Uniwersytet Warszawski
Wydział Fizyki

Maciej E. Marchwiany

Opis własności nanostruktur niskowymiarowych za pomocą algorytmów uczenia maszynowego

Rozprawa doktorska
napisana pod kierunkiem
prof. dr hab.
Jacka A. Majewskiego
oraz promotora pomocniczego
dr Magdaleny Popielskiej
(Birowskiej)

Spis treści

Lista rysunków

Lista tablic

Lista symboli

W pracy stosowano oznaczenia:
$\hat{H}$ - hamiltonian układu,
$\hat{H}_{KS}$ - hamiltonian dla układu Kohna-Shama,
E - energia układu,
F - funkcjonał energii,
E_n - energia n-tego stanu własnego,
E_{xc} - energia wymianno-korelacyjna,
E_{xc}^{LDA} - energia wymianno-korelacyjna w przybliżeniu LDA,
E_{xc}^{GGA} - energia wymianno-korelacyjna w przybliżeniu GGA,
E_{xc}^{PBE0} - energia wymianno-korelacyjna w przybliżeniu PBE0,
E_{xc}^{B3LYP} - energia wymianno-korelacyjna w przybliżeniu B3LYP,
E_{xc}^{HSE} - energia wymianno-korelacyjna w przybliżeniu HSE,
E_X^{HF} - energia wymiany w przybliżeniu Hartree-Hocka,
E_X^{LDA} - energia wymiany w przybliżeniu LDA,
E_X^{GGA} - energia wymiany w przybliżeniu GGA,
E_C^{LDA} - energia korelacji w przybliżeniu LDA,
E_C^{GGA} - energia korelacji w przybliżeniu GGA,
E_x^{EXX} - energię wymiany obliczana ściel (ang. Exact Exchange),
$E_X^{SR,HF}$ - energia wymiany oddziaływania krótkozasięgowego w przybliżeniu Hartree-Focka,
$E_X^{SR,PBE}$ - energia wymiany oddziaływania krótkozasięgowego w przybliżeniu PBE,
$E_X^{LR,PBE}$ - energia wymiany oddziaływania długozasięgowego w przybliżeniu PBE,
E_C^{PBE} - energia wymiany w przybliżeniu PBE,
E_{I-I} - energia - oddziaływania jąder atomowych,
$\hat{T}$ - operator energii kinetycznej,
T_S - energię kinetyczną układu nieoddziałujących elektronów,
U - potencjał oddziaływania elektron - elektron,
U_H - energia oddziaływania kulombowskiego rozkładu ładunków,
V - potencjał jąder atomowych oraz zewnętrzny potencjał,
V_H - potencjał Hartree,
V_ext - potencjał zewnętrzny,
$V_e^{PP}xt$ - pseudopotencjał układu elektronów,
V_α^{PP} - pseudopotencjał jądra atomowego,
V_{XC} - potencjał wymienno - korelacyjny,
V_{KS} - potencjał w układzie Kohna-Shama,
$\hat{p}_i$ - operator pędu i-tego elektronu,
m - masa elektronu,

Ψ - wieloelektronowa unormowana funkcją falową układu,
Ψ_n - wieloelektronowa unormowana funkcją falową odpowiadająca n-temu stanowi własnemu,
Φ_s - funkcja falowa układu N nieoddziałujących elektronów,
ϕ - jednoelektronowa funkcja falowa,
$\vec{R_i}$ - położenie i-tego atomu,
$\vec{r_i}$ - położenie i-tego elektronu,
e - ładunek elektronu,
ρ - gęstość elektronowa,
Z_α - liczba atomowa atomu α,
L_x, L_y, Lz - stałe sieci,
$\sigma_{\alpha,\beta}$ - tensor naprężeń komórki elementarnej,
y_i - zmienna zależna,
x_i - zmienna niezależna,
ϵ - szum losowy,
$ODDs$ - szansa,
$E(y)$ - wartość oczekiwana zmiennej losowej y,
$Var(y)$ - wariancja zmiennej losowej,
σ_y - odchylenie standardowe zmiennej losowej y,
$cor(x, y)$ - korelacja zmiennych losowych x i y,
M - macierz korelacji,
$r(x, y)$ - współczynnik korelacji r-Pearsona pomiędzy zmiennymi x i y,
λ^k - moment rzędu k zmiennej losowej x,
μ^k - moment centralny rzędu k zmiennej losowej x,
$H(x)$ - entropia informacji zmiennej x,
MSE - błąd średniokwadratowy (ang. mean square error),
MAE - błąd modułów (ang. mean absolute error),
$R2$ - współczynnik determinacji (ang. R2 score),
EV - miara wyjaśnienia wariancji (ang. explained variance),
$logit$ - funcja logitowa,
β_i - współczynniki dopasowania i-tej zmiennej,
D_k - Metryka Cook'a,
$R(\beta_1, \beta_2, \ldots, \beta_M)$ - czynnik regularyzacyjny,

Streszczenie

W pracy zastosowano algorytmy uczenia maszynowego i sztucznej inteligencji do opisu i przewidywania własności nanostruktur niskowymiarowych. Opisano proces budowania modeli bazujących na algorytmach uczenia maszynowego w odniesieniu do układów jednowymiarowych: nanorurek borowych i nanorurek azotku boru oraz materiałów dwuwymiarowych MXenes. Układy te są w centrum zainteresowania badaczy ze względu na uch niezwykłe własności oraz potencjalne zastosowania w medycynie oraz farmacji.

Zbudowano modele modele pozwalające na realistyczne przewidywanie całkowitej energii, energii kohezji oraz toksyczności badanych związków. Badania wymagały stworzenia szeregu modeli w oparciu o algorytmy uczenia maszynowego. Wykorzystano najbardziej zaawansowane algorytmy oraz zaadaptowano je na potrzeby problemów przedstawionych w dysertacji. Ponadto przetestowano wszystkie dostępne metody reprezentacji geometrii układów krystalicznych.

Podczas prac wykorzystano unikalne połączenie danych eksperymentalnych z danymi teoretycznymi pochodzącymi z symulacji numerycznych. Dane teoretyczne pochodziły z otwartych baz danych zawierających wyniki symulacji oraz z ponad 8500 przeprowadzonych podczas pisania rozprawy symulacji numerycznych. Otrzymane wyniki opierają się na analizie algorytmami sztucznej inteligencji unikalnych zbiorów danych. Ponadto zastosowane techniki umożliwiły wyjaśnienie stworzonych modeli oraz sposobu opisywania przez nie procesów fizycznych. Pozwoliło to na poszerzenie wiedzy o badanych układach. Oryginalne i nowatorskie zastosowanie sztucznej inteligencji w badaniach naukowych. Szczególnie ważne wydają się wyniki opisujące mechanizm toksyczności materiałów MXenes.

Abstract

Machine learning and artificial intelligence algorithms have been used to describe and predict the properties of low-dimensional nanostructures. The process of building predictive models based on machine learning algorithms has been employed to: (i) one-dimensional systems: boron nanotubes and boron nitride nanotubes and, two (ii)-dimensional MXenes materials. These systems are presently of interest to researchers due to their unique properties and potential application in medicine and pharmacy.

Predictive models for total system energy, cohesion energy and toxicity have been built. Conducting the research required numerous models based on machine learning algorithmshave been employed to the problems presented in the dissertation. Moreover, all available methods of representing the geometry of the crystal systems ave been tested.

During the work, a unique combination of data from the experiment and theoretical data from numerical simulations have been used. Theoretical data come from open databases containing simulation results and from over 8,500 numerical simulations carried out during the dissertation preparation. The obtained results are based on the analysis of the unique data sets using artificial intelligence algorithms. In addition, the developed methodology allows for explaination of the physical processes and broadens the knowledge about the considered systems. The results describing the toxicity mechanism of MXenes materials seem to be particularly novel and important.

Rozdział 1

Wstęp

Ostatnie lata przyniosły gwałtowny rozwój narzędzi opartych na analizie statystycznej oraz uczeniu maszynowym, służących do analizy dużych zbiorów danych. Ten rozwój powodowany jest głównie zwiększeniem ilości gromadzonych danych oraz wzrostem dostępnej mocy obliczeniowej. Ogromne znaczenie ma również szybko postępujący rozwój algorytmiki, który pozwolił na znaczącą optymalizację istniejących rozwiązań. W szczególności, wsteczna propagacja błędu [1] zaimplementowana na kartach graficznych (GP GPU) przyczyniła się do rozwoju głębokich sieci neuronowych. Osiągnięty postęp na tyle przyczynił się do wzrostu istotności wnioskowania wspomaganego komputerowo, że niebawem może stanowić kolejny etap w ewolucji badań naukowych. Za odkryciami empirycznymi, teoretycznymi czy numerycznymi, ostatnio metody sztucznej inteligencji stają się kolejną paradygmatem nauki [2]. Ten trend jest także coraz silniej widoczny w fizyce materii skondensowanej. Nauki matematyczno-przyrodnicze nie są jednak naturalnym obszarem stosowania algorytmów sztucznej inteligencji. W pracy pokazano, że narzędzia uczenia maszynowego pozwalają na przewidywanie istotnych wielkości fizycznych, fizyko-chemicznych, czy biologicznych, oraz na stawianie i weryfikację hipotez niedostępnych dla tradycyjnych metod analizy danych. Przedstawiono także specyficzne techniki pozwalające na stosowanie algorytmów uczenia maszynowego dla małych zbiorów danych.

W fizyce materii skondensowanej najczęściej stosuje się algorytmy uczenia maszynowego do: przewidywania energii układu [3, 4], obliczania potencjału atomowego [5–7], oraz wyznaczania przerwy energetycznej [8–11]. Jednak te najpopularniejsze zastosowania nie wyczerpują potencjału nowych metod. Sztuczna inteligencja w badaniach naukowych ma dwa zasadnicze zastosowania:

— przewidywanie wyniku procesu,
— analiza przebiegu procesu.

Przez przewidywanie wyniku procesu rozumiemy możliwość przewidywania wyniku eksperymentu lub symulacji numerycznych bez konieczności ich wykonywania. Przewidywanie wyniku procesu jest bardzo szeroko stosowanie i może być zaadoptowane do opisu praktycznie każdego zjawiska fizycznego. Algorytmy uczenia maszynowego umożliwiają bardzo dokładne przewidywanie wyniku. Jednak ze względu na swoją złożoność lub budowę nie pozwalają na głębsze zrozumienie istoty badanego procesu lub choćby interpretację składowych budowanego modelu.

Analiza przebiegu procesu pozwala na zrozumienie istoty badanego zjawiska. Opiera się ona na stosowaniu prostszych i łatwych do interpretacji

algorytmów lub wyznaczeniu istotności zmiennych niezależnych modelu oraz relacji pomiędzy nimi. Zrozumienie zjawiska jest pogłębiane o opis jego przebiegu w funkcji rozważanych zmiennych oraz opis wpływu każdej zmiennej jego przebiegu.

W pracy zaprezentowano możliwości stosowania narzędzi opartych na algorytmach uczenia maszynowego w analizie danych eksperymentalnych oraz wyników symulacji numerycznych nawet dla stosunkowo niedużej liczby obserwacji. Oczywiście nie można mówić wówczas, o budowaniu dokładnego modelu, ale możliwe jest wnioskowanie i dogłębna analiza posiadanych danych. Opisane metody pozwalają na pogłębione zrozumienie mechanizmów badanego procesu w oparciu o wyniki analizy istotności zmiennych oraz jakości zbudowanych modeli.

Przedstawione wyniki bazują na danych pochodzących z wielu źródeł: prac eksperymentalnych zaczerpniętych z literatury, wyników symulacji komputerowych pozyskanych z zewnętrznych baz danych, oraz samodzielnie przeprowadzonych symulacji numerycznych. Pozwala to na zaprezentowanie możliwości algorytmów uczenia maszynowego dla wszystkich obecnie dostępnych źródeł danych. Tak szerokie spektrum stosowanych danych jest unikalne i nie spotykane w literaturze. Ponadto, przedstawiono wyniki opracowanych modeli sztucznej inteligencji oraz wnioski jakie można wysnuć z analizy istotności dostępnych zmiennych.

W pracy zaprezentowano proces budowania modelu opartego na algorytmach uczenia maszynowego oraz analizę jakości szeregu modeli i sposobów reprezentacji układów fizycznych. Wybrano i przetestowano algorytmy dedykowane do wielkości i rodzaju posiadanych danych oraz najpopularniejsze reprezentacje układów fizycznych. Przedstawiono także szereg technik pozwalających na poprawę jakości modelu dla małych zbiorów danych. W tej pracy wykorzystano algorytmy uczenia maszynowego do przewidywania energii całkowitej układu, energii kohezji oraz cytotoksyczności związków. Badania zostały wykonane na trzech klasach związków: nanorurkach boru, nanorurkach azotku-boru oraz materiałów dwuwymiarowych MXenes.

Praca jest podzielona na pięć rozdziałów. Rozdział pierwszy opisuje podstawy teoretyczne metod zastosowanych do wykonanych symulacji numerycznych w reżimie teorii funkcjonału gęstości. Rozdział drugi zawiera podstawy metod uczenia maszynowego: podstawowe pojęcia statystyki, terminologię i metodologię uczenia maszynowego oraz opis wykorzystywanych algorytmów. Rozdział trzeci opisuje techniki charakterystyczne dla algorytmów uczenia maszynowego stosowane w fizyce materii skondensowanej, takie jak reprezentacja geometrii układu, techniki pracy z małymi zbiorami danych czy niezbalansowanymi zbiorami danych. Kolejny rozdział zawiera opis otrzymanych wyników oraz dyskusję poprawności opracowanych modeli przy użyciu wielu algorytmów uczenia maszynowego oraz reprezentacji układów fizycznych.

Częściowe wyniki dotyczące przewidywania cytotoksyczności nanomateriałów dwuwymiarowych MXenes zostały opublikowane w [12].

Rozdział 2

Podstawy DFT

2.1. Teoria funkcjonału gęstości

Przewidywania własności układów oddziaływujących cząstek, w szczególności elektronów, jest jednym z największych wyzwań fizyki teoretycznej i chemii kwantowej [13, 14], o znaczeniu praktycznym, którego nie sposób przecenić.

Teoria funkcjonału gęstości [15–17] (ang. Density Functional Theory - DFT) jest ogólną metodą pozwalającą na znalezienie stanu podstawowego układu oddziałujących cząstek w zewnętrznym potencjale i znajduje zastosowanie zarówno w fizyce jądrowej jak materii skondensowanej. W układach materii skondensowanej, takich jak molekuły, kryształy, nanocząstki, głównym problemem jest znalezienie stanu podstawowego N elektronów tworzących struktury, co w zasadzie jest jednym z głównych problemów mechaniki kwantowej. W mechanice kwantowej stany kwantowe układu N elektronów o położeniach $(\vec{r}_1, \vec{r}_2, \ldots, \vec{x}_N)$ zadane są przez równanie Schrödingera:

$$\hat{H}\Psi_n(\vec{r}_1, \ldots, \vec{r}_N) = E_n\Psi_n(\vec{r}_1, \ldots, \vec{x}_N), \tag{2.1}$$

gdzie $\hat{H}$ jest hamiltonianem układu, $\Psi_n(\vec{r}_1, \vec{r}_2, \ldots, \vec{x}_N)$ wieloelektronową funkcją falową układu N elektronów, a E_n energią stanów własnym układu. Stan o najniższej energii E_0 nazywamy stanem podstawowym układu.

Hamiltonian $\hat{H}$ układu N oddziałujących elektronów w potencjale zewnętrznym $\hat{V}$ składa się z operator energii kinetycznej elektronów $\hat{T}$, oraz operator oddziaływania pomiędzy elektronami $\hat{U}$:

$$\hat{H} = \hat{T} + \hat{U} + \hat{V}, \tag{2.2}$$

W układach materii skondensowanej potencjał zewnętrzny dla N elektronów określony jest przez kulombowskie oddziaływanie elektronów z współtworzącymi układ M jądrami atomowymi. Atomy posiadają liczby atomowe $(Z_1, \ldots, Z_\alpha, \ldots, Z_M)$ oraz znajdują się odpowiednio w położeniach $(\vec{R}_1, \ldots, \vec{R}_\alpha, \ldots, \vec{R}_M) = \{\vec{R}\}$:

$$V(\vec{r}_1, \vec{r}_2, \ldots, \vec{x}_N; \{\vec{R}\}) = \sum_i^N \sum_\alpha^M \frac{-e^2 Z_\alpha}{|\vec{r}_i - \vec{R}_\alpha|}. \tag{2.3}$$

Oczywiście na układ mogą zostać nałożone całkowicie zewnętrzne (również dla jąder) pola elektryczne czy magnetyczne, które wnoszą odpowiedni wkład do potencjału zewnętrznego $\hat{V}$, ale teraz w tym krótkim wprowadzeniu teorii funkcjonału gęstości zostaną pominięte. Operator energii kinetycznej $\hat{T}$ jest sumą operatorów energii kinetycznej poszczególnych elektronów:

$$\hat{T} = \sum_i \frac{\hat{p}_i^2}{2m}, \tag{2.4}$$

gdzie $\hat{p}_i$ jest operatorem pędu i-tego elektronu o masie m. Operator oddziaływania pomiędzy elektronami U jest sumą oddziaływania kulombowskiego par elektronów:

$$\hat{U} = \sum_{i<j} \frac{e^2}{|\vec{r}_i - \vec{r}_j|}. \tag{2.5}$$

Powyższe równanie Schrödingera nie ma ścisłego rozwiązania nawet dla dwóch oddziałujących elektronów, więc cały rozwój fizyki materii skondensowanej, chemii kwantowej i nauki o materiałach nie mógłby być możliwy bez metod przybliżonych pozwalających na otrzymanie kontrolowanie dokładnych przybliżonych rozwiązań. Na tym polu teoria funkcjonału gęstości (DFT) odegrała niezwykle istotną rolę.

2.2. Twierdzenie Hohenberga-Kohna

Podstawową wielkością w teorii DFT jest jednoelektronowa gęstość, która może być otrzymana z wieloelektronowej funkcji falowej poprzez scałkowanie po N-1 położeniach elektronów :

$$\rho(\vec{r}) = N \int \dots \int |\Psi(\vec{r}, \vec{r}_2, \dots, \vec{r}_N)|^2 d^3r_2 \dots d^3r_N, \tag{2.6}$$

gdzie dla uproszczenia pominęliśmy zmienne spinowe, a gęstość jest unormowana do N elektronów, $\int \rho(\vec{r}) d^3r = N$.

W roku 1964, P. Hohenberg i W. Kohn [15] pokazali, że znajomość jednocząstkowej gęstości elektronowej $\rho(\vec{r})$ pozwala jednoznacznie wyznaczyć energię niezdegenerowanego stanu podstawowego układu N elektronów. Energia układu N elektronów o danej gęstości jednoelektronowej $\rho(\vec{r})$ jest funkcjonałem tej gęstości, t.j. $E = E[\rho]$. Funkcjonał energii $E[\rho]$ można zapisać wydzielając trywialną część energii układu elektronów w potencjale zewnętrznym w następującej formie:

$$E[\rho] = \int d\vec{r} \rho(\vec{r}) V_{ext}(\vec{r}) + F[\rho], \tag{2.7}$$

gdzie $F[\rho]$ jest uniwersalnym funkcjonałem (nie zależnym od V_{ext}) zawierającym wkład do energii całkowitej układu od energii kinetycznej i energii kulombowskiego oddziaływania elektronów. Najlepszym sposobem wprowadzenia funkcjonału $F[\rho]$ jest użycie uniwersalnej definicji Mel Levy [18, 19]:

$$F[\rho] = min_{\Psi \to \rho} < \Psi | \hat{T} + \hat{U} | \Psi >, \tag{2.8}$$

gdzie minimalizacja przeprowadzona jest po wszystkich funkcjach wieloelektronowych całkujących się do gęstości jednoelektronowej ρ. Dokonując minimalizacji tylko $\hat{T}$ i $\hat{U}$ można w analogiczny sposób zdefiniować funkcjonał energii kinetycznej oddziałujących elektronów $T[\rho]$ i funkcjonał kulombowskiego oddziaływania $U[\rho]$, co pozwala zapisać funkcjonał $F[\rho]$ jako sumę dwóch wkładów: $F[\rho] = T[\rho] + U[\rho]$. Jeżeli przez E_0 oznaczymy energię stanu podstawowego układu elektronów, a przez ρ_0 odpowiadającą jej gęstość jednocząstkową, wówczas twierdzenie Hohenberga-Kohna mówi, że:

$$\begin{aligned} &(i) E[\rho] \geqslant E[\rho_0], \\ &(ii) E[\rho] = E_0 \iff \rho = \rho_0. \end{aligned} \tag{2.9}$$

Dla stanu podstawowego układu N elektronów $\hat{H}\Psi_0 = E_0\Psi_0$. Jeżeli gęstość ρ_0 jest otrzymana z funkcji wieloelektronowej Ψ_0, to jak łatwo widać zachodzi relacja $E_0 =< \Psi_0|\hat{H}|\Psi_0 >= E[\rho_0]$, stanowiąca koncepcyjną podstawę teorii funkcjonału gęstości. Zamiast szukać wieloelektronowej funkcji dla układu N elektronów, która minimalizuje wartość oczekiwaną hamiltonianu (czyli energię) możemy szukać funkcji $\rho(\vec{r})$ trzech zmiennych rzeczywistych $\vec{r}$, która minimalizuje funkcjonał energii $E[\rho]$. Twierdzenie Hohenberga-Kohna ma charakter twierdzenia o istnieniu. Wskazuje na istnienie funkcjonału gęstości jednocząstkowej, który zminimalizowany ze względu na gęstość ρ pozwoli na otrzymanie energii stanu podstawowego układu N oddziałujących elektronów. Praca Hohenberga i Kohna nie podaje przepisu tej minimalizacji, a próby jej przeprowadzenia okazały się niezwykle trudne i nieskuteczne. Praktyczne rozwiązanie problemu zostało przedstawione w roku 1965 przez W. Kohna i L. Shama w pracy [20, 21]. Metoda Kohna-Shama stanowi obecnie podstawę realizacji obliczeń w ramach DFT i została zaimplementowana w tuzinach komercyjnych i ogólnie dostępnych pakietów obliczeniowych.

2.3. Metoda Kohna-Shama

Poniżej przedstawiono główne założenia metody Kohna-Shama, której głównym punktem jest wprowadzenie układu elektronów nieoddziałujących o gęstości ρ_{KS}, które poruszają się w efektywnym potencjale zewnętrznym V_{KS}. Kohn i Sham podali praktyczny przepis jak wyznaczyć V_{KS} w taki sposób, żeby $\rho_0 = \rho_{KS}$, czyli gęstości stanu podstawowego układów oddziałującego i nieoddziałującego były równe. Implikuje to wyznaczenie w dalszych krokach również energii stanu podstawowego E_0 dla oddziałujących elektronów. Nie jest ona jednak równa energii całkowitej układu elektronów nieoddziałujących. Hamiltonian dla układu Kohna-Shama N nieoddziałujących elektronów można zapisać w następującej postaci:

$$\hat{H}_{KS} = \sum_i^N \frac{\hat{p}_i^2}{2m} + \sum_i^N V_{KS}(\vec{r}_i). \tag{2.10}$$

Funkcja falowa Φ_s dla układu N nieoddziałujących elektronów jest wyznacznikiem Slatera stanów jednoelektronowych ϕ :

$$\Phi_s(\vec{r}_1, \ldots, \vec{r_N}) = \frac{1}{\sqrt{N}} \begin{vmatrix} \phi_1(\vec{r}_1) & \phi_1(\vec{r}_2) & \ldots \\ \phi_2(\vec{r}_1) & \phi_2(\vec{r}_2) & \ldots \\ \vdots & \vdots & \ddots \end{vmatrix}. \tag{2.11}$$

Gęstość jednolektronową układu N nieoddziałujących elektronów (zgodnie z definicją podaną powyżej) można więc wyrazić przez sumę gęstości jednolektronowych:

$$\rho_S(\vec{r}) = \sum_i^N \phi_i^*(\vec{r})\phi_i(\vec{r}), \tag{2.12}$$

która zgodnie z założeniami Kohna-Shama musi być równa: $\rho_S(\vec{r}) = \rho(\vec{r})$.

Zgodnie ze schematem Mel Levy można zdefiniować również energię kinetyczną dla układu nieoddziałujących elektronów:

$$T_S[\rho] = min_{\Phi\to\rho} < \Phi|\hat{T}|\Phi > . \tag{2.13}$$

Warto podkreślić, że $T_S[\rho]$ różni się od energii kinetycznej układu oddziałujących elektronów $T[\rho]$ nawet dla identycznej gęstości jednoelektronowej, ponieważ minimalizacja w definicji tych wielkości przebiega po innych funkcjach wieloelektronowych, wyznaczniki Slatera Φ dla $T_S[\rho]$, a ogólne wieloelektronowe funkcje falowe Ψ dla $T[\rho]$. Funkcjonał energii $F[\rho]$ można teraz zapisać w postaci:

$$F[\rho] = T_S[\rho] + U_H[\rho] + (U[\rho] - U_H[\rho] + T[\rho] - T_S[\rho]), \tag{2.14}$$

gdzie $U_H[\rho]$ jest klasyczną energią oddziaływania kulombowskiego rozkładu ładunków elektrycznych o rozkładzie $\rho(\vec{r})$, a wielkość w nawiasie określa się mianem energii wymienno-korelacyjnej $E_{xc}[\rho]$:

$$U_H[\rho] = \frac{1}{2}\int\int \frac{\rho(\vec{r})\rho(\vec{r}')}{|\vec{r}-\vec{r}'|} d^3r d^3r', \tag{2.15}$$

$$E_{xc}[\rho] = U_H[\rho] - U[\rho] + T[\rho] - T_S[\rho]. \tag{2.16}$$

Funkcjonał $F[\rho]$ przyjmuje więc postać:

$$F[\rho] = T_S[\rho] + U[\rho] + E_{xc}[\rho], \tag{2.17}$$

gdzie energia wymienno-korelacyjna E_{xc} jest jedyną nieznaną *explicite* składową funkcjonału i wymaga wprowadzenia jak najlepszych przybliżeń, w celu przeprowadzenia wiarygodnych ilościowo obliczeń. Zagadnieniu wyboru odpowiedniego funkcjonału E_{xc} poświęcona jest ogromna literatura. W chwili obecnej istnieje ponad 200 różnych przybliżeń funkcjonału E_{xc} [22], jedne częściej stosowane w obliczeniach chemicznych, inne w obliczeniach fizycznych.

Ponieważ w realizacji Kohna-Shama teorii funkcjonału gęstości DFT gęstość $\rho(\vec{r})$ wyraża się przez funkcje jednoelektronowe ϕ_i, minimalizację energii całkowitej ze względu na ρ można zamienić na minimalizację energii ze względu na ϕ, co prowadzi do równań Kohna-Shama dla orbitali jednocząstkowych ϕ_i:

$$\left(-\frac{\hbar^2}{2m}\nabla^2 + V_{ext}(\vec{r}) + V_H(\vec{r}) + V_{xc}(\vec{r})\right)\phi_i(\vec{r}) = \epsilon_i\phi_i(\vec{r}), \tag{2.18}$$

gdzie $V_H(\vec{r})$) jest potencjałem kulombowskim, a $V_{xc}(\vec{r})$ potencjałem wymienno-korelacyjnym równym pochodnej funkcjonalnej funkcjonału wymienno- korelacyjnego ze względu na gęstość elektronową ρ:

$$V_H(\vec{r}) = \int \frac{\rho(\vec{r}')}{|\vec{r}-\vec{r}'|} d^3r', \tag{2.19}$$

$$V_{xc}(\vec{r}) = \frac{\partial E_{xc}[\rho]}{\partial\rho(\vec{r}))}. \tag{2.20}$$

Lokalny potencjał w równaniu Kohna-Shama określany jest mianem potencjału Kohna-Shama $V_{KS}(\vec{r})$:

$$V_{KS}(\vec{r}) = V_{ext}(\vec{r}) + V_H(\vec{r}) + V_{xc}(\vec{r}), \tag{2.21}$$

jest to efektywny potencjał, w którym poruszają się elektrony "nieoddziałujące" pomocniczego układu Kohna-Shama zdefiniowanym poprzez hamiltonianie $\hat{H}_{KS}$ (2.10). Ponieważ potencjał kulombowski i wymienno-korelacyjny (odpowiednio V_H i V_{xc}) zależą od gęstości ρ, a ta z kolei zadana jest poprzez orbitale jedoelektronowe ϕ_i, równania Kohna-Shama muszą być rozwiązane w sposób samouzgoniony.

Wprowadzenie realizacji Kohna-Shama teorii funkcjonału gęstości, przemieniło DFT z formalnej teorii dotyczącej energii stanu podstawowego układu N oddziałujących elektronów, w niezwykle efektywną metodę obliczeniową dla układów molekularnych, nanostruktur oraz układów periodycznych. Miało również fundamentalne znaczenia dla rozwoju chemii, fizyki materii skondensowanej, czy nauki o materiałach. Schemat obliczeniowy Kohna-Shama oparty jest na dwóch punktach: (i) znalezieniu dobrego przybliżenia dla funkcjonału E_{xc}, co definiuje dokładność przewidywań teoretycznych, oraz (ii) opracowaniu metodologii numerycznej dla rozwiązania równań Kohna-Shama, co określa precyzję obliczeń.

2.4. Funkcjonały wymienno-korelacyjne

Omówienie wszystkich używanych obecnie funkcjonałów wymienno- korelacyjnych E_{xc} nie mieści się w zakresie tej pracy doktorskiej, niemniej wyróżnione zostaną najważniejsze klasy przybliżeń. Historycznie pierwszym przybliżeniem prowadzącym do rozsądnych wyników było przybliżenie LDA (ang. Local Density Approximation). Oparte jest ono na założeniu, że niejednorodna gęstość $\rho(\vec{r})$ może być traktowana lokalnie (dla punktu $\vec{r}$) jako gęstość jednorodnego gazu elektronowego o gęstości n. Energia wymiany jednorodnego gazu elektronowego jest znana (proporcjonalna do $n^{4/3}$), a energia korelacji jednorodnego gazu elektronowego może zostać otrzymana z obliczeń Monte Carlo [23] i odpowiednio sparametryzowana jako analityczna funkcja n [24]. Łatwo można otrzymać energie wymiany $\varepsilon_x(n)$ i korelacji $\varepsilon_c(n)$ dla jednorodnego gazu elektronowego o gęstości n na cząstkę, i konsekwentnie energię wymienno-korelacyjną na cząstkę $\varepsilon_{xc}(n) = \varepsilon_x(n)+\varepsilon_c(n)$. Wtedy przybliżenie LDA dla funkcjonału gęstości otrzymuje się z ε_{xc} przez scałkowanie po objętości układu niejednorodnego:

$$E_{xc}^{LDA}[\rho] = \int \varepsilon(\rho(\vec{r}))\rho(\vec{r})d^3r. \tag{2.22}$$

Uogólnieniem funkcjonału LDA jest funkcjonał GGA (ang. Generalized Gradient Approximation):

$$E_{xc}^{GGA}[\rho] = \int f_{xc}(\rho(\vec{r}), \nabla\rho)\rho(\vec{r})d^3r, \tag{2.23}$$

gdzie funkcja podcałkowa f_{xc} zależy nie tylko od gęstości elektronowej w punkcie $\vec{r}$, ale również od gradientu gęstości. W literaturze istnieje wiele funkcjonałów GGA różniących się wyborem funkcji f_{xc}, jak na przykład (PW86) pochodzący od J. P. Perdue i W. Yue [25]. Funkcja f_{xc} ustalana jest na podstawie reguł sumacyjnych i numerycznego fitowania do rozwiązań otrzymanych metodami CI (ang. Configuration Interaction) [26] czy kwantowej metody Monte Carlo QMC (ang. Quantum Monte Carlo) [27, 28].

W następnej wersji funkcjonału Perdew-Wang (PW91) funkcja $f_x(\rho)$ opisująca część wymienną funkcjonału przybiera postać [29, 30]:

$$f_x(\rho) = \frac{3k_F}{4\pi}\frac{1 + 0.1965s\ sinh^{-1}(7.796s) + (0.274 - 0.151e^{-100s^2})s^2}{1 + 0.1964s\ sinh^{-1}(7.796s) + 0.004s^4}, \tag{2.24}$$

gdzie $k_F = (3\pi^2\rho)^{1/3}$, a $s = \frac{|\nabla\rho|}{2k_F\rho}$.

Funkcjonał wymienno-korelacyjny z uproszczoną funkcją f_x przez J. P. Perdew, K. Burke i M. Ernzerhof (PBE), obecnie bardzo szeroko stosowany zarówno w obliczeniach molekularnych jak i dla układów periodycznych, jest niewątpliwie najpopularniejszym funkcjonałem wymienno-korelacyjnym typu GGA. Ostatnio używane są też funkcjonały meta-GGA [31, 32], w których w funkcji podcałkowej f_{xc} wprowadza się dodatkowo zależność od gęstości energii kinetycznej $\tau(\vec{r})$.

W działaniach zmierzających do otrzymania jak najdokładniejszego przybliżenia dla E_{xc}, należy wymienić dwa podejścia, które wykorzystują cechy metody Kohna-Shama pozwalające na uściślenie wyrażenia dla energii wymiany. Ponieważ w metodzie Kohna-Shama pomocniczy układ jest opisany przez wyznacznik Slatera funkcji jednoelektronowych (rozwiązań równań Kohna-Shama), można podać ścisłe wyrażenie na energię wymiany E_x^{EXX} (ang. Exact Exchange) [33, 34]:

$$E_x^{EXX}[\rho] = -\frac{1}{2}\sum_i \int\int d^r d^3r' \phi_i^*(\vec{r}) \left(\sum_j \frac{\phi_j(\vec{r})\phi_j^*(\vec{r'})}{|\vec{r} - \vec{r'}|} \right) \phi_i(\vec{r'}). \tag{2.25}$$

$E_x^{EXX}[\rho]$ jest funkcjonałem gęstości poprzez $\phi[\rho]$, ale *explicite* zależność $E_x^{EXX}[\rho]$ od ρ nie jest znana. Warto zauważyć również, że chociaż formalnie wyrażenie dla E_x^{EXX} jest identyczne z wyrażeniem na energię wymiany w metodzie Hartree-Focka E_x^{H-F}, to wartości liczbowe energii wymiany E_x^{EXX} i E_x^{H-F} są różne ze względu na to, że w każdym z tych wyrażeń występuję inne orbitale jednoelektronowe; w E_x^{EXX} rozwiązania równań Kohna-Shama ϕ_i, a E_x^{H-F} rozwiązania Hartree-Focka. W ten sposób w funkcjonale wymiany i korelacji jedyną składową wymagającą wprowadzenia przybliżeń jest funkcjonał korelacji $E_c[\rho]$, a funkcjonał ze ścisłą wymianą możemy oznaczyć następująco; $E_{xc}^{EXX}[\rho] = E_x^{EXX}[\rho] + E_c[\rho]$. W równaniach Kohna-Shama otrzymanych z $E_{xc}^{EXX}[\rho]$ potencjał Kohna-Shama $V_{K-S}(\vec{r})$ pozostaje lokalnym potencjałem zmiennej przestrzennej $\vec{r}$. Używając funkcjonału $E_{xc}^{EXX}[\rho]$ przeprowadzono szereg obliczeń dla układów molekularnych i periodycznych. Prowadzi to do wyników obliczeń w lepszej zgodności z danymi doświadczalnymi, jednak nakład numeryczny jest typowo 60-100 razy większy niż z użyciem funkcjonałów typu GGA. Niemniej dla opracowywania przybliżeń dla funkcjonału korelacji możliwość uwzględnienia ścisłej wymiany ma olbrzymie znaczenie.

Drugą (oprócz EXX) metodą, gdzie oddziaływanie wymienne jest uwzględnione w sposób podobny do metody Hartree-Focka jest tak zwana uogólniona metoda Kohna-Shama GKS (ang. Generalized Kohn-Sham) [34].

W metodzie GKS w funkcjonale $F[\rho]$ wydziela się nie tylko funkcjonał energii kinetycznej nieoddziałujących elektronów (jak w metodzie Kohna-Shama), ale także funkcjonał $F_\alpha^S[\rho]$ zawierający część operatora $\hat{U}$, zdefiniowany następująco:

$$F_\alpha^S[\rho] = min_{\Phi\to\rho} < \Phi|\hat{T} + \alpha\hat{U}|\Phi >, \tag{2.26}$$

gdzie minimalizacja jest przeprowadzona po wszystkich wyznacznikach Slatera całkujących się do danej gęstości jednoelektronowej $\rho(\vec{r})$, a $0 \leqslant \alpha \leqslant 1$. Funkcjonał Hohenberga-Kohna $F[\rho]$ przyjmuje postać:

$$F[\rho] = F_\alpha^S[\rho] + F^R[\rho], \tag{2.27}$$

gdzie funkcjonał $F^R[\rho]$ zawiera wszystkie nieznane wkłady do $F[\rho]$ i wymagające odpowiednich przybliżeń. Oczywiście dla $\alpha = 0$ metody Kohna-Shama i GKS są równoważne. Uogólnione równania Kohna-Shama na orbitale ϕ_i w metodzie GKS zawierają nielokalny operator wymiany, podobnie do metody Hartree-Focka i w odróżnieniu do metody EXX. Oczywiście tak ogólne sformułowanie otwiera drogę do realizacji licznych przybliżeń funkcjonału gęstości. Funkcjonały konstruowane w oparciu o GKS noszą nazwę potencjałów hybrydowych, ponieważ poprzez odpowiedni wybór stałej α mogą praktycznie w dowolnym stosunku mieszać nielokalny wkład do energii wymiany z lokalnym przybliżeniem wziętym np. z funkcjonałów LDA lub GGA [35].

Prostym funkcjonałem hybrydowym jest funkcjonał (PBE0) [36], gdzie część wymienno-korelacyjna funkcjonału jest opisana formułą:

$$E_{xc}^{PBE0} = \frac{3}{4}E_x^{HF} + \frac{1}{4}E_x^{PBE} + E_c^{PBE}. \tag{2.28}$$

Inne funkcjonały hybrydowe są skonstruowane na podstawie bardziej skomplikowanych wyrażeń, jak np. popularny funkcjonał (B3LYP) (Becke, three - parameter, Lee-Yang-Parr) [37, 38], gdzie część wymienno-korelacyjna przybiera postać:

$$\begin{aligned} E_{xc}^{B3LYP} = E_x^{LDA} + a_0(E_x^{HF} - E_x^{LDA}) + \\ +a_x(E_x^{GGA} - E_x^{LDA}) + E_c^{LDA} + a_c(E_c^{GGA} - E_c^{LDA}), \end{aligned} \tag{2.29}$$

ze stałymi $a_0 = 0.20$, $a_x = 0.72$, $a_c = 0.81$.

