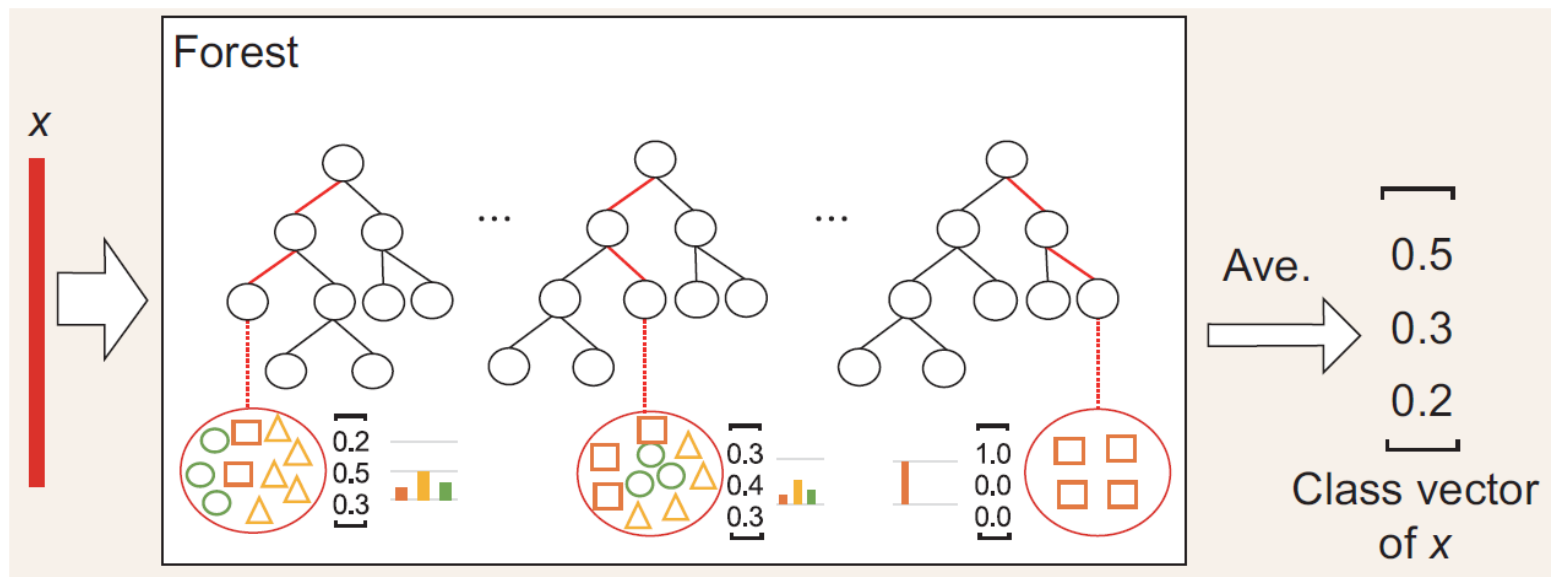
# Architektura (2)



- Dwa typy lasów losowych:
  - Klasyczny las losowy rozważający pierwiastek z liczby cech kandydatów w każdym węźle.
  - Ekstremalnie randomizowane drzewa (*Extra trees*).
- Każdy las składa się z 500 drzew.