Często w konstrukcji potencjałów hybrydowych oddziaływanie kulombowskie wchodzące do operatora $\hat{U}$ dzielone jest na krótkozasięgowe i długozasięgowe i tylko krótkozasięgowe uwzględniane w energii wymiany E_x^{HF} (czy F_α^S). Tak jest w funkcjonałach (HSE) [39]:

$$E_{xc}^{HSE} = \alpha E_x^{SR,HF}(\omega) + (1-\alpha)E_x^{SR,PBE}(\omega) + E_x^{LR,PBE}(\omega) + E_c^{PBE}, \tag{2.30}$$

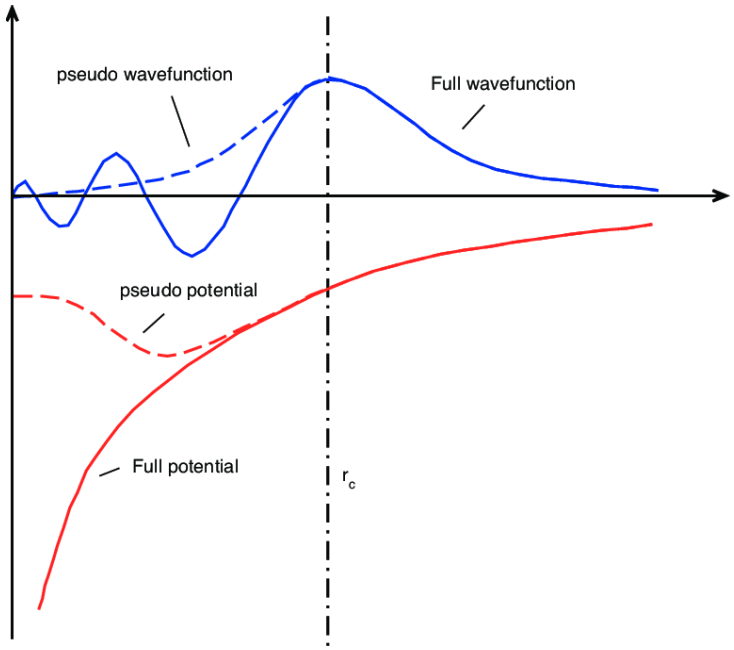
gdzie SR i LR oznaczają odpowiednio krótkozasięgową (ang. short range) i długozasięgową (ang. long range) część oddziaływania kulombowskiego, a α i ω są arbitralnie ustalonymi stałymi. Dla $\alpha = \frac{1}{4}$, oraz $\omega = 0.2$, funkcjonał jest oznaczany jako (HSE06) [39].

2.5. Metody rozwiązania równań Kohna-Shama

W poprzednich paragrafach przedstawiono drogę opracowania funkcjonału wymiany i korelacji, który pozwoliłby na efektywne obliczenia z tak zwaną chemiczną dokładnością (1 kcal/mol), przez Johna Perdew określoną jako drabina Jakuba DFT [40].

Metody rozwiązania równań Kohna-Shama można podzielić na dwie kategorie, w zależności jak traktowane są elektrony inertnych powłok rdzeni atomowych. Jeżeli są one zaliczone do liczby N elektronów, to wtedy mówimy o metodach AE (ang. all electron). W metodach AE potencjał zewnętrzny dla układu elektronów jest określony przez sumę potencjałów kulombowskich pochodzących od jąder atomowych o liczbach atomowych Z_α:

$$V_{ext}(\vec{r}) = -\sum_\alpha \frac{Z_\alpha e^2}{|\vec{R}_\alpha - \vec{r}|}, \tag{2.31}$$



Rysunek 2.1: Pseudopotencjał. Rysunek pochodzi z [44]

gdzie sumowanie po α przebiega po wszystkich jądrach atomowych w położeniach $\{\vec{R}_\alpha\}$.

W drugiej grupie metod, do układu N elektronów włączane są tylko elektrony atomowe powłok walencyjnych i oddziaływanie elektronu walencyjnego z jądrem atomowym i elektronami rdzenia wokół tego jądra opisane jest przez pseuopotencjał $\hat{V}_\alpha^{PP}(\vec{r})$, który najczęściej jest nielokalny. Wtedy potencjał zewnętrzny dla układu elektronów (teraz walencyjnych) jest suma pseudopotencjałów atomowych:

$$\hat{V}_{ext}^{PP}(\vec{r}) = \sum_\alpha \hat{V}_\alpha^{PP}(\vec{r} - \vec{R}_\alpha). \tag{2.32}$$

Istnieje wiele metod generacji pseudopotencjałów [41–43], a ich największą korzyścią w obliczeniach wieloatomowych jest zastąpienie osobliwego potencjału kulombowskiego wokół rdzenia przez gładki pseudopotencjał, co ilustruje rysunek 3.1.

Wprowadzenie pseudopotencjału pozwala na efektywny rozkład funkcji falowych w równaniach Kohna-Shama na stosunkowo niewielką liczbę fal płaskich stanowiących niezwykle wygodną bazę. W chwili obecnej stosowane są trzy grupy pseudopotencjałów: pseudopotencjały zachowujące normę NCPP (ang. Norm Concerving PseudoPotentials) [45], pseudopotencjały ultra miękkie USPP (ang. Ultra Soft PseudoPotentials) [46], oraz pseudopotencjały projektorowe PAW (Projector Augmented Waves) [47].

Pseudopotencjały tych trzech klas różnią się sposobem uwzględniania ekranowania jądra atomowego przez elektrony rdzeniowe. Za najdokładniejszy sposób opisu tego ekranowania, a co za tym idzie za najlepsze pseudopotencjały uważa się wygenerowane metodą PAW. Metoda PAW może być uważana za metodę typu AE z zamrożonymi stanami elektronów rdzeniowych takimi jak w swobodnych atomach. Lata doświadczeń z obliczeniami pseudopotencjałowymi sprawiły, że niemal wszystkie kody numeryczne posiadają

obecnie bazy pseudopotencjałów dla większości pierwiastków układu okresowego [48]. Ułatwia to znacząco przeprowadzenie standardowych obliczeń.

Wybór metody rozwiązania równań Kohna-Shama, AE czy PP, zależy od fizycznego problemu, który chcemy rozwiązać na poziomie teorii DFT. Jeżeli celem jest przykładowo zbadanie procesów fotoemisji z powłok rdzeniowych należy wybrać metodę AE. Jeżeli prowadzone są obliczenia własności materiałów, o których decydują elektrony walencyjne, możemy wybrać metody pseudopotencjałowe. Oba podejścia zostały zaimplementowane w całym szeregu kodów numerycznych. Metoda AE została zastosowana w kodach: używających metody LAPW (jak exciting!, Elk, Fleur, Wien2k), lub bazy numerycznych orbitali (FHI-aims). Metodę pseuopotencjału zaimplementowano w kodach: używających fal płaskich jako bazy (CASTEP, VASP, Quantum Espresso, PWPAW, GPAW, abinit, Da Capo, fhi98md, DOD-pw, PWscf), oraz orbitali atomowych jako bazy (SIESTA, Dmol, AFD-band). W obliczeniach DFT przeprowadzonych na potrzeby niniejszej dysertacji używano kodów pseudopotencjałowych VASP [49–52] i Quantum Espresso [53–55].

Szersze omówienie zagadnień rozwiązywania równan Kohna - Shama można znaleźć w książce Richarda M. Martina [41] lub opracowaniach [42, 56].

2.6. Oddziaływanie jąder

We wcześniejszych paragrafach przedstawiono możliwość wyznaczenia energii stanu podstawowego układu N oddziałujących elektronów w polu potencjału zewnętrznego pochodzącego od jąder atomowych o liczbach atomowych $\{Z_\alpha\}$ lub ekranowanych przez elektrony rdzeniowe jąder, czyli dodatnich jonów o liczbach atomowych $\{Z_\alpha^{val}\}$ w położeniach $\{\vec{R}_\alpha\}$. Dodatnie jądra czy jony oddziaływając ze sobą poprzez potencjał kulombowski i ich energia wzajemnego oddziaływania jest równa

$$E_{I-I} = \frac{1}{2}e^2 \sum_\alpha \sum_\beta \frac{Z_\alpha Z_\beta}{|\vec{R}_\alpha - \vec{R}_{beta}|}, \tag{2.33}$$

gdzie użyto odpowiednich wartości liczb atomowych Z_α (tzn. liczby atomowej lub jonowej). Energia E_{I-I} jest obliczana w stosowanych w pracy doktorskiej kodach (VASP i Quantum Espresso) używając metody Ewalda [53]. Całkowita energia układu elektronów i jonów jest równa $E_{tot} = E_{el} + E_{I-I}$. Żeby obliczyć równowagowe położenia jąder lub jonów, należy znaleźć położenia $\{\vec{R}_\alpha^0\}$, w których wszystkie siły $\vec{F}_\alpha := \frac{\partial E_{tot}}{\partial \vec{R}_\alpha}$ działające na jądra (czy jony) są równe zero. Wkłady do sił pochodzące od E_{I-I} liczone są bezpośrednim rachunkiem, a wkłady od E_{el} stosując twierdzenie Hellmanna-Feynmana [57]. Podobnie można wyznaczyć optymalny kształt komórki elementarnej, żądając żeby elementy tensora naprężeń

$$\sigma_{\alpha,\beta} = \frac{\partial E_{tot}}{\partial \epsilon_{\alpha\beta}} \tag{2.34}$$

równe pochodnej energii całkowitej po składowych tensora odkształcenia, również znikały. W kodach VASP czy Quantum Espresso obliczanie sił czy składowych tensora naprężeń jest standardową procedurą prowadzącą do wyznaczenia zoptymalizowanej geometrii układu, która była używana w proce-

durach uczenia maszynowego ML (ang. Machine Learning) stosowanych w niniejszej rozprawie.

Rozdział 3

Podstawy pojęcia uczenia maszynowego

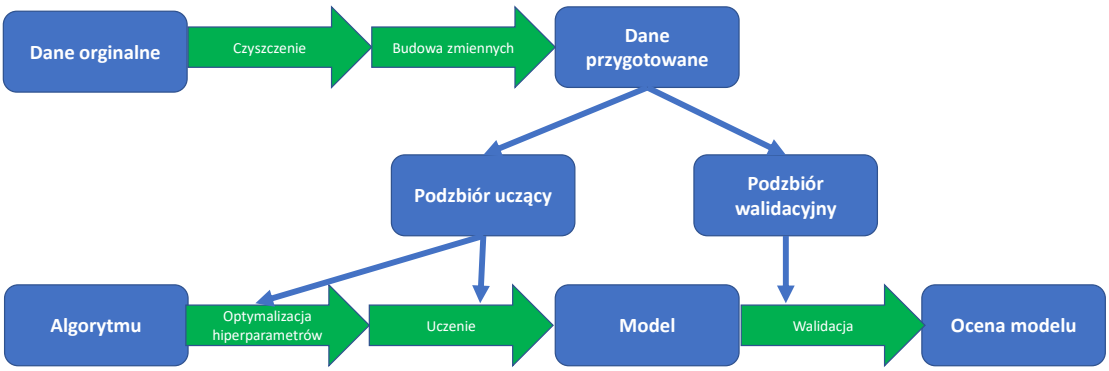
3.1. Metody uczenia maszynowego a metody statystyczne

Termin uczenie maszynowe zawiera w sobie bardzo szeroką klasę metod oraz algorytmów: od najprostszych algorytmów dopasowywania prostej do punktów z obserwacji, czy podziału przestrzeni obserwacji płaszczyzną, po złożone algorytmy sztucznej inteligencji i głębokie sieci neuronowe. Jednak wszystkie techniki szeroko pojętego uczenia maszynowego opierają się na metodach wnioskowania statystycznego i znajdowaniu prawidłowości w dużych zbiorach danych. Podstawą wszystkich algorytmów uczenia maszynowego jest statystyka. Niemniej można wyodrębnić dwie zasadnicze różnice pomiędzy uczeniem maszynowym, a metodami statystycznymi, mianowicie charakterystyka danych użytych do analizy oraz sposób budowania modelu.

Po pierwsze, statystyka może być wykorzystywana na stosunkowo małej próbie z populacji, podczas gdy uczenie maszynowe wymaga dużych zbiorów danych. Jednocześnie liczba zmiennych opisujących dane zagadnienie, a więc używana w budowaniu modelu, może być wielokrotnie większa dla algorytmów uczenia maszynowego. Przez co znaczna część czasu podczas budowania modelu opartego na algorytmach uczenia maszynowego jest poświęcana na dobór optymalnego zestawu zmiennych lub budowę nowych zmiennych oraz na redukcji wymiarowości problemu. Prawidłowość ta jest obserwowana zarówno podczas stosowana tradycyjnych algorytmów uczenia maszynowego jak i w modelach opartych na głębokich sieciach neuronowych. W pierwszym podejściu preprocesowanie zmiennych wykonywane jest "ręcznie", podczas gdy głębokie sieci neuronowe same uczą się istotności zmiennych oraz budują nowe na podstawie oryginalnych zmiennych. Proces ten zajmuje znaczącą część całkowitego czasu uczenia modelu.

Po drugie, różne są także sposoby modelowania procesów oboma narzędziami. Statystyka wymaga zbudowania modelu probabilistycznego opisywanego zjawiska na podstawie wcześniej zdobytej wiedzy oraz szeregu założeń o przebiegu procesu. Uczenie maszynowe nie wymaga żadnej wiedzy o procesie, ani dodatkowych założeń. Statystyka pozwala na dokładne zbadanie zależności pomiędzy zmiennymi i dokładny opis obserwowanego procesu. Podczas gdy uczenie maszynowe skupia się na jak najdokładniejszym przewidywaniu wyniku procesu.

Mimo różnic pomiędzy modelami statystycznymi, a modelami uczenia maszynowego, praca z oboma narzędziami jest podobna. Na dostępnym zbio-



Rysunek 3.1: Przebieg procesu budowania modelu dla algorytmów uczenia maszynowego

rze zaobserwowanych przypadków buduje się model oparty na sztucznej inteligencji. Proces ten dla algorytmów uczenia maszynowego nazywa się uczeniem modelu w nawiązaniu do sztucznej inteligencji oraz podkreśleniu odrębności od metod statystycznych. Ponadto nie jest budowany model statystyczny przebiegu procesu i nie można określić statystycznych przedziałów ufności. W związku z tym konieczne jest stosowanie dodatkowych danych do walidacji poprawności modelu.

Praca z algorytmami opartami na uczeniu maszynowym ma za cel zbudowanie modelu zależnego od szeregu zmiennych niezależnych, na podstawie których obliczana jest przybliżona wartość zmiennej zależnej (w ogólności możliwe jest przewidywanie szeregu zmiennych niezależnych, ale dla przejrzystości rozważania zostaną ograniczone do jednej zmiennej zależnej). Proces budowania modelu składa się z następujących kroków:

1. przygotowanie zmiennych niezależnych dla modelu - w czasie tego etapu wykonywane jest tzw. "czyszczenie danych" (ang. data cleaning) oraz budowane są nowe zmienne niezależne;
2. budowa modelu - w czasie tego etapu wykonuje się optymalizację hiperparametrów algorytmu oraz uczenie modelu;
3. ocena jakości modelu - etap ten polega na obliczeniu metryk jakości modelu na zbiorze walidacyjnym.

Każdy z etapów dzieli się zwykle na mniejsze podetapy. Ponadto, każdy z etapów najczęściej jest powtarzany wielokrotnie w celu uzyskania najlepszego modelu zgodnie z kryteriami etapu trzeciego. Budowa modelu statystycznego składa się z dwóch etapów:

1. określenie klasy rozważanych statystyk,
2. dobór parametrów modelu.

Natomiast w pracy z modelami statystycznymi najczęściej nie poświęca się uwagi budowaniu zmiennych, a jedynie dopasowuje się parametry modelu lub wykonuje się testy zgodności danych z założonym rozkładem. Budowanie

modelu statystycznego sprowadza się zatem do analizy zależności pomiędzy zmienną zależną a zmiennymi niezależnymi dostępnymi w zbiorze danych. Działania te są także wykonywane dla uczenia maszynowego, ale większy nacisk kładzie się na zbudowanie optymalnego zestawu zmiennych niezależnych. Zatem nawet sam proces budowania modeli różni uczenie maszynowe i statystykę.

3.2. Podział metod uczenia maszynowego

Algorytmy oparte na uczeniu maszynowym i sztucznej inteligencji stanowią bardzo obszerną klasę metod analizy danych. Zawierają bardzo zróżnicowane metody, od prostych po niezwykle skomplikowane i złożone. Ponadto, mają wiele zastosowań w różnych klasach problemów. Jedne nadają się tylko do pojedynczego problemu, podczas gdy inne można stosować niemal w każdym przypadku. Różnorodność ta oraz liczebność metod wymaga dokładnej ich klasyfikacji tak, aby łatwo można było wybrać najlepszy algorytm do rozwiązania danego problemu.

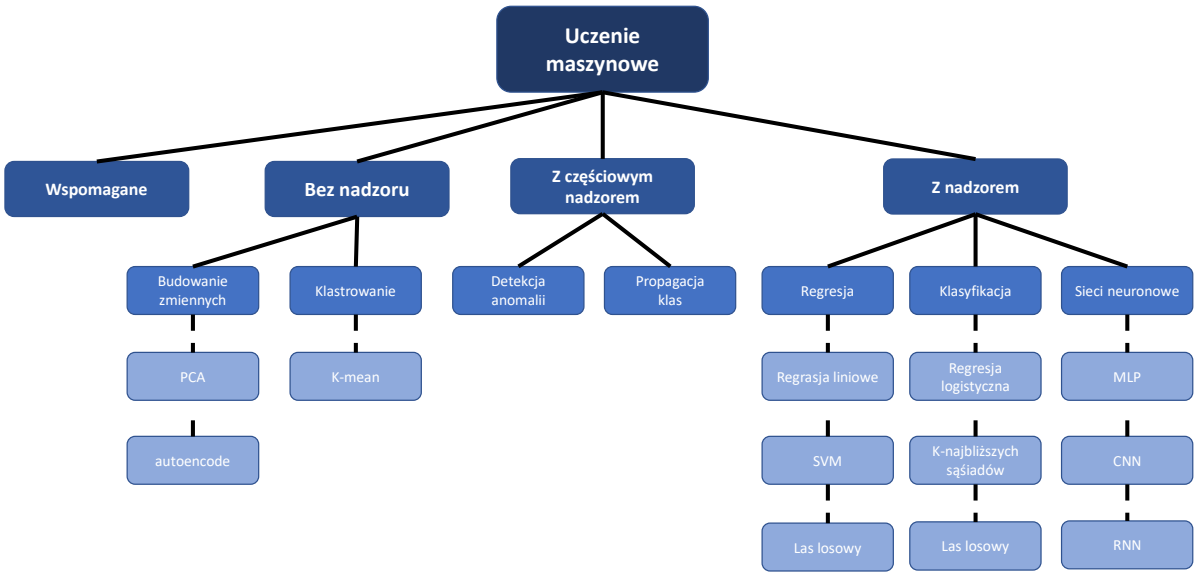
Podstawowym kryterium podziału algorytmów uczenia maszynowego jest podział ze względu na rodzaj posiadanych danych. Najbardziej klasyczna klasyfikacja obejmuje:

— uczenie z nadzorem (ang. supervised learning),
— uczenie bez nadzoru (ang. unsupervised learning),
— uczenie częściowo z nadzorem (ang. semi supervised learning),
— uczenie wspomagane (ang. reinforcement learning).

O uczeniu z nadzorem mówimy wtedy, gdy wykonywana jest analiza danych zawierających zmienne niezależne oraz zmienną zależną (lub kilka zmiennych zależnych). Zmienne niezależne opisują parametry analizowanego eksperymentu podczas, gdy zmienne zależne opisują jego wyniki. Dane tego typu są często określane jako opisane lub oznaczone (ang. labeled). Uczenie z nadzorem można podzielić ze względu na typ zmiennej zależnej na regresje i klasyfikacje.

Jeśli zmienna zależna jest typu rzeczywistego, wówczas mamy do czynienia z regresją odpowiadającą dopasowaniu krzywej do punktów pomiarowych. Jeśli natomiast zmienna zależna jest liczbą całkowitą, opisywana przez skończoną liczbę przedziałów lub klas, wówczas zagadnienie jest określane jako problem klasyfikacji. Podział ten jest jednoznaczny, ale czasem stosuje się podejścia zamieniające problem regresji na klasyfikację (poprzez dyskretyzację zmiennej zależnej) lub klasyfikację na regresję (dopuszcza się przyjmowanie przez zmienna zależną wartości niecałkowitych). Pierwszy typ przekształcenia jest często stosowany, gdy nie jest wymagana dokładna wartość liczbowa zmiennej zależnej. Na przykład często nie jest istotna dokładna wartość przerwy energetycznej, a jedynie fakt czy materiał jest przewodnikiem czy izolatorem. Drugi używa się, gdy klasy mają wewnętrzny porządek i możliwe jest określenie relacji pomiędzy nimi lub ich posortowanie.

Należy jeszcze podkreślić, że konieczne jest uwzględnienie pewnej niedoskonałości procesu zbierania próbek, czy losowości w samym badanym proce-



Rysunek 3.2: Podział metod uczenia maszynowego wraz z przykładowymi algorytmami

sie. Dokładna wartość zmiennej zależnej Y_i jest obarczona pewnym błędem losowym (szum losowy) ϵ_i. Rzeczywista wartość użyta w modelu (y_i) jest równa:

$$y_i = Y_i + \epsilon_i, \tag{3.1}$$

przy czym zakłada się, że dla każdego pomiaru i, ϵ_i pochodzi z tego samego rozkładu i ma rozkład normalny.

Uczenie bez nadzoru opiera się na danych zawierających jedynie wartości zmiennych niezależnych. Uczenie bez nadzoru dzieli się na dwa zagadnienia:

— klastrowanie,
— inżyniera zmiennych.

Klastrowanie polega na znajdowaniu regularności i podobieństw w nieopisanych danych lub próbie automatycznego znalezienia kategorii w danych. Wykonywana jest analiza podobieństwa pomiędzy zmiennymi niezależnymi różnych pomiarów i na tej podstawie budowane są klastry. Inżyniera zmiennych to zbiór technik mających na celu zmniejszenie wymiarowości problemu poprzez usunięcie części nieistotnych zmiennych niezależnych lub stworzenie nowego zestawu zmiennych lepiej opisujących badane zagadnienie.

Algorytmy uczenia maszynowego z częściowym nadzorem mają zastosowanie, gdy w dużym zbiorze danych jedynie część danych jest opisana przez zmienną zależną. Metody te opierają się na wielokrotnym budowaniu modeli i stopniowym klasyfikowaniu nieopisanych danych.

Wśród algorytmów uczenia maszynowego można znaleźć także bardzo egzotyczne przypadki jak algorytmy uczenia wspomaganego (ang. reinforced learning). Są one używane do modelowania przebiegu złożonych procesów

zawierających wiele kroków, po których znany jest jedynie końcowy wynik procesu bez informacji o wpływie poszczególnych kroków na końcowy wynik. Algorytmy te trudno jest jednoznacznie przypisać do jednej z powyższych kategorii jednak są one metodami dedykowanymi do bardzo wąskich zagadnień i nie dają się łatwo uogólnić.

3.3. Podstawowe pojęcia statystyki i teorii informacji

Algorytmy uczenia maszynowego opierają się na wnioskowaniu statystycznym i można je traktować jako rozszerzenie technik statystyki na bardzo duże zbiory danych zawierające wiele zmiennych niezależnych. Stąd zrozumienie algorytmów uczenia maszynowego wymaga znajomości podstawowych pojęć ze statystyki [58–60], które zostały tutaj wymienione.

zdarzenie elementarne - wynik doświadczenia losowego,

populacja - zbiór wszystkich zdarzeń elementarnych możliwych podczas obserwacji danego procesu,

próba - podzbiór populacji,

zmienna losowa - funkcja przypisująca zdarzeniu elementarnemu wartość liczbową,

rozkład prawdopodobieństwa - funkcja określająca prawdopodobieństwo każdej wartości zmiennej losowej,

szansa - stosunek prawdopodobieństwa sukcesu do prawdopodobieństwa porażki. Jeśli prawdopodobieństwo sukcesu wynosi p, wówczas szansa ($ODDs$) jest równa:

$$ODDs = \frac{p}{1-p}, \tag{3.2}$$

wartość oczekiwana $E(y)$ zmiennej losowej y jest zdefiniowana jako:

$$E(y) = \int yp(y)dy, \tag{3.3}$$

gdzie $\int$ jest całką uogólnioną, w której całkowanie jest zastępowane sumowaniem dla zmiennych dyskretnych, a p jest rozkładem prawdopodobieństwa zmiennej losowej y,

wartość średnia $\bar{y}$ zmiennej losowej y jest estymatorem wartości oczekiwanej i jest zdefiniowana jako:

$$\bar{y} = \frac{1}{N}\sum_{i=1}^{N} y_i, \tag{3.4}$$

gdzie sumowanie przebiega po wszystkich obserwacja zmiennej y z próby, N to liczebność próby, a y_i jest wartością i-tej obserwacji,

wariancja zmiennej losowej y jest zdefiniowana jako:

$$\mathrm{Var}(y) = E\left([y - E(y)]^2\right). \tag{3.5}$$

Równanie (3.5) jest równoważne wyrażeniu:

$$\mathrm{Var}(y) = E(y^2) - E(y)^2. \tag{3.6}$$

odchylenie standardowe zmiennej losowej y jest równe:

$$\sigma_y = \sqrt{Var(y)}. \tag{3.7}$$

moment λ^k rzędu k zmiennej losowej x jest zdefiniowany jako:

$$\lambda^k = E(x^k). \tag{3.8}$$

moment centralny μ^k rzędu k zmiennej losowej x jest zdefiniowany jaki:

$$\mu^k = E\left((x - E(x))^k\right). \tag{3.9}$$

kowariancja zmiennych losowych x i y jest zdefiniowana jako:

$$cov(x, y) = E(x - \bar{x}) \cdot E(y - \bar{y}). \tag{3.10}$$

współczynnik korelacji r-Pearsona jest miarą zależności liniowej pomiędzy zmiennymi x i y. Przybiera wartości z zakresu $[-1, 1]$, przy czym wartości dodatnie mówią o dodatniej korelacji (wraz ze wzrostem wartości jednej zmiennej rośnie wartość drugiej), ujemne wartości oznaczają ujemną korelację (wraz ze wzrostem wartości jednej zmiennej wartość druga maleje). Współczynnik korelacji r-Pearsona x i y wyraża się wzorem:

$$r(x, y) = \frac{cov(x, y)}{\sigma_x \sigma_y}. \tag{3.11}$$

Ponadto, dla zrozumienia algorytmów uczenia maszynowego przydatna jest znajomość podstawowych definicji z zakresu teorii informacji takich, jak [61]:

entropia informacji - średnia ilość informacji niesiona przez pojedynczą wiadomość z konkretnego źródła. Entropia informacji $H(x)$ jest zdefiniowana wzorem:

$$H(X) = -\sum_i p(x_i) log_2 \left(p(x_i)\right), \tag{3.12}$$

gdzie sumowanie przebiega po wszystkich możliwych wartościach zmiennej losowej x, a $p(x_i)$ - jest prawdopodobieństwem wystąpienia wartości x_i zmiennej losowej x.

3.4. Metryki

Wśród metod uczenia maszynowego dostępne jest wiele algorytmów, część z nich jest bardzo ogólna, część dedykowana jedynie dla wąskiej grupy problemów. Wybór odpowiedniego algorytmu wymaga dużego doświadczenia oraz znajomości poszczególnych algorytmów. Co więcej, aby zbudować dobry model, konieczne jest jeszcze wybranie optymalnej wartości szeregu hiperparametrów charakterystycznych dla każdego algorytmu. W związku z powyższym niezbędna jest ocena jakości działania testowanego algorytmu i możliwość obiektywnego porównywania wielu metod i wartości hiperparametrów. Do tego celu definiuje się szereg metryk używanych w zależności od rodzaju rozważanego problemu. Najczęściej stosowanymi są [62]:

— MSE - błąd średniokwadratowy (ang. mean square error),
— MAE - błąd modułów (ang. mean absolute error),
— R2 - współczynnik determinacji (ang. R2 score),
— EV - miara wyjaśnienia wariancji (ang. explained variance),
— oob - ang. out-of-bag,
— dokładność - (ang. accuracy),
— precyzja - (ang. precision),

— czułość - (ang. sensitivity),
— współczynnik F1 (ang. F1 score).

Błąd średniokwadratowy (ang. mean square error - MSE) określa jak bardzo wynik regresji różni się od oczekiwanej wartości. Metryka ta jest szczególnie wrażliwa na pojedyncze skrajne wartości, natomiast mniej wrażliwa na niewielkie odchylenia od wartości dokładnych. MSE jest zdefiniowana jako:

$$MSE(y) = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2, \tag{3.13}$$

gdzie sumowanie przebiega po wszystkich obserwacjach w próbie, $\hat{y}_i$ jest zaobserwowaną wartością w i-tym pomiarze, a y_i jest przewidzianą wartością w i-tym pomiarze.
Wartości MSE leżą w przedziale $[0, \infty)$. Wartości bliższe zeru oznaczają lepszą jakość przewidywania modelu.

Błąd modułów (ang. mean absolute error = MAE) jest zdefiniowany podobnie jak MSE i także ma zastosowania do regresji. W odróżnieniu od MSE nie zaniedbuje małych odchyleń od wartości zadanej, zarówno małe jak i duże odchylenia są traktowane tak samo. Metryka ta jest zdefiniowana jako:

$$MAE(y) = \frac{1}{N}\sum_{i=1}^{N}|\hat{y}_i - y_i|. \tag{3.14}$$

Wartości MSE są zawarte w przedziale $[0, \infty)$. Wartości bliższe zeru oznaczają lepszą jakość przewidywania modelu.

Współczynnik determinacji (ang. R2 score - R2) jest podobny do MSE i także jest wrażliwy na wartości mocno odbiegające od wartości zadanej, ale w przeciwności do MSE i MAE jest unormowany i przyjmuje wartości z przedziału $[0, 1]$. Unormowanie tej metryki pozwala na obiektywne porównywanie modeli bez konieczności posiadania wiedzy dziedzinowej lub choćby informacji o zbiorze wartości zmiennej zależnej (przykładowo błąd 0.1 ev w zależności od badanej wielkości może być istotny lub nie). Metryka ta jest szczególnie popularna w środowisku analityków danych. Jest ona zdefiniowana jako:

$$R^2(y) = 1 - \frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{\sum_{i=1}^{N}(\bar{y}_i - y_i)^2}. \tag{3.15}$$

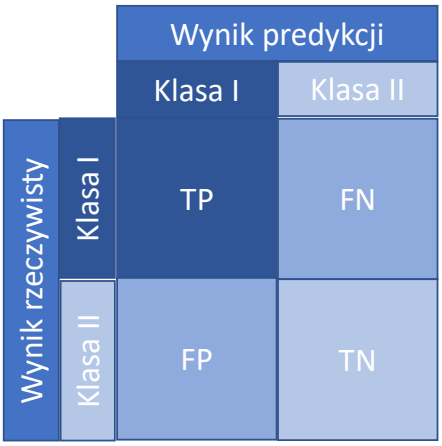
Wartości bliższe jedynce oznaczają lepszy model.

Miara wyjaśnienia wariancji (ang. explained variance score - EV) jest metryką stosowaną do regresji i pokazuje jak dużą moc uogólniania ma model i jak dużą część zmienności zmiennej zależnej był w stanie opisać. Jest ona zdefiniowana jako:

$$EV(y, \bar{y}) = 1 - \frac{\mathrm{Var}(y - \bar{y})}{\mathrm{Var}(y)}. \tag{3.16}$$

EV przyjmuje wartości z przedziału $[0, 1]$, wartości bliższe jedynce oznaczają lepszą jakość przewidywania modelu.

oob (ang. out-of-bag) jest metryką jakości modelu zarówno regresji jak i klasyfikacji. Może być obliczona dla algorytmów lasu losowego (patrz 3.7.5)



Rysunek 3.3: Macierz poprawności klasyfikacji

lub algorytmów uczenia w zespole (patrz 3.7.7). Oba te algorytmy podczas uczenia budują wiele podzbiorów zbioru treningowego i na każdym z nich budują modele, które są następnie agregowane. Następnie na pozostałych podzbiorach zbioru treningowego obliczane są odpowiednie metryki.

Zbudowanie uniwersalnej metryki dla problemu klasyfikacji jest znacznie trudniejsze niż dla regresji. Główną komplikacją jest liczba klas. Problemy przedstawiane w tej pracy ograniczają się do klasyfikatora binarnego (gdzie możliwe są tylko dwie klasy), pozwala to na znaczne uproszczenie omawianych metryk. W przypadki klasyfikacji dwóch klas mamy do czynienia z czterema możliwościami poprawności wyniku:

TP - prawdziwie dodatni (ang. True positive) oznacza, że obserwacja z pierwszej klasy została poprawnie sklasyfikowana,
TN - prawdziwie ujemny (ang. True negative) oznacza, że obserwacja z drugiej klasy została poprawnie sklasyfikowana,
FP - fałszywie dodatni (ang. False positive) oznacza, że obserwacja z klasy drugiej została zaklasyfikowana jako obserwacja z klasy pierwszej,
FN - fałszywie ujemny (ang. False negative) oznacza, że obserwacja z klasy pierwszej została zaklasyfikowana jako obserwacja z klasy drugiej.

Przypadki te są przedstawiane w postaci macierzy poprawności klasyfikacji (znajduje się ona na rysunku 3.3). Oznaczenia TP, TN, FP, FN są używane, także jako liczba obserwacji sklasyfikowanych odpowiednio jako TP, TN, FP, FN. Konwencja ta będzie stosowana podczas definiowania metryk. Wówczas metryki dla klasyfikacji mogą zostać zdefiniowane jako:

Dokładność (*Acc*):

$$Acc = \frac{TP + TN}{TP + TN + FP + FN}, \tag{3.17}$$

oznacza część poprawnie sklasyfikowanych obserwacji.

Precyzja (Pr):

$$Pr = \frac{TP}{TP + FP}, \tag{3.18}$$

oznacza część poprawnie sklasyfikowanych obserwacji klasy pierwszej do wszystkich obserwacji klasy pierwszej.

Czułość (Se):

$$Se = \frac{TP}{TP + FN}, \tag{3.19}$$

oznacza część poprawnie sklasyfikowanych obserwacji klasy pierwszej do wszystkich obserwacji sklasyfikowanych jako klasa pierwsza.

F1 score $(F1)$:

$$F1 = 2 * \frac{Se * Pr}{Se + Pr}, \tag{3.20}$$

jest bardziej wiarygodna niż czułość lub precyzja jeśli klasy są nie równo liczne.

Oczywiście możliwe jest zdefiniowanie znacznie większej liczby metryk jako dowolną kombinację TP, TN, FP, FN, jednak powyżej zdefiniowane metryki są najczęściej używane w literaturze. Definicje metryk są silnie niesymetryczne ze względu na poprawność klasyfikacji pierwszej klasy. Wynika to ze względów historycznych oraz ich zastosowań w medycynie i automatyzacji diagnostyki. W diagnostyce konsekwencją błędnego sklasyfikowania pacjenta zdrowego jako chorego jest jedynie przeprowadzenie dokładniejszych badań, podczas gdy klasyfikacja pacjenta chorego jako zdrowego wiąże się ze znacznym opóźnieniem leczenia. Zatem pewne błędy niosą za sobą znacznie poważniejsze konsekwencje.

W przypadku klasyfikacji bardzo istotna jest liczebność obserwacji obu klas. Opisywane metryki dobrze sprawdzają się, gdy liczebność obu klas jest porównywalna. Podczas, gdy stosunek liczebności jednej klasy do drugiej przekracza 3:1, konieczne jest zastosowanie metryk uwzględniających niezbalansowanie klas (ang. imbalanced classification) [63–65].

3.5. Podzbiory danych

Zdefiniowane powyżej metryki wymagają określenia na jakim zbiorze danych będą obliczane. Jednocześnie wybór zbioru walidacyjnego będzie kluczowy podczas oceny jakości modelu. Najczęściej podczas budowania modelu liczba danych jest ograniczona i niemożliwe jest pozyskanie dodatkowych danych w celu walidacji, Dane walidacyjne mocno różniące się od treningowych zaniżą wartość metryk, podczas gdy dane pochodzące ze zbioru treningowego poprawią wartości metryk. Walidacja modelu musi się zatem odbywać na danych z poza zbioru treningowego jednak podobnych do danych w nim zawartych.

Najpopularniejszą metodą wyboru zbioru walidacyjnego jest wyodrębnienie go ze posiadanych danych. Wykonuje się losowego podziału danych na dane treningowe i testowe. Zbiór walidacyjny w zależności od ilości posiadanych danych stanowi 10 do 20 procent wszystkich danych. Dla algorytmów wymagających wyboru hiperparametrów lub porównania różnych zbiorów zmiennych niezależnych wyodrębnia się dodatkowy podzbiór walidacyjny. Służący on do określania jak zmiany parametrów modelu lub zmiennych wpływa na jakość jego przewidywań, na zbiorze testowym określa się natomiast zdolność uogólniania modelu dla najlepszego zestawu parametrów. Zbór testowy stanowi od 5 do 20 procent zbioru danych,

Zatem podczas budowania modelu pracuje się z trzema podzbiorami danych treningowym, walidacyjnym i testowym. Żadne ze zbiorów nie mogą mieć część wspólnej. Ponadto w różnych podzbiorach nie mogą występować obserwacje o tych samych wartościach zmiennych niezależnych, nawet jeśli opisują one różne obserwacje i jedynie wybór reprezentacji oraz zmiennych niezależnych prowadzi od powtórzenia wartości.

Opisana procedura podziałów danych nie gwarantuje najlepszej estymacji metryk. W najprostszym przypadku podziały wykonywane są losowo, co pozwala na niezerowe prawdopodobieństwo, że zbór treningowy będzie posiadał tylko obserwacje z jednej klasy lub skrajne wartość zmiennych niezależnych. Dokładniejszą metodą jest zastosowanie testowania krzyżowego (kroswalidacji, ang. cross validation). Szczególny procedury zostały opisane w następnym rozdziale.

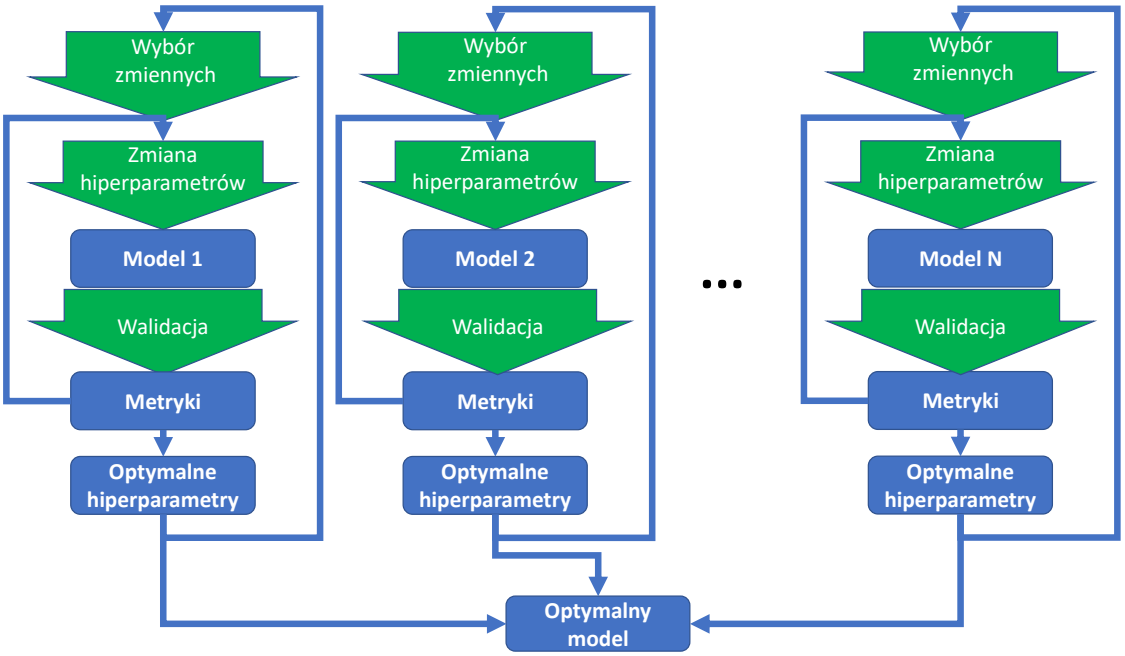
3.6. Testowanie krzyżowe i wyznaczanie hiperparametrów

Proces uczenia modelu jest w zasadzie ostatnim etapem bardziej skompilowanego procesu. Proces budowania modelu można podzielić na trzy etapy:

1. wybór algorytmu,
2. optymalizacja hiperparametrów,
3. walidacja modelu.