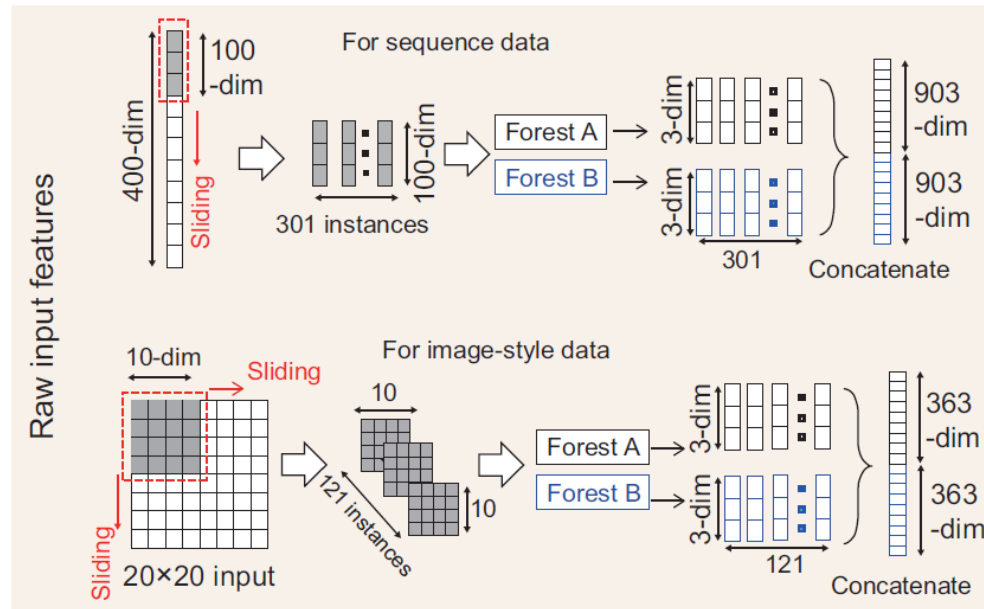


# Generowanie wektora reprezentacji



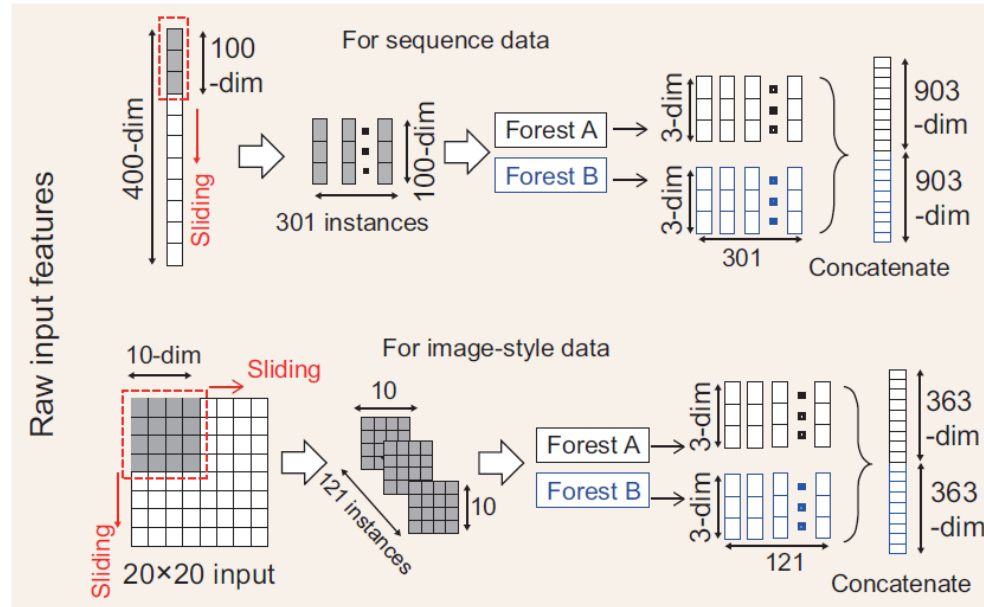
- Procedura przypomina standardową inferencję.
- Reprezentacja każdego wektora jest efektem uśrednienia wyników z  $k$  uruchomień danego lasu losowego.
- Po nauczaniu całego nowego poziomu, następuje ewaluacja na zbiorze walidacyjnym.
- Brak istotnej poprawy po dodaniu nowego poziomu powoduje zakończenie treningu.