Etapy pierwszy i drugi przeprowadzane są w dwóch zagnieżdżonych pętlach. Budowa modelu rozpoczyna się od wybrania zestawu algorytmów pozwalających na rozwiązanie zadanego problemu. Następnie dla każdego z modelu optymalizuje się hiperparametry i określa jakość modelu z najlepszymi hiperparametrami. Etap trzeci polega na przetestowaniu modelu dla nowych danych i określenia końcowej jakości modeli.

Sam proces określania jakości modeli nie jest jednak trywialny. Konieczne jest wybranie metryki lub metryk opisujących jakość modelu oraz oszacowanie zdolności przewidywania modelu. Istnieje kilka podejść oszacowania jakości modelu. Jedną z możliwości jest zbudowanie modelu na całym posiadanym zbiorze danych, a następnie określenie wartości metryk dla danych z tego samego zbioru. Prowadzi to jednak do zbudowania modelu, który będzie dokładnie odtwarzał zbiór treningowy, ale nie będzie miał żadnej zdolności uogólniania dla nowych danych. Podejście to, choć czasem stosowane, nie po-

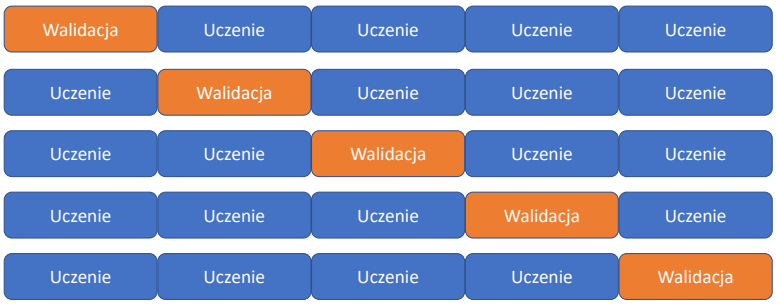


Rysunek 3.4: Schemat wyboru najlepszego modelu.

zwala na rzetelne porównywanie modeli i prowadzi do przeuczenia modelu. Lepszym rozwiązaniem jest wyodrębnienie z całego zbioru danych podzbioru danych i użycia go do wyznaczenia metryk jakości modelu. Technika ta jest określana jako prosta walidacja, a wyodrębniony podzbiór nazywa się zbiorem walidacyjnym. Tu także pojawia się ryzyko, że podział będzie preferował jedną klasę modeli i o otrzymanej wartości metryki będzie w największym stopniu decydował podział i zawartość zbioru walidacyjnego. Jednym z rozwiązań jest wielokrotna walidacja modelu i zebranie cząstkowych wartości metryk do jednej liczby. Innym rozwiązaniem jest zastosowanie testowania krzyżowego (ang. cross validation). Polega ono na podzieleniu zbioru treningowego na N rozłącznych podzbiorów. Następnie jeden podzbiór używa się do obliczenia metryk, a pozostałe $N-1$ do uczenia modelu. Procedurę powtarza się tak, aby każdy podzbiór był użyty do obliczenia metryk. Po obliczeniu wszystkich metryk liczy się ich średnią i odchylenie standardowe, a obie wartości są używane do określenia jakości modelu. Wybór wartości parametru N jest arbitralny, w literaturze najczęściej spotyka się:

— 5-krotne testowanie krzyżowe, gdzie $N = 5$,
— 10-krotne testowanie krzyżowe, gdzie $N = 10$,
— Leave-one-out, gdzie $N =$ wielkość próby.

Podczas wykonywania testowania krzyżowego każdorazowe uczenie modelu musi być wykonane od początku bez wiedzy o wynikach z pozostałych iteracji [62]. Ponadto, jeśli w czasie budowania modelu stosowane są jakieś dodatkowe kroki, jak normalizacja zmiennych czy redukcja wymiarowości, należy je wykonywać dla każdej iteracji oddzielnie aby oszacowanie było wiarygodne.



Rysunek 3.5: Podział zbioru treningowego podczas 5-krotnego testowania krzyżowego

Optymalizacja hiperparametrów jest najczęściej wykonywana poprzez równomierne próbkowanie całej przestrzeni parametrów (ang. grid search). Technika ta polega na wybraniu zbioru wartości każdego hiperparametru. Następnie budowane są modele dla każdej kombinacji wartości hiperparametrów. Z otrzymanych modeli wybiera się najlepszy zgodnie z używaną metryką.

W związku z wielokrotnym uczeniem modelu procedura 'grid search' jest bardzo kosztowna. Ponadto wymaga doświadczenia oraz często wiedzy dziedzinowej podczas określenia zakresu badanych parametrów. 'Grid search' okazuje się jednak niezwykle skuteczna i często jedyną dostępna techniką. Stosowane są także inne algorytmy optymalizacji bezgradientowej takie jak: algorytmy genetyczne [66] czy inteligencja rozproszona [67]. Wprowadzają one dodatkową warstwę komplikacji do problemu oraz wymagają wyboru kolejnego zestawu hiperparametrów.

Ostateczna walidacja modelu powinna odbywać się na danych, które nie zostały użyte do uczenia modelu. Dlatego najczęściej z całego zbioru danych wydziela się zbór walidacyjny. Jego wielkość stanowi zwykle około 10% całego zbioru.

3.7. Podstawowe algorytmy - klasyfikacja

3.7.1. Regresja logistyczna (ang. logistic regression)

Regresja logistyczna [68, 69] jest najprostszym algorytmem klasyfikacji, który zasadniczo pozwala na dokonanie rozróżnienia pomiędzy dwoma klasami, ale może zostać rozszerzony na więcej klas. W literaturze można znaleźć wiele sposobów definiowania tej metody. Jednym z najbardziej fundamentalnych jest definicja oparta na obliczeniu szansy ($ODDs$). Szansa może być powiązana z prawdopodobieństwem p pierwszej klasy za pomocą tzw. funkcji logitowej:

$$logit(p) := \ln \frac{p}{1-p} = \ln ODDs. \tag{3.21}$$

Przekształcenie odwrotne ma postać:

$$p = \frac{1}{1 + e^{-logit(p)}}. \tag{3.22}$$

Jeśli założymy, że relacja pomiędzy zmiennymi niezależnymi $x_1, x_2, \ldots x_N$ a $logit(p)$ jest liniowa (patrz 3.8.1) to:

$$logit(p) = \beta_0 + \beta_1 x_1 + \cdots + \beta_N. \tag{3.23}$$

Wówczas, możemy wyznaczyć prawdopodobieństwo pierwszej klasy jako:

$$p = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \cdots + \beta_N x_N)}}. \tag{3.24}$$

W wyniku tej procedury otrzymujemy funkcję przekształcającą zmienne niezależne na prawdopodobieństwo zaobserwowania pierwszej klasy. Decyzja o przypisaniu danej obserwacji do pierwszej lub drugiej klasy następuje na podstawie otrzymanego prawdopodobieństwa. Próg przejścia pomiędzy klasami zależy od konkretnego przypadku i może być zmieniany w zależności od oczekiwanej czułości klasyfikatora. Algorytm regresji logistycznej często jest rozszerzany o dodatkową klasę "nie zaklasyfikowane" w przypadku, gdy obliczone prawdopodobieństwo jest bliskie wartości granicznej.

Regresja logistyczna może zostać wyprowadzona także jako rozwinięcie regresji liniowej [70, 71]. Wówczas konieczne jest przeskalowanie zakresu wartości z $[-\infty, \infty]$ do $[0, 1]$. Wykonuje się to za pomocą funkcji sigmoidalnej $f(x) = \frac{1}{1+e^{-x}}$. Po prostych przekształceniach otrzymać można funkcję $logit$. Wyprowadzenie to nie jest istotnie różne od prezentowanego, jednak historycznie stosowano funkcję szansy [71].

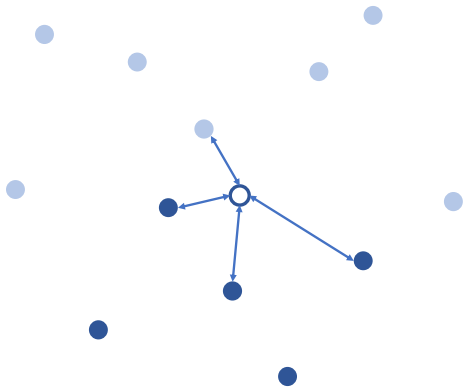
Regresja logistyczna może być rozszerzona na przypadek klasyfikacji wielu klas. Wówczas oblicza się prawdopodobieństwo każdej klasy osobno i wybiera klasę o najwyższym prawdopodobieństwie.

3.7.2. Algorytm k-najbliższych sąsiadów (ang. k-nearest neighbors)

Algorytm k - najbliższych sąsiadów jest modelem nieparametrycznym, czyli nie można jawnie wypisać zależności pomiędzy zmienną zależną, a zmiennymi niezależnymi. Algorytm opiera się na założeniu, że w gęsto spróbkowanej przestrzeni zdarzeń obserwacje należące do jednej klasy tworzą klastry lub innymi słowy są blisko siebie. Dodatkowo zakłada się, że posiadany zbiór obserwacji jest reprezentatywny i nowe obserwacje są jedynie odchyleniami od znanych obserwacji. Algorytm wygląda następująco:

1. policzyć odległość niesklasyfikowanej obserwacji i od wszystkich obserwacji znajdującymi się w zbiorze treningowym;
2. znaleźć k najbliższych sąsiadów;
3. jako przewidzianą klasę obserwacji i-tej wybrać najliczniejszą klasę w sąsiedztwie obserwacji.

Metoda ta nie wymaga uczenia modelu i może być użyta od razu po zdefiniowaniu zbioru treningowego. Jest jednocześnie kosztowna, gdyż dla każdej niesklasyfikowanej obserwacji, obliczana jest odległość od wszystkich sklasyfikowanych obserwacji w zbiorze danych. Koszt obliczeniowy szybko rośnie wraz ze zwiększaniem liczebności zbioru treningowego oraz wymiarowości problemu. Algorytm k-najbliższych sąsiadów w praktyce może być stosowany jedynie w przestrzeniach niskowymiarowych, na skutek tzw. 'Przekleństwa Wymiarowości' (ang. Curse of dimensionality) [72].



Rysunek 3.6: Algorytm k-najbliższych sąsiadów

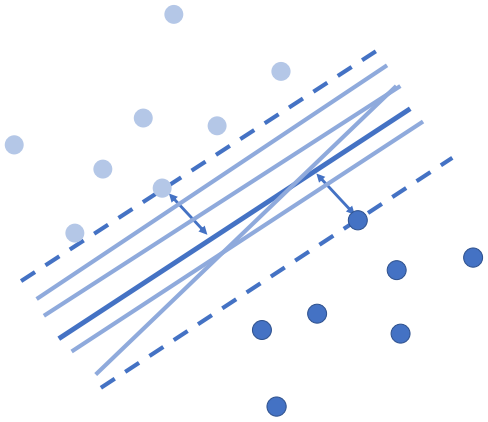
Podczas stosowania algorytmu k-najbliższych sąsiadów konieczne jest dokonanie dwóch wyborów: (i) wyboru metryki w przestrzeni parametrów oraz (ii) wartości parametru k. W praktyce zwykle stosuje się jedynie normę L2. Wybór k jest mniej oczywisty: k = 1 prowadzi do bardzo niestabilnego modelu, podczas gdy duże wartości k znacząco obniżają dokładność klasyfikacji. Preferowane są nieparzyste wartości k, aby uniknąć sytuacji równej liczebności obu klas wśród najbliższych sąsiadów, która prowadzi do niejednoznaczności wyniku.

Algorytm k-najbliższych sąsiadów nie jest ograniczany liczbą klas w zbiorze treningowym, o ile każda klasa jest reprezentowana w bazie danych. Ponadto algorytm może zostać rozszerzony na regresję, wówczas krok 3 algorytmu zostaje zastąpiony przez obliczenie średniej spośród sąsiadów.

3.7.3. Metoda wektorów nośnych (ang. support vector machine)

Metoda wektorów nośnych (ang. Support Vector Machine - SVM) jest algorytmem pozwalającym na wyznaczenie płaszczyzny rozdzielającej dwa zbiory, czyli pozwala na rozwiązywanie zagadnienia klasyfikacji. SVM może zostać rozszerzony także na przypadek regresji.

Podczas wyznaczania krzywej rozdzielającej dwie klasy, możliwa jest duża niejednoznaczność. Przykład takiej sytuacji przedstawia rysunek 3.7. Dla przedstawionej sytuacji możliwe są przesunięcia równoległe krzywej oraz zmiana kąta nachylenia krzywej. Niejednoznaczność ta stała się punktem wyjścia do zdefiniowania metody wektorów nośnych. Definiuje się obszar marginesu jako przestrzeń wewnętrzną, której krzywa rozdzielająca klasy może być przesuwana równolegle bez zmiany działania modelu na zbiorze treningowym. Jakość modelu jest określana jako wielkość marginesu. Zatem parametry krzywej optymalizuje się tak, żeby odległość krzywej od najbliższej obserwacji była maksymalna.



Rysunek 3.7: Niejednoznaczność rozdziału klas, poszczególne klasy zaznaczono różnymi kolorami.

Dla przypadku separacji liniowej, krzywa rozdzielająca klasy ma postać:

$$f(x) = \beta_0 + \sum_{i=1}^{N} \beta_i x_i = 0. \tag{3.25}$$

Wartości x dla których $f(x) > 0$ odpowiadają pierwszej klasie, a wartości x takie że $f(x) < 0$ drugiej. Można także zdefiniować marginesy: po lewej od linii rozdzielającej klasy (m_1) oraz po prawej (m_2) jako:

$$m_1 : \beta_0 + \sum_{i=1}^{N} \beta_i x_i = -M, \tag{3.26}$$

$$m_2 : \beta_0 + \sum_{i=1}^{N} \beta_i x_i = M. \tag{3.27}$$

Szerokość marginesu to odległość między marginesami m_1 i m_2. Jeśli klasy będziemy definiowane przez $y = 1$ i $y = -1$, wówczas można klasy rozdzielać przez:

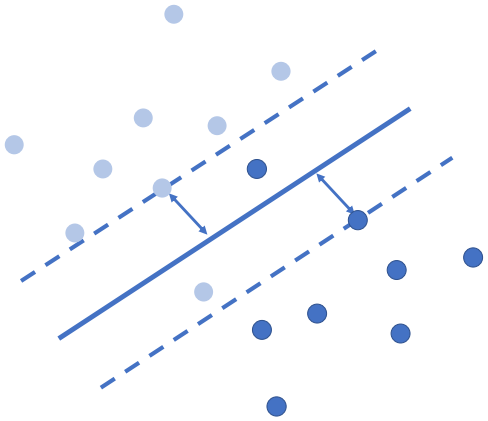
$$y(\beta_0 + \sum_{i=1}^{N} \beta_i x_i) > 0. \tag{3.28}$$

W przypadku, gdy obie klasy nie są separowane płaszczyzną z rozpatrywanej klasy (np. prostą w przypadku dwuwymiarowym), wprowadza się zmienne osłabiające ζ. Sytuacja taka została przedstawiona na rysunku 3.8. Zmiana ta pozwala, aby część punktów znalazła się wewnątrz marginesu lub po niewłaściwej stronie płaszczyzny rozdzielającej klasy. Wówczas marginesy m_1 i m_2 dla przypadku liniowego mają postać:

$$m_1 : \beta_0 + \sum_{i=1}^{N} \beta_i x_i = M(-1 + \zeta), \tag{3.29}$$

$$m_2 : \beta_0 + \sum_{i=1}^{N} \beta_i x_i = M(1 - \zeta). \tag{3.30}$$

Wartość parametru ζ jest miarą, która określa, jak wiele punktów może być źle sklasyfikowanych przez model.



Rysunek 3.8: Klasy nie separujące się.

SVM może być rozszerzone do przypadku nielinowego poprzez transformację Ψ do przestrzeni więcej wymiarowej (zostanie to omówione w rozdziale 3.8.2). Możliwe jest zastąpienie funkcji separacji przez:

$$f(x) = \beta_0 + \sum_{j=0}^{M} \alpha_i \langle x, x_j \rangle. \tag{3.31}$$

gdzie $\sum_{j=0}^{M}$ przebiega po całym zbiorze treningowym, $\langle x_i, x_j \rangle$ jest iloczynem skalarnym pomiędzy x_i i x_j, a α_i jest parametrem dopasowania
Wówczas nieliniowość można zapisać jako:

$$f(x) = \beta_0 + \sum_{j=0}^{M} \langle \Psi(x), \Psi(x_j) \rangle. \tag{3.32}$$

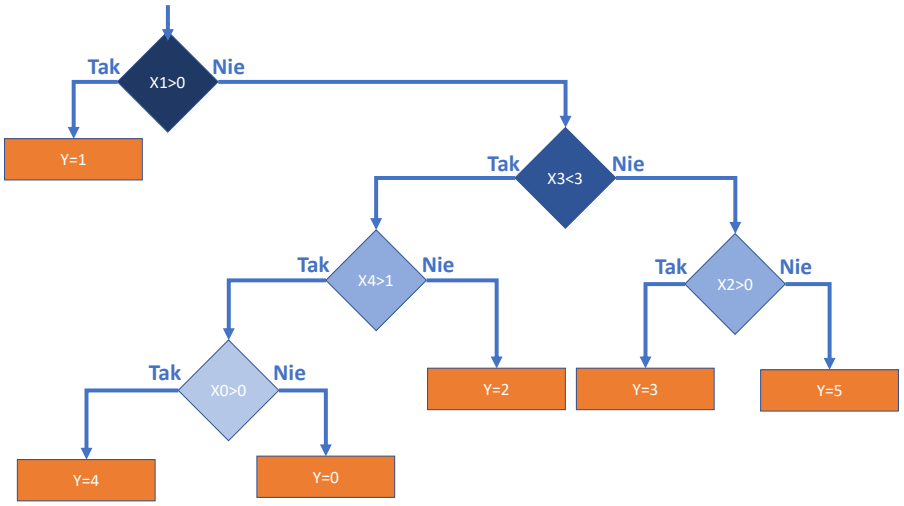
Ponieważ, nie jest konieczna znajomość postaci przetransformowanej funkcji, a jedynie wartości iloczynu skalarnego, obliczanie transformacji zmiennych może być zastąpione transformacją iloczynu skalarnego. Taka forma umożliwia zastosowanie jądra operatora (ang. kernel):

$$\langle \Psi(x), \Psi(x_j) \rangle = K(x, x_j). \tag{3.33}$$

Zastosowanie jąder operatorów pozwala na zmniejszenie kosztu obliczeniowego oraz łatwe wprowadzenie nieliniowości do modelu. Najczęściej spotykane jądra operatora to:

— liniowy: $K(x, x_j) = \langle x, x_j \rangle$,
— rbf: $K(x, x_j) = exp(-\gamma \left\| x - x_j \right\|^2)$,
— sigmoid: $K(x, x_j) = tanh(\gamma \langle x, x_j \rangle + r_c)$.

gdzie γ i r_c są parametrami danego jądra operatora.



Rysunek 3.9: Przykład działania drzewa decyzyjnego

3.7.4. Drzewo decyzyjne (ang. decision tree)

Idea drzewa decyzyjnego wywodzi się z teorii procesów biznesowych i prób automatyzacji procesu podejmowania decyzji w czasie przebiegu procesu. Drzewo jest zbudowane z szeregu połączonych ze sobą węzłów decyzyjnych oraz liści. Każdy węzeł decyzyjny posiada przynajmniej dwie gałęzie łączące je z kolejnymi węzłami lub liśćmi. Liście odpowiadają decyzjom lub w przypadku klasyfikacji przyporządkowanym klasom. Pierwszy węzeł decyzyjny nazywany jest korzeniem. Drzewo decyzyjne przechodzone jest od korzenia do liści. Na każdym napotkanym węźle podejmowana jest decyzja, którą gałęzią będzie kontynuowane przechodzenie drzewa. Decyzja o wyborze ścieżki jest podejmowana na podstawie wartości jednej zmiennej niezależnej. W czasie rozważań dla uroszczenia ograniczymy się do drzew binarnych, gdzie każdy węzeł ma dwie gałęzie. Opis drzew decyzyjnych może być łatwo uogólniony na drzewa o więcej niż dwóch gałęziach ale drzewa binarne są najczęściej stosowane i najczytelniejsze.

Najważniejszą cechą drzew losowych jest dzielenie przestrzeni zmiennych niezależnych na regiony zgodnie z regułami zapisanymi w węzłach, a następnie przypisywanie regionom klas. Każdy kolejny poziom drzewa decyzyjnego odpowiada wprowadzeniu kolejnych granic w przestrzeni zmiennych niezależnych.

Podczas budowy drzewa decyzyjnego konieczne jest określenie kryteriów wyboru kolejnej gałęzi na każdym węźle decyzyjnym. Istnieje kilka algorytmów budowania drzewa losowego: ID3 [73], Gini Index (CART) [74], Chi-Square (CHAID)[75], czy Reduction in Variance [76]. Jednak najbardziej popularnym jest ID3. Opiera się on na entropii informacji. Każdy węzeł jest budowany tak, aby wybrana zmienna niezależna i kryterium wyboru ścieżki prowadziło do zwiększenia entropii informacji po przejściu przez węzeł. Innymi słowy maksymalizowany jest przyrost informacji.

Kolejne węzły do drzewa są dodawane tak długo, aż uzyskana zostanie zadana dokładność modelu lub maksymalna głębokość drzewa. Niepożądane

jest jednak zbudowanie zbyt dużego drzewa. Otrzymuje się wówczas przeuczony model, który będzie jedynie powtarzał dane ze zbioru treningowego bez jakiejkolwiek zdolności uogólniania dla nowych danych. Istnieje wiele technik ograniczania wielkości drzewa najpopularniejsze to ograniczanie liczby węzłów decyzyjnych oraz przycinanie drzewa (ang. pruning). Pruning polega na przechodzeniu drzewa od liści do korzenia. Każdy napotkany węzeł jest zastępowany liściem najliczniejszej klasy, po czy obliczana jest dokładność nowego drzewa, jeśli zmiana jest niewielka eliminacja węzła jest akceptowana.

Dodatkową funkcjonalnością drzew decyzyjnych jest oszacowanie istotności zmiennych dla modelu. Podczas budowania modelu tworzony jest szereg węzłów decyzyjnych i są one uszeregowane zgodnie z malejącą ich istotnością dla modelu zgodnie z przyjętą normą. Mechanizm ten pozwala na określenie istotności każdej zmiennej. Ze względu na dużą wrażliwość drzew decyzyjnych na szum w zbiorze treningowym oraz tendencję do przeuczania informacja o istotności zmiennych z pojedynczego drzewa w praktyce nigdy nie jest wykorzystywana, jest ona jednak używana w lesie losowym.

3.7.5. Las losowy (ang. random forest)

Rozszerzeniem drzewa decyzyjnego jest las losowy [77]. Główną wadą drzew decyzyjnych jest ich mała elastyczność, łatwość przeuczania oraz mała zdolność uogólniania. Rozwiązaniem tych problemów jest zastosowanie wielu drzew losowych jednocześnie.

Algorytm uczenia drzewa decyzyjnego jest w pełni deterministyczny i użycie kilku drzew decyzyjnych dla tych samych danych spowoduje stworzeniem szeregu identycznych drzew. Aby tego uniknąć, można dokonać podziału zbioru treningowego na podzbiory dla każdego drzewa i przeprowadzić uczenie każdego z nich na innym podzbiorze danych. Mimo, że technika ta tworzy różne drzewa, są one jednak skorelowane ze sobą. Korelację usuwa się poprzez losowania od jakich parametrów zależy pojedyncze drzewo.

Metoda budowania lasu losowego pochodzi od techniki bagging [78]. Ze zbioru treningowego jest losowany k-elementowy podzbiór, na którym budowany jest model. Procedura ta jest powtarzana p-krotnie. Dodać należy, że każde losowanie podzbioru wykonuje się niezależnie od pozostałych, czyli można tu myśleć o losowaniu obserwacji ze zwracaniem. W wyniku czego otrzymujemy zbiór p modeli wytrenowanych tylko na części danych. Wynikiem końcowym modelu jest najczęściej przewidywana klasa przez zbiór p modeli. Technikę 'bagging' można stosować także dla regresji, wówczas jako wartość końcowa brana jest średnia arytmetyczna przewidywań poszczególnych modeli. Kolejnym krokiem budowania lasu losowego jest zastosowanie techniki 'bagging' dla zmiennych niezależnych wykorzystywanych do budowania modelu. Losowany jest wiec podzbiór zmiennych, jedno losowanie jest wykonywane dla jednego podzbioru treningowego.

Zastosowanie wielu drzew losowych pozwala na uniknięcie przeuczenia modelu oraz zmniejszenie jego wrażliwości na szum w zbiorze treningowym. Zastosowanie losowego podziału zmiennych niezależnych usuwa korelację pomiędzy drzewami.

Ponieważ las losowy składa się z szeregu niezależnych drzew losowych możliwe jest precyzyjne określenie istotności zmiennych. Istotność zmiennych jest wyznaczana osobno dla każdego drzewa po czym liczona jest średnia wartość istotności zmiennej wśród drzew, które jej używały do budowy modelu. Duża liczba drzew pozwala na wyznaczenie rzeczywistej hierarchii istotności zmiennych. Podkreślić należy jednak, że wartość liczbowa istotności zmiennych zwracana przez las losowy nie może być porównywana pomiędzy modelami, a można ją używać jedynie do selekcji zmiennych istotnych w ramach pojedynczego modelu lub zestawu zmiennych.

3.7.6. Ekstremalne losowe drzewa (ang. Extreme random tree)

Algorytm ekstremalne losowe drzewa (ang. Extreme random tree, ERT) [79] jest wariantem lasu losowego. Ma on rozwiązać problemy kosztowności obliczeniowej podczas uczenia oraz problem tendencji modelu do przeuczania. Kosztowność podczas budowania lasu losowego wynika z konstrukcji poszczególnych drzew, gdzie konieczne jest wielokrotne wyznaczanie najistotniejszej zmiennej oraz optymalnego jej podziału. Tendencja do przeuczania także pochodzi z konstrukcji drzew, które uczą się podziału przestrzeni na zbiorze testowym i jeśli drzewo będzie odpowiednio duże będzie w stanie dokładnie odtworzyć zbiór treningowy. Rozwiązaniem obu tych problemów jest extreme random tree. Najważniejsza różnica pomiędzy lasem losowym a ekstremalne losowymi drzewami polega na tym, że w ERT drzewa są budowane całkowicie losowo. Dla każdego węzła decyzyjnego ERT zmienna jest wybierana losowo oraz jej podział także jest losowy. Druga różnica polega na tym, że do budowania drzew bierze się cały zbiór treningowy, a nie tylko jego podzbiór.

3.7.7. Gradient boosting i AdaBoost

Gradient boosting [80] i AdaBoost [81] są algorytmami uczenia w zespole (ang. Ensemble learning). Opierają się na założeniu, że możliwe jest zbudowanie zestawu prostych klasyfikatorów, które wspólnie zbudują złożony model. Oba algorytmy są bardzo do siebie podobne. Różnią się przede wszystkim sposobem uczenia. Oba algorytmy opierają się na technice boosting, która polega na wielokrotnym losowaniu nowej próby ze zbioru treningowego.

3.8. Podstawowe algorytmy - regresja

Regresja ma za zadnie zbudowanie modelu w najlepszy sposób przewidującego wartość zmienne ciągłej zależnej. Na proces ten można patrzeć jak na dopasowywanie krzywej do zmierzonych danych. Występuje jednak kilka różnic pomiędzy dopasowywaniem krzywej do punktów, a uczeniem maszynowym. Najważniejszą różnicą jest wielkość danych. Dane używane w uczeniu maszynowym mają znacznie większą objętość niż zmienne klasycznie dostępne w fizyce czy chemii. Ponadto wymiarowość samej przestrzeni zmiennych niezależnych jest znacznie większa niż w problemach dopasowywania krzywej. Drugą różnicą jest podejście do samych danych, w uczeniu maszynowym

stosuje się bardzo zaawansowane techniki czyszczenia zbioru danych oraz wyboru zmiennych, na których będzie budowany model. Kolejną istotną zmianą jest zmiana w procesie budowy modelu. W wielu zastosowaniach nie jest istotne jak działa model, czyli jaka jest zależność funkcyjna zmiennej zależnej od poszczególnych zmiennych. Dlatego w uczeniu maszynowym często stosowane są modele nieparametryczne i używa się ich bez próby zrozumienia zjawiska, które jest modelowane. Takie podejście nie jest spotykane w środowisku naukowym. Jednak informacje zdobyte podczas budowania modelu pozwalają jakościowo rozumieć zachodzący proces oraz często dostarczają cząstkowych informacji o naturze badanego zjawiska, a sama analiza istotności zmiennych pozwala wyciągnąć wnioski o istocie analizowanego procesu.

3.8.1. Regresja liniowa (ang. linear regression)

Najprostszą metodą przewidywania wartości zmiennej rzeczywistej jest regresja liniowa [62]. Zakłada ona liniową zależność pomiędzy zmienną zależną y, a zmiennymi niezależnymi $x_1, x_2, \ldots, x_N$:

$$y = \beta_0 + \sum_{i=1}^{N} \beta_i x_i. \tag{3.34}$$

gdzie β_i są współczynnikami dopasowania i-tej zmiennej.

Jeśli wprowadzimy następujące oznaczenia dla M-elementowego zbioru treningowego:

$$\vec{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_M \end{pmatrix}, \tag{3.35}$$

$$\hat{X} = \begin{pmatrix} 1 & x_{11} & \ldots & x_{1N} \\ 1 & x_{21} & \ldots & x_{2N} \\ \vdots & \ddots & \vdots & \\ 1 & x_{M1} & \ldots & x_{MN} \end{pmatrix}, \tag{3.36}$$

$$\vec{\beta} = \begin{pmatrix} \beta_1 \\ \beta_2 \\ \vdots \\ \beta_N \end{pmatrix}, \tag{3.37}$$

To wówczas:

$$\vec{y} = \hat{X}\vec{\beta}. \tag{3.38}$$

Można znaleźć minimum błędu średniokwadratowego poprzez obliczenie pierwszej pochodnej. Po prostych przekształceniach otrzymujemy:

$$\vec{\beta} = \left(\hat{X}^T\hat{X}\right)^{-1}\hat{X}^T Y,$$

oraz

$$\beta_0 = \bar{y} - \vec{\beta}\bar{x}.$$

Model ten mimo swojej prostoty jest często stosowany i daje zaskakująco dobre wyniki. Ponadto ze względu na swoją prostotę jest łatwy w interpretacji i jest on używany jako punkt wyjścia do bardziej skompilowanych

algorytmów. Liniowa regresja rozszerzona o regularyzację (patrz 3.10.6) jest także często używana do badania istotności zmiennych.

3.8.2. Regresja wielomianowa (ang. polynomial regression)

Rozszerzeniem liniowej regresji jest regresja wielomianowa. Zakłada ona, że zależność pomiędzy zmienną zależną i zmiennymi niezależnymi jest wielomianowa:

$$y = w(x_1, x_2, \ldots, x_N, K),$$

gdzie $w(x_1, x_2, \ldots, x_N, K)$ jest wielomianem stopnia K od zmiennych $x_1, x_2, \ldots, x_N$.

Najczęstszym sposobem interpretacji modelu wielomianowego jest traktowanie go jako rozszerzony model liniowy. Należy podkreślić, że rozszerzenie potęgowe nie wprowadza nowych zmiennych niezależnych, a jedynie dodanie dodatkowych zmiennych będących potęgą zmiennych oryginalnych. Na zbiorze zmiennych $x_1, x_2, \ldots, x_N$ dokonuje się transformacji zamieniającej je na zbiór zmiennych będących kombinacją wielomianową oryginalnych zmiennych. Na przykład dla zmiennych x_1, x_2 i $K = 3$:

$$x_1, x_2 \rightarrow x_1, x_2, x_1^2, x_2^2, x_1^3, x_2^3, x_1^2 x_2, x_1 x_2^2, x_1 x_2.$$

Przekształcenie to pozwala w dalszym ciągu używać regresji liniowej jednocześnie dopasowując do danych zależność wielomianową. Transformacja do przestrzeni wyżej wymiarowej niestety prowadzi do dużej liczby współczynników dopasowania. Dla N zmiennych i wielomianu rzędu K liczba nowych zmiennych jest rzędu N^K. Liczba ta znacznie utrudnia interpretację modelu oraz prowadzenie obliczeń. Ponadto stosowanie wielomianów wyższego rzędu wiąże się z koniecznością stosowania dużego zbioru danych.

3.8.3. Uogólnienie regresji liniowej

Analogicznie jak w przypadku regresji wielomianowej do regresji liniowej można wprowadzać dodatkowe zmienne będące transformacją oryginalnych zmiennych. Wówczas zamiast transformacji zmiennych poprzez wielomian można wykonać dowolną ich transformację i wprowadzić nowe zmienne.

$$y = F(x_1, x_2, \ldots, x_N),$$

gdzie $F(x_1, x_2, \ldots, x_N)$ jest funkcją transformującą zmienne $x_1, x_2, \ldots, x_N$.

Należy jednak podkreślić, że wprowadzanie dodatkowych zmiennych do modelu może znacząco pogorszyć jego jakość. Nowe zmienne są zależne od zmiennych oryginalnych i nie niosą dodatkowej informacji, a jedynie pozwalają na bardziej skomplikowaną zależność pomiędzy zmienną zależną a zmiennymi niezależnymi. Ponadto takie rozwiązanie może prowadzić do obciążenia używanych estymatorów i niedoszacowania błędów modelu. Dodatkowo metoda ta zwiększa wymiarowość problemu, a co za tym idzie zwiększa się minimalna wielkość zbioru treningowego oraz stosowane algorytmy stają się mniej stabilne i wolniej zbiegają. Co więcej, większa liczba dopasowywanych parametrów modelu wiąże się z koniecznością uczenia modelu na większym zbiorze danych.

3.8.4. Drzewa decyzyjne/las losowy (ang. decision tree/random forest)

Drzewa losowe (patrz 3.7.4) oraz las losowy (patrz 3.7.5) mogą być także wykorzystywane do problemu regresji. Konieczne są dwie zmiany w stosunku do klasyfikacji:

— zmiana sposobu agregacji wyniku końcowego,
— zmiana metody budowania drzew decyzyjnych.

Zmiana sposobu agregacji polega na zamianie mechanizmu głosowania na obliczanie średniej arytmetycznej. Zmiana metody budowy drzew losowych polega na zamienieniu metryki opisującej jakość modelu (np. entropia informacji) na obliczanie błędu średniokwadratowego. Zatem każdy węzeł drzewa jest budowany w ten sposób, aby wybrana zmienna niezależna oraz jej podział prowadził do minimalizacji błędu średniokwadratowego modelu.

Drzewa decyzyjne i las losowy dla regresji mają takie same własności jak drzewa decyzyjne i las losowy wykorzystywany do klasyfikacji, w szczególności mechanizm określania istotności zmiennych.

3.9. Podstawowe algorytmy - inne typy uczenia

3.9.1. Algorytm centroidów (ang. k-means clustering)

Algorytm centroidów znany także jako k-średnich (ang. K-mean) jest najpopularniejszym algorytmem klastrowania. Celem algorytmu jest znalezienie prawidłowości w nieopisanym zbiorze danych. Wynikiem tej metody jest znalezienie M podgrup/klastrów w zbiorze punktów. Algorytm wygląda następująco:

1. losowane jest M punktów w przestrzeni stanowiących początkowe środki klastrów,
2. obliczane są odległości punktów od środków klastrów,
3. każdy punkt jest przyporządkowany do klastra, którego środek jest najbliżej,
4. dla każdego klastra liczony jest nowy środek jako geometryczny środek punktów do niego przynależnych,
5. obliczana jest odległość punktów od środka klastrów, do którego zostały przyporządkowane, następnie sumowane są te odległości do wartości D,
6. jeśli $D < \epsilon$ zakończenie procedury, w przeciwnym przypadku idź do punktu 2.

Kroki 2-6 są rachunkiem samouzgodnionym prowadzącym do znalezienia klas najbardziej podobnych do siebie obiektów. Jako miarę podobieństwa bierze się odległość od przeciętnego przedstawiciela danej klasy.

Jedynym hiperparametrem algorytmu centroidów jest liczba klas M. Są dwa sposoby ustalenia ilości klas: albo posiada się wiedzę dziedzinową i informacja o liczbie klas jest znana od początku, albo należy ją optymalizować tak, jak każdy inny hiperparametr. Istnieją, także algorytmy pozwalające

uniknąć konieczności określenia liczby klas na przykład metody hierarchiczne czy DBSCAN [82].

3.9.2. Propagacja etykiet (ang. label propagation algorithm)

Algorytm propagacji etykiet [83] jest podstawowym algorytmem uczenia częściowo z nadzorem. Pozwala on na zbudowanie metody znając wartość zmiennej zależnej jedynie dla części obserwacji ze zbioru treningowego. Algorytm klasyfikacji wygląda następująco:

1. każdemu punktowi nie posiadającemu klasy jest przypisywana unikalna klasa,
2. klasy są propagowane iteracyjne pomiędzy punktami w sposób następujący: każdy punkt ma klasę najliczniej reprezentowaną wśród sąsiadów. Krok ten jest powtarzany tak długo, aż kolejna iteracja nic nie zmienia.

Algorytm jest rozszerzeniem algorytmu centroidów. Największą różnicą jest krok 1, w którym punkty nie posiadające własnej klasy są inicjowane tak, że każdy punkt ma osobną klasę.

Algorytm propagacji etykiet nie wymaga określania liczby klas jak w algorytmie centroidów ich liczba jest określana na podstawie klas w zbiorze treningowym. Podczas stosowania tej metody konieczne jest zapewnienie dobrej jakości danych wejściowych. W zbiorze treningowym każda klasa musi być reprezentowana.

3.9.3. Detekcja anomalii (ang. anomaly detection)

Detekcja anomalii jest techniką uczenia maszynowego dedykowaną do wykrywania rzadkich zdarzeń lub punktów odbiegających od reszty zbioru. Istnieje wiele metod wykrywania anomalii. Najpopularniejsze z nich to:

— jedno-klasowy SVM (ang. One-class support vector machine.) [84],
— las izolacyjny (ang. isolation forest) [85],
— metryka Cook'a (ang. Cooks metric)[86].

Dwie pierwsze metody są rozszerzeniami algorytmu SVM i lasu losowego. Jedno-klasowy SVM polega na zbudowaniu modelu wektorów nośnych na nieanomalnych obserwacjach. Następnie używa się modelu do określenia czy nowy punkt jest w grupie punktów normalnych czy nie. Jeśli wynik istotnie odbiega od wartości zaobserwowanej mówi się, że mamy do czynienia z anomalią. W algorytmie lasu izolacyjnego buduje się drzewa całkowicie losowo, losowość dotyczy zarówno zmiennych w węzłach jak i ich podziału (podobnie jak w extreme random trees (patrz 3.7.6)). Dla każdej obserwacji oblicza się średnią długość drogi od korzenia do liścia i na tej podstawie określa czy obserwacja jest anormalna.