# Skanowanie wieloziarniste (1)



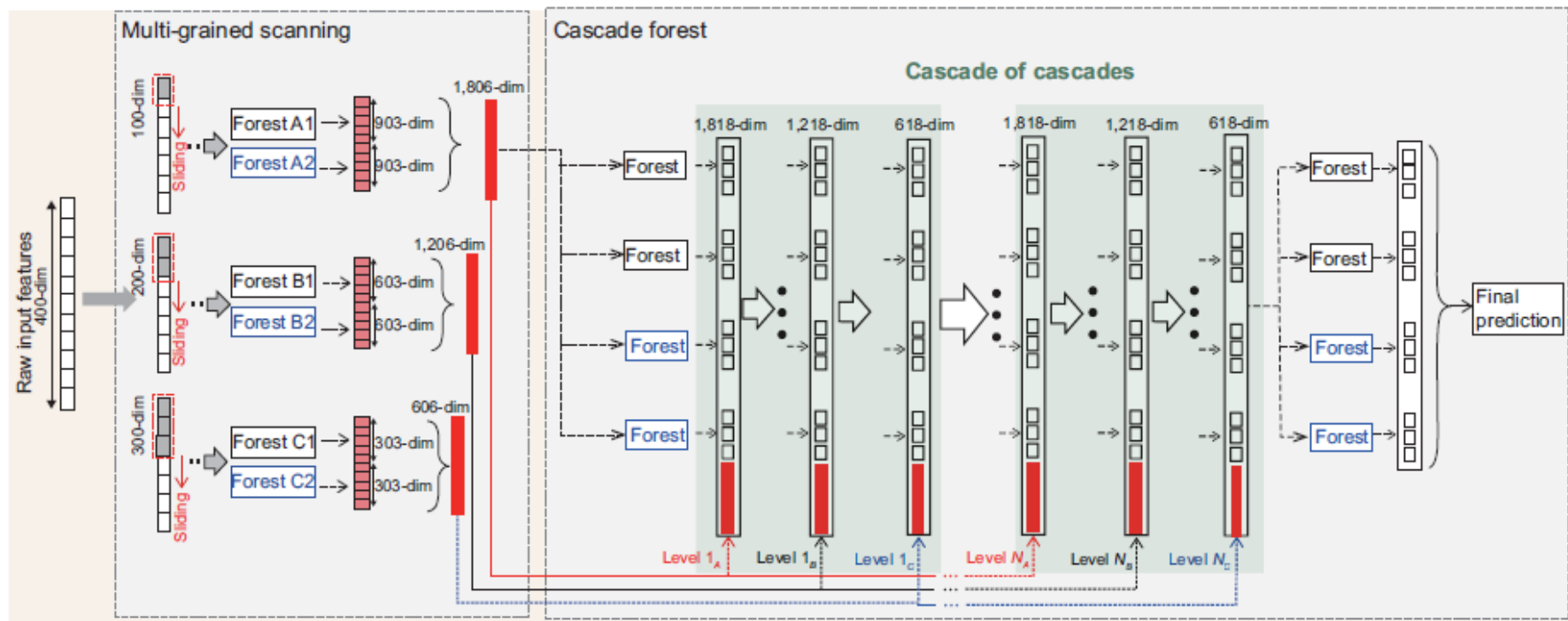
- Inspiracja umiejętnością sieci neuronowych do nauki relacji między cechami, np. wykrywanie wzorców sieciami konwolucyjnymi.
- Przesuwające się okno patrzy tylko na część wejścia.
- Wszystkie obserwacje etykietowane są oryginalnymi etykietami.

# Skanowanie wieloziarniste (2)



- Może skutkować błędami w etykietach.
  - Przykład: wycięcie kawałka zdjęcia, w którym nie ma kota.
- Technika celowego tworzenia błędnych etykiet nazywa się *flipping output* [2] i, paradoksalnie, może poprawić jakość zespołu modeli.

# Różne rozmiary okna



Główna architektura uwzględnia różne długości przesuwających się okien.

# Hiperparametryzacja: sieć neuronowa vs Deep forest

**Table 1.** Summary of hyper-parameters and default settings of gcForest. Bold font highlights hyper-parameters with relatively larger influence; '?' indicates default value unknown, or a general requirement for different settings for different tasks.

Deep neural networks (e.g. convolutional neural networks)	gcForest
Type of activation functions: Sigmoid, ReLU, tanh, linear, etc.	Type of forests: Completely random forest, random forest, etc.
Architecture configurations: <b>No. hidden layers:</b> ? <b>No. nodes in hidden layer:</b> ? <b>No. feature maps:</b> ? <b>Kernel size:</b> ?	Forest in multi-grained scanning: <b>No. forests:</b> {2} <b>No. trees in each forest:</b> {500} Tree growth: till pure leaf, or reach depth 100 <b>Sliding window size:</b> { $\lfloor d/16 \rfloor$ , $\lfloor d/8 \rfloor$ , $\lfloor d/4 \rfloor$ }
Optimization configurations: <b>Learning rate:</b> ? Dropout: {0.25/0.50} <b>Momentum:</b> ? <b>L1/L2 weight regularization penalty:</b> ? Weight initialization: Uniform, glorot-normal, glorot-uni, etc. Batch size: {32/64/128}	Forest in cascade: <b>No. forests:</b> {8} <b>No. trees in each forest:</b> {500} Tree growth: till pure leaf

Table 2: Comparison of test accuracy on MNIST

<b>gcForest</b>	<b>99.26%</b>
LeNet-5	99.05%
Deep Belief Net	98.75% [Hinton <i>et al.</i> , 2006]
SVM (rbf kernel)	98.60%
Random Forest	96.80%