Metryka Cook'a pozwala na detekcję anomalii w zbiorze. Budowany jest szereg modeli zredukowanych poprzez usunięcie ze zbioru treningowego jednej obserwacji. Następnie obliczana jest różnica pomiędzy przewidywaniami zredukowanego modelu Y_i^k a modelu pełnego Y_i:

$$D_k = \frac{\sum_i (Y_i^k - Y_i)^2}{p \cdot MSE(Y^k)}, \tag{3.39}$$

gdzie sumowanie przebiega po całym zbiorze treningowym, a p jest liczbą parametrów użytych do budowy modelu.

Duża wartość metryki oznacza, że punkt k ma bardzo duży wpływ na model i najprawdopodobniej istotnie odbiega od pozostałych, więc można go traktować jako anomalię. Metryka Cook jest wyznaczana dla każdej obserwacji. Wiąże się wiec to z ogromnym kosztem obliczeniowym.

3.10. Budowa zmiennych

3.10.1. Metody budowania zmiennych

Budowa modelu opartego na algorytmach uczenia maszynowego składa się z trzech część: wyboru algorytmu, doboru hiperparametrów oraz wyborze zmiennych niezależnych opisujących obserwacje. Wszystkie ta elementy są ze sobą powiązane i żaden nie może być wykonany bez pozostałych. Wybór algorytmu nie jest trudny. Istnieje ograniczona liczba algorytmów pozwalających na pracę z zadanym problemem i już niewielkie doświadczenie badacza pozwala na właściwy wybór grupy potencjalnie obiecujących algorytmów. Mimo, że sam proces uczenia może być czasochłonny i może wymagać dużej mocy obliczeniowej to jednak jest to zagadnienie czysto techniczne. Bardziej złożonym procesem jest optymalizacja hiperparametrów modelu. Zwiększa one jedynie kosztowność samego procesu budowania modelu, ale najczęściej nie jest to zagadnienie trudne. Wybór właściwych zmiennych lub ich wytworzenie jest znacznie bardziej skompilowane. W ogólności nie jest możliwe napisanie algorytmu tworzenia zmiennych niosących informacje o zmiennej zależnej, są jednak techniki udoskonalania oraz redukcji ilości zmiennych używanych podczas budowaniu modelu.

3.10.2. Rodzaje zmiennych i ich preprocesing

Zmienne można podzielić ze względu na typ danych, które zawierają:

— zmienne ciągłe,
— zmienne kategoryczne,
— zmienne opisowe.

Zmienne ciągłe to zmienne rzeczywiste z zadanego przedziału. Zmienne kategoryczne to albo zmienne dyskretne albo zmienne opisywane przez skończona liczbę klas. Zmienne opisowe to zmienne posiadające dużo danych w formacie tekstowym. Także bardziej złożone formaty inputów (jak obrazki) można przekształcić na zestaw zmiennych o zdefiniowanych powyżej typach.

Preprocesing danych polega na przygotowaniu ich do pracy z modelem uczenia maszynowego. Dla zmiennych ciągłych wykonuje się: normalizację, standaryzację albo dyskretyzację. Normalizacja polega na zamianie zakresu zmiennej na przedział $[0, 1\rangle$. Standaryzacja oznacza, transformację przesuwającą średnią wartość zmiennej do wartości 0 oraz przeskalowanie wariancję zmiennej do wartości 1. Dyskretyzacja polega na podziale zakresu zmien-

nej na k przedziałów i do opisu wartości zmiennej używa się identyfikatora przedziału.

Zmienne kategoryczne przepisuje się używając techniki etykietowania lub kodowania jednorazowego. Etykietowanie polega na przypisaniu każdej kategorii wartości liczbowej. Metody tej używa się najczęściej jeśli kolejne klasy mogą zostać przetłumaczone na intensywność obserwacji, na przykład poziom bólu pacjenta. Podczas kodowania jednorazowego tworzony jest wektor o długości równej ilości możliwych wartości zmiennej niezależnej. Jeśli zmienna przyjmuje wartość z danej kategorii to odpowiednie pole w wektorze przyjmuje wartość jeden, a pozostałe wartość zero.

Zmienne opisowe wymagają narzędzi przetwarzania języka naturalnego (ang. natural language processing - NLP). Tematyka ta przekracza zakres tej pracy oraz w analizowanych zbiorach danych nie ma tego typu zmiennych, preprocesing dla tego typu zmiennych nie będzie omawiany.

Preprocesing danych jest często wykonywany jednocześnie z czyszczeniem danych (ang. data cleaning).

3.10.3. Czyszczenie zmiennych (ang. data cleaning)

Praca z dowolnym zbiorem danych wymaga, aby dane były dobrej jakości. W różnych działach uczenia maszynowego termin ten oznacza różne warunki. W tej pracy ograniczymy się do przypadków regresji i klasyfikacji. Proces poprawiania jakości danych określa się jako czyszczenie danych.

Zbiory danych bardzo często posiadają dużo błędnych lub niekompletnych wpisów. Podstawowym problemem jest brak części danych we wpisie (brakuje wartości jednej lub kilku zmiennych niezależnych). Istnieje kilka technik radzenia sobie z tym zjawiskiem:

— usunięcie wpisu,
— uzupełnienie brakujących pól wartością średnią brakującej zmiennej,
— uzupełnienie brakujących pól wartością średnią brakującej zmiennej przy czym uśrednienie odbywa się w ramach danej klasy,
— wprowadzenie dodatkowej wartości przekazującej modelowi informację o braku danych.

Najczęściej usuwa się niekompletne wpisy ze zbioru treningowego.

Następnym problemem jest występowanie wartości "NaN" czyli wartość nieokreślona. Technika usuwania tego typu błędnych wpisów jest taka sama jak dla braku danych. Najczęściej wpis z wartością "NaN" usuwa się lub wpis taki traktuje się jako specjalną wartość zmiennej.

Ostatnią grupą problemów z wpisami jest błędny format danych, wielkości z poza zakresu lub inne problemy z zapisem obserwacji. Takie wpisy są zasadniczo albo usuwane albo ręcznie poprawiane, aby pasowały do reszty danych.

Podczas czyszczenia danych usuwa się także wszystkie duplikaty wpisów oraz zmienne nieniosące żadnej informacji, czyli zawierające tylko jedną wartość lub niepowtarzalną wartość dla każdej obserwacji. Często podczas tego kroku wykorzystuje się podstawowe techniki analizy danych oraz wiedzę

dziedzinową do usunięcia zmiennych nieniosących żadnej informacji, jak na przykład informacja o osobie wykonującej pomiar.

Sprawdzane są także jednostki używane dla poszczególnych zmiennych i uzgadniany jest jednolity zbór jednostek wykorzystywany przez wszystkie zmienne.

3.10.4. Korelacje pomiędzy zmiennymi

Podczas budowania zestawu zmiennych sprawdza się szereg ich parametrów. Pierwszą sprawdzaną cechą jest wzajemna korelacja. Zmienne mocno skorelowane mogą być zastępowane przez pojedynczą zmienną, gdyż informacja o zmiennej zależnej jest powtórzona w obu zmiennych. Przykładem takich zmiennych mogą być prędkość i czas przebycia zadanej drogi. Do określania korelacji pomiędzy zmiennymi używa się najczęściej współczynnika korelacji r-Pearsona. Najpopularniejszą reprezentacją korelacji pomiędzy zmiennymi jest macierz korelacji, która zawiera korelację pomiędzy wszystkimi parami zmiennymi w zbiorze treningowym (najczęściej umieszcza się w niej zarówno zmienne zależne jak i niezależne).

Podkreślić należy dwie cechy tej metody. Po pierwsze współczynnik korelacji r-Pearsona opisuje liniową korelację pomiędzy zmiennymi, więc nieliniowe zależności pomiędzy zmiennymi są źle opisywane lub nie są opisywane. Jeśli zależność pomiędzy zmiennymi jest nieliniowa współczynnik korelacji będzie pokazywał znacznie mniejszą zależność niż jest w rzeczywistości. Drugą cechą jest brak zależności pomiędzy brakiem korelacji między zmiennymi, a ich istotnością dla modelu. Zmienne silnie skorelowane są nieistotne dla modelu, ale brak korelacji nie implikuje istotności zmiennych. Ponadto, częstą praktyką jest używanie do budowania modelu zmiennych skorelowanych ze względu na posiadaną wiedzę dziedzinową lub chęć przetestowania hipotezy.

Należy także pamiętać, że współczynnik korelacji nie daje wglądu w prawdziwą naturę problemu, a jedynie pewne informacje o zależności pomiędzy zaobserwowanymi liczbami. Obliczany współczynnik korelacji odnosi się jedynie do zależności wewnątrz posiadanego zbioru danych, a nie całej populacji. Można łatwo tak wybrać punkty ze zbioru danych, aby dwie skorelowane zmienne miały korelację o przeciwnym znaku niż w rzeczywistości. Podczas analizy danych z zewnętrznego zbioru danych nie mamy żadnej kontroli nad próbkowaniem, więc mogą pojawić się w nim efekty będące artefaktami sposobu zbierania danych.

3.10.5. Metody wyboru istotnych zmiennych

Algorytmy uczenia maszynowego są stosowane dla przestrzeni o wysokiej wymiarowości, co w połączeniu z częstym stosowaniem modeli nieparametrycznych prowadzi do dobrze działających modeli. Z drugiej strony modele te nie wnoszą nic do głębszego zrozumienia problemu. O ile takie podejście nie jest problemem w zastosowaniach biznesowych, gdzie ważniejsze od zrozumienia jest dokładne przewidzenie przebiegu zjawiska, to w fizyce teoretycznej równie ważne jest wskazanie związku przyczynowo-skutkowego. Jeśli możliwa jest praca z prostym algorytmem parametrycznym, jak regresja liniowa lub wieloliniowa, możliwe jest pełne zrozumienie istoty badanego zjawiska. Jed-

nak jeśli jedynym dobrze działającym modelem jest model nieparametryczny, zrozumienie zjawiska na podstawie takiego modelu jest bardzo trudne lub wręcz niemożliwe. Wówczas analiza istotności zmiennych dla takiego modelu pozwala na wyciągnięcie pewnych informacji o istocie badanego zjawiska.

Najczęściej stosowane metryki dla wyboru istotnych zmiennych to:

— współczynnik istotności zmiennych z lasu losowego,
— test statystyczny p-wartość (ang. p-value).

Las losowy pozwala na określenie istotności zmiennej dla modelu ze względu na jej położenie w węzłach drzew losowych. Informacja ta stanowi ranking istotności zmiennych dla modelu. Jednak wartość liczbowa nie ma znaczenia i liczby te nie mogą być porównywalne pomiędzy modelami. Istotność zmiennej z lasu losowego pozwala określić kolejność istotności zmiennych oraz stwierdzić w sposób jakościowy, które zmienne można usunąć z modelu. Jeśli w budowaniu modelu używa się zmiennych skorelowanych, algorytm lasu losowego jako istotną zmienną może opisać jedną lub drugą zmienną albo podzielić wartość współczynnika istotności pomiędzy obie zmienne. Ponadto lasy losowe o małej liczbie drzew wykazują dużą niestabilność w zwracanych wartościach liczbowych współczynnika istotności zmiennych.

Drugą metodą badania istotności zmiennych jest rozważenie hipotezy o nieistotności zmiennej dla modelu i wykonanie testu statystycznego dla tej hipotezy. Określana jest wówczas p-wartość i na jej podstawie dokonywany jest wybór o istotności zmiennej.

3.10.6. Metody redukcji wymiarowości przestrzeni cech

Techniki wyznaczania istotności zmiennych prowadzą do upraszczania modelu przez usuwanie zmiennych nieistotnych lub mało istotnych dla modelu. Modele zawierające mniej zmiennych posiadają wiele zalet. Po pierwsze, są dużo łatwiejsze do budowania i stabilniejsze podczas procesu uczenia. Po drugie, działają znacznie szybciej niż modele zależne od wielu zmiennych. Po trzecie, ułatwiony jest proces zbierania obserwacji, gdyż trzeba zebrać mniej danych, co także zmniejsza koszt archiwizacji i przechowywania danych. Po czwarte, modele takie mogą być łatwiejsze do interpretacji.

W literaturze można spotkać wiele technik redukcji wymiarowości przestrzeni cech. Najpopularniejsze algorytmy to:

— analiza głównych składowych (ang. principal component analysis - PCA) [87],
— regularyzacja (ang. regularization) [70],
— t-SNE [88],
— analiza niezależności zmiennych [70],
— autoencoder [89].

Analiza głównych składowych jest jedną z najbardziej uniwersalnych technik redukcji wymiarowości. Pierwszym krokiem jest wyznaczanie macierzy korelacji $\hat{M}$, której elementami są współczynniki korelacji r-Pearsona pomiędzy wartościami zmiennych niezależnych:

$$\hat{M}_{i,j} = r(x_i, x_j). \tag{3.40}$$

Następnie rozwiązywane jest zagadnienie własne dla macierzy $\hat{M}$. Z otrzymanych wektorów własnych wybiera się k o największym module wartości własnej. Stanowią one nową bazę zmiennych, na którą są rzutowane oryginalne zmienne niezależne. PCA wymaga, aby zmienne przed wyznaczeniem macierzy $\hat{M}$ były ustandaryzowane.

Regularyzacja polega na wymuszeniu na modelu podczas procesu uczenia zmniejszenia wartości parametrów dopasowania. Do funkcji kosztu E dodaje się człon regularyzacyjny $R(\beta_1, \beta_2, \ldots, \beta_M)$. Regularyzację najczęściej stosuje się razem z regresją liniową, wielomianową lub regresją logistyczną. Możliwe jest także zastosowanie tej techniki w innych modelach parametrycznych. Jeśli jako funkcja kosztu używany jest błąd średniokwadratowy, funkcja kosztu może zostać rozszerzona do postaci:

$$E = \frac{1}{N}\sum_{i=1}^{N}(\hat{y}_i - y_i)^2 + \lambda R(\beta_1, \beta_2, \ldots, \beta_N). \tag{3.41}$$

Parametr λ jest nieujemnym współczynnikiem opisującym siłę regularyzacji, a jego wielkość odpowiada ze siłę wytłumiania współczynników modelu. Jako człon regularyzacyjny stosuje się [62]:

— regresję Ridge z normą L1:

$$R(\beta_1, \beta_2, \ldots, \beta_N) = \sum_{j=1}^{M}|\beta_j|, \tag{3.42}$$

— LASSO z normą L2:

$$R(\beta_1, \beta_2, \ldots, \beta_N) = \sum_{j=1}^{M}|\beta_j|^2, \tag{3.43}$$

— 'Elastic net' z normami L1 i L2:

$$R(\beta_1, \beta_2, \ldots, \beta_N) = \lambda_1\sum_{j=1}^{M}|\beta_j| + \lambda_2\sum_{j=1}^{M}|\beta_j|^2. \tag{3.44}$$

Zastosowanie normy L1 prowadzi do zerowania większości współczynników β_i. Efekt ten zależy od wartości parametru λ, ale najczęściej już małe jego wartości prowadzą do zerowania większości współczynników. Zachowanie takie jest często wykorzystywane do redukcji wymiarowości problemu oraz detekcji istotnych zmiennych. Metryka L1 jest niestabilna i może łatwo prowadzić do bardzo źle działających modeli.

Norma L2 nie zeruje współczynników β_i ale prowadzi do znacznego zmniejszenia ich wartości (do wielkości rzędu 10^{-6}), jest jednak znacznie stabilniejsze od normy L1 oraz zazwyczaj model daje lepsze wyniki.

Połączeniem obu tych technik jest 'Elastic net', gdzie zastosowanie normy L1 ma wymusić zerowanie współczynników podczas, gdy norma L2 ma zapewnić stabilność metody.

Możliwe jest stosowanie także innych technik jak autoencoder czy t-SNE. Techniki te są stosowane głównie do danych będących obrazkami lub wyłącznie podczas przygotowywania wykresów.

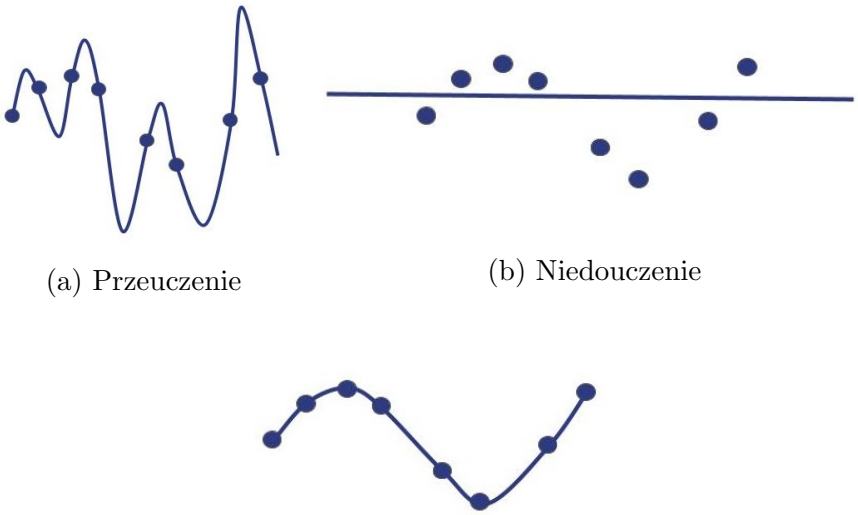
3.11. Techniki wyboru algorytmu

Budowane modelu opartego na metodach uczenia maszynowego wymaga zachowania szczególnej ostrożności podczas wyboru stosowanego algorytmu w kontekście liczby dostępnych danych i liczby zmiennych niezależnych. Każdy algorytm posiada inną liczbę dopasowywanych parametrów i inną zależność ilości tych parametrów od liczby zmiennych niezależnych. Nie chodzi tu o hiperparametry, których dobór wykonuje się najczęściej poprzez testowanie krzyżowe, ale o parametry optymalizowane podczas uczenia. Każdy kolejny dopasowywany parametr zwiększa zapotrzebowanie na dane w zbiorze treningowym. Techniki budowania zmiennych (w tym wyboru istotnych zmiennych) pozwalają na redukcję wymiarowości problemu, jednak nie można ich stosować w oderwaniu od procesu wyboru algorytmu.

Uczenie modelu polega na dopasowaniu parametrów algorytmu do obserwacji. Procedura uczenia może prowadzić do dwóch skrajnych sytuacji: model może być przeuczony (nadmierne dopasowanie, ang. overfitting) lub niedouczony (niedopasowanie, ang. underfitting). O przeuczeniu mówimy wówczas, gdy dopasowana funkcja zbyt dokładnie oddaje obserwacje. Wówczas model nie potrafi uogólniać i dokonywać wiarygodnego przewidywania, jedynie dokładnie odtwarzać zbiór treningowy. Często w takim przypadku mówi się o pamięci modelu lub zbyt dużej elastyczności dopasowywanej funkcji. Model jest niedouczony, gdy dopasowywana funkcja jest zbyt prosta i nie oddaje szczegółów w zbiorze treningowym lub, gdy błąd dopasowania parametrów modelu jest duży. Najprostszym przykładem obu zjawisk jest dopasowywanie wielomianami do obserwacji będących parabolą z szumem losowym. Przykładem niedouczenia modelu jest próba dopasowania prostej do obserwacji (rysunek 3.10a), a przeuczenia modelu próba dopasowania wielomianami stopnia dwunastego (rysunek 3.10b).

Podczas wyboru algorytmu i zestawu zmiennych najczęściej używa się testowania krzyżowego do obliczania metryk porównywania modeli oraz istotności zmiennych. Podczas pracy z tą techniką należy pamiętać o dwóch szczególnych przypadkach:

— procedura testowania krzyżowego musi obejmować całą procedurę budowania modelu od skalowania zmiennych, przez wybór zmiennych, po budowanie modelu i jego testowania. Często stosowanie testowania krzyżowego ogranicza się jedynie do samego procesu uczenia i walidacji modelu, czyli obliczania metryk określających jakość modelu. Prowadzi to do przeuczenia modelu oraz małej wiarygodności wyników. Poprawną procedurą jest podział zbioru danych na fragmenty i wielokrotne powtarzanie całej procedury budowania modelu włącznie z wyborem istotnych zmiennych oraz skalowaniem, które musi być wykonywane na podzbiorze treningowym.
— podczas testowania algorytmów opartych na lesie losowym lub technikach wielokrotnego próbkowania nie ma potrzeby wykonywania testowania krzyżowego. Podczas uczenia modelu (budowania pojedynczego drzewa) zbiór treningowy jest losowany z całego zbioru treningowego zatem część oryginalnego zbioru nie jest wykorzystywana. Nieużyte obserwacje



(a) Przeuczenie

(b) Niedouczenie

(c) Poprawne dopasowanie

Rysunek 3.10: Schematyczna ilustracja problemu dopasowywania modeli: 3.10a model przeuczony, 3.10b model niedouczony, 3.10c model poprawny.

mogą posłużyć jako zbór testowy, na nim obliczana jest metryka oob (ang. out-of-bag). Pozwala ona na znaczne przyśpieszenie procesu testowania modeli poprzez uniknięcie wielokrotnego uczenia podczas testowania krzyżowego.

Przedstawione algorytmy i techniki mogą być stosowane dla dowolnego typu problemów. Istnieją, także metody popularne tylko w konkretnych zastosowaniach. Stosowanie sztucznej inteligencji w fizyce materii skondensowanej również wymusza stosowanie specyficznych metod, szczególnie istotne jest reprezentowanie geometrii układów molekularnych czy krystalicznych oraz położeń atomów w sposób odpowiedni dla algorytmów uczenia maszynowego.

Rozdział 4

Techniki charakterystyczne dla fizyki materii skondensowanej

4.1. Stosowane zestawy zmiennych

4.1.1. Rodzaje zmiennych

Zmienne używane do budowanie modeli opartych na algorytmach uczenia maszynowego w fizyce materii skondensowanej można podzielić na trzy kategorie:

— zmienne eksperymentalne - pochodzą z pomiaru i obejmują dowolne własności, jak poziom domieszkowania czy wartość przerwy energetycznej,
— zmienne obliczeniowe - otrzymywane w wyniku symulacji numerycznych, jak gęstość elektronowa czy energia,
— zmienne położeniowe - są to zmienne budowane na podstawie położeń atomów w układzie, które mogą być znane na przykład z dyfrakcji rentgenowskiej (ang. X-ray diffraction) lub symulacji numerycznych. Stanowią one szczególny przypadek zmiennych obliczeniowych. Z racji na ich istotność oraz różnorodność należy je wyodrębnić jako oddzielną kategorię.

Dla budowania modelu nie ma znaczenia jaki rodzaj zmiennych będzie użyty (o ile niosą one niezbędną informacje). Dla modelu matematycznego zmienne niezależne są wektorem wartości, a ich pochodzenie czy znaczenie poszczególnych pól nie wpływa na zachowanie algorytmu. Natomiast cechy te są bardzo istotne dla fizycznej interpretacji wyników oraz możliwości zastosowania modelu. Jeśli model ma bardzo dobrą dokładność ale wymaga zmiennych, których uzyskanie jest kosztowne i czasochłonne, prawdopodobnie taki model będzie w praktyce mało użyteczny. Podczas budowania modelu łatwo wpaść w pułapkę, kiedy stosowane zmienne wymagają przeprowadzenia eksperymentu lub obliczeń, których wynikiem jest zmienna zależna.

Co więcej, często konieczne jest korzystanie z wielu baz danych lub danych literaturowych i trudnością jest znalezienie zestawu zmiennych występujących we wszystkich źródłach. Zasadniczą zaletą danych tego typu jest możliwość użycia tych danych wprost do budowy modelu bez konieczności ich dodatkowych przekształceń.

Istnieje wiele otwartych baz danych pozwalających na uzyskanie wyników obliczeń bez konieczności prowadzenia długich symulacji. Najbardziej popularne z nich to:

— The Materials Project [90, 91],
— OMDB [92],

— C2DB [93],
— aNANt [94],
— OCSEP [95],
— Materials Cloud [96].

Problem z tak uzyskanymi danymi jest podobny jak z danymi eksperymentalnymi. Konieczne jest uzgodnienie zestawu zmiennych pomiędzy wynikami z różnych baz danych. Często dane z jednej bazy nie są w pełni zgodne. Co więcej, często obliczenia są wykonywane na różnym poziomie teorii i z różną precyzją. Mimo tych niedogodności uzyskanie danych obliczeniowych jest znacznie mniej kłopotliwe niż pozyskanie danych eksperymentalnych.

Znacznie trudniej pracuje się ze zmiennymi położeniowymi. Współrzędne atomów nie są dobrymi zmiennymi do budowania modelu, jednak są najwygodniejsze ze względu na łatwość dostępu i niezależność od zewnętrznych baz danych. Ponadto pozwalają na badania bez konieczności wykonywania długotrwałych i kosztowych obliczeń czy doświadczeń. Dodatkowo łatwo jest wygenerować położenia dla dowolnego związku i na ich podstawie przewidywać własności związku. Z tego powodu skupimy się na sposobach opisywania położeń atomów na potrzeby algorytmów uczenia maszynowego.

4.1.2. Własności zmiennych

Aby zmienne stosowane w modelu dobrze opisywały układ fizyczny muszą posiadać szereg właściwości:

— Liczba zmiennych nie może zależeć od wielkości układu. Jest to fundamentalna własność zestawu zmiennych opisujących układ atomowy. Jest ona konieczna, aby móc zbudować uniwersalny model dla wielu różnorodnych systemów. Algorytmy uczenia maszynowego wymagają takiej samej liczby zmiennych niezależnych dla każdej obserwacji ze zbioru treningowego.
— Niezmienniczość ze względu na permutacje kolejności atomów: oprócz stałej liczby zmiennych konieczne jest, aby kolejność atomów w układzie nie zmieniała wartości zmiennych niezależnych.
— Jednoznaczność: zmienne muszą jednoznacznie opisywać układ fizyczny i być unikalne dla układu. Niepożądane jest, aby dwa różne układy fizyczne były reprezentowane przez te same wartości zmiennych, może to prowadzić do niekontrolowanych artefaktów w zbiorze treningowym i sztucznego zwielokrotnienia obserwacji w bazie danych. Czego wynikiem może być przeszacowanie dokładności modelu, gdyż różne obserwacje w zbiorze treningowym i testowym mogą mieć te same wartości zmiennych niezależnych.
— Zachowanie transformacji oryginalnego układu: konieczne jest aby zmienne odzwierciedlały własności układu. Zatem jeśli przykładowo układ posiada jakąś symetrię, to tą samą symetrię musi zachowywać układ zmiennych niezależnych.
— Łatwość obliczenia: jest to wymóg bardziej praktyczny niż formalny. Zmienne muszą być łatwe i szybkie do policzenia, tak aby koszt wyznaczenia zmiennych niezależnych nie był porównywalny z wyznaczeniem zmiennej zależnej tradycyjnymi metodami.

— Rozszerzalność: zmienne muszą być łatwo rozszerzalne na układy nie wchodzące w skład zbioru treningowego.

Wymienione własności są konieczne dla poprawnego funkcjonowania modelu opartego na algorytmach uczenia maszynowego. Istnieją jednak techniki oparte na głębokich sieciach neuronowych (DNN - ang. deep neural networks) pozwalające na korzystanie ze zmiennych nie posiadających jednej lub kilku z tych cech. Można więc nauczyć sieć neuronową, że kolejność atomów w układzie nie ma znaczenia dla jego własności. Innym rozwiązaniem jest zastosowanie sieci rekurencyjnych [97] lub sieci grafowych [98] pozwalających pracować ze zmienną liczbą zmiennych niezależnych oraz raz zmianami ich kolejności. Podejście takie znacznie ułatwia pracę i jest jedną z największych zalet DNN, wymaga jednak dziesiątków tysięcy obserwacji w zbiorze treningowym oraz dużego doświadczenia w budowaniu sieci neuronowych. W pracy z danymi fizycznymi zazwyczaj konieczne jest korzystanie z małych zbiorów danych, więc zastosowanie głębokich sieci neuronowych nie jest możliwe.

4.1.3. Najpopularniejsze zmienne położeniowe

Najpopularniejszymi zmiennymi położeniowymi są:

— Macierz Kulombowska (ang. Coulomb Matrix) [99],
— Macierz Ewalda (ang. Ewald Matrix) [100],
— Macierz sinusoid (ang. Sine Matrix) [99],
— Funkcje symetrii (ang. Symmetry functions) [101],
— Ważone funkcje symetrii (ang. Weight symmetry functions) [102],
— Odcisk atomowy (ang. Atomic fingerprints) [103],
— Wielociałowa reprezentacja tensorowa (ang. Many-body tensor representation) [104],
— Gładkie nałożenie położeń atomów (ang. Smooth Overlap of Atomic Positions) [105],
— Momenty rozkładów.

Macierz Kulombowska

Zestaw zmiennych opisywany przez Macierz Kulombowską został zbudowany tak, aby opisywać najsilniejsze oddziaływania pomiędzy molekułami - oddziaływania elektrostatyczne. Pierwszym krokiem budowania zmiennych jest wyznaczenie macierzy oddziaływań kulombowskich $\hat{M}$. Jej elementy obliczane są następująco:

$$M_{ij} = \begin{cases} \frac{1}{2} Z_i^{2.4} & \text{dla } i = j \\ \frac{Z_i Z_j}{R_{ij}} & \text{dla } i \neq j \end{cases} \tag{4.1}$$

gdzie Z_i jest ładunkiem jądra i-tego atomu w układzie, a R_{ij} jest odległością pomiędzy atomami i oraz j. Taka definicja macierzy $\hat{M}$ sprawia, że wielkość macierzy, a przez to liczba zmiennych opisujących układ jest zmienna wraz ze zmianą liczby atomów w układzie. Aby usunąć tą niedogodność macierz $\hat{M}$ jest rozszerzana. Znajdowany jest największy rozważany układ (w zbiorze treningowym) i jego wielkość determinuje wymiar $\hat{M}$, dla mniejszych systemów macierz $\hat{M}$ jest dopełniania zerami. Pozostaje jeszcze problem wrażliwości zmiennych na permutacje kolejności atomów. Aby uzyskać tą cechę

macierz $\hat{M}$ jest sortowana. Dla każdego wiersza obliczana jest suma elementów, następnie wiersze i kolumny są sortowane według rosnącej ich sumy. Macierz Kulombowska jest wygodną reprezentacją układu fizycznego oraz mało wymagającą obliczeniowo. Dodatkowo mechanizm reprezentacji wielu typów pierwiastków w układzie oraz brak zależności liczby zmiennych od liczby pierwiastków w układzie wynika z definicji zmiennych. Jednak dla zbiorów treningowych zawierających duże układy oraz zbiorów o dużej rozpiętości liczby atomów w układach, jest mało wydajna. Ponadto, liczba zmiennych koniecznych do opisania układu jest proporcjonalna do kwadratu liczby atomów, co wymusza posiadanie ogromnych zbiorów treningowych dla dużych układów. Należy także podkreślić, że ta reprezentacja została stworzona dla molekuł i nie uwzględnia periodyczności układów krystalicznych.

Macierz Ewalda

Macierz Ewalda jest rozszerzeniem Macierzy Kulombowskiej na układy periodyczne. Oddziaływanie kulombowskie w elementach macierzy $\hat{M}$ zostało zastąpione przez sumowanie Ewalda [106]. Elementy macierzowe mają postać:

$$M_{ij} = \begin{cases} \frac{1}{2} Z_i^{2.4} & \text{dla } i = j \\ x_{ij}^k + x_{ij}^d + x_{ij}^0 & \text{dla } i \neq j \end{cases} \tag{4.2}$$

gdzie poszczególne wkłady do elementów M_{ij} zdefiniowane są następująco:

$$x_{ij}^k = Z_i Z_j \sum_L \frac{erfc(\alpha |r_i - r_j + L|)}{|r_i - r_j + L|},$$

$$x_{ij}^d = \frac{Z_i Z_j}{\pi V} \sum_G \frac{e^{-|G|^2/(2a)^2}}{|G|^2} \cos\left(G \cdot (r_i - r_j)\right),$$

$$x_{ij}^0 = -\frac{\alpha}{\sqrt{\pi}}(Z_i^2 + Z_j^2) - \frac{\pi}{2V\alpha^2}(Z_i + Z_j)^2.$$

gdzie r_i jest położeniem i-tego atomu, $\sum_L$ i $\sum_G$ przebiegają po całej przestrzeni rzeczywistej i Fouriera, a V to objętość komórki elementarnej,
α jest parametrem metody.
Dalsze kroki są takie same jak dla Macierzy Kulombowskiej. Obliczenie wartości zmiennych dla Macierzy Ewalda jest bardziej kosztowne niż dla Macierzy Kulombowskiej ale uwzględniona jest periodyczność układu.

Macierz sinusoid

Macierz sinusoid jest uproszczeniem Macierzy Ewalda w celu poprawy wydajności wyznaczania zmiennych niezależnych. Macierz $\hat{M}$ jest zdefiniowana jako:

$$M_{ij} = \begin{cases} \frac{1}{2} Z_i^{2.4} & \text{dla } i = j \\ \frac{Z_i Z_j}{|B * \sum_{k=(x,y,z)} \vec{e}_k (\pi B^{-1} \cdot (\vec{r}_i - \vec{r}_j)|} & \text{dla } i \neq j \end{cases}, \tag{4.3}$$

gdzie B to macierz zbudowana z wektorów sieci, a $\vec{e}_k$ jest wektorem jednostkowym w kierunku k.
Reprezentacja ta pozwala na uwzględnienie periodyczności w układzie przy jednoczesnym ograniczeniu kosztu obliczeniowego.

Funkcje symetrii

Funkcje symetrii są rodziną funkcji używanych do opisywania układów molekularnych. W skład tych zmiennych wchodzi kilka grup funkcji: $G_j^1, G_j^2, G_j^3, G_j^4$. Do ich zdefiniowania konieczna jest funkcja odcięcia f_c obliczana jako:

$$f_c(r) = \begin{cases} \frac{1}{2}\left(\cos\left(\frac{\pi r}{R_c}\right)+1\right) & \text{dla } r \leqslant R_c \\ 0 & \text{dla } r > R_c \end{cases}, \tag{4.4}$$

lub:

$$f_c(r) = \begin{cases} \operatorname{tgh}^3\left(1-\frac{r}{R_c}\right) & \text{dla } r \leqslant R_c \\ 0 & \text{dla } r > R_c \end{cases}, \tag{4.5}$$

gdzie R_c jest promieniem odcięcia powyżej, którego nie rozważamy oddziaływania pomiędzy atomami.

Wówczas funkcje symetrii mogą zostać zdefiniowane następująco:

$$G_j^1 = \sum_{i=1}^{N} f_c(R_{ij}), \tag{4.6}$$

$$G_j^2 = \sum_{i=1}^{N} e^{-\eta(R_{ij}-R_s)^2} f_c(R_{ij}), \tag{4.7}$$

$$G_j^3 = 2^{1-\zeta} \sum_{k\neq i} \sum_{l\neq i,k} [(1+\lambda\cos(\theta_{kil}))^{\zeta} e^{-\eta(R_{ik}^2+R_{il}^2+R_{kl}^2)} f_c(R_{ik}) f_c(R_{il}) f_c(R_{kl})], \tag{4.8}$$

$$G_j^4 = 2^{1-\zeta} \sum_{k\neq i} \sum_{l\neq i,k} [(1+\lambda\cos(\theta_{kil}))^{\zeta} e^{-\eta(R_{ik}^2+R_{il}^2)} f_c(R_{ik}) f_c(R_{il})], \tag{4.9}$$

gdzie R_{ij} jest odległością pomiędzy parą atomów $(i,\ j)$, R_s jest to parametr rodziny funkcji związanym z centrowaniem funkcji, a η, ζ, λ są parametrami funkcji z danej rodziny.

Możliwe jest zdefiniowanie także funkcje symetrii wyższych rzędów. Jednak w praktyce stosuje się jedynie G_i^2 i G_i^4. Wynika to z faktu, że oddziaływania w molekułach modeluje się głównie jako funkcję odległości pomiędzy atomami.

Funkcje symetrii posiadają wszystkie własności jakie są wymagane od zestawu zmiennych niezależnych. Mają jednak kilka poważnych wad. Po pierwsze, zależą od dużej liczby parametrów, których wartości należy zoptymalizować. Po drugie, konieczne jest stosowanie bardzo wielu funkcji bazowych. Przyjmuje się, że dla układu zawierającego jeden pierwiastek potrzeba kilkaset funkcji, aby poprawnie opisać układ. Po trzecie fFunkcje symetrii muszą być osobno dopasowywane dla każdej pary pierwiastków występujących w układzie, co powoduje, że dla zbiorów treningowych zawierających układy posiadające wiele różnych pierwiastków stosowanie funkcji symetrii jest niemożliwe ze względu na ich liczebność.

Ważone funkcje symetrii

Rozszerzeniem funkcji symetrii'na układy wielopierwiastkowe są ważone funkcje symetrii. Powstają one przez dodanie wagi do pierwotnych funkcji, opisujących różne pierwiastki w układzie:

$$G_j^2 = \sum_{i=1}^{N} g(Z_i) e^{-\eta(R_{ij}-R_s)^2} f_c(R_i j), \tag{4.10}$$

$$G_j^4 = 2^{1-\zeta} \sum_{k \neq i} \sum_{l \neq i,k} h(Z_k, Z_l)[(1 + \lambda \cos(\theta_{kil}))^{\zeta} e^{-\eta(R_{ik}^2 + R_{il}^2)} f_c(R_{ik}) f_c(R_{il})], \tag{4.11}$$

gdzie Z_i to liczba atomowa i-tego atomu, $g(Z_i)$ to waga związaną z liczbą atomową i-tego atomu, może być zdefiniowane jako $g(Z_j) = Z_j$ [102], $h(Z_i, Z_j)$ jest wagą związaną z liczbami atomowymi i-tego i j-tego atomu, może być zdefiniowane jako $h(Z_i, Z_j) = Z_i Z_j$.

Zastosowanie wag pozwala na wyeliminowanie konieczności budowania zestawu funkcji symetrii dla każdej pary pierwiastków występujących w zbiorze treningowym. Prowadzi to zatem do budowania prostszych i bardziej przejrzystych modeli. Tak znaczne ograniczenie liczby zmiennych istotnie skraca także czas uczenia i działania modelu niezależnie od używanego algorytmu.

Odciski atomowe

Odciski atomowe mogą być definiowane jako rodziny trzech funkcji $S_{i,k}$, $V_{i,k}$ i $T_{i,k}$:

$$S_{i,k} = c_k \sum_{i \neq j} \exp\left(-\frac{1}{2}\left(\frac{r_{ij}}{\sigma_k}\right)^2\right) f_c(r_{ij}), \tag{4.12}$$

$$V_{i,k}^{\alpha} = c_k \sum_{i \neq j} \frac{r_{ij}^{\alpha}}{r_{ij}} \exp\left(-\frac{1}{2}\left(\frac{r_{ij}}{\sigma_k}\right)^2\right) f_c(r_{ij}), \tag{4.13}$$

$$T_{i,j}^{\alpha,\beta} = c_k \sum_{i \neq j} \frac{r_{ik}^{\alpha} r_{ij}^{\beta}}{r_{ij}^2} \exp\left(-\frac{1}{2}\left(\frac{r_{ij}}{\sigma_k}\right)^2\right) f_c(r_{ij}), \tag{4.14}$$

gdzie r_{ij} to odległość pomiędzy parą atomamów (i, j), $\alpha, \beta = x, y, z$ (składowe kartezjańskie), r_{ij}^{α} jest modułem różnicy składowej α położeń atomamów i oraz j, a σ_k jest parametrem funkcji i jest dobierany jak hiperparametr.