Table 3: Comparison of test accuracy on ORL

	5 image	7 images	9 images
<b>gcForest</b>	<b>91.00%</b>	<b>96.67%</b>	<b>97.50%</b>
Random Forest	91.00%	93.33%	95.00%
CNN	86.50%	91.67%	95.00%
SVM (rbf kernel)	80.50%	82.50%	85.00%
<i>k</i> NN	76.00%	83.33%	92.50%

## ORL

- 400 obrazów 40 osób w skali szarości
- Rozdzielczość 92 x 112
- 5, 7 lub 9 obrazów per osoba użytych do treningu.

Table 4: Comparison of test accuracy on GTZAN

<b>gcForest</b>	<b>65.67%</b>
CNN	59.20%
MLP	58.00%
Random Forest	50.33%
Logistic Regression	50.00%
SVM (rbf kernel)	18.33%

Table 5: Comparison of test accuracy on sEMG data

<b>gcForest</b>	<b>71.30%</b>
LSTM	45.37%
MLP	38.52%
Random Forest	29.62%
SVM (rbf kernel)	29.62%
Logistic Regression	23.33%

- GTZAN
  - 10 rodzajów muzyki.
  - 100 30-sekundowych ścieżek per każdy rodzaj.
- sEMG
  - 1800 nagrań.
  - 3000 cech per nagranie.
  - Klasyfikacja 6 możliwych ruchów ręki.

Table 6: Comparison of test accuracy on IMDB

<b>gcForest</b>	<b>89.16%</b>
CNN	89.02% [Kim, 2014]
MLP	88.04%
Logistic Regression	88.62%
SVM (linear kernel)	87.56%
Random Forest	85.32%

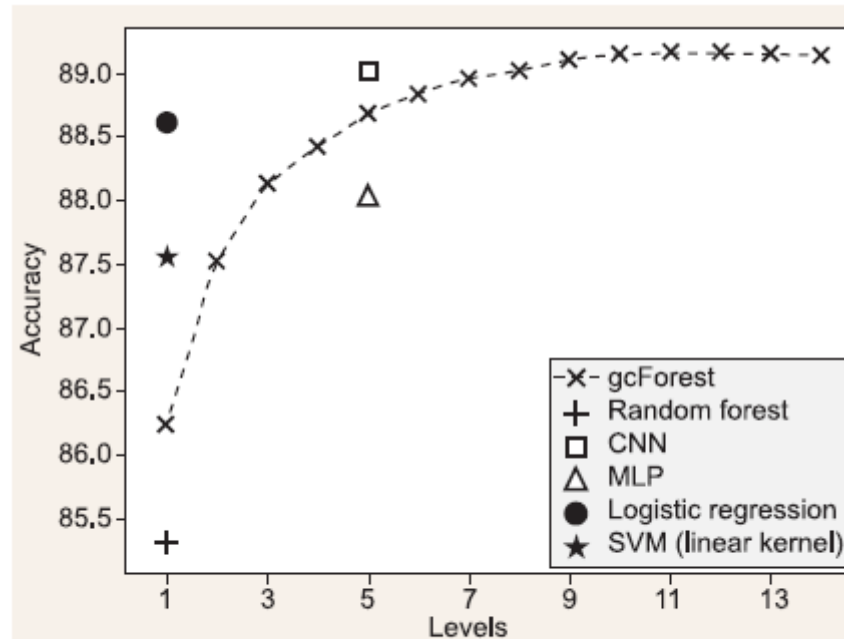
Table 7: Comparison of test accuracy on low-dim data

	LETTER	ADULT	YEAST
<b>gcForest</b>	<b>97.40%</b>	<b>86.40%</b>	<b>63.45%</b>
Random Forest	96.50%	85.49%	61.66%
MLP	95.70%	85.25%	55.60%

- IMDB
  - 50 tys. recenzji filmów zakodowanych przy pomocy TF-IDF.
- Zbiory danych UCI
  - LETTER: 16 cech, 20k obserwacji.
  - ADULT: 14 cech, 49k obserwacji.
  - YEAST: 8 cech, 1484 obserwacje.