Odciski atomowe opisują lokalną charakterystykę układu, traktując każdą współrzędną oddzielnie. Wynika to z faktu, że zostały zaprojektowane do przewidywania oddziaływań elektrostatycznych pomiędzy atomami, a więc wektora sił kulombowskich w kartezjańskiej przestrzeni położeń. Przy przewidywaniu wartości, które łatwo separują się na niezależne składowe wektora, takie budowanie zmiennych jest naturalne. Jednak podczas pracy ze zmiennymi skalarnymi, jak całkowita energia układu, rozbicie na składowe prowadzi do stworzenia wielu zmiennych, które same nie niosą informacji o oddziaływaniach, a jedynie złożone ze sobą pozwalają na poprawny opis układu. Utrudnia to interpretację wyników, w szczególności analizę istotności zmiennych.

Wielociałowa reprezentacja tensorowa

Wielociałowa reprezentacja tensorow jest rozszerzeniem macierzy kulombowskiej. Definiuje się szereg funkcji f_k, które w ogólności mogą zostać zapisane jako:

$$f_k(x, z) = \sum_{i=1}^{N} w_k(i) D(x, g_k(i)) \Pi_{j=1}^{k} C_{z_j, Z}, \tag{4.15}$$

gdzie k to numer funkcji, zwykle $k = 1, 2, 3, 4$, sumowanie $\sum_{i=1}^{N}$ jest sumowaniem uogólnionym po wszystkich atomach w układzie (dla $k = 1$),

parach atomów (dla $k = 2$), trójkach atomów (dla $k = 3$) i tak dalej, z_j to liczba atomowa pierwiastka j w układzie, w_k jest elementem M_{kk} z Macierzy Kulombowskiej, D to funkcja rozmywająca, zwykle używana jest funkcja gaussowska, g_k jest funkcją definiowaną różnie dla różnych wartości k: dla $k = 1$ jest to liczba atomowa, dla $k = 2$ jest odwrotnością odległości pomiędzy atomami, dla $k = 3$ jest kątem pomiędzy atomami, dla $k = 4$ jest kątem torsyjnym, a C_{ij} jest parametrem zależnym od pierwiastka.

Powyższa definicja funkcji f_k prowadzi do znacznego skompilowania zmiennych niezależnych (dla $k > 2$), co sprawia, że reprezentacja ta jest trudna do zastosowania.

Lokalna wielociałowa reprezentacja tensorowa

Istnieje, także lokalna wersja wielociałowej reprezentacji tensorowej, która różni się od oryginału tym, że:

— nie jest używa funkcja f_1 ,
— funkcje zarówno rozmycia jak i wag nie muszą być centrowane na atomach.

Rozszerzenie to pozwala na opisywanie bardziej specyficznych własności układu. Należy jednak zachować ostrożność podczas wyboru centrów funkcji, aby nie zwielokrotniać posiadanych już informacji o systemie, co prowadzić może do pominięcia innych istotnych informacji.

Gładkie nałożenie położeń atomów

Gładkie nałożenie położeń atomów jest przedstawieniem położeń atomów jako gęstość atomów w przestrzeni. Położenie każdego atomu jest rozmywane przez funkcję gausaowską z harmoniką sferyczną. Gęstość p w układzie jest opisywana przez:

$$p_{nn'l}^{Z_1 Z_2}(r) = \pi \sqrt{\frac{8}{2l+1}} \sum_m c_{nlm}^{Z_1}(r) c_{n'lm}^{Z_2}(r), \tag{4.16}$$

gdzie n i n' to główne liczby kwantowe, l jest stopniem harmoniki sferycznej, Z_1 i Z_2 to liczby atomowe pierwiastków w układzie, a c_{nlm}^z to rdzenie funkcji, zdefiniowane następująco:

$$c_{nlm}^z(r) = \int \int \int_{R^3} dV g_n(r) Y_{lm}(\theta, \phi) \rho^Z(r), \tag{4.17}$$

gdzie g_n jest radialną funkcją bazy, Y_{lm} jest harmoniką sferyczną, a ρ^Z jest gęstością atomów pierwiastka Z, rozmytą funkcją gaussowską.
Jako funkcje rozmycia g_n używa się zwykle funkcji gaussowskich [107] lub wielomianowych [108].

Inne rodzaje zmiennych

Jako zmienne położeniowe można także brać wielkości tradycyjnie używane w statystyce do budowania zmiennych niezależnych: momenty (statystyczne) wielkości opisujących atomy w układzie takie jak odległość pomiędzy atomami, czy zastępowanie rozkładu gęstości poprzez lokalne przybliżenia lub rozwinięcia w bazie. Podejścia takie nie są jednak popularne i zazwyczaj nie dają lepszych wyników niż omawiane wcześniej.

Dobór najlepszych zmiennych niezależnych jest kluczowy dla dobrych wyników algorytmów uczenia maszynowego. Większość opisywanych rodzin zmiennych zależy także od szeregu hiperparametrów. Podczas selekcji najlepszego zestawu zmiennych trzeba przetestować kilka algorytmów uczenia maszynowego oraz zoptymalizować ich hiperparametry. Obie optymalizacje hiperparametrów (rodziny zmiennych oraz algorytmu) wykonuje się przy użyciu testowania krzyżowego. Optymalizacja hiperparametrów na obu poziomach jest zależna od siebie i najczęściej trzeba przebadać wszystkie kombinacje hiperparametrów dla zmiennych i modeli. W oczywisty sposób generuje to ogromną liczbę przypadków do przetestowania. Konsekwencją czego jest duże zapotrzebowanie na moc obliczeniową oraz konieczność opracowania metodyki prowadzenia badań i przechowywania wyników dla dużej liczby parametrów.

4.2. Praca z małymi zbiorami danych

Próba stosowania algorytmów uczenia maszynowego w analizie zjawisk fizycznych napotyka na dwie istotne trudności:

1. konieczność zbudowania istotnych zmiennych niezależnych dla modelu,
2. niewielki zasób dostępnych danych.

Pierwszy problem został omówiony w poprzednim rozdziale. Dla każdego zagadnienia konieczne jest wykonanie oddzielnej analizy i wyboru najodpowiedniejszych zmiennych. Problem drugi można spotkać we wszystkich zastosowaniach uczenia maszynowego w fizyce. W zagadnieniach analizy zjawisk fizycznych jest on jednak bardzo widoczny i dotyka praktycznie każdego przypadku. Powszechnie uważa się, że zbór treningowy jest wystarczająco liczny dla zastosowania algorytmów uczenia maszynowego, jeśli zawiera tysiąc obserwacji. Definicja ta wydaje się przeszacowana, dla dobrej jakości modelu wystarczy posiadać 5 do 10 obserwacji na jeden parametr modelu.

Jak zostało omówione w rozdziale 4.1.1, dane dostępne do analizy pochodzą z dwóch źródeł: albo z eksperymentu albo z symulacji numerycznych. W obu przypadkach uzyskanie pojedynczej obserwacji jest procesem długotrwałym i kosztownym. Najczęściej też nie ma możliwości w żaden sposób poszerzyć posiadanej puli obserwacji. Oprócz kosztowności dodatkową trudnością jest charakter prowadzonych badań.

Istnieje szereg metod rozwiązywania problemu mało licznych zbiorów danych. Podczas stosowania algorytmów opartych na uczeniu maszynowym są to:

— zdobycie dodatkowych danych,
— stosowanie prostych modeli,
— stosowanie wielokrotnego próbkowania,
— wygenerowanie syntetycznych danych,
— dokładne czyszczenie danych,
— przeanalizowanie istotności zmiennych,
— stosowanie modeli uśrednionych.

Zdobycie większej ilości danych jest najlepszym rozwiązaniem ale najczęściej nie ma możliwości na rozszerzenie posiadanego zbioru danych.

Użycie prostych modeli pozwala na wyeliminowanie problemu przeuczenia modelu dla niewielkiej liczby zmiennych [109]. Dla małych zbiorów danych zazwyczaj stosuje się modele liniowe lub wieloliniowe z regularyzacją. Procesy można lokalnie przybliżyć liniowo więc regresja liniowa działa nadspodziewanie dobrze w wielu przypadkach. Zastosowanie regularyzacji pozwala także na ograniczenie ilości dopasowywanych parametrów i upraszcza model. Zastosowanie liniowych modeli jest jednak niewystarczające w przypadku bardziej złożonych procesów oraz konieczności uzyskania dużej dokładności modelu.

Wielokrotne próbkowanie jest techniką często używane w statystyce. Polega ona na wielokrotnym losowaniu z próby próbek na których buduje się model. W praktyce stosuje się dwa sposoby próbkowania: boosting [110] oraz bagging [111]. W literaturze spotyka się dwa podejścia: buduje się szereg prostych modeli na podstawie próby wylosowanej z oryginalnej próby albo generuje się próbę przez wielokrotne losowanie obserwacji z oryginalnej próby i budowie jednego modelu. Metody te zmniejszają co prawda niepewność estymacji, ale losowania wykonywane są na małej puli obserwacji, więc jeśli jakieś aspekt procesu nie jest w niej widoczny model także nie będzie w stanie go odzwierciedlić.

Generowanie syntetycznych danych jest metodologią podobną do wielokrotnego próbkowania. Polega ono na tworzeniu syntetycznych obserwacji na podstawie oryginalnej próby. Istnieje wiele technik, w większości dedykowanych do szczególnych zagadnień jak analiza obrazów. W przypadku danych niebędących obrazkami lub tekstem najczęściej stosuje się dwa algorytmy: dodawanie losowego szumu do oryginalnych obserwacji lub SMOTE [112]. Obie techniki są podobne, ale różnią się sposobem dodawania losowości do oryginalnych obserwacji podczas generowania syntetycznych danych. W pierwszej metodzie losuje się szum dla każdej zmiennej niezależnie ale z jednego rozkładu, podczas gdy w SMOTE nowy punkt jest tworzony z oryginalnej obserwacji, do której dodaje się wektor będący liniową kombinacją najbliższych sąsiadów pomnożoną przez losową liczbę. Obie techniki dobrze nadają się dla zmiennych ciągłych, jednak dla zmiennych dyskretnych lub kategorycznych prowadzą do niefizycznych wartości zmiennych niezależnych przez co model staje się trudny do interpretacji.

Podczas czyszczenia danych należy zwrócić szczególną uwagę na obserwacje mocno odstające od pozostałych, gdyż w przypadku małych zbiorów treningowych są one dużo mocniej odzwierciedlane w modelu niż dla dużych zbirów treningowych. Jednocześnie należy pamiętać, że takie właśnie punkty mogą prowadzić do odkrycia nowych zjawisk, więc należy zachować szczególną ostrożność podczas usuwania punktów z próby.

Dla małych zbiorów danych kluczowe jest wykonanie selekcji istotnych zmiennych oraz ograniczenie modelu do jak najmniejszej ich liczby. Pozwala to na zmniejszenie liczby dopasowywanych parametrów modelu, a co za tym idzie możliwości przeuczenia modelu. Dla dużych zbiorów danych model zawierający dodatkowe mało istotne zmienne niezależne będzie działać tak samo dobrze jak model uproszczony, jednak dla małego zbioru treningowego nieuproszczony model będzie miał tendencję do przeuczania.

Stosowanie modeli uśrednionych polega na zbudowaniu wielu prostych modeli opisujących jedynie część procesu i połączenie ich w jeden bardziej ogólny. Najczęściej stosowane algorytmy to las losowy [77], Gradient boosting [80] i AdaBoost [81], omówione w rozdziale 3.7.7.

4.3. Niezbalansowane zbiory danych

Problem niezbalansowanych zbiorów danych występuje w przypadku klasyfikacji obserwacji. Polega on na nierównolicznym występowaniu obserwacji z różnych klas. Dla uproszczenia organiczymy się do przypadku dwóch klas, ale problem i opisywane techniki mogą być łatwo uogólnione na przypadek większej liczby klas.

Podczas budowania klasyfikatora niezbędne jest zbalansowanie rozkładu klas w celu uniknięcia preferencji przez model najliczniejszej klasy. Przyjmuje się, że jeśli stosunek liczebności klas jest mniejszy niż 1:2 zbiór danych jest wciąż zbalansowany. Oznacza to, że mniej liczna klasa stanowi około 30% wszystkich obserwacji. Problem niezbalansowanych danych jest szczególnie widoczny dla małych zbiorów danych, przez co kluczowy w zastosowaniach algorytmów uczenia maszynowego w fizyce.

Rozważmy przypadek, gdy stosunek liczebności klas wynosi 1:4. Wówczas jeśli model będzie przewidywał wszystkie przypadki jako należące do liczniejszej klasy, otrzymana dokładność wyniesie 80%. Jeśli różnica w liczebności jest większa, dokładność modelu przewidującego jedynie liczniejszą klasę będzie coraz wyższa. Prowadzi to do modelu pozbawionego zdolności rozróżniania klas, zwracającego zawsze jedną wartość.

Istnieje szereg technik balansowania danych lub pracy z niezbalansowanym zbiorem danych:

— zwiększenie liczebności mniej licznej klasy przez jej rozszerzenie poprzez wielokrotne próbkowanie,
— zwiększenie liczebności mniej licznej klasy poprzez wygenerowanie syntetycznych danych,
— zmniejszenie liczebności liczniejszej klasy przez usunięcie z niej losowych obserwacji,
— zastosowanie dedykowanej metryki dla niezbalansowanych danych,
— zastosowanie algorytmów ważonych.

Zwiększenie liczebności mniej licznej klasy przez wielokrotne próbkowanie może być stosowane jedynie dla dużych zbiorów danych. Dla małych zbiorów danych nie przynosi ono pożądanego rezultatu, gdyż prowadzi do przeuczonego modelu i nie poprawia jego zdolności uogólniania. Wynika to z faktu, że mało liczne zbiory danych nie pozwalają na zbudowanie dobrej statystyki.

Wygenerowanie syntetycznych danych jest lepszym rozwiązaniem niż wielokrotne próbkowanie, nawet dla małych zbiorów danych, gdyż pozwala na budowanie modelu posiadającego lepszą zdolność uogólniania. Do mniej licznej klasy stosuje się te same techniki jak podczas rozszerzania małych zbiorów danych (patrz rozdział 4.2) czyli poprzez dodanie losowego szumu lub użycie algorytmu SMOTE [112].

Zmniejszenie liczebności liczniejszej klasy prowadzi do zbalansowanych klas, ale znacząco zmniejsza liczebność zbioru danych. Zatem może być stosowana jedynie dla bardzo dużych zbiorów danych.

Zastosowanie dedykowanych metryk jest uniwersalną techniką niezależnie od wielkości zbioru danych. Rozszerzenie metryki polega na dodaniu do jej definicji wagi odwrotnie proporcjonalnych do liczebności klas. Przez ten zabieg błędy klasyfikacji mniej licznej klasy są bardziej istotne podczas uczenia modelu przez co unika się preferowania liczniejszej klasy. Przykładowo dla dokładności stosuje się zbalansowaną dokładność (ang. balanced accuracy) [113], definiowaną jako: $Acc_b = \frac{1}{2}(\frac{TP}{TP+FP} + \frac{TN}{TN+FN})$ Pozwala ona wiarygodnie porównywać wyniki modeli dla niezbalansowanych zbiorów danych. Jeśli metryka ta zostanie użyta podczas uczenia modelu wówczas mówimy o algorytmach ważonych.

4.4. Algorytmy charakterystyczne dla uczenia maszynowego w fizyce materii skondensowanej

Jednym z najczęściej stosowanych algorytmów uczenia maszynowego w analizie procesów fizycznych w fizyce materii skondensowanej jest Regresja grzbietowa (ang. Kernel Ridge Regression, KRR) [114]. Metoda ta polega na stworzeniu metryki podobieństwa pomiędzy obserwacjami. W metodzie KRR oblicza się jak bardzo dany punkt jest podobny do punktów ze zbioru treningowego i na podstawie tego podobieństwa wynik jest określany jako średnia ważona:

$$y(x) = \sum_{i=1}^{N} \alpha_i exp\left(\frac{1}{2}\left(\frac{|x - x_i|}{\sigma}\right)^2\right), \tag{4.18}$$

gdzie α_i jest parametrem modelu, proporcjonalnym do wartości y_i, a σ - parametrem metryki. Metoda KRR może być interpretowana jako regresja liniowa z zastosowaniem jądra operatora podobnie jak w algorytmie wektorów rozpinających. Funkcja $exp\left(\frac{1}{2}\left(\frac{|x-x_i|}{\sigma}\right)^2\right)$ może być rozumiana, więc jako metryka podobieństwa lub jako jądro (ang. kernel) .

Metoda KRR bazuje na podobieństwie pomiędzy punktami i wymaga dużej liczby obserwacji równomiernie próbkujących badaną przestrzeń. Jest to największa trudność stosowania tej metody podczas analizy zjawisk w fizyce materii skondensowanej.

Rozdział 5

Przewidywanie własności nanomateriałów

5.1. Nanorurki borowe

5.1.1. Opis układu

Bor jest jednym z najciekawszych pierwiastków w układzie okresowym. Wynika to z jego położenia pomiędzy metalami i niemetalami oraz tendencji do tworzenia ogromnej liczby struktur [115–117], takich jak cząsteczki B_{12}, kryształy, struktury dwuwymiarowe jak również struktury jednowymiarowe nanorurki [118, 119] czy nanodruty [120].

Nanostruktury boru posiadają duży potencjał aplikacyjny [121, 122], co wynika z ich właściwości: wysokiej twardości, niskiej gęstości oraz wysokiej temperatury topnienia [123–125]. Są one wykorzystywane jako budulec baterii, katalizatorów, oraz wytrzymałej elektroniki [126], a także w nanourządzeniach elektronicznych oraz optoelektronicznych [127–132].

Jedną z najbardziej obiecujących struktur są nanorurki borowe [118, 119]. Jako struktury jednowymiarowe dają doskonałą możliwość badania przewodnictwa elektrycznego oraz cieplnego [133]. Ponadto, nanostruktury jednowymiarowe są stosowane w tranzystorach polowych, diodach elektroluminescencyjnych (LED), fotodetektorach, ogniwach słonecznych oraz czujnikach [134].

Nanorurki borowe zostały po raz pierwszy przewidziane teoretycznie przez I. Boustani i A. Quandt [135], a w 2004 r. [136] po raz pierwszy zsyntetyzowane. Najczęściej zakładane geometrie nanorurek borowych to zwinięta struktura dwuwymiarowa boru - borofen (ang. borophene), który jest strukturą analogiczną do grafenu [137, 138]. Należy zaznaczyć, iż istnieje wiele innych możliwych geometrii nanorurek borowych. Nie są one jednak przebadane z powodu dużych nakładów obliczeniowych. Procedura zwijania borophene'u prowadzi do znacznego ograniczenia liczby możliwych geometrii, co przyczyniło się w pewnej mierze do jej popularności. Wymaga ona określenia dwóch parametrów: sposobu zwijania borofenu oraz określenia ewentualnych zmian w geometrii. Pierwszy parametr jest analogiczny jak w przypadku grafenu i jest opisywany przesunięciem wzdłuż wektorów sieci o całkowitą liczbę długości wiązania. Drugi parametr opisuje defekty. W strukturze zwijanego borofenu, w literaturze rozważa się głównie defekty punktowe czyli usunięcie atomów z węzłów sieci. Podejście to mimo znacznego uproszczenia problemu prowadzi jednocześnie do zmniejszenia przestrzeni możliwych geometrii struktur boru. Znacznie lepszym rozwiązaniem byłoby wykonanie szeregu optymalizacji geometrii przy użyciu teorii funkcjonału gęstości dla

dużej liczby niezależnych i różnych początkowych geometrii. Podejście takie jednak ze względu na wymagania obliczeniowe jest trudne do zrealizowania. Rozwiązaniem tego problemu może być zastosowanie algorytmów uczenia maszynowego.

Celem prac prezentowanych w tym rozdziale było stworzenie modelu opisującego zależność energii nanorurki borowej od jej geometrii. Model taki może okazać się bardzo przydatny podczas przeszukiwania przestrzeni możliwych konformacji w celu znalezienia optymalnej geometrii. Zastosowanie algorytmów uczenia maszynowego prowadzi do skrócenia czasu poszukiwań optymalnych położeń atomów poprzez wstępną selekcje układów o potencjalnie optymalnych geometriach. Podczas prac nie ograniczaliśmy badanych geometrii nanorurek boru do zwiniętego borofenu. Prezentowane wyniki powstały dla geometrii nanorurek definiowanej przez jej grubość oraz parametry komórki elementarnej: jej długość oraz liczbę atomów.

5.1.2. Generowanie zbioru danych

Pierwszym krokiem budowania modelu opisującego energię nanorurek borowych było stworzenie zbioru treningowego. Wygenerowano 4000 losowych geometrii nanorurek. Zakresy parametrów zostały zebrane w tabeli 5.1. Parametry zostały wybrane zgodnie z danymi z publikacji [137, 139].

Tabela 5.1: Wartości parametrów użytych do generowania zbioru treningowego dla nanorurek borowych.

Parametry	zakres
Liczba atomów w komórce elementarnej	2 - 10
promień R nanorurki	1,5 nm - 8 nm
długość D nanorurki	1 nm - 4 nm

Dla każdej nanorurki była losowana liczba atomów N, jej długość D oraz promień R. Następnie losowano położenia na powierzchni walca o promieniu R i długości D. Losowanie było wykonywane tak, aby atomy były równomiernie rozmieszczone na powierzchni.

Na podstawie wylosowanej geometrii tworzony był plik wsadowy dla pakietu numerycznego Quantum Espresso [53–55].

Przed rozpoczęciem generowania zbioru treningowego i wykonaniem symulacji dobrano parametry obliczeń: siatkę k-punktów, promień odcięcia energii kinetycznej (cutoff), rozmycia gausowskiego w strefie Brillouin'a (degauss parameter) oraz wielkości superkomórki. Wyniki testów przedstawiają odpowiednio wykresy A.1, A.2, A.3 oraz A.4 umieszczonych w uzupełnieniu A.

Ponieważ celem badań było znalezienie zależności pomiędzy energią, a geometrią nanorurki nie wykonywano optymalizacji geometrii, a jedynie obliczono energię całkowitą. Podczas obliczeń sprawdzano zbieżność algorytmu tak, aby w zbiorze treningowym znajdowały się jedynie uzbieżnione wyniki.

5.1.3. Przewidywanie energii

W celu zbudowania najlepszego modelu opartego na algorytmach uczenia maszynowego przetestowano szereg algorytmów omówionych w poprzednich rozdziałach. Wypisano je w tabeli 5.2 wraz z oznaczeniami:

Tabela 5.2: Używane algorytmy uczenia maszynowego oraz ich oznaczenia

Nazwa algorytmu	oznaczenie
Regresja liniowa	regLIN
Regresja liniowa z regularyzacją L1	regLIN-L1
Regresja liniowa z regularyzacją L2	regLIM-L2
Lasy losowe	regRF
SVM z linową funkcją jądra	SVM-lin
SVM z funkcją jądra rbf	SVM-rbf
SVM z sigmoidalną funkcją jądra	SVM-sig
Ekstremalnie losowe drzewa	XRT
XGBoost	XGB
Adaboost oparty na regresji liniowej	Ada-lin
Adaboost oparty na drzewach losowych	Ada-tree
Regresja grzbietowa z jądrem liniowym	KRR-lin
Regresja grzbietowa z jądrem rbf	KRR-rbf

W tabeli 5.3 zebrano badane zmienne położeniowe oraz ich oznaczenia.

Każdy model został zbudowany dla każdego zestawu zmiennych położeniowych. W celu walidacji jakości modelu użyto dwóch metryk: R2 oraz MES. Jako wyznacznik jakości modelu badano dwa warianty wybranych metryk: obliczaną po zakończeniu procesu uczenia na zbiorze treningowym oraz obliczaną za pomocą 10-krotnego testowania krzyżowego (CV). Pierwsza miara pozwala kreślić czy model jest w stanie odwzorować badane zjawisko. Druga miara pozwala wiarygodnie porównywać modele. Otrzymane wyniki dla metryk R2 i MSE przedstawiono odpowiednio w tabelach 5.4 i B.1.

Przedstawione wyniki pokazują bardzo dobrą zdolność uogólniania modelu (model może dokładnie szacować energię dla nowych układów). Niemal wszystkie testowane algorytmy uzyskały wynik metryki R2 podczas 10-krotnego testowania krzyżowego powyżej 0,90. Dokładniejsza analiza pokazuje, że dla jakości modelu większe znaczenie ma jakość wybranych zmiennych niż sam algorytm. Dla zmiennych budowanych przez Macierz Kulombowską model liniowy daje prawie taką samą dokładność jak modele oparte na lasach losowych czy ADAboost. Wynik ten może być także związany z definicją samej Macierzy Kulombowskiej, która jest zdefiniowana jako macierz oddziaływań elektrostatycznych pomiędzy atomami. Oddziaływania elektrostatyczne stanowią znaczący wkład do energii całkowitej układu. Stąd wysokie wartości metryk dla tego zestawu zmiennych pokrywają się z intuicją fizyczną. Jednocześnie wyniki ESM są istotnie gorsze niż CM, co pozwala sądzić, że efekty związane z periodycznością układu nie są istotne dla opisu energii lub model jest w stanie nauczyć się periodyczności układu i wprowadzić poprawki do modelu.

Tabela 5.3: Używane zmienne opisujące geometrię nanostruktur oraz ich oznaczenia

Nazwa zmiennych	oznaczenie
Macierz Kulombowska	CM
Macierz Ewalda	ESM
Macierz sinusoid	SM

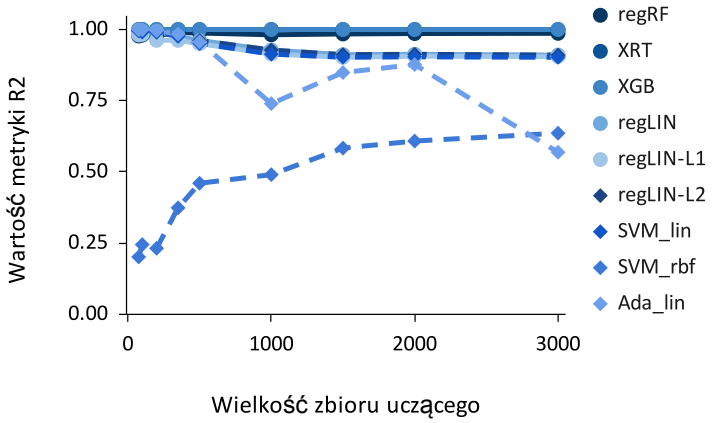
Tabela 5.4: Wartość metryki R2 przewidywania energii całkowitej dla nanorurek borowych obliczona na **zbiorze treningowym** oraz **testowania krzyżowego (CV)**. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

	zbiór treningowy			CV		
model	ESM	CM	SM	ESM	CM	SM
regRF	0,98	0,98	0,97	0,92	0,95	0,80
XRT	1,00	1,00	1,00	0,92	0,94	0,78
regLIN	0,55	0,87	0,83	0,54	0,95	0,66
regLIN-L1	0,54	0,86	0,82	0,52	0,94	0,65
regLIN-L2	0,55	0,87	0,83	0,53	0,95	0,66
SVM-lin	X	0,87	0,83	X	0,95	0,66
SVM-rbf	X	0,10	X	X	X	X
SVM-sig	1,00	0,99	0,99	X	X	X
Ada-lin	0,51	0,84	0,61	0,49	0,90	0,50
Ada-tree	1,00	1,00	1,00	0,92	0,95	0,78
KRR-lin	X	0,79	0,62	X	0,67	0,77
KRR-rbf	0,00	0,03	x	X	X	X

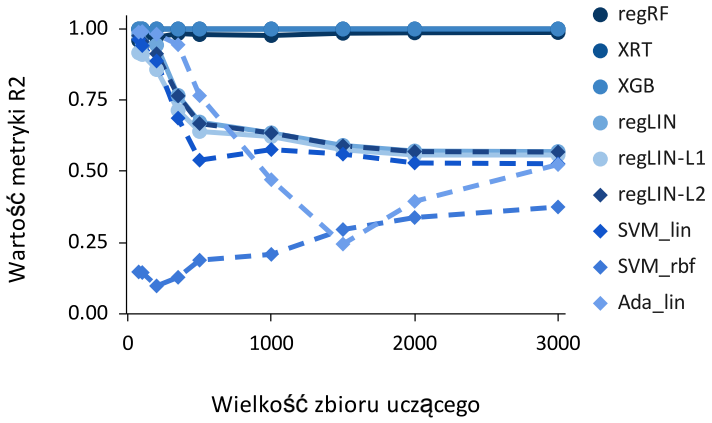
Przebadano stabilność modeli ze względu na liczbę obserwacji w zbiorze treningowym w zakresie od 50 do 4000 obserwacji. Dla zmiennych: Macierzy Kulombowskiej, Macierzy Ewalda oraz Macierzy sinusoid wyniki zostały przedstawione na rysunku 5.1. Jako miarę jakości modelu wybrano metrykę R2 obliczaną po zakończeniu procesu uczenia modelu. Wyniki uzyskane na podstawie 10-krotnego testowania krzyżowego zostały przedstawione na wykresie 5.2 dla Macierzy Kulombowskiej, Macierzy Ewalda oraz Macierzy sinusoid.

Wszystkie badane kombinacje algorytmów oraz zestawów zmiennych dają bardzo wysokie wartości metryki R2 już dla najmniejszych zbiorów treningowych. Wyjątek stanowią jedynie modele SVM-rbf, SVM-sig, które nie opisują poprawnie zjawiska (wartość metryki R2 zbiega do 0,7). Ocena metryki R2 za pomocą testowania krzyżowego oraz dla procesu uczenia daje podobne wyniki. Należy jedynie podkreślić, że stosowanie metody testowania krzyżowego dla małych zbiorów danych istotnie przeszacowuje wartości metryk, zatem realna zdolność uogólniania modeli zbudowanych na małym zbiorze danych jest znacznie gorsza.

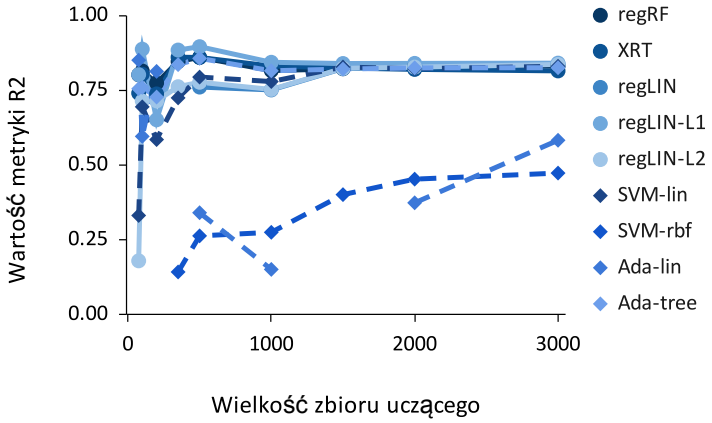
Prezentowane wyniki pozwalają dodatkowo określić jakości zmiennych po-



(a) CM

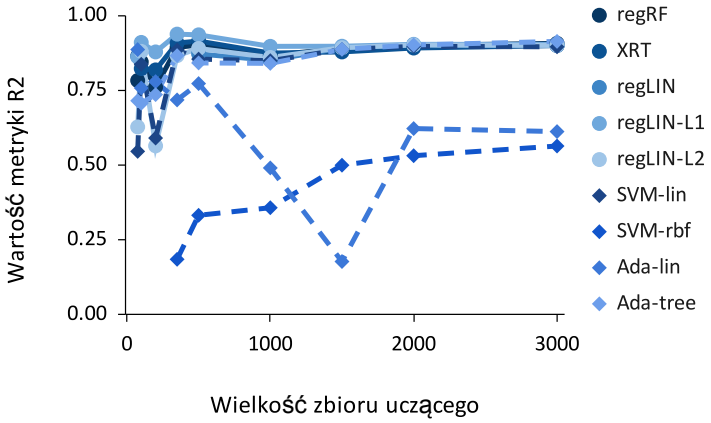


(b) ESM

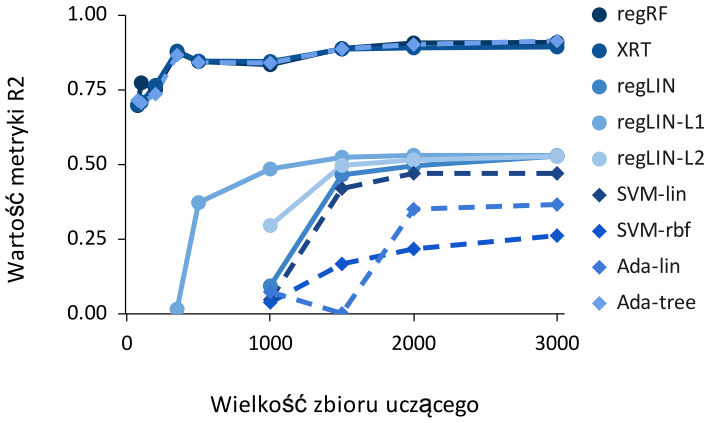


(c) SM

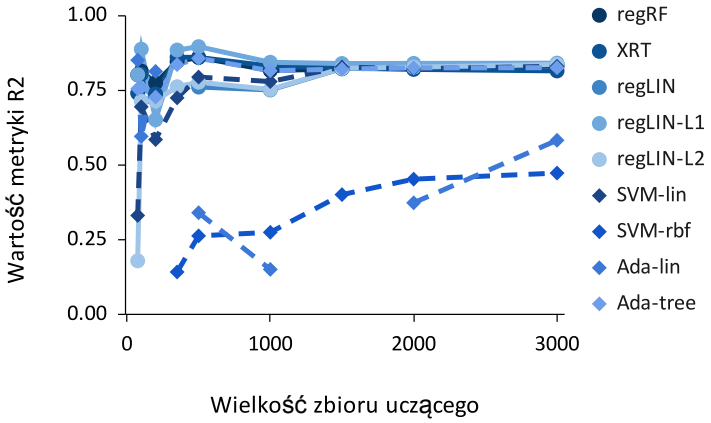
Rysunek 5.1: Metryka R2 w funkcji wielkości zbioru treningowego dla nanorurek borowych dla układów opisanych przy pomocy zmiennych: (a) Macierzy Kulombowskiej (CM), (b) Macierzy Ewalda (ESM), (c) Macierzy sinusoid (SM).



(a) CM



(b) ESM



(c) SM

Rysunek 5.2: Metryka R2 z 10-krotnego testowania krzyżowego w funkcji wielkości zbioru treningowego dla nanorurek borowych dla układów opisanych przy pomocy zmiennych: (a) Macierzy Kulombowskiej (CM), (b) Macierzy Ewalda (ESM), (c) Macierzy sinusoid (SM).

łożeniowych. Najlepsze rezultaty uzyskano dla rodziny zmiennych Macierzy Kulombowskiej

5.1.4. Przewidywanie energii kohezji

Kolejnym krokiem prac nad zrozumieniem fizyki nanorurek borowych było zbudowanie modelu opartego na algorytmach uczenia maszynowego pozwalającego na przewidywanie stabilności nanorurek w funkcji położeń atomów w układzie. Jednym ze sposobów określenia stabilności układu jest wyznaczenie znaku energii kohezji E_{koh} na atom:

$$E_{koh} = \frac{1}{N}(E_{tot} - \sum_{i=1}^{N} E_i), \tag{5.1}$$

gdzie E_{tot} jest całkowitą energią układu, $\sum_{i=1}^{N}$ jest sumą po wszystkich atomach w układzie, N jest liczbą atomów w układzie, a E_i jest energią i-tego izolowanego atomu.

Zbór treningowy składał się z 4000 losowych geometrii oraz przypisanych im energii kohezji obliczonych za pomocą DFT. Parametry użyte do generowania geometrii układów znajdują się w tabeli 5.1. Podczas obliczeń nie wykonywano optymalizacji geometrii. Miało to zmaksymalizować obszar próbkowania przestrzeni geometrii i pozwolić na zbudowanie ogólniejszego modelu.

Przetestowane algorytmy uczenia maszynowego oraz rodziny zmiennych położeniowych zebrano odpowiednio w tabelach 5.2 i 5.3.

Podobnie jak dla przewidywania energii całkowitej układu jako metryki jakości modeli wybrano R2 i MSE. Obie metryki dla wszystkich kombinacji model-zestaw zmiennych zostały wyznaczone na zbiorze treningowym oraz przy użyciu 10-krotnego testowania krzyżowego. Otrzymane wyniki dla metryk R2 i MSE przedstawiono odpowiednio w tabelach 5.5 i B.2.

Podczas przewidywania energii kohezji nanorurek borowych najwyższą dokładność uzyskały modele oparte na drzewach losowych. Także modele regresji liniowej dają dobrą dokładność. Modele bazujące na bardziej złożonych metodach takich jak SVM oraz KRR z nieliniowymi jądrami operatora nie opisują poprawnie energii kohezji. Podkreślić należy, że rozszerzenia modelu liniowego jak regularyzacja lub zastosowanie AdaBoost znacząco poprawia zdolność uogólniania modelu liniowego. Wyniki pokazują także, że nie ma konieczności uwzględniania periodyczności układu, gdyż CM daje lepsze wyniki niż ESM dla wszystkich modeli. Wartości metryki R2 podczas uczenia modelu są zbliżone do 1, natomiast podczas testowania krzyżowego są na poziomie 0,8 - 0,9 dla najlepszych modeli. Prezentowane wyniki pozwalają wnioskować o dużej zdolności uogólniania modeli. Jednocześnie wyniki są gorsze niż dla przewidywania energii całkowitej. Wynika to z faktu, że energia kohezji nie może zostać bezpośrednio zapisana jako ważona suma energii oddziaływania pomiędzy atomami, które to stanowią elementy Macierzy Kulombowskiej.

Tabela 5.5: Wartość metryki R2 przewidywania energii kohezji dla nanorurek borowych obliczona podczas **zbiorze treningowym** oraz **testowania krzyżowego (CV)**. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

	uczenie			CV		
model	ESM	CM	SM	ESM	CM	SM
refRF	0,91	0,97	0,93	0,60	0,89	0,55
XRT	1,00	1,00	1,00	0,52	0,87	0,56
XGBoost	0,92	0,98	0,95	0,66	0,87	0,63
regLIN	0,21	0,80	0,64	X4	0,58	0,83
regLIN-L1	0,43	0,77	0,61	0,15	0,64	0,80
regLIN-L2	0,45	0,80	0,64	0,09	0,58	0,83
SVM-lin	0,34	0,75	0,56	X	0,44	0,74
SVM-rbf	0,08	0,08	0,08	X	X	X
SVM-sig	0,99	0,99	0,99	X	X	X
Ada-lin	0,30	0,69	0,46	X	0,34	0,63
Ada-tree	1,00	1,00	1,00	0,51	0,90	0,52
KRR-lin	X	0,79	0,62	X	0,67	0,77
KRR-rbf	0,00	0,16	0,06	X	X	X

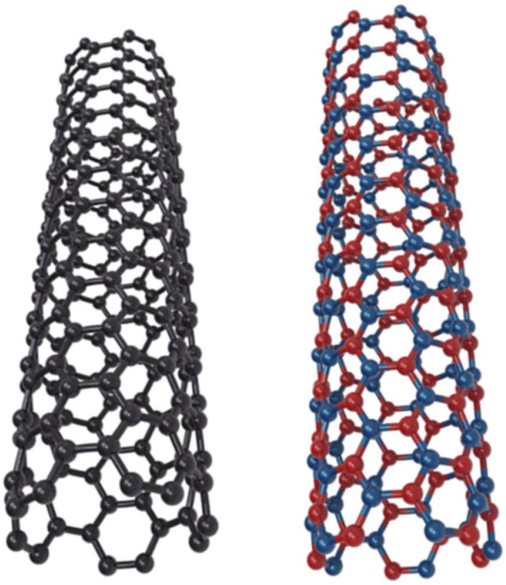
5.1.5. Wnioski

W niniejszym rozdziale przedstawiono wyniki badań nad opisem energii oraz stabilności nanorurek borowych za pomocą algorytmów uczenia maszynowego. Prace zostały wykonane na danych pochodzących z przeprowadzonych 4000 symulacji numerycznych w programie Quantum Espresso. Baza danych zawierała geometrię układów oraz ich energię całkowitą i energię kohezji. Rezultatem badań są stworzone modele energii całkowitej oraz energii kohezji. W czasie prac przetestowano szereg algorytmów uczenia maszynowego oraz wiele sposobów reprezentacji geometrii nanorurek. Najlepsze modele posiadają dokładność na poziomie 0,95 dla energii całkowitej (mierzone metryką R2 obliczaną za pomocą 10-krotnego testowania krzyżowego) oraz 0,90 dla energii kohezji. Najwyższą dokładność osiągnęły modele oparte na drzewach losowych dla reprezentacji położeń atomów za pomocą Macierzy Kulombowskiej.

5.2. Nanorurki azotku boru

5.2.1. Opis układu

Azotek boru posiada bardzo dobre własności elektryczne oraz optyczne przy jednocześnie dużej stabilności fizyko-chemicznej. Może występować w postaci dwuwymiarowych warstw, nanorurek lub objętościowych kryształów o różnych strukturach kryptograficznych [140]. Struktury azotku boru mogą posiadać różną symetrię: heksagonalną, kubiczną lub wurcytu, występują także odmiany amorficzne [141]. Najstabilniejszą formą jest struktura hek-



Rysunek 5.3: Porównanie geometrii jednowarstwowych nanorurek węglowych i azotku-boru. Rysunek pochodzi z [144].

sagonalna. Azotek boru ze względu na swoje właściwości znajduje szereg zastosowań aplikacyjnych.

Jedną z najciekawszych struktur azotku-boru są nanorurki. Ich istnienie zostało przewidziane teoretycznie w 1984 [142], a zostały zsyntezowane w 1995 [143]. Nanorurki azotku boru są bardzo podobne do nanorurek węglowych. Geometria obu układów opiera się na zwiniętej monowarstwie o geometrii sieci hexagonalnej [144] lub kilku takich warstw dla nanorurek wielowarstwowych. Do tej pory nie zostały wykonane dokładne badania stabilności innych geometrii ułożenia atomów w nanorurce. Również własności fizyko-chemiczne obu nanorurek są zbliżone jednak nanorurki azotku boru są stabilniejsze, przez co lepiej nadają się do zastosowań przemysłowych i medycznych.

Nietoksyczność [145] oraz wysoka biokompatybilność nanorurek azotku boru umożliwia ich potencjalne zastosowanie do aplikacji biomedycznych, terapeutycznych i diagnostycznych [146], na przykład, mogą mieć zastosowanie w onkologii [147, 148] oraz jako nośnik leków [149].

Otwartym pozostaje pytanie jaka jest optymalna geometria nanorurek azotku-boru. Azotek boru to układ dwu składnikowy, stąd konieczne jest uwzględnienie składu procentowego obu pierwiastków oraz ich rozmieszczenia w nanorurce. Fakt ten znacząco komplikuje problem znajdowania optymalnej geometrii nanorurek azotku boru. Dlatego, tak jak w przypadku czystych nanorurek borowych zastosowanie algorytmów uczenia maszynowego pozwala przyśpieszyć proces szukania optymalnej geometrii, a co za tym idzie istotnie skrócić czas uzyskania wyniku.

5.2.2. Generowanie zbioru danych

Zbiór treningowy został zbudowany analogicznie jak zbiór treningowy dla nanorurek borowych (patrz 5.1.2). Wygenerowano 4000 losowych geometrii

Tabela 5.6: Wartości parametrów użytych do generowania zbioru treningowego.

Parametry	zakres
Liczba atomów B w komórce elementarnej	2 . . . 10
Liczba atomów N w komórce elementarnej	1 . . . 5
promień nanorurki R	1,5 nm . . . 5 nm
długość nanorurki D	1 nm . . . 4 nm

nanorurek. Zakres parametrów generatora zostały zebrane w tabeli 5.6. Parametry zostały wybrane zgodnie z danymi z literatury [143].

5.2.3. Przewidywanie energii

Analogicznie jak dla nanorurek borowych wykonano testy stabilności algorytmów uczenia maszynowego w funkcji wielkości zbioru treningowego oraz liczby zmiennych używanych do budowania modelu. Uzyskane wyniki są zgodne z wynikami nanorurek borowych. Ponieważ wyniki są bardzo zbliżone do wyników z rysunków 5.1 i 5.2, zostały tutaj pominięte.

W celu uzyskania najlepszego modelu przetestowano szereg algorytmów uczenia maszynowego zebranych w tabeli 5.2, natomiast stosowane zmienne wypisano w tabeli 5.3.

Otrzymane wyniki dla metryk R2 i MSE przedstawiono odpowiednio w tabelach 5.7 i C.1. Obie metryki były wyznaczane na zbiorze treningowym i testowanie przy użyciu 10-krotnego testowania krzyżowego.

Tabela 5.7: Wartość metryki R2 obliczona na **zbiorze treningowym** oraz **testowania krzyżowego (CV)** podczas przewidywania energii całkowitej nanorurki azotku-boru. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

	zbiór treningowy			CV		
model	ESM	CM	SM	ESM	CM	SM
regRF	0,97	0,99	0,97	0,78	0,89	0,80
XRT	1,00	1,00	1,00	0,76	0,92	0,82
XGB	0,90	0,98	0,94	0,78	0,93	0,85
regLIN	0,02	0,94	0,87	0,38	0,92	0,85
regLIN-L1	0,47	0,93	0,87	0,41	0,93	0,86
regLIN-L2	0,48	0,94	0,87	0,39	0,92	0,85
SVM-lin	X	0,94	0,87	X	0,92	0,85
SVM-rbf	X	X	X	X	X	X
SVM-sig	1,00	0,98	0,99	X	X	X
Ada-lin	0,23	0,93	0,78	0,27	0,89	0,80
Ada-tree	1,00	1,00	1,00	0,76	0,90	0,81
KRR-lin	0,62	0,85	0,67	0,70	0,67	0,13
KRR-rbf	0,71	0,70	0,70	X	X	X

Otrzymane wyniki pokazują bardzo dobrą zdolność uogólniania modelu. Niemal wszystkie modele uzyskały wynik metryki R2 podczas testowania krzyżowego powyżej 0,90. Wyniki są analogiczne do wyników otrzymanych dla nanorurek borowych. W przypadku nanorurek azotku boru analogicznie do przypadku nanorurek borowych, także CM daje najlepsze wyniki. Pozwala to na podtrzymanie tezy, że możliwe jest uwzględnianie periodyczności układu nawet używając zmiennych nie posiadających tej informacji.

Badane kombinacje algorytmów oraz zestawów zmiennych dają bardzo wysokie wartości metryki R2 już dla niewielkich zbiorów treningowych. Wyjątek stanowią jedynie modele SVM-rbf i SVM-sig. Wartość metryki R2 zbiega dla nich do wartości około 0,7. Najlepsze rezultaty uzyskano dla Macierzy Kulombowskiej. Wyniki są bardzo zbliżone do wyników uzyskanych dla nonorurek borowych.

5.2.4. Przewidywanie energii kohezji

Przeanalizowano także stabilność nanorurek azotku-boru w stosunku do swobodnych atomów. Dane, z których składał się zbiór treningowy wygenerowano analogicznie jak dla opisu energii całkowitej. Zbiór składał się z 4000 obserwacji. Parametry generowania geometrii znajdują się w tabeli 5.6. Przetestowane algorytmy uczenia maszynowego wypisano w tabeli 5.2, stosowane zmienne wypisano w tabeli 5.3.

Otrzymane wartości metryk R2 i MSE na zbiorze treningowym oraz na podstawie 10-krotnego testowania krzyżowego zostały zebrane odpowiednio w tabeli 5.8 i C.2.

Tabela 5.8: Wartość metryki R2 obliczona na zbiorze treningowym oraz CV podczas przewidywania energii kohezji nanorurki azotku-boru. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

	zbiór treningowy			CV		
model	ESM	CM	SM	ESM	CM	SM
refRF	0,96	0,98	0,93	0,87	0,66	0,12
XRT	1,00	1,00	1,00	0,81	0,65	0,19
XGBoost	0,98	0,98	0,94	0,84	0,68	0,02
regLIN	X	0,85	0,67	X	0,65	0,15
RegLIN-L1	0,66	0,83	0,64	0,76	0,67	0,25
RegLIN-L2	0,68	0,85	0,67	0,75	0,65	0,15
SVM-lin	0,55	0,81	0,61	0,62	0,63	0,19
SVM-rbf	X	0,01	X	X	X	X
SVM-sig	0,99	0,99	0,99	X	X	X
Ada-lin	0,64	0,73	0,42	0,80	0,26	X
Ada-tree	1,00	1,00	1,00	0,78	0,74	0,08
KRR-lin	0,62	0,85	0,67	0,70	0,67	0,13
KRR-rbf	0,71	0,70	0,70	X	X	X

Otrzymane wyniki są zbliżone do wyników nanorurek borowych, jednak dla wszystkich modeli i zmiennych wartość R2 obliczana za pomocą

10-krotnego testowanie krzyżowego jest niższa. Zachowane są zależności dokładności modeli i zmiennych. Dla nanorurek azotku boru najlepsze wyniki uzyskują modele oparte na drzewach losowych oraz modele oparte o rozszerzenia regresji liniowej. Także dla tych układów zmienne nieperiodyczne prowadzą do dokładniejszych modeli niż zmienne uwzględniające periodyczność układu.

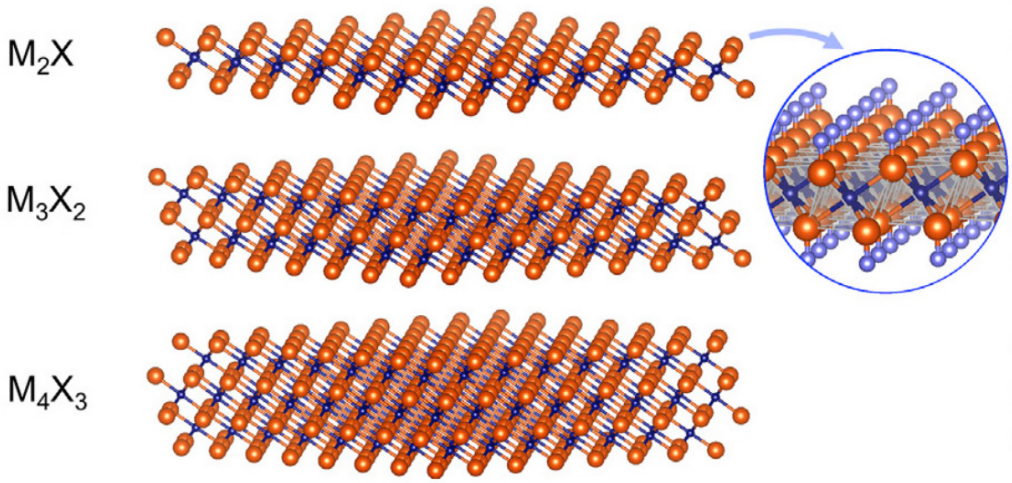
5.2.5. Wnioski

W tym rozdziale przedstawiono badania nad modelami sztucznej inteligencji opisującymi energię całkowitą oraz energię kohezji nanorurek azotku boru. Prace zostały wykonane na danych pochodzących z 4000 przeprowadzonych symulacji numerycznych za pomocą Quantum Espresso. Baza danych zawierała geometrię układów, ich energię całkowitą oraz energię kohezji. Przetestowanie algorytmów uczenia maszynowego oraz reprezentacji geometrii pozwoliło na stworzenie modeli o dokładności na poziomie 0,92 dla energii całkowitej oraz 0,87 dla energii kohezji (obliczanych za pomocą metryki R2 przy użyciu 10-krotnego testowania krzyżowego). Najwyższą dokładność osiągnęły modele oparte na lesie losowym dla reprezentacji położeń atomów za pomocą macierzy kulombowskiej lub macierzy ewalda.

5.3. Materiały warstwowe z rodziny MXenes

5.3.1. Opis układu

Termin MXenes [150] odzwierciedla unikalną dwuwymiarową strukturę materiałów, opisywaną formułą chemiczną $M_{n+1}X_nT_z$, gdzie M jest metalem przejściowym, X oznacza węgiel lub azot, $n = 1, 2, 3, \ldots$, a T_z odpowiada grupom funkcyjnym na powierzchni (np. -OH, -O, -F). Geometrię MXenes przedstawia rysunek 5.4.



Rysunek 5.4: Schemat geometrii struktur dwuwymiarowych MXenes.

Badania nad materiałami MXenes szybko się rozwijają od momentu zsyntetyzowania pierwszego przedstawiciela $Ti_3C_2T_z$ w 2011 roku przez Naguib *et al.* [151] i później dalszych [152]. Teoretyczne przewidywania nowych MXenes oraz analiza ich stabilności są przedstawione w szeregu prac [153]. Wiele teoretycznie przewidzianych struktur MXenes zostało zsyntetyzowane [154] i stale rośnie potencjał zastosowań aplikacyjnych tych związków [155, 156].

Badania nad właściwościami MXenes prowadzone są bardzo dynamicznie i poszukiwane są nowe zastosowania. W szczególności zastosowania w medycynie są bardzo obiecujące. Pierwsze wyniki wskazywały na przeciwbakteryjne właściwości $Ti_3C_2T_z$ [157] wraz z założonym mechanizmem działania nano-ostrza (ang. nano-blade mechanism) [158], oraz brak tego efektu dla Ti_2CT_z [159]. Jednak pogłębione badania dotyczące właściwości i struktury materiału ostatecznie wykazały brak właściwości antybakteryjnych czystego $Ti_3C_2T_z$ [160]. W przypadku toksyczności MXenes pierwsze badania *in vitro* dotyczące wielowarstwowego $Ti_3C_2T_z$ wykazały potencjalne zagrożenie związane z wytwarzaniem reaktywnych form tlenu (ROS) [159]. Dalsze prace uwidoczniły różnice we właściwościach toksykologicznych w świetle stechiometrii MXenes (tj. $Ti_3C_2T_z$ lub Ti_2CT_z) [161]. Okazało się również, że na cytotoksyczność oraz stabilność związków wpływ mają także parametry makroskopowe takie jak grubości płatków [162]. Obecnie stajemy przed wyzwaniem szybkiego i wiarygodnego wyodrębniania najbardziej obiecujących przedstawicieli MXenes o najwyższym potencjale zastosowania w medycynie i najniższych zagrożeniach cytotoksykologicznych. W niniejszym rozdziale przedstawimy proces budowania modeli opisujących energię kohezji oraz cytotoksyczności MXenes.

5.3.2. Zbiory danych

Badania rozpoczęto od zebrania zbioru danych pozwalającego na zbudowanie modelu opisującego właściwości cytotoksyczne rodziny materiałów MXenes. Zebrane dane pochodzą z dwóch źródeł: eksperymentów i obliczeń numerycznych. Dane doświadczalne zawierają wyniki pomiarów cykotoksyczności zsyntetyzowanych MXenes oraz informacje o badanych próbkach. Dane teoretyczne pochodzą z symulacji numerycznych MXenes i zawierają informację o geometrii oraz energii całkowitej układu.

Podział ten został zachowany podczas budowania modeli. Wykonano niezależne prace dla danych doświadczalnych oraz teoretycznych, następnie połączono oba zbiory w jeden i na nim także wykonano analizę.

Zbiór I - dane eksperymentalne

Zbiór I zawiera dane eksperymentalne pochodzące z literatury. W tabeli 5.9 zebrano informacje o pochodzeniu wszystkich danych eksperymentalnych.

W tabeli D.1 przedstawiono wybrane zmienne opisujące układ wraz z zakresem ich wartości. Na podstawie znalezionych w literaturze informacji o cytotoksyczności oraz parametrach mierzonych związków wybrano zmienne niezależne opisane we wszystkich publikacjach. Następnie dla zmiennych kategorycznych wykonano etykietowanie oraz dla zmiennych opisujących funk-

Tabela 5.9: Szczegółowe informacje o pochodzeniu danych doświadczalnych dotyczących cytotoksyczności MXenes

MXenes:	
Ti_3C_2	Refs.[163–185]
Ta_4C_3	Refs. [186–188]
Nb_2C	Refs.[189–192]
Ti_2C	Ref. [193]
Mo_2C	Refs. [194]
$Mo_{1.33}C$	Ref.[195]
Nb_4C_3	Refs. under publications
V_2C	Refs. [183, 196]
Ti_2N	Ref. [162]
Ti_4N_3	Ref. [197]

cjonalizację powierzchni zastosowano One-hot-encoder. Tabela opisująca dane oraz ich pochodzenie została zamieszczona w artykule [12].

Zbiór I zawierał oryginalnie 8 zmiennych niezależnych. Wstępne analizy pokazały istotność funkcjonalizacji powierzchni, więc zostały one wyodrębnione jako osobne zmienne. Występowanie grupy funkcyjnej na powierzchni było traktowane jako osobna zmienna jeśli pierwiastek pojawiał się przynajmniej w 10% wszystkich obserwacji. Doprowadziło to, do znacznego wzrostu liczby rozważanych zmiennych, ale jak to zostanie pokazane w kolejnych rozdziałach są one kluczowe dla modelu. Zmienne te pozwalają na interpretację fizyczną wyników oraz postawienie hipotezy o pochodzeniu cytotoksyczności badanych układów. Podczas budowania modelu korzystano z 17 zmiennych niezależnych. Wszystkie zmienne były kategoryczne. Zmienna zależną dla zbioru I jest cytotoksyczność.

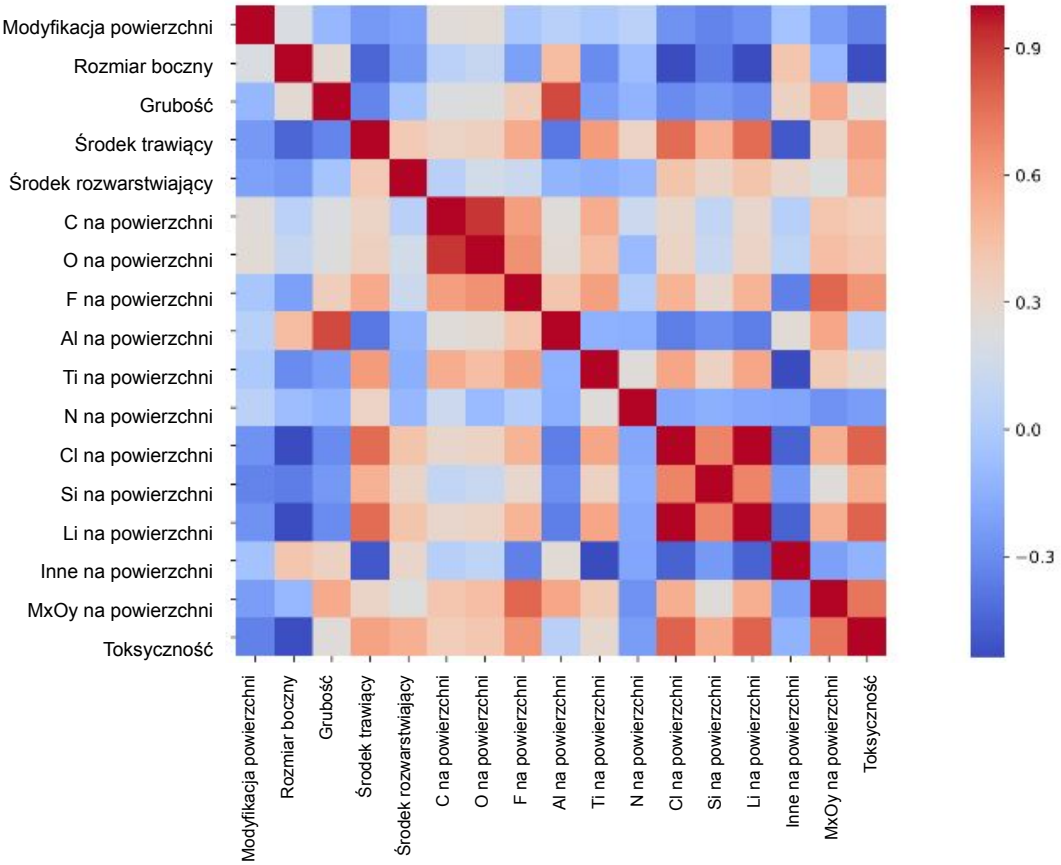
Stosunek ilości zmiennych do liczby obserwacji w zbiorze treningowym wymagał szczególnie dokładnego wyboru istotnych zmiennych. Analiza istotności zmiennych została przedstawiona w kolejnym rozdziale.

Zbadano korelację pomiędzy zmiennymi niezależnymi w celu określenia możliwych zależności pomiędzy nimi i eliminacji zmiennych zależnych. Macierz korelacji została przedstawiona na rysunku 5.5.

Analizując macierz korelacji można zaobserwować silne korelacje pomiędzy częścią zmiennych niezalanych oraz zmienną zależną. Szczególnie mocno widoczna jest korelacja pomiędzy obecnością grup funkcyjnych na powierzchni (zwłaszcza chloru, litu oraz M_xO_y) oraz cytotoksycznością. Zależności te zostały potwierdzone w budowanych modelach.

Podczas analizy korelacji pomiędzy zmiennymi należy pamiętać o dwóch kwestiach:

— analizowana jest korelacja istniejąca w zbiorze danych, a nie w całej populacji. Więc należy ostrożnie formułować tezy, szczególnie w przypadku małych zbiorów danych,
— analizowana jest korelacja liniowa, więc jeśli zależność pomiędzy zmiennymi jest bardziej skomplikowana może być ona niewidoczna. Ponadto analizowana jest zależność jedynie pomiędzy parą zmiennych. Zależności



Rysunek 5.5: Macierz korelacji zmiennych opisujących cytotoksyczność materiałów MXenes w zbiorze I. Kolorami zostały zaznaczono wartości korelacji pomiędzy zmiennymi, używając konwencji: kolor czerwony silna dodatnia korelacja, kolor niebieski silan korelacja ujemna

pomiędzy grupami zmiennych wymagają innego podejścia na przykład zastosowania biblioteki MDFS [198, 199].

Oryginalny zbiór I jest niezbalansowany. Stosunek obserwacji nietoksycznych do toksycznych wynosi 3:1. Wymusza to stosowanie dedykowanych metod dla niezbalansowanych danych w celu otrzymania wiarygodnych wyników oraz modelu o dobrych własnościach uogólniania.

Zbiór II - dane teoretyczne

Zbiór teoretyczny zawiera zmienne będące wynikiem symulacji numerycznych. Wszystkie dane pochodzą z bazy danych Materials Project [90, 91]. Obliczenia były wykonane programem VASP [49–52]. Oryginalny zbiór posiadał 61 obserwacji. Każda obserwacja zawierała informację o komórce elementarnej, położeniach i rodzajach atomów w komórce elementarnej, wartość przerwy energetycznej, energii kohezji (ang. Formation Energy), momentu magnetycznego oraz strukturę pasmową. Zbiór nie zawierał natomiast informacji o cytotoksyczności (dla części obserwacji można ją uzyskać ze zbioru I)

Do budowy modeli użyto zmiennych:

— Macierz kulombowska,
— Macierz ewalda,
— Macierz sinusoid,
— Funkcje symetrii,
— Ważone funkcje symetrii,
— Odcisk atomowy,

które zostały opisane w rozdziale 4.1.3.

Podczas pracy ze zbiorem II oraz generowania nowych zmiennych niezależnych trudnością była duża rozpiętość wielkości układów. Komórki elementarne zawierały od 2 do 30 atomów, co w przypadku zmiennych budowanych przez macierze prowadzi do 900 zmiennych niezależnych dla największego układu. Ponadto znaczna większość układów zawierała mniej niż 5 atomów, co oznacza że dla większości obserwacji 875 z 900 zmiennych miało wartość zero. Wynikiem tego jest mała stabilność modeli opartych na zbiorze teoretycznym oraz duże trudności przy uczeniu modelu i jego walidacji.

Macierz korelacji pomiędzy zmiennymi typu odciski atomowe przedstawiono na rysunku 5.6. Dla czytelności nazwy zmiennych zostały zastąpione numeracją.

Pokazuje on słabą korelację pomiędzy poszczególnymi zmiennymi niezależnymi. W związku z liczbą zmiennych niezależnych w pozostałych rodzinach zmiennych położeniowych nie jest możliwe wykonanie czytelnej wizualizacji pełnej macierzy korelacji. Otrzymane wyniki także pokazują słabą korelację pomiędzy zmiennymi niezależnymi.

Zbiór III - połączone zbiory I i II

Zbiór III jest rozszerzeniem zbioru I o dane ze zbioru II. Do obserwacji ze zbioru I dodano zmienne oparte na położeniach atomów w komórce elementarnej. Zastosowano dwa rodzaje zmiennych pozycyjnych: Funkcje symetrii i Ważone funkcje symetrii. Celem połączenia obu zbiorów było sprawdzenie jak duży wpływ na cytotoksyczność ma geometria układu.

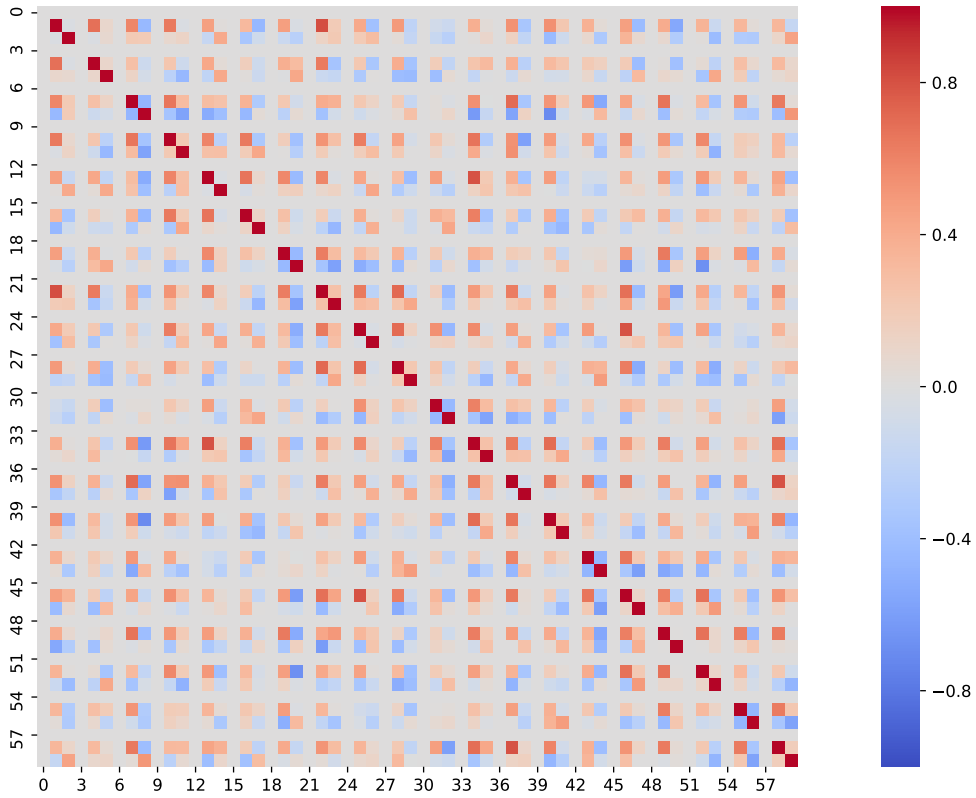
Zbadano także korelację pomiędzy zmiennymi ze zbioru I, a zmiennymi ze zbioru II. Największa wartość korelacji wynosi 0,3. Macierz korelacji pomiędzy zmiennymi przedstawiono na rysunku 5.7.

5.3.3. Przewidywanie cytotoksyczności

Zbiór I

W celu wyodrębnienia najistotniejszych zmiennych dla modelu zastosowano dwie techniki: analizę istotności zmiennych obliczaną za pomocą lasu losowego oraz redukcję wymiarowości przez regularyzację L1. Podczas pracy z obiema technikami należy zoptymalizować hiperparametry: liczbę drzew w lesie losowym lub wartość współczynnika regularyzacji. Oba parametry zostały zoptymalizowane za pomocą procedury grid search, opisanej w rozdziale 3.6. Wybrano wartość 1000 dla liczby drzew losowych oraz 0,1 dla współczynnika regularyzacji.

Wyniki istotności zmiennych otrzymane z lasu losowego przedstawia rysunek 5.8.

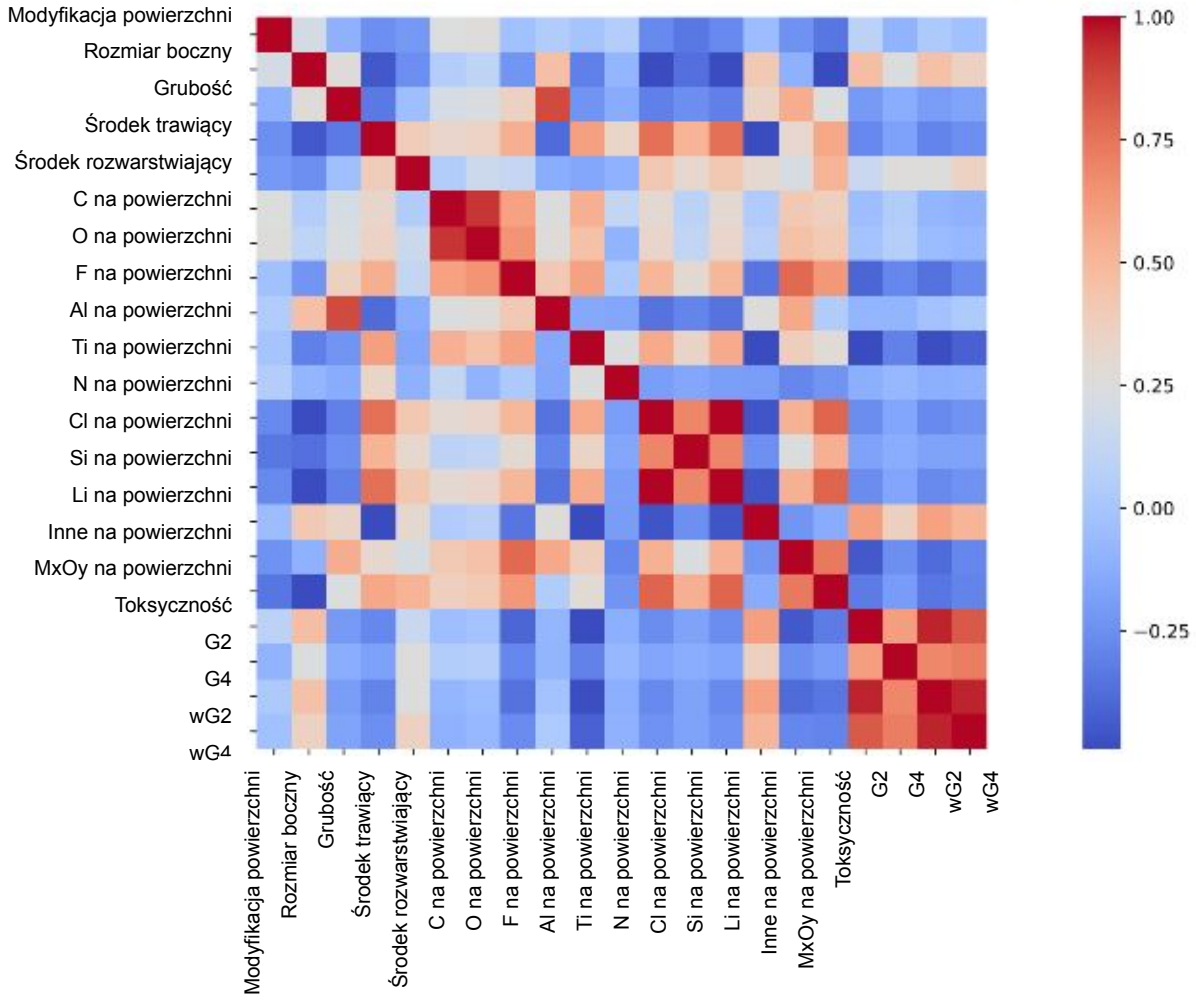


Rysunek 5.6: Macierz korelacji zmiennych opisujących cytotoksyczność materiałów MXenes w zbiorze II, kolorami zostały zaznaczono wartości korelacji pomiędzy zmiennymi, używając konwencji: kolor czerwony silna dodatnia korelacja, kolor niebieski silan korelacja ujemna. Dla czytelności nazwy zmiennych zostały zastąpione ich numerami w zbiorze treningowym.

Zmienne można podzielić na trzy grupy według istotności dla modelu: bardzo istotne dla modelu, istotne dla modelu i nieistotne.

Zmienne bardzo istotne to: obecność tlenku metalu (M_xO_y) i litu na powierzchni. Zmienne istotne to: modyfikacja powierzchni, środek trawiący, wielkość siatki, obecność chloru na powierzchni. Do zmiennych nieistotnych można zaliczyć pozostałe zmienne. Wartości istotności zmiennych z lasu losowego pozwalają powiedzieć, że najistotniejszą zmienną jest obecność M_xO_y na powierzchni, drugą jest obecność litu na powierzchni. Kolejne cztery zmienne mają bardzo zbliżone wartości istotności i nie można określić jaka jest dokładnie kolejność ich istotności w modelu. Podkreślić należy fakt, że rozdzielenie klas istotności jest bardzo wyraźne, co ułatwia budowanie modelu.

Wyniki z modelu liniowego z regularyzacją L1 nie mówią o istotności zmiennej, a jedynie mogą wyznaczyć zmienne nieistotne. Podczas uczenia modelu parametry dopasowania zmiennych nieistotnych dla modelu są zero-

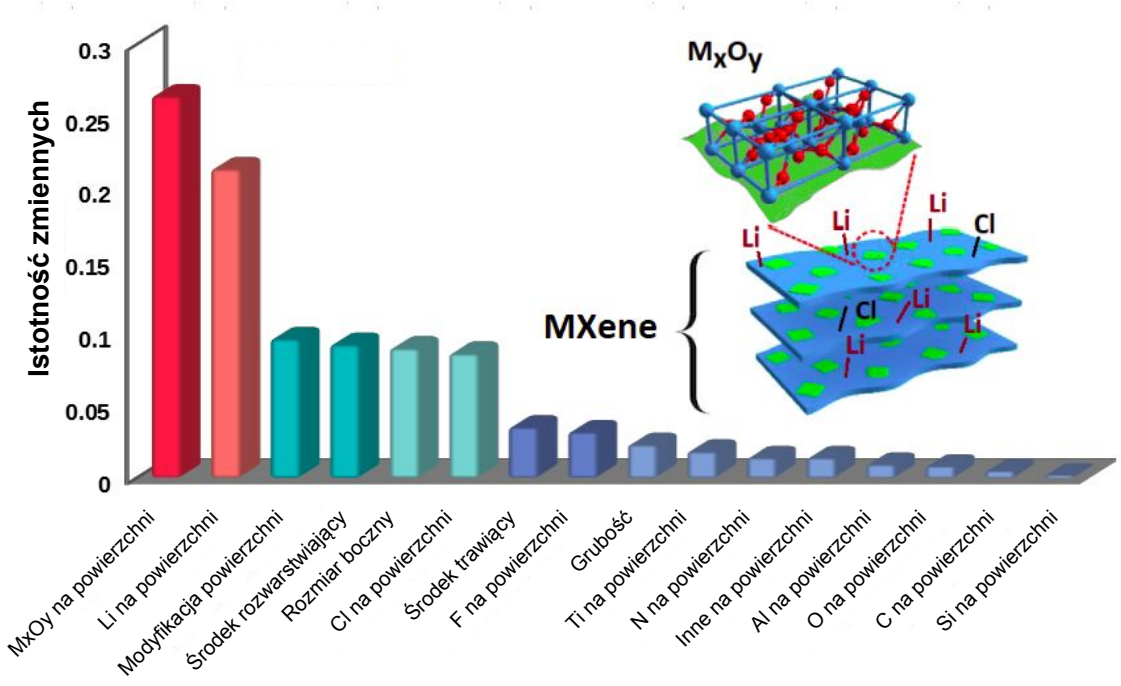


Rysunek 5.7: Macierz korelacji zmiennych opisujących cytotoksyczność materiałów MXenes w zbiorze III, kolorami zostały zaznaczono wartości korelacji pomiędzy zmiennymi, używając konwencji: kolor czerwony silna dodatnia korelacja, kolor niebieski silan korelacja ujemna

wane. Ilość zerowanych parametrów zależy od wartości współczynnika regularyzacji. Za istotne zmienne uważa się te, które mają niezerowy współczynnik dopasowania. Wartość liczbowa współczynnika dopasowania nie mówi o istotności zmiennej, gdyż zależy ona od zakresu wartości zmiennej. Jedynie jeśli wszystkie zmienne są unormowane można próbować określać poziom ich istotności. Jednak dla zmiennych kategorycznych nie można takiej możliwości. Zmienne o niezerowych współczynnikach dopasowania to: obecność M_x0_y na powierzchni, modyfikacja powierzchni, środek trawiący, wielkość siatki, obecność chloru na powierzchni oraz obecność litu na powierzchni.

Dla obu metod najistotniejsze zmienne są takie same. Fakt ten pozwala używać regresji logistycznej oraz liniowych zależności cytotoksyczności od zmiennych niezależnych.

Do budowania zmiennych wykorzystano także algorytm PCA (patrz 3.10.6).



Rysunek 5.8: Istotność zmiennych opisujących cytotoksyczność materiałów MXenes w zbiorze I uzyskana za pomocą lasu losowego. Rysunek pochodzi z [12].

Wyniki uzyskane za jego pomocą zostały przedstawione w dalszej części tego rozdziału.

Podczas budowania modelu przetestowano szereg algorytmów, zostały one zebrane w tabeli 5.10.

Tabela 5.10: Używane algorytmy uczenia maszynowego oraz ich oznaczenia

Nazwa algorytmu	oznaczenie
Regresja logistyczna z regularyzacją L1	regLOG-l1
Regresja logistyczna z regularyzacją L2	regLOG-l1
Lasy losowe	regRF
SVM z liniową funkcją jądra	regSVM-lin
SVM z funkcją jądra rbf	regSVM-rbf
SVM z sigmoidalną funkcją jądra	regSVM-sig
Ekstremalnie losowe drzewa	XRT
XGBoost	XGBoost
Adaboost oparty na regresji logistycznej	ADA-LOG
Adaboost oparty na drzewach losowych	ADA-tree
Regresja grzbietowa z jdrem liniowym	KRR-lin

W celu uzyskania najdokładniejszego przewidywania cytotoksyczności zbudowano szereg modeli opartych na różnych algorytmach oraz liczbie zmiennych. Zbiór I jest niezbalansowany dlatego przetestowano metody dedykowa-

ne dla takich zbiorów: algorytmy ważone i rozszerzanie zbioru algorytmem SMOTE (opis metod znajduje się w rozdziale 4.2). Wyniki obu metod porównano z wynikami niezbalansowanego zbioru danych.

Niezbalansowany zbiór I Pierwszym zestawem zbudowanych modeli były modele budowane na niezbalansowanym zbiorze I. W celu porównania różnych modeli użyto 10-cio krotnego testowania krzyżowego. Wyniki zostały przedstawione w tabeli 5.11.

Tabela 5.11: Dokładność przewidywania cytotoksyczności MXenes dla niezbalansowanego zbioru I oraz zbioru zbalansowanego przez SMOTE i przez wykorzystanie algorytmów ważonych oraz zbioru III z wykorzystanie algorytmów ważonych.

	Zbiór I			Zbiór III
	niezbalansowane	zbalansowane dane:		
ML algorytm	dane	Weight	SMOTE	Weight
regRF	0,747	0,826	0,833	0,845
regLOG-l1	0,700	0,776	0,783	0,845
regLOG-l2	0,720	0,798	0,783	0,727
regSVM-lin	0,867	0,725	0,808	0,781
regSVM-rbf	0,662	0,917	0,933	0,876
regSVM-sig	0,555	0,543	0,4042	0,545
ADA-tree	0,700	0,826	0,833	0,768
ADA-LOG	0,753	0,810	0,800	0,800
XRT	0,722	0,776	0,750	0,793
XGBoost	0,747	0,836	0,808	0,836
KRR-lin	0,872	0,751	0,812	0,814

Dla niezbalansowanego zbioru danych wszystkie algorytmy działają na pograniczu klasyfikacji wszystkich obserwacji jako nietoksyczne. Odpowiada to dokładność na poziomie 0,75. Najgorsze wyniki uzyskał model regSVM-sig i były one na poziomie 0,5. Najwyższą dokładność uzyskał model regSVM-lin na poziomie 0,87. Na podstawie wyników trudno wyciągnąć jakiekolwiek wnioski poza nieliniową zależnością pomiędzy cytotoksycznością a zmiennymi niezależnymi.

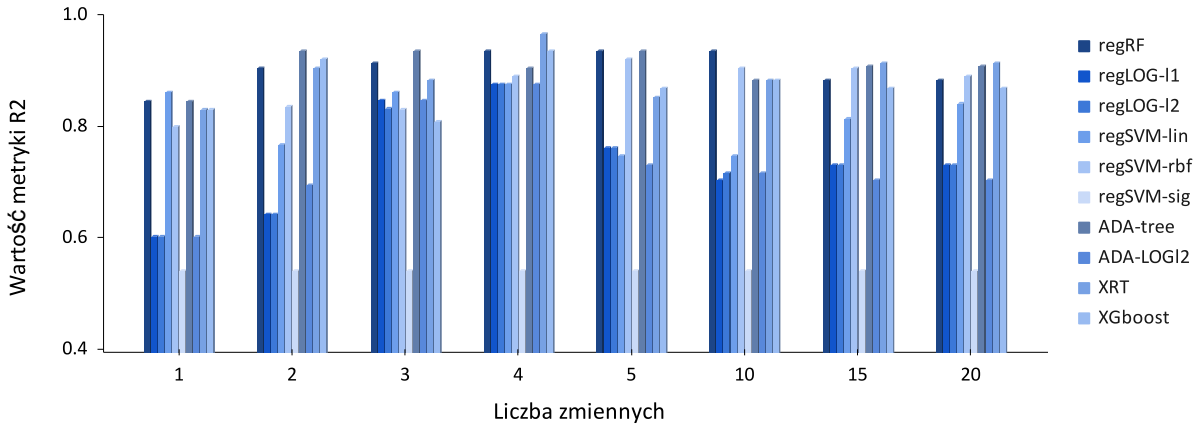
Zbalansowany zbiór I - algorytmy ważone Następnym krokiem było zastosowanie algorytmów ważonych. Pozwalają one na usunięcie niezbalansowania zbioru danych bez konieczności ingerencji w sam zbiór. Wyniki zostały przedstawione w tabeli 5.11. Niemal wszystkie wyniki są lepsze niż w przypadku niezbalansowanych danych, co zgadza się z oczekiwaniami. Najwyższą dokładność uzyskał model regSVM-rbf na poziomie 0,92. Świadczy to o silnej nieliniowej zależności cytotoksyczności od zmiennych niezależnych. Najniższą dokładność ma model regSVM-sig i nie widać zmiany w stosunku do niezbalansowanych danych. Modele oparte na regresji logistycznej i XRT poprawiły nieznacznie dokładność w stosunku do niezbalansowanych danych, ale w dalszym ciągu są na pograniczu dokładności.

Zbalansowany zbiór I - algorytm SMOTE Przetestowano, także działanie algorytmu SMOTE do rozszerzenia mniej licznej klasy. Otrzymane wyniki przedstawiono w tabeli 5.11. Większość algorytmów ma lepszą dokładność od algorytmów ważonych. Najwyższą dokładność miał model regSVM-rbf na poziomie 0,93. Najniższą dokładność ma model regSVM-sig i jest ona niższa znacznie niż dla pozostałych metod.

Pomimo lepszej dokładności stosowanie algorytmu SMOTE prowadzi do trudności w interpretacji wyników. SMOTE generuje wartości zmiennych dyskretnych, które dla wygenerowanych obserwacji są niecałkowite. Prowadzi to do zmiennych opisujących obecność pierwiastka na powierzchni o wartości niecałkowitych na przykład: lit występuje na powierzchni w stopniu 0,6 a nie występuje w stopniu 0,4. Z tego powodu zdecydowano się nie stosować algorytmu SMOTE.

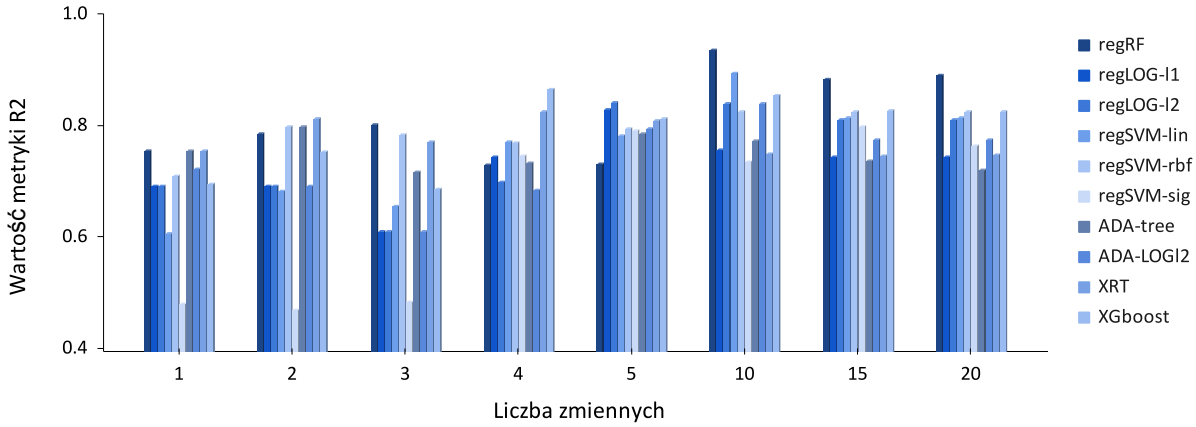
Dalsze prace nad zbiorem danych były wykonywane przy zastosowaniu algorytmów ważonych.

Zależność wyników od liczby zmiennych Zbadano zależności pomiędzy dokładnością klasyfikacji, a liczbą zmiennych niezależnych używanych w modelu. Zastosowano dwie metody zmniejszania wymiarowości problemu: wybór zmiennych zgodnie z istotnością obliczaną przez las losowy oraz tworzenie zmiennych algorytmem PCA. Regularyzacja L1 pozwala także na redukcję wymiarowości problemy, ale nie pozwala wprost kontrolować liczby zmiennych używanych w modelu. Dla każdej z technik zbadano dokładność jako funkcje liczby zmiennych. Wyniki zostały przedstawione na rysunku 5.9 dla zmiennych wybieranych lasem losowym i na rysunku 5.10 dla zmiennych generowanych praz PCA.



Rysunek 5.9: Wartość metryki R2 w funkcji liczby najistotniejszych zmiennych lasu losowego podczas klasyfikacji cytotoksyczności MXenes.

W obu przypadkach zastosowanie dwóch lub trzech zmiennych pozwala uzyskać bardzo wysoką dokładność modelu. Zwiększenie liczby zmiennych powyżej pięciu prowadzi do zmniejszenia dokładności. Wynika to z przeuczania modelu, który dla tej samej liczby obserwacji musi dopasować więcej współczynników przez co samo dopasowanie ma mniejszą dokładność. Na rysunku



Rysunek 5.10: Wartość metryki R2 w funkcji liczby zmiennych generowanych przez PCA podczas klasyfikacji cytotoksyczności MXenes.

5.9 widać, że dla znacznej części algorytmów po użyciu dwóch najistotniejszych oryginalnych zmiennych dodawanie kolejnych nie poprawia znacząco dokładności. Podczas, gdy wszystkie algorytmy oparte na regresji logistycznej (regLOG-L1, regLOG-L2, regSVM-lin, ADA-LOG) potrzebują około pięciu zmiennych aby uzyskać dokładność na poziomie 0,85. Podobną obserwacje można znaleźć dla PCA na wykresie 5.10. Dla wszystkich metod najgorszą dokładność uzyskuje regSVM-sig, a jego dokładność bardzo słabo zależy od liczby zmiennych. Niezależnie od wybranego algorytmu selekcja oryginalnych zmiennych daje lepsze wyniki niż zmienne budowane za pomocą PCA. Pozwala to na wysunięcie wniosku o dobrym wyborze oryginalnych zmiennych, a próby budowania syntetycznych zmiennych nie powodują poprawy działania modelu. Wynika to także ze specyfiki algorytmu PCA, który buduje zmienne w taki sposób aby były one możliwie najmniej skorelowane, jednak w tym przypadku za wynik odpowiada konkretna kombinacja dwóch/trzech zmiennych, a nie ich liniowa kombinacja, która jest wynikiem działania PCA.

Zbiór II

Zbór II zawiera wyniki symulacji numerycznych, zatem nie zawiera informacji o cytotoksyczności. Niemożliwe jest zatem zastosowanie uczenia z nadzorem (ang. Supervised Learning). Możliwe jest jednak zastosowanie klastrowania obserwacji i na podstawie danych ze zbioru I wykonanie walidacji wyniku (oba zbiory mają część wspólną). Wykorzystano algorytm centroidów z dwoma klastrami do analizy danych. Użyto zmiennych położeniowych:

— Macierz kulombowska,
— Macierz ewalda,
— Macierz sinusoid,
— Ważone funkcje symetrii,
— Odcisk atomowy.

Z pośród wszystkich obserwacji ze zbioru II pięć pokrywa się z obserwacjami ze zbioru I czyli znana jest ich cytotoksyczność. Na tej podstawie można oszacować poprawność działania modelu. Stosowane metody grupują

obserwacje i bez dodatkowej informacji nie można określić, która klasa odpowiada związkom toksycznym, a która nietoksycznym. Model kastrowania uważa się za poprawny jeśli wszystkie obserwacje toksyczne (znane ze zbioru I) są w jednej klasie, a wszystkie obserwacje nietoksyczne w drugiej. Wyniki testowanych modeli i zmiennych zostały przedstawione na rysunku 5.11.

	Cr2C	Ta4C3	Ti4C3N	Zr3N2	Mo2N	Ti3N2	V3C2	V4C3	Mo3N2	Mo2C	Cr3N2	Cr3N2
Mocna kowariancja	1	1	1	-1	1	1	1	1	1	1	-1	1
Jednoklasowy SVM	1	1	1	1	1	1	1	1	1	-1	-1	-1
Las izolacyjny	1	1	1	1	1	1	1	1	1	1	1	-1
	Ta3C2	Cr3N2	Ta2N	Nb2N	Mo4C3	Cr3N2	Zr2N	Nb4C3	Hf3N2	V4C3	Ta2N	Ti2N
Mocna kowariancja	1	1	1	1	1	1	1	1	1	1	-1	1
Jednoklasowy SVM	-1	-1	1	1	1	1	1	1	1	1	1	-1
Las izolacyjny	1	1	1	1	1	-1	1	1	1	1	-1	1
	Mo2N	Nb2C	Ti2N	Sc2C	Hf3N2	Nb2C	Cr3C2	Cr3N2	Cr3N2	Zr4C3N	Cr3N2	Hf2N
Mocna kowariancja	1	1	1	1	-1	1	1	1	1	1	1	-1
Jednoklasowy SVM	-1	-1	-1	1	1	1	1	1	1	1	1	1
Las izolacyjny	1	1	1	1	1	1	1	1	-1	1	-1	-1
	V2C	Mo2C	Ti2C	Nb2N	Sc4C3	Nb4C3	V2C	V2N	Mo3C2	Hf4N3	Hf2CN	Nb4C3
Mocna kowariancja	1	1	1	1	-1	1	1	1	1	1	1	1
Jednoklasowy SVM	-1	1	1	1	1	1	1	1	1	1	-1	-1
Las izolacyjny	1	1	1	1	1	1	1	1	1	1	1	1
	Hf4C3N	Mo2C	Cr3N2	Ta4N3	Ti3C2	Cr3C2	Cr2N	V2C	Ta2C	Cr3N2	Hf4N3	Ta4C3
Mocna kowariancja	1	1	1	-1	1	1	1	1	1	1	-1	-1
Jednoklasowy SVM	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1	-1
Las izolacyjny	1	1	1	-1	1	1	1	1	1	1	-1	-1

Rysunek 5.11: Przewidywana cytotoksyczność MXenes przy użyciu uczenia bez nadzoru. Kolor tła oznacza zmierzoną cytotoksyczność, znak i kolor wartości oznacza wynik działania modelu. Wartości oznaczone na niebiesko oznaczają związki niecytoksyczne, na pomarańczowo cytotoksyczne.

Żaden testowany model dla żadnego zestawu zmiennych nie zaklasyfikował wszystkich obserwacji toksycznych do jednej klasy, a wszystkie obserwacje nietoksyczne do drugiej.

Otrzymane wyniki pozwalają wysunąć tezę, że cytotoksyczność związków MXenes nie zależy od ich geometrii.

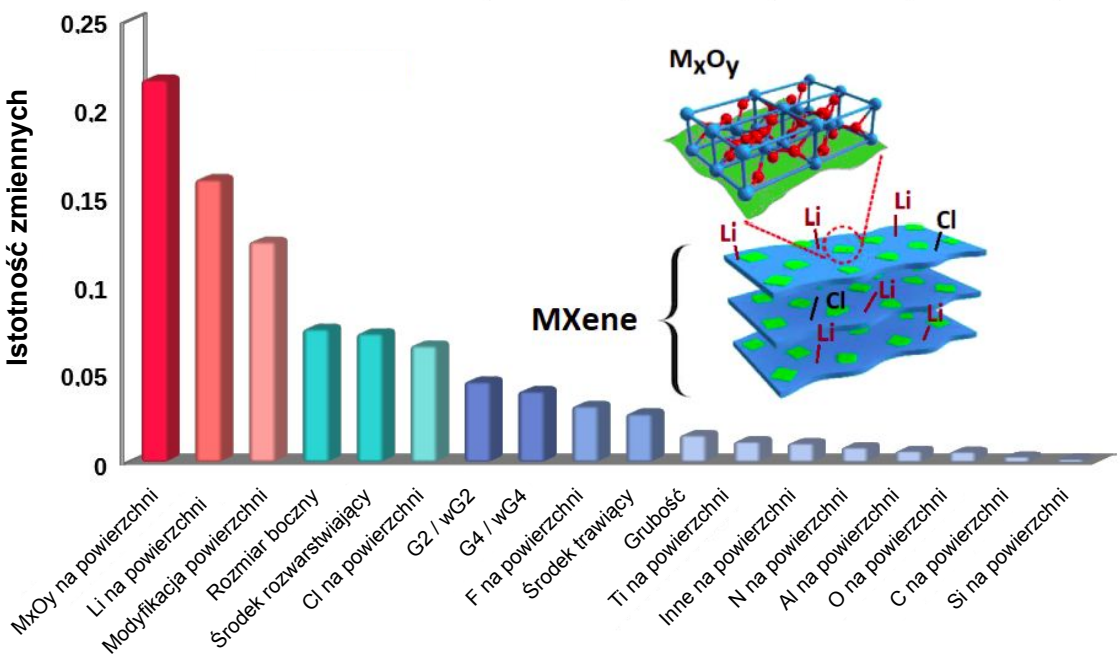
Przetestowano także algorytmy detekcji anomalii. Ponieważ w zbiorze I znacząca większość obserwacji odpowiada jednej klasie starano się sprawdzić czy w zbiorze II także jedna klasa jest istotnie przeważająca w stosunku do drugiej. Przetestowano algorytmy: Jednoklasowa maszyna wektorów nośnych (ang. One-Class SVM) [200], las izolacyjny (ang. Isolation Forest) [201] oraz mocna kowariancja (ang. robust covariance) [202]. Ich wyniki także nie pozwalają na wiarygodne stosowanie modelu.

Zbiór III

Zbór stanowi rozszerzenie zbioru I o zmienne opisujące geometrię ze zbioru II. Jako zmienne opisujące układ wybrano Funkcje symetrii i Ważone funkcje symetrii. Wyniki dokładności dla algorytmów ważonych przy 10-krotnego testowania krzyżowego przedstawia tabela 5.11 Wyniki nieznacznie różnią się od rezultatów dla zbioru I, ale nie można mówić o istotniej różnicy pomiędzy dokładnością pomiędzy oboma zbiorami.

Wykonano także analizę istotności zmiennych za pomocą lasu losowego. Wyniki zostały przedstawione na rysunku 5.12.

Wyniki istotności zmiennych są zbliżone do wyników w zbiorze I, w szczególności kolejność zmiennych i poziom ich istotności jest zachowany. Jedynie



Rysunek 5.12: Istotność zmiennych opisujących cytotoksyczność materiałów MXenes w zbiorze III uzyskana za pomocą lasu losowego. Rysunek pochodzi z [12].

granica pomiędzy poziomami istotności zmiennych jest mniej ostra. Na wykresie 5.12 jako zmienne SF i gSF podano wartość dla najistotniejszej zmiennej z obu rodzin. W obu przypadkach zmienne są na pograniczu zmiennych istotnych dla modelu.

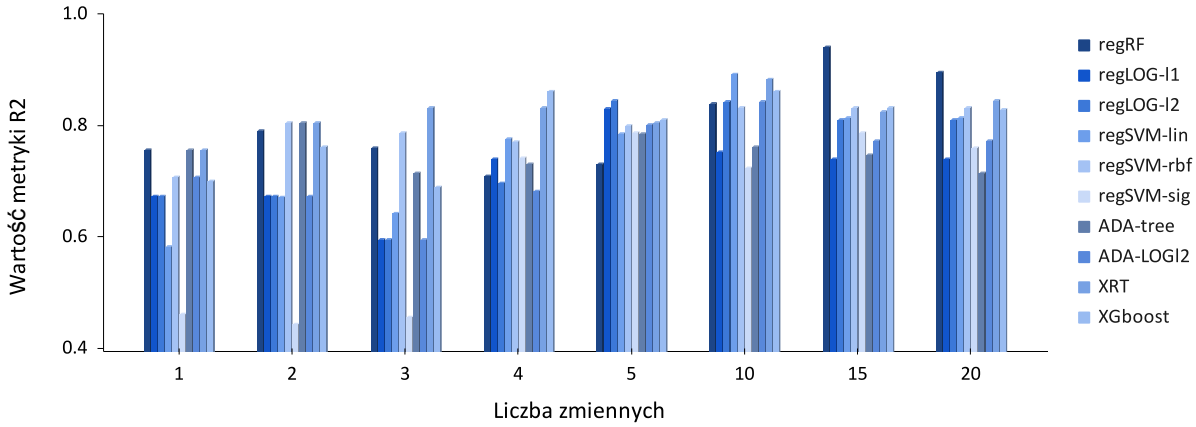
Przeanalizowano także wpływ liczby zmiennych na poprawność klasyfikacji. Wyniki zostały przedstawione na rysunkach 5.13 oraz 5.14. Zmienne były wybierane zgodnie z rosnącą istotnością według lasu losowego oraz przez algorytm PCA. Wyniki nie różnią się od tych dla zbioru I.

Prezentowane wyniki świadczą, że na cytotoksyczność MXenes kluczowy wpływ ma powierzchnia, w szczególności obecność M_xO_y oraz litu i chloru, a także modyfikacje powierzchniowa.

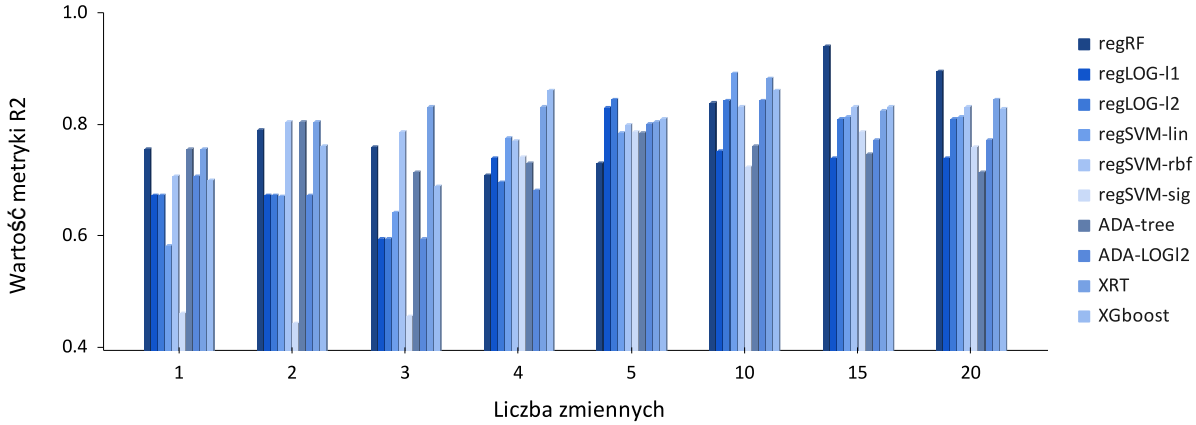
Przewidywanie cytotoksyczności dla niezbadanych związków

Zbudowane modele pozwalają na przewidzenie cytotoksyczności dla nowych związków. Jako najlepszy model wybrano regSVM-rgf dla zmiennych ze zbioru I. Na podstawie danych literaturowych zebrano wartości zmiennych konieczne do zasilenia modelu. Wyniki zostały zebrane w tabeli 5.12 (tabela została opublikowana w pracy [12]).

Otrzymane wyniki pozwolą na uniknięcie długotrwałych i kosztownych testów cytotoksyczności dla związków przewidywanych jako cytotoksyczne z dużym prawdopodobieństwem.



Rysunek 5.13: Wartość metryki R2 w funkcji liczby zmiennych wybieranych zgodnie z rosnącą istotnością z lasu losowego podczas przewidywania cytotoksyczności MXenes.



Rysunek 5.14: Wartość metryki R2 w funkcji liczby zmiennych generowanych przez PCA podczas przewidywania cytotoksyczności MXenes.

5.3.4. Przewidywanie energii kohezji

W celu przewidzenia czy dany związek jest stabilniejszy niż zbiór swobodnych atomów, co wpływa na możliwość jego zsyntetyzowana, przebadano modele opisujący energię kohezji. Przeanalizowano te same algorytmy uczenia maszynowego, które były testowane podczas badań nad nanorurkami. Lista algorytmów została zebrana w tabeli 5.2 oraz zmiennych położeniowych:

— Macierz kulombowska (CM),
— Macierz ewalda (ESM),
— Macierz sinusoid (SM),
— Funkcje symetrii: tylko rodziny zmiennych G2 (G2) tylko rodziny zmiennych G4 (G4), cała baza (SF),
— Ważone funkcje symetrii: tylko rodziny zmiennych G2 (wG2), tylko rodziny zmiennych G4 (wG4), cała baza (wSF) ,
— Odcisk atomowy (AF),

Tabela 5.12: Klasyfikacja cytotoksyczności MXenes wraz z jej prawdopodobieństwem dla związków nie zawartych w zbiorze treningowym.

MXene [cyt.]:	Prawdopodobieństwo	M_xO_y	Li	metoda syntezy	zanieczyszczenia powieszchniowe
Ti_3C_2 [174]	0,18	Brak	Obecny	LiF/HCl; Us.	Brak
Ti_3C_2 [175]	0,05	Brak	Brak	HF	Au
Ti_3C_2 [176]	0,18	Brak	Obecny	LiF/HCl; Us.	Brak
Ti_3C_2 [177]	0,10	Brak	Brak	HF; Us.	Brak
Ti_3C_2 [178]	0,10	Brak	Brak	HF; Us.	Brak
Ti_3C_2 [179]	0,06	Brak	Obecny	LiF/HCl	APTES+CEA
Ti_3C_2 [180]	0,07	Brak	Obecny	LiF/HCl; Us.	DNA, Pt, Pd
Ti_3C_2[181]	0,05	Brak	Brak	HF; Us.	Ag
Ti_3C_2 [182]	0,07	Brak	Obecny	LiF/HCl	L-ACC
Ti_3C_2 [183]	0,07	Brak	Obecny	LiF/HCl; Us.	PAS
Ti_3C_2 [184]	0,87	Obecny	Brak	LiF/HCl	Brak
Ti_3C_2 [185]	0,88	Obecny	Brak	LiF/HCl	Brak
V_2C [183]	0,05	Brak	Obecny	LiF/HCl; Us.	PAS
V_2C [196]	0,05	Brak	Brak	NaF/HCl	Brak
Nb_2C [191]	0,04	Brak	Brak	NaF/HCl	Brak
Nb_2C [192]	0,04	Brak	Brak	HF; 60^o	Brak
Ti_2N [162]	0,05	Brak	Brak	KF/HCl; Us.	Brak
$Mo_{1.33}C$ [195]	0,04	Brak	Brak	HF; TBAOH	No
Ti_4N_3[197]	0,03	Brak	Obecny	KF/LiF/NaF, TBAOH	No

Parametry użyte dla Funkcje symetrii i Ważonych funkcje symetrii są zebrane w tabeli 5.13, zostały one wybrane zgodne z sugestiami podanymi przez J. Behler w [101].

Tabela 5.13: Wartość parametrów użytych do wygenerowania Funkcje symetrii i Ważonych funkcje symetrii dla MXenes.

parametr	zakres wartości
R_c	6
N	10
R_s	$1, \ldots, N$
η^2	$\frac{1}{R_c}N^{1/N}, \ldots, \frac{1}{R_c}N^{N/N}$
λ	-1, 1
ζ	1, 2, 4, 16

Parametry użyte do wygenerowania Odcisku atomowego zostały wybrane zgodnie z [103] i znajdują się w tabeli 5.14.

Tabela 5.14: Wartość parametrów użytych do wygenerowania Odcisku atomowego dla związków MXenes.

parametr	zakres wartości
R_c	6
N	10
σ	$1, 5. \ldots, 11, 5$

Przetestowano dokładność przewidywań modeli oraz ich zdolności uogólniania. Zbadano dokładność modelu na zbiorze treningowym oraz na podstawie 10-cio krotnego testowania krzyżowego. Pierwszy test pozwala określić, czy model może prawidłowo odzwierciedlać zależność energii kohezji jako funkcję wybranych zmiennych niezależnych. Drugi opisuje zdolność uogólniania modelu dla obserwacji nieużytych podczas uczenia modelu. Jako miarę poprawności użyto dwóch metryk R2 oraz MSE. Wyniki dla metryki R2 zostały zebrane w tabeli 5.15, a wyniki dla metryki MSE w tabeli E.1. Modele były budowane na całym zestawie zmiennych (bez redukcji wymiarowości).

Tabela 5.15: Wartość metryki R2 dla przewidywania energii MXenes obliczona podczas uczenia modelu. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

R2	ESM	CM	SM	G2	G4	wG2	wG4	SF	wSF	AF
regRF	0,81	0,86	0,85	0,78	0,74	0,87	0,80	0,81	0,87	0,82
regLIN	0,98	1,00	0,99	0,27	0,19	0,26	0,19	0,42	0,41	0,72
regLIN-L1	0,69	0,68	0,65	0,00	0,00	0,20	0,19	0,00	0,25	0,15
regLIN-L2	0,97	0,99	0,99	0,22	0,11	0,26	0,19	0,31	0,41	0,63
regSVM-lin	0,88	X	X	0,14	0,09	0,17	X	0,23	0,12	0,60
regSVM-rbf	0,96	0,96	0,96	0,32	0,05	0,96	0,96	0,25	0,96	0,96
regSVM-sig	X	X	X	X	X	X	X	X	X	X
ADA-tree	0,88	0,89	0,89	0,79	0,74	0,87	0,83	0,89	0,89	0,89
ADA-LIN	0,63	0,88	0,84	0,35	0,15	0,35	0,08	0,70	0,64	0,53
XRT	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
KRR-lin	0,96	0,98	0,99	0,03	X	0,23	X	0,15	0,40	0,60
KRR-rbf	0,32	0,32	0,32	0,27	0,07	0,36	0,32	0,23	0,32	0,33

Otrzymane wyniki pozwalają wnioskować, że możliwe jest opisywanie energii kohezji jako funkcji zmiennych położeniowych. Dla wszystkich testowanych algorytmów zmienne G2, G4 mają najniższą wartość metryki R2, zatem najgorzej opisują energię kohezji. Dokładność poprawia się przy zastosowaniu obu rodzin funkcji jednocześnie (SF). Analogiczne obserwacje zachodzą dla zmiennych wG2, wG4 i wSF. Świadczy to o konieczności analizy nie tylko odległości pomiędzy atomami lub tylko zależności kątowej w lokalnym otoczeniu atomu, ale obu tych czynników jednocześnie (zmienne SF, wSF, AF). Jednocześnie, rozważanie układu jako całości pozwala na rozważanie jedynie oddziaływań pomiędzy atomami (zmienne ESM, CM, SM) bez dodatkowych informacji o geometrii układu. Ponadto CM dają lepsze wyniki niż jego rozszerzenie na przypadki periodyczne poprzez zmienne ESM. Wynik ten jest zgodny z obserwacją F. Feber [100]. Dla części modeli metryka R2 jest ujemna (oznaczona przez X w tabeli 5.15), co oznacza że model nie odzwierciedla istoty zjawiska i nie jest w stanie opisywać zależności pomiędzy energią kohezji a zmiennymi położeniowymi. Najgorsze wyniki otrzymano dla algorytmów SVM-sig oraz KRR-rbf. Dla zmiennych ESM, CM, SM modele liniowe mają bardzo wysoką wartość metryki R2. Wynika to z konstrukcji zmiennych. Zawierają one sumy cząstkowe energii kulombowskiej, która stanowi znaczną część całkowitej energii oddziaływania pomiędzy ato-

mami, widocznej w energii kohezji. Wartości MSE dla modeli o wysokiej wartości metryki R2 są mniejsze niż 0,01 eV. Otrzymane wyniki pokazują, że zmienne położeniowe mogą być używane do opisywania energii kohezji, ale jednocześnie w związku z wielkością zbioru treningowego mogą możliwe jest przeuczenie modeli.

Wyniki metryk R2 i MES uzyskane za pomocą dziesięciokrotnego testowania krzyżowego zostały zebrane odpowiednio w tabeli 5.16 i E.2.

Tabela 5.16: Wartość metryki R2 dla przewidywania energii MXenes obliczona dla dziesięciokrotnego testowania krzyżowego. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

R2	ESM	CM	SM	G2	G4	wG2	wG4	SF	wSF	AF
regRF	0,01	0,37	0,21	0,18	X	X	X	0,11	X	X
regLIN	X	X	X	X	0,13	X	X	X	X	X
regLIN-L1	X	0,02	X	X	X	X	X	X	X	X
regLIN-L2	X	X	X	X	X	0,06	X	X	X	X
regSVM-lin	X	X	X	0,02	X	X	X	X	X	X
regSVM-rbf	X	X	X	X	X	0,17	X	0,09	X	X
regSVM-sig	X	X	X	X	X	X	X	X	X	X
ADA-tree	0,25	X	0,20	X	X	0,30	X	X	X	X
ADA-LIN	X	X	X	X	X	X	X	X	X	X
XRT	X	X	0,23	0,38	X	X	X	X	X	0,20
KRR-lin	X	X	X	X	X	X	X	X	X	X
KRR-rbf	X	X	X	0,06	X	X	X	X	X	X

Wszystkie testowane kombinacje modeli oraz zmiennych położeniowych dają bardzo niskie wartości metryki R2. W większości przypadków wartość metryki R2 jest ujemna, a więc model nie opisuje poprawianie zależności pomiędzy zmienną zależną a zmiennymi niezależnymi. Wartości MSE są o co najmniej rząd wielkości wyższe niż dla zbioru treningowego. Świadczy to o przeuczeniu modelu. Otrzymane wyniki pokazują, także bardzo niską zdolność uogólniania zbudowanych modeli.

Aby zweryfikować wyniki sprawdzono również wartość oob (ang. out-of-bag) podczas budowania modelu opartego na lasie losowym. Otrzymane wyniki znajdują się w tabeli 5.17. Ponieważ oob odpowiada metryce R2 otrzymane

Tabela 5.17: Wartość oob dla przewidywania energii MXenes uzyskana podczas uczenia lasu losowego. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

	CM	SM	G2	G4	wG2	wG4	SF	wSF	AF
oob(R2)	0.22	0.14	X	X	0.12	0.00	X	0.23	X

rezultaty potwierdzają wyniki obliczone przy użyciu testowania krzyżowego.

Możliwe jest kilka powodów takiego zachowania modelu:

— model jest przeuczony,

— zbiór treningowy jest za mało liczny,
— stosowane modele nie opisują poprawnie zależności pomiędzy zmienną zależną a zmiennymi niezależnymi,
— wybrano zły zestaw zmiennych niezależnych,
— źle wybrano metrykę lub metodologię budowania i walidacji modeli.

Pierwszą przyczyną takiej postaci otrzymanych wyników może być przeuczenie modelu. Spowodowane jest ono zbyt dużą liczbą parametrów dopasowywanego modelu w stosunku do liczby obserwacji w zbiorze treningowym. Liczba parametrów modelu jest funkcją liczby zmiennych używanych do budowania modelu. W celu uproszczenia modelu zmniejszono liczbę zmiennych wykorzystując metodę PCA. Zależność wartości metryki R2 podczas uczenia modelu w funkcji liczby zmiennych niezależnych przedstawiono na rysunku F.2.

Otrzymane wyniki pokazują, że algorytmy oparte na drzewach decyzyjnych, niezależnie od rodzaju wybranych zmiennych, potrzebują 5 do 10 rodzaju zmiennych położeniowych, aby poprawnie opisać zjawisko (na zbiorze treningowym) . Natomiast algorytmy oparte na regresji liniowej potrzebują znacznie więcej zmiennych. Zatem możliwe jest zmniejszenie liczby zmiennych opisujących geometrię układu.

Analogiczne testy przeprowadzono stosując 10-cio krotne testowanie krzyżowe, a otrzymane wyniki przedstawiono na rysunku F.3.

Prezentowane wyniki pochodzące z testowania krzyżowego pokazują, że zmniejszenie liczby zmiennych nie wpływa na jakość modelu. Może to wynikać ze specyfiki algorytmu PCA oraz typu badanych układów. Liczba atomów w pojedynczej obserwacji w zbiorze treningowym jest w zakresie od trzech do trzydziestu, co odpowiada od 9 do 900 zmiennym dla CM, ESM, ES (ponieważ macierze te mają rząd równy liczbie atomów, a wiec liczba zmiennych zależy od kwadratu liczby atomów). Dodatkowo tylko pojedyncze obserwacje zawierają więcej niż 5 atomów. W związku z tym parametry dopasowania odpowiadające zmiennym położeniowym powyżej 25 są dopasowywane na podstawie kilku lub pojedynczych obserwacji, co prowadzi do bardzo dużych błędów wyznaczenia wartości parametrów modelu. Zastosowanie algorytmu PCA nie rozwiązuje problemu, gdyż nowe zmienne są obliczane na podstawie korelacji pomiędzy oryginalnymi zmiennymi, zatem wartości zmiennych powyżej 25 praktycznie nie pozostaje w korelacji z żadną lub prawie żadną inną zmienną. Zatem po zastosowaniu PCA największy wpływ na wartości nowych zmiennych mają zmienne posiadające niezerowe wartości jedynie w małej części obserwacji.

W celu zmniejszenia wpływu rozszerzania zerami macierzy CM, ESM i SM (a zatem i zmiennych powyżej 25), ręcznie wybrano 16 największych wartości zmiennych dla każdej obserwacji i użyto ich jako nowe zmienne położeniowe. Wyniki zostały zebrane w tabeli 5.18.

Otrzymane wartości metryki R2 podczas uczenia modelu pokazują, że jedynie algorytmy oparte na drzewach losowych są w stanie opisać zależność energii kohezji przy stosowaniu nowych zmiennych. Jednocześnie wyniki uzyskane z testowania krzyżowego w dalszym ciągu nie pozwalają wnioskować o poprawności działania modelu oraz jego zdolności uogólniania.

Tabela 5.18: Wartość metryki R2 dla przewidywania energii MXenes obliczona podczas uczenia modelu oraz testowania krzyżowego. Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

	zbiór treningowy			CV		
model	ESM	CM	SM	ESM	CM	SM
regRF	0,79	0,79	0,82	X	0,05	X
regLIN	0,26	0,37	0,38	X	X	X
regLIN-L1	0,26	0,37	0,38	X	X	X8
regLIN-L2	0,26	0,37	0,38	X	0,00	X
ADA-tree	0,87	0,88	0,90	X	X	X
ADA-LIN	0,42	0,59	0,45	X	0,05	X
XRT	1,00	1,00	1,00	0,02	X	X

Kolejną przyczyną niskiej zdolności uogólniania modeli może być wielkość zbioru treningowego. Przetestowane algorytmy wybrano w taki sposób aby liczba dopasowywanych parametrów była niewielka, co sprawia że mogą być stosowane dla małych zbiorów danych. Jednak sama liczba obserwacji powoduje, że budowa modelu jest trudna i może być obciążona dużymi błędami wynikającymi z próbkowania przestrzeni zmiennych niezależnych. Badane układy są stosunkowo do siebie podobne (można mówić o klastrach w przestrzeni zmiennych niezależnych, co zostało potwierdzone podczas analizy zbioru treningowego II). Jednocześnie niemożliwe jest uzyskanie większej liczby obserwacji, gdyż zbór II posiada teoretyczną geometrię wszystkich możliwych MXenes. W celu zwiększenie liczebności zbioru treningowego oraz poprawy spróbkowania przestrzeni wygenerowano syntetyczne dane za pomocą dodawania szumu losowego. Podczas generowania syntetycznych danych należy pamiętać, żeby zwielokrotnić jedynie podzbiór treningowy, podczas gdy zbiór walidacyjny powinien pozostać niezmieniony. Zwielokrotnienie danych przed podziałem na podzbiory treningowy i walidacjny prowadzi do powstanie korelacji pomiędzy zbiorami przez co wyniki z walidacji są niewiarygodne. Wielkość zbioru treningowego była zwielokrotniona pięcio-, dziesięcio- oraz dwudziestokrotnie w stosunku do oryginalnego zbioru. Szum do obserwacji był dodawany na dwa sposoby:

— szum był dodawany do położeń atomów, na podstawie których były generowane zmienne niezależne dla modelu (geometry),
— szum był dodawany bezpośrednio do zmiennych niezależnych (położeniowych) używanych w modelu (feature).

Otrzymane wyniki zostały przedstawione w tabelach 5.19 i 5.20.

Wyniki nie pokazują poprawy jakości modeli po zwielokrotnieniu danych w czasie uczenia modelu. Wyniki w dalszym ciągu sugerują przeuczenie modeli oraz ich brak zdolności uogólniania.

Wykonano także analogiczne testy dla zmiennych budowanych za pomocą algorytmu PCA dla liczby zmiennych 1, 2, 5, 10, 15, 20, 25, 30, 35, 50 i zwielokrotnienia podzbioru treningowego pięcio- dziesięcio- oraz dwudziestokrotnie. Rezultaty testowania krzyżowego pokazują, że jedynie XRT oraz

Tabela 5.19: Wartość metryki R2 dla przewidywania energii MXenes obliczona podczas uczenia modelu przy zwielokrotnieniu podzbioru treningowego 5, 10 i 20 razy. Szum dodawano do zmiennych niezależnych modelu (feature) oraz położeń atomów w układach (geometry).

	ESM			CM			SM		
zwielokrotnienie	x5	x10	x20	x5	x10	x20	x5	x10	x20
feature									
regRF	0,94	0,94	0,94	0,93	0,97	0,94	0,94	0,94	0,94
regLIN	0,96	0,95	0,96	0,98	0,73	0,97	0,98	0,98	0,98
regLIN-L1	0,69	0,69	0,68	0,73	0,97	0,73	0,70	0,70	0,70
regLIN-L2	0,96	0,96	0,96	0,98	0,86	0,97	0,98	0,97	0,97
ADA-tree	0,89	0,90	0,89	0,85	0,98	0,86	0,89	0,90	0,90
ADA-LIN	0,98	0,98	0,95	0,98	1,00	0,98	0,98	0,98	0,98
XRT	1,00	1,00	1,00	1,00	0,97	1,00	1,00	1,00	1,00
KRR-lin	0,96	0,95	0,95	0,97	0,39	0,97	0,98	0,97	0,97
KRR-rbf	0,42	0,50	0,56	0,36	0,37	0,39	0,35	0,40	0,40
geometry									
regRF	0,95	0,95	0,95	0,97	0,97	0,97	0,96	0,96	0,95
regLIN	0,99	0,99	0,98	1,00	1,00	1,00	1,00	1,00	1,00
regLIN-L1	0,69	0,69	0,69	0,74	0,74	0,74	0,71	0,71	0,71
regLIN-L2	0,99	0,99	0,99	0,99	0,99	0,99	1,00	1,00	1,00
ADA-tree	0,90	0,92	0,92	0,94	0,83	0,93	0,95	0,95	0,94
ADA-LIN	0,84	0,97	0,99	1,00	1,00	1,00	1,00	1,00	1,00
XRT	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00	1,00
KRR-lin	0,98	0,98	0,98	0,99	0,99	0,99	0,99	0,99	0,99
KRR-rbf	0,92	0,98	0,99	0,92	0,98	0,99	0,92	0,98	0,99

regresja liniowa z regularyzacją L1 (regLIN-L1) mają zadowalającą dokładność oraz zdolność uogólniania. regLIN-L1 dla zwielokrotnienia dziesięcio- i dwudziestokrotnego przy liczbie zmiennych powyżej 5 (dla zmiennych ESM i CM) daje wartość metryki R2 równą 0,13. Wyniki dla XRT przedstawia rysunek F.1.

Wyniki dla XRT nie pokazują żadnej prawidłowości czy asymptotycznego zachowania, zatem niemożliwe jest utrzymanie hipotezy o zdolności uogólniana modelu mimo, że najwyższy współczynnik R2 wynosi 0,39.

Przeprowadzono także testy zależności dokładności modelu od amplitudy szumu podczas generowania syntetycznych danych. Żadna testowana wartość nie poprawiła istotnie dokładności modeli. Ponieważ dla wszystkich testowanych wartości wyniki nie różniły się znacząco od prezentowanych w tabelach 5.19 i 5.20, nie zostały zamieszczone w tej pracy.

Przetestowane zmienne położeniowe były stosowane z powodzeniem w literaturze [99, 100, 103, 104, 203–214]. Autorzy tych prac niezależnie czy budowali modele przewidujące energię czy przerwę energetyczną podają jedynie MAE lub MSE bez podawania R2. W prezentowanych wynikach MAE i MSE podczas testowania krzyżowego jest na poziomie odpowiednio 0,5 i 0,2. Ponadto autorzy operują na znacznie większych zbiorach danych o liczebności

Tabela 5.20: Wartość metryki R2 dla przewidywania energii MXenes obliczona podczas testowania krzyżowego przy zwielokrotnieniu podzbioru treningowego 5, 10 i 20 razy. Szum dodawano do zmiennych niezależnych modelu (feature) oraz położeń atomów w układach (geometry).Wartość X oznacza ujemną wartość metryki, co odpowiada modelowi nie opisującemu zjawiska.

	ESM			CM			SM		
zwielokrotnienie	x5	x10	x20	x5	x10	x20	x5	x10	x20
feature									
regRF	X	X	X	X	X	X	X	X	X
regLIN	X	X	X	X	X	X	X	X	X
regLIN-L1	X	X	X	X	X	X	X	X	X
regLIN-L2	X	X	X	X	X	X	X	X	X
ADA-tree	X	X	X	X	X	X	X	X	X
ADA-LIN	X	X	X	X	0,01	0,01	X	X	X
XRT	X	X	X	X	X	X	X	X	X
KRR-lin	X	X	X	X	X	X	X	X	X
KRR-rbf	X	X	X	X	X	X	X	X	X
geometry									
regRF	X	X	X	X	X	X	X	X	X
regLIN	X	X	X	X	X	X	X	X	X
regLIN-L1	X	X	X	X	X	X	X	X	X
regLIN-L2	X	X	X	X	X	X	X	X	X
ADA-tree	X	X	X	X	X	X	X	X	X
ADA-LIN	X	X	X	X	X	X	X	X	X
XRT	X	X	X	X	X	X	X	X	X
KRR-lin	X	X	X	X	X	X	X	X	X
KRR-rbf	X	X	X	X	X	X	X	X	X

rzędu kilku tysięcy oraz o większym podobieństwie pomiędzy obserwacjami. W cytowanych pracach badane układy składały się z małej liczby pierwiastków oraz rozpiętość liczby atomów jest mniejsza niż w przypadku MXenes przez co problem jest lepiej postawiony.

5.3.5. Porównanie wyników przewidywania energii kohezji dla nanorurek borowych, nanorurek azotku-boru oraz MXenes

Otrzymane wyniki dla nanorurek borowych oraz azotku boru są znacznie lepsze niż wyniki przewidywania energii MXenes, pomimo podobnych liczebności zbiorów danych (50 dla nanorurek borowych i 61 dla MXenes). Wynika to z kilku powodów:

— Nanorurki borowe oraz azotku boru są układami znacznie prostszymi niż MXenes. Zbudowane są tylko z jednego lub dwóch rodzaju pierwiastków, co zmniejsza różnicę wartości ładunków poszczególnych atomów, a co za tym idzie i amplitudę samych zmiennych położeniowych. Ponadto można używać mniejszej liczby zmiennych.

— Testowane nanorurki borowe i azotku boru mają mniejszą amplitudę liczby atomów. Nanorurki użyte do budowania modelu składały się z 2 do 10 atomów podczas, gdy zbiór MXenes zawierał obserwacje o wielkości 2 do 30 atomów. Przekłada się to na liczbę zmiennych położeniowych. Przykładowo CM liczy 100 zmiennych dla nanorurek, a dla MXenes 900.
— Rozkład wielkości układów w obu zbiorach treningowych był inny. Dla nanorurek borowych i azotku boru był on równomierny, więc w zbiorze treningowym największych układów było około 12%, podczas gdy dla MXenes największe układy stanowiły około 2% całego zbioru danych. W połączeniu z dużą liczbą zmiennych uzupełnianych zerami prowadzi to do bardzo słabego spróbkowania dużej części przestrzeni zmiennych niezależnych dla MXenes.
— Wariancja wartości zmiennych położeniowych jest mniejsza dla nanorurek borowych i azotku boru niż dla MXenes.
— Geometria nanorurek jest mniej zróżnicowana niż geometria MXenes. Modele oparte na algorytmach uczenia maszynowego znajdują podobieństwa pomiędzy obserwacjami i na tej podstawie potrafią dokonywać uogólnienia. Zbyt duża różnorodność w danych nie pozwala na zbudowanie wiarygodnego modelu.

5.3.6. Wnioski

Przedstawione badania pokazują możliwości zastosowania algorytmów uczenia maszynowego do analizy własności struktur dwuwymiarowych na przykładzie MXenes. Przeanalizowano trzy zbiory danych: dane doświadczalne zawierające makroskopowy opis próbek oraz ich cytotoksyczność, dane teoretyczne zawierające geometrię układów wraz z ich energią oraz zbiór doświadczalno-teoretyczny stanowiący połączenie danych doświadczalnych i teoretycznych. Zbudowano szereg modeli opartych na algorytmach sztucznej inteligencji oraz przetestowano metody reprezentacji geometrii układów. Zbiór doświadczalny oraz jego rozszerzenie o dane teoretyczne dają bardzo zbliżoną dokładność, co pozwoliło postawić hipotezę o nieistotności geometrii układów MXenes na ich cytotoksyczność. Na podstawie analizy istotności zmiennych określono najważniejsze cechy wpływające na cytotoksyczność MXenes: obecność tlenków metali (MxOy) i litu na powierzchni. Najdokładniejszy z zbudowanych modeli, o dokładności obliczanej metryką R2 na poziomie 0,92, został użyty do przewidzenie cytotoksyczności nieprzebadanych jeszcze eksperymentalnie związków MXenes. Modele stworzone jedynie na bazie danych teoretycznych nie pozwoliły zbudować wiarygodnego modelu cytotoksyczności MXenes. Uprawdopodobnia to hipotezę o braku wpływu geometrii związków MXenes na ich cytotoksyczność.

Rozdział 6

Szczegóły techniczne

6.1. Biblioteki uczenia maszynowego

Wszystkie obliczenia związane z uczeniem maszynowym zostały zaimplementowane w języku Python [215]. Sam język pozwala na wykonywanie podstawowych operacji i obliczeń, jednak jego największą siłą są dodatkowe biblioteki. Znacznie rozszerzają one możliwości Pythona i umożliwiają szybkie oraz wydajne tworzenie aplikacji. W czasie tworzenia modeli opartych na uczeniu maszynowym korzystano z szeregu bibliotek:

Numpy [216] jest biblioteką numeryczną pozwalającą na prowadzenie wydajnych obliczeń na wektorach. Rozszerza ona język Python pozwalając pracować na wektorach i macierzach oraz zapewnia wydajną implementację najczęściej stosowanych funkcji matematycznych, obliczanie funkcji na wszystkich elementach wektora. Wydajne stosowanie biblioteki Numpy wymaga wektoryzacji obliczeń i przepisanie formuł matematycznych tak, aby były wykonywane na całym wektorze danych jednocześnie.

pandas [217] jest najpopularniejszą biblioteką Pythona do operowania na danych. Pozwala na wczytywanie szeregu formatów danych stosowanych w uczeniu maszynowym w tym plików csv (ang. Comma Separated Value) [218]. Dane są przechowywane w strukturach - DataFrame pozwalających na pracę z interfejsem zbliżonym do dostępnego w bazach SQL. Sprawia to, że wczytanie i podstawowe czyszczenie danych jest proste i wygodne. Jednak bardziej zaawansowane operacje na danych są kłopotliwe i wymagają dużego nakładu pracy, dlatego najczęściej wykonuje się je za pomocą biblioteki Numpy. Biblioteka pandas zapewnia ponadto metody czyszczenia danych, usuwania brakujących wpisów, łączenie kilku DataFrame'ów oraz obliczania podstawowych wielkości statystycznych. Jest rozszerzeniem biblioteki Numpy i obiekty z obu bibliotek są zgodne. Wydajność pandas jest porównywalna do Numpy jednak posiada duże narzuty pamięciowe podczas przechowywania danych. DataFrame wymaga prawie dwa razy więcej pamięci niż odpowiadająca mu macierz zapisana w Numpy.

scikit-learn [219] jest podstawową biblioteką do klasycznego uczenia maszynowego w Pythonie. Zapewnia większość najpopularniejszych algorytmów uczenia z nadzorem oraz bez nadzoru oraz szereg metod pomocniczych jak liczenie metryk czy testowanie krzyżowe. Biblioteka jest zgodna z pandas DataFrame oraz może być używana bezpośrednio z Numpy. Niewątpliwą zaletą biblioteki jest wygoda używania i szerokie spektrum zaimplementowanych metod.

ASE (The Atomic Simulation Environmen) [220] jest biblioteką pozwalającą na prowadzenie oraz analizę wyników symulacji molekularnych. Pozwala także na wczytywanie i tworzenie plików wsadowych oraz wynikowych do najpopularniejszych pakietów obliczeniowych w fizyce ciała stałego oraz chemii kwantowej takich jak: Abinit, Castep, CP2K, Elk, Quantum Espresso, Gaussian, Gromacs, Lammps, NWChem, CASP oraz pliki .xyz. Jest to bardzo potężne i rozbudowane narzędzie. W tej pracy było wykorzystywane jedynie do wczytywania plików wynikowych z symulacji kodami Quantum Espresso oraz VASP. Podczas pracy z tą biblioteką napotkano problemy z generowaniem plików inputowych oraz modyfikacją wczytanych danych, co znacznie ograniczyło jej stosowalność. Ponadto tworzenie obiektu zawierającego dane o układzie jest niejasne i możliwe jedynie dla prostych oraz małych układów, a wiele istotnych szczegółów zostało pominięte w dokumentacji.

DScribe [221, 222] jest biblioteką pozwalającą na budowanie zmiennych położeniowych z wczytanych danych. Niestety wspiera jedynie format ASE i stosowanie jej inaczej niż poprzez wczytanie pliku wsadowego lub wynikowego z symulacji jest bardzo trudne i uciążliwe.

Szczegółowy opis prac z poszczególnymi narzędziami znajduje się w dalszej części pracy.

6.2. Stworzone kody

Opisywane kody zostały zamieszczone w Dodatku G.

6.2.1. Podstawowe kody

Budowa modelu opartego na uczeniu maszynowym jest w ogólności procesem złożonym, składającym się z wielokrotnych powtórzeń i doboru wielu hiperparametrów. Listing 1 zawiera najprostszy przypadek zastosowania regresji liniowej przy użyciu biblioteki scikit-learn, która zapewnia szereg klas dla różnych algorytmów uczenia maszynowego. Użycie wybranego algorytmu wymaga wywołania konstruktora klasy z listą parametrów charakterystyczną dla danego konstruktora.

Każda klasa posiada dwie metody: *fit* oraz *predict*. Odpowiadają one uczeniu oraz wykonaniu przewidywania dla danego algorytmu. Metoda fit jest wywoływana z parametrami: X i Y. X jest DataFrame zawierającym zmienne niezależne, wiersze odpowiadają kolejnym obserwacjom. Y jest wektorem wartości zmiennej zależnej. Metoda *predict* jest wywoływana z parametrem X, gdzie X jest DataFrame zawierającym zmienne niezależne.

Biblioteka scikit-learn zapewnia także funkcje pomocnicze takie jak obliczanie metryk za pomocą testowania krzyżowego. Listing 2 zawiera przykładowy kod obliczania metryki R2 podczas 5-krotnego testowania krzyżowego.

6.2.2. Budowa zmiennych położeniowych

Zmienne położeniowe mogą być budowane dzięki bibliotekom DScribe oraz ASE. Przykład wczytywania plików VASP'a oraz zbudowanie zmiennych CM został przedstawiony w listingu 3.

Oprócz Macierzy Kulombowskiej, Macierzy Ewalda oraz Macierzy sinusoid zmienne położeniowe były implementowane bez użycia dedykowanych bibliotek. Listing 4 zawiera implementację Funkcji symetrii G2.

Kod liczący funkcji symetrii G4 jest znacznie bardziej skompilowany i wymaga obliczenia nie tylko odległości, ale także kątów pomiędzy atomami. Z racji objętości kodu nie został on zamieszczony w tej pracy.

6.2.3. Zrównoleglenie obliczeń

W czasie budowania modelu testowano szereg modeli oraz zestawów zmiennych niezależnych. Uczenie i testowanie każdego modelu był przeprowadzone zupełnie niezależne od pozostałych modeli. Obliczenia tego typu mogą być łatwo zrównoleglane. Do zrównoleglania użyto biblioteki MPI4py [223] będącej implementacją standardu MPI (ang. Message Passing Interface) [224] dla Pythona. MPI4py pozwala na wykonywanie rozproszonych obliczeń na wielu węzłach obliczeniowych. Wykorzystuje on mechanizmy komunikacji i przesyłania danych zgodny z MPI dla języków C/C++. MPI4py posiada obiektowe API dostępne, także w MPI dla C++, nie jest one jednak popularne. MPI4py pozwala na wykorzystanie wszystkich technik używanych w tworzeniu aplikacji równoległych w modelu pamięci rozproszonej. Listing 5 zawiera najważniejsze fragmenty kodu równoległych obliczenia.

Implementacja podana w listingu 5 zakłada, że liczba modeli jest podzielna przez liczbę procesów. Założenie to może być jednak łatwo usunięte. Operacja ta jednak zmniejsza czytelność kodu, dlatego została pominięta w tej pracy. Jednocześnie liczba budowanych modeli jest na tyle niewielka, że wykorzystanie podczas obliczeń liczby procesorów równej liczbie modeli nie jest wyzwaniem dla współczesnego klastra.

6.3. Obliczenie kwantowo-mechaniczne

Wszystkie obliczenia kwantowo - mechaniczne były wykonane w Interdyscyplinarnym Centrum Modelowania Matematycznego i Komputerowego Uniwersytetu Warszawskiego (ICM UW) w ramach grantu G80-15. W czasie prac uruchomiono ponad 8500 zadań. Oprócz właściwych obliczeń kwantowo-mechanicznych przeprowadzono także szereg testów w celu optymalizacji parametrów obliczeń. Zbadano także skalowalność Quantum Espresso w celu wyznaczenia najlepszej liczby używanych procesorów. Czasy obliczeń zostały przedstawione na rysunku A.5.

Kod dobrze skaluje się jedynie dla kilku procesorów. Wynika to z wielkości układu. Dla odpowiednio dużego układu Quantum Espresso skaluje się na tysiące rdzeni obliczeniowych [225].

Rozdział 7

Podsumowanie

W ostatnich latach uczenie maszynowe (ang. machine learning) stało się czwartym paradygmatem nauki i szybko znajduje rozliczne zastosowania w naukach przyrodniczych, między innymi w fizyce materii skondensowanej, nauce o materiałach, i nanotechnologii. Głównym zadaniem uczenia maszynowego jest przyspieszenie procesu projektowania nowych nanomateriałów o pożądanych funkcjonalnościach oraz bazujących na nich urządzeń i procesów, które przyczyniłyby się do rozwiązania najbardziej pilnych wyzwań ludzkości. Mimo dynamicznego rozwoju zastosowań uczenia maszynowego na polu nauki o materiałach, nie zostały wypracowane dotychczas standardowe procedury, i w zasadzie każdy problem wymaga zastosowania specyficznej metodologii.

Niniejszej praca stanowi studium zastosowań metodologii uczenia maszynowego do problemu przewidywania własności niskowymiarowych nanostruktur na podstawie znanych cech składników (np. atomów) i ich wzajemnej konfiguracji. Podstawowym problemem w tym kontekście jest wiarygodne przewidzenie stabilności nowych nanostruktur. W pracy skoncentrowano się na badaniu nanorurek borowych, nanorurek azotku-boru, oraz dwu-wymiarowych węglików tytanu - materiałów MXenes. Zbadano stabilność tych nanostruktur badając ich energię całkowitą oraz energię kohezji. W przypadku materiałów MXenes przeprowadzono również badania cytotoksyczności. Przeprowadzone w dysertacji badania bazowały na danych eksperymentalnych zaczerpniętych z różnych repozytoriów jak również rozległych obliczeń w ramach teorii funkcjonału gęstości (DFT) przeprowadzonych podczas badań.

Ponieważ teoria DFT jest dobrze zadomowiona w środowisku teoretyków materii skondensowanej i istnieje cały szereg znakomitych monografii poświęconych DFT, w pracy doktorskiej zaprezentowano tylko bardzo zwięzły opis teorii oraz praktycznych implementacji DFT. Z drugiej strony, metodologia uczenia maszynowego nie jest powszechnie znana w środowisku teoretyków materii skondensowanej i zdecydowano się na bardziej szczegółowe przedstawienie całego pakietu zagadnień związanych z uczeniem maszynowym (w rozdziale 3) oraz specyficznych zastosowań algorytmów uczenia maszynowego do problemów materii skondensowanej (w rozdziale 4). Zasadniczą część rozprawy stanowi rozdział 5, gdzie zaprezentowano wyniki badań w oparciu o metodologię uczenia maszynowego do wyżej wspomnianych własności badanych nanostruktur.

Opisując wyniki badań, pokazano techniki rozwiązywania problemów spotykanych podczas stosowania algorytmów uczenia maszynowego na rzeczywistych danych fizycznych. Ponadto opisano szereg zmiennych położeniowych

pozwalających zapisywać geometrie układów w sposób zgodny z wymaganiami algorytmów sztucznej inteligencji. Zaprezentowano też wyniki rozlicznych testów algorytmów uczenia maszynowego dostosowanych do specyfiki problemów fizyki materii skondensowanej.

W szczególności zbudowano model opisujący energie całkowita oraz energie kohezji nanorurek borowych oraz azotku-boru. Podczas tworzenie zbioru danych niezbędnego do uczenia modeli przeprowadzono ponad osiem tysięcy symulacji kwantowo - mechanicznych. Na podstawie zgromadzonych danych przetestowano dwanaście algorytmów uczenia maszynowego oraz dziesięć klas zmiennych położeniowych. Ponadto dla każdej pary: model - zmienna położeniowa wykonano optymalizacje hiperparametrów stosowanego algorytmu oraz zbadano jakość powstałego modelu. Najlepsza kombinacja algorytmów oraz zmiennych posiada dokładność mierzona metryka R2 obliczana podczas testowania krzyżowego na poziomie 0.95. Otrzymane wyniki pokazują, że możliwe jest stworzenie modelu opartego na algorytmach uczenia maszynowego posiadającego zarówno dużą dokładność jak i zdolność uogólniania podczas predykcji energii w oparciu o geometrie układu. Ponadto analiza wyników pokazała, że dobrze dobrane zmienne pozwalają na stosowanie bardzo prostych modeli jak regresja wieloliniowa, co pozwala na prowadzenie dalszych analiz oraz głębsze zrozumienie także bardziej złożonych własności badanych nano-układów.

Jednak próba przewidywania energii kohezji MXenes, na podstawie geometrii układów, nie przyniosła oczekiwanych wyników. Udało się zbudować modele oparte na algorytmach uczenia maszynowego, które odzwierciedlały energie układu, ale nie posiadały one zdolności uogólniania. Największymi trudnościami podczas pracy były: duża różnorodność obserwacji w zbiorze treningowym (dużą liczba występujących w nim pierwiastków oraz rozpiętość liczby atomów w układzie) przy jednocześnie małej liczbie obserwacji. Przetestowano możliwości generacji syntetycznych danych, ale zastosowane algorytmy nie poprawiły istotnie otrzymywanej dokładności.

Modele przewidujące cytotoksyczność MXenes osiągnęły dokładność (obliczona dziesięciokrotnym testowaniem krzyżowym) na poziomie 0,9. Wartość ta jest bardzo wysoka dla tak małego zbioru danych. Ponadto analiza istotności zmiennych pokazała, że cytotoksyczność nie zależy od geometrii MXenes, ale raczej od obecności litu oraz tlenków metali na powierzchni Mxenes. Stosując algorytmy uczenia bez nadzoru wykonano analizę cech charakteryzujących geometrie układu pod kątem wpływu na cytotoksyczność. Nie znaleziono jednak cech geometrii rozważnych związków pozwalających na wyodrębnienie klastrów związków cytotoksycznych w badanym zbiorze danych. Wykonano także przewidywanie cytotoksyczności związków dotychczas niezbadanych eksperymentalnie. Wyniki analiz cytotoksyczności prezentowane w tej pracy zostały wcześniej opublikowane w [12].

Modele oparte na zmiennych położeniowych pokazały, że uwzględnianie periodyczności badanych układów (nanorurki borowe i azotku boru oraz MXenes) nie wpływa na jakość modelu i do pełnego opisu układu wystarczają informacje o położeniach atomów w komórce elementarnej.

Otrzymane wyniki pokazują potencjał algorytmów opartych na uczeniu maszynowym w zastosowaniach do problemów fizyki materii skondensowanej.

Zgodnie z intuicją, dokładność i stosowalność tych metod jest silnie skorelowana z wielkością i jakością posiadanych danych. Dla dużych zbiorów danych dobrej jakości można zbudować bardzo dokładne modele, podczas gdy dane słabej jakości prowadza do modeli o niskiej dokładności. Fakt ten podkreśla konieczność dokładnego testowania tworzonych modeli. Niemniej, jak pokazano w tej pracy, nawet dla niewielkich zbiorów danych algorytmy uczenia maszynowego mogą być z powodzeniem stosowane w problemach fizyki materii skondensowanej i stanowić potężne narzędzie analizy danych.

Metodologia uczenia maszynowego opracowana na potrzeby badań specyficznych nanostruktur rozważanych w dysertacji oraz dokładne testy ogólnych algorytmów uczenia maszynowego, szczegółowo opisane w niniejszej rozprawie, mogą stanowić drogowskaz do dalszych badań własności różnych nowatorskich nanomateriałów.

Bibliografia

1. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning Representations by Back-propagating Errors. *Nature* **323,** 533–536 (1986).
2. Hey, T., Tansley, S. & Tolle, K. *The Fourth Paradigm: Data-Intensive Scientific Discovery* ISBN: 978-0-9825442-0-4 (Microsoft Research, Oct. 2009).
3. Hansen, K. *et al.* Assessment and Validation of Machine Learning Methods for Predicting Molecular Atomization Energies. *Journal of Chemical Theory and Computation* **9,** 3404–3419 (2013).
4. Rupp, M., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O. A. Fast and accurate modeling of molecular atomization energies with machine learning. *Physical Review Letters* **108,** 058301 (2012).
5. Hansen, K. *et al.* Machine Learning Predictions of Molecular Properties: Accurate Many-Body Potentials and Nonlocality in Chemical Space. *The Journal of Physical Chemistry Letters* **6.** PMID: 26113956, 2326–2331 (2015).
6. Chmiela, S. *et al.* Machine learning of accurate energy-conserving molecular force fields. *Science Advances* **3** (2017).
7. Chmiela, S., Sauceda, H. E., Müller, K.-R. & Tkatchenko, A. Towards Exact Molecular Dynamics Simulations with Machine-Learned Force Fields. *Nature Communications* **9** (Feb. 2018).
8. Huang, Y. *et al. Machine learning band gap and alignment of nitride semiconductors* Oct. 2018.
9. Dong, Y. *et al.* Bandgap prediction by deep learning in configurationally hybridized graphene and boron nitride. *npj Computational Materials* **5,** 26 (Feb. 2019).
10. Rajan, A. C. *et al.* Machine-Learning-Assisted Accurate Band Gap Predictions of Functionalized MXene. *Chemistry of Materials* **30,** 4031–4038 (2018).
11. Zhuo, Y., Mansouri Tehrani, A. & Brgoch, J. Predicting the Band Gaps of Inorganic Solids by Machine Learning. *The Journal of Physical Chemistry Letters* **9.** PMID: 29532658, 1668–1673 (2018).
12. Marchwiany, M., Birowska, M., Popielski, M., Majewski, J. & Jastrzębska, A. Surface-Related Features Responsible for Cytotoxic Behavior of MXenes Layered Materials Predicted with Machine Learning Approach. *Materials* **13** (2020).
13. Stefanucci, G. & van Leeuwen, R. *Nonequilibrium Many-Body Theory of Quantum Systems: A Modern Introduction* (Cambridge University Press, 2013).

14. Martin, R. M., Reining, L. & Ceperley, D. M. *Interacting Electrons: Theory and Computational Approaches* (Cambridge University Press, 2016).
15. Hohenberg, P. & Kohn, W. Inhomogeneous Electron Gas. *Phys. Rev.* **136,** B864–B871 (3B 1964).
16. Parr, R. G. *Density Functional Theory of Atoms and Molecules* in *Horizons of Quantum Chemistry* (eds Fukui, K. & Pullman, B.) (Springer Netherlands, Dordrecht, 1980), 5–15. ISBN: 978-94-009-9027-2.
17. Reinhold, J. D. R. Salahub, M. C. Zerner (eds). The challenge of d and f electrons. Theory and computation. ACS symposium series 394. American Chemical Society, Washington, DC 1989, ISBN 0-8412-1628-2, (ACS Symposium Series, ISSN 0065-6156; 394), 405 Seiten, $107.95. *Crystal Research and Technology - CRYST RES TECH* **25,** 624–624 (June 1990).
18. Levy, M. & Perdew, J. P. *The Constrained Search Formulation of Density Functional Theory* 11–30 (Springer US, 1985).
19. Levy, M. in *Density Functional Theory of Many-Fermion Systems* (ed Löwdin, P.-O.) 69–95 (Academic Press, 1990).
20. Kohn, W. & Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **140,** A1133–A1138 (4A 1965).
21. Kohn, W. & Sham, L. J. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* **140,** A1133–A1138 (4A 1965).
22. Mardirossian, N. & Head-Gordon, M. Thirty years of density functional theory in computational chemistry: An overview and extensive assessment of 200 density functionals. *Molecular Physics* **115,** 1–58 (June 2017).
23. Ceperley, D. & Alder, B. Ground-State of the Electron-Gas by A Stochastic Method. *Phys. Rev. Lett.* **45,** 566– (Aug. 1980).
24. Perdew, J. P. & Zunger, A. Self-interaction correction to density-functional approximations for many-electron systems. *Phys. Rev. B* **23,** 5048–5079 (10 May 1981).
25. Perdew, J. P. & Yue, W. Accurate and simple density functional for the electronic exchange energy: Generalized gradient approximation. *Phys. Rev. B* **33,** 8800–8802 (12 June 1986).
26. Szabo, A. & Ostlund, N. S. *Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory* First (Dover Publications, Inc., 1996).
27. Saavedra, F. & Kalos, M. Bilinear diffusion quantum Monte Carlo methods. *Physical review. E, Statistical, nonlinear, and soft matter physics* **67,** 026708 (Mar. 2003).
28. *Quantum Monte Carlo methods in physics and chemistry / edited by M.P. Nightingale and Cyrus J. Umrigar.* (Kluwer Academic, 1999).
29. Perdew, J. *et al.* Atoms, Molecules, Solids, and Surfaces: Applications of the Generalized Gradient Approximation for Exchange and Correlation. *Physical review. B, Condensed matter* **46,** 6671–6687 (Oct. 1992).

30. Perdew, J. P., Burke, K. & Wang, Y. Generalized gradient approximation for the exchange-correlation hole of a many-electron system. *Phys. Rev. B* **54,** 16533–16539 (23 Dec. 1996).
31. Amovilli, C. & Floris, F. M. Shannon Entropy in Atoms: A Test for the Assessment of Density Functionals in Kohn-Sham Theory. *Computation* **6.** ISSN: 2079-3197 (2018).
32. Hao, P. *et al.* Performance of meta-GGA Functionals on General Main Group Thermochemistry, Kinetics, and Noncovalent Interactions. *Journal of Chemical Theory and Computation* **9,** 355–363 (2013).
33. Städele, M., Majewski, J. A., Vogl, P. & Görling, A. Exact Kohn-Sham Exchange Potential in Semiconductors. *Phys. Rev. Lett.* **79,** 2089–2092 (11 1997).
34. Seidl, A., Görling, A., Vogl, P., Majewski, J. & Levy, M. Generalized Kohn-Sham schemes and the band-gap problem. *Physical Review. B, Condensed matter* **53,** 3764–3774 (Mar. 1996).
35. Lin, P., Ren, X. & He, L. Efficient Hybrid Density Functional Calculations for Large Periodic Systems Using Numerical Atomic Orbitals. *Journal of Chemical Theory and Computation* **17,** 222–239 (Dec. 2020).
36. Perdew, J. P., Burke, K. & Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* **77,** 3865–3868 (18 Oct. 1996).
37. Raghavachari, K. *Perspective on "Density functional thermochemistry. III. The role of exact exchange"* (eds Cramer, C. J. & Truhlar, D. G.) 361–363. ISBN: 978-3-662-10421-7 (Springer Berlin Heidelberg, Berlin, Heidelberg, 2001).
38. Lee, C., Yang, W. & Parr, R. G. Development of the Colle-Salvetti correlation-energy formula into a functional of the electron density. *Phys. Rev. B* **37,** 785–789 (2 Jan. 1988).
39. Heyd, J. & Scuseria, G. Hybrid Functional Based on a Screened Coulomb Potential. *J. Chem. Phys.* **118** (May 2003).
40. Perdew, J. P. & Schmidt, K. Jacob's ladder of density functional approximations for the exchange-correlation energy. *AIP Conference Proceedings* **577,** 1–20 (2001).
41. Probert, M. Electronic Structure: Basic Theory and Practical Methods, by Richard M. Martin. *Contemporary Physics - CONTEMP PHYS* **52,** 77–77 (Jan. 2011).
42. Singh, D. & Nordström, L. *Planewaves, Pseudopotentials and the LAPW Method, Second Edition* ISBN: 978-0-387-28780-5 (Dec. 2005).
43. Schwerdtfeger, P. The Pseudopotential Approximation in Electronic Structure Theory. *ChemPhysChem* **12,** 3143–3155 (2011).
44. Råsander, M. *A Theoretical Perspective on the Chemical Bonding and Structure of Transition Metal Carbides and Multilayers* PhD thesis (Jan. 2010).
45. Troullier, N. & Martins, J. L. Efficient pseudopotentials for plane-wave calculations. *Phys. Rev. B* **43,** 1993–2006 (3 Jan. 1991).
46. Vanderbilt, D. Soft self-consistent pseudopotentials in a generalized eigenvalue formalism. *Phys. Rev. B* **41,** 7892–7895 (11 Apr. 1990).

47. Blöchl, P. E. Projector augmented-wave method. *Phys. Rev. B* **50,** 17953–17979 (24 Dec. 1994).
48. Corso, A. Pseudopotentials periodic table: From H to Pu. *Computational Materials Science* **95,** 337–350 (Dec. 2014).
49. Kresse, G. & Hafner, J. *Ab initio* molecular dynamics for liquid metals. *Phys. Rev. B* **47,** 558–561 (Jan. 1993).
50. Kresse, G. & Hafner, J. Ab initio molecular-dynamics simulation of the liquid-metal–amorphous-semiconductor transition in germanium. *Phys. Rev. B* **49,** 14251–14269 (20 May 1994).
51. Kresse, G. & Furthmüller, J. Efficiency of ab-initio total energy calculations for metals and semiconductors using a plane-wave basis set. *Comput. Mater. Sci.* **6,** 15–50. ISSN: 0927-0256 (1996).
52. Kresse, G. & Furthmüller, J. Efficient iterative schemes for ab initio total-energy calculations using a plane-wave basis set. *Phys. Rev. B* **54,** 11169–11186 (Oct. 1996).
53. Giannozzi, P. *et al.* Advanced capabilities for materials modelling with QUANTUM ESPRESSO. *Journal of Physics: Condensed Matter* **29,** 465901 (2017).
54. Giannozzi, P. *et al.* QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials. *Journal of Physics: Condensed Matter* **21,** 395502 (19pp) (2009).
55. www.quantum-espresso.org.
56. Wills, J. *et al. Full-Potential Electronic Structure Method: Energy and Force Calculations with Density Functional and Dynamical Mean Field Theory* ISBN: 978-3-642-15143-9 (Springer Science Business Media, Jan. 2010).
57. Feynman, R. P. Forces in Molecules. *Phys. Rev.* **56,** 340–343 (4 Aug. 1939).
58. Bontempi, G. *Handbook on "Statistical foundations of machine learning" (2nd edition)* (Feb. 2021).
59. Abebe, A., Daniels, J., McKean, J. & Kapenga, J. Statistics and data analysis. *Michigan: Western Michigan University* (2001).
60. Dangeti, P. *Statistics for Machine Learning: Techniques for Exploring Supervised, Unsupervised, and Reinforcement Learning Models with Python and R* ISBN: 1788295757 (Packt Publishing, 2017).
61. MacKay, D. J. C. *Information Theory, Inference, and Learning Algorithms* (2003).
62. Hastie, T., Tibshirani, R. & Friedman, J. *The Elements of Statistical Learning* (Springer New York Inc., New York, NY, USA, 2001).
63. He, H. & Ma, Y. *Imbalanced Learning: Foundations, Algorithms, and Applications* 1st. ISBN: 1118074629 (Wiley-IEEE Press, 2013).
64. Guo, H. & Viktor, H. L. Learning from Imbalanced Data Sets with Boosting and Data Generation: The DataBoost-IM Approach. *SIGKDD Explor. Newsl.* **6,** 30–39. ISSN: 1931-0145 (2004).
65. Fernández, A. *et al. Learning from Imbalanced Data Sets* ISBN: 978-3-319 -98073-7 (Jan. 2018).
66. Deb, K. An introduction to genetic algorithms. *Sadhana* **24,** 293–315. ISSN: 0973-7677 (Aug. 1999).

67. Kennedy, J. *Particle Swarm Optimization* (eds Sammut, C. & Webb, G. I.) 760–766 (Springer US, Boston, MA, 2010).
68. Hosmer, D. W. & Lemeshow, S. *Applied logistic regression* ISBN: 0471356328, 9780471356325 (John Wiley and Sons, 2000).
69. Tolles, J. & Meurer, W. Logistic Regression: Relating Patient Characteristics to Outcomes. *JAMA* **316,** 533 (Aug. 2016).
70. James, G., Witten, D., Hastie, T. & Tibshirani, R. *An Introduction to Statistical Learning: With Applications in R* ISBN: 1461471370 (Springer Publishing Company, Incorporated, 2014).
71. Hastie, T., Tibshirani, R. & Friedman, J. *The elements of statistical learning: data mining, inference and prediction* (Springer, 2009).
72. Bellman, R. *Adaptive Control Processes* (Princeton University Press, 1961).
73. Quinlan, J. R. Induction of decision trees. *Machine Learning* **1,** 81–106. ISSN: 1573-0565 (Mar. 1986).
74. Dodge, Y. *Gini Index* 231–233 (Springer New York, New York, NY, 2008).
75. Quinlan, J. *C4.5: Programs for Machine Learning* (Morgan Kaufmann, 1993).
76. Breiman, L., Friedman, J. H., Olshen, R. A. & Stone, C. J. *Classification and Regression Trees* (Wadsworth and Brooks, Monterey, CA, 1984).
77. Ho, T. K. *Random Decision Forests* in *Proceedings of the Third International Conference on Document Analysis and Recognition (Volume 1) - Volume 1* (IEEE Computer Society, USA, 1995), 278.
78. Breiman, L. Bagging Predictors. *Machine Learning* **24,** 123–140. ISSN: 1573-0565 (Aug. 1996).
79. Geurts, P., Ernst, D. & Wehenkel, L. Extremely randomized trees. *Machine Learning* **63,** 3–42. ISSN: 1573-0565 (Apr. 2006).
80. Breiman, L. *Arcing the edge* tech. rep. (1997).
81. Freund, Y. & Schapire, R. E. *A Short Introduction to Boosting* in *In Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence* (Morgan Kaufmann, 1999), 1401–1406.
82. Ester, M., Kriegel, H.-P., Sander, J. & Xu, X. *A Density-Based Algorithm for Discovering Clusters a Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise* in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining* (AAAI Press, Portland, Oregon, 1996), 226–231.
83. Raghavan, N., Albert, R. & Kumara, S. Near Linear Time Algorithm to Detect Community Structures in Large-Scale Networks. *Physical review. E, Statistical, nonlinear, and soft matter physics* **76,** 036106 (Oct. 2007).
84. Noumir, Z., Honeine, P. & Richard, C. *On simple one-class classification methods