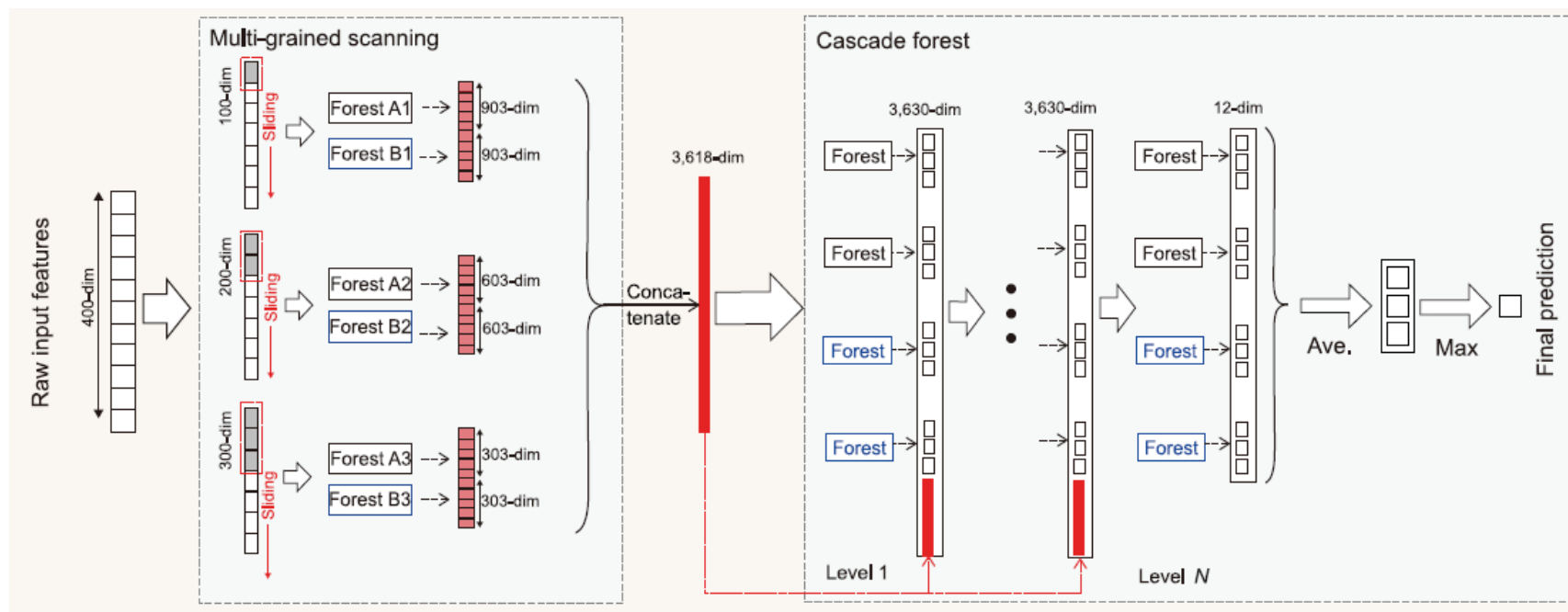


# Wyniki: wpływ liczby poziomów



**Figure 6.** Performance tendency on IMDB.

# Konkatenacja wszystkich długości okien



	MNIST	ORL	GTZAN	sEMG
gcForest	99.26%	97.50%	65.67%	71.30%
variant	98.96%	98.30%	65.67%	55.93%
	IMDB	LETTER	ADULT	YEAST
gcForest	89.16%	97.40%	86.40%	63.45%
variant	89.32%	97.25%	86.17%	63.23%

- Pierwsze (?) podejście bazujące na idei głębokiego uczenia bez różniczkowalnych modułów.
  - Tym samym bez użycia gradientowych metod optymalizacji.
- „Głębokie uczenie” bez wstecznej propagacji błędu.
- Istotnym kierunkiem dalszych badań jest rozszerzenie wektora reprezentacji.
  - W najprostszym przypadku można uwzględniać wektory z poprzednich warstw, wektory rodzeństwa, itp.
  - Jest potencjał, np. bazujący na autoenkoderze przy pomocy lasu losowego [3].

Q & A

1. Ghosh, D., & Cabrera, J. (2021). Enriched random forest for high dimensional genomic data. *IEEE/ACM transactions on computational biology and bioinformatics*, 19(5), 2817-2828.
2. Breiman, L. (2000). Randomizing outputs to increase prediction accuracy. *Machine Learning*, 40, 229-242.
3. Feng, J., & Zhou, Z. H. (2018, April). Autoencoder by forest. In *Proceedings of the AAAI conference on artificial intelligence* (Vol. 32, No. 1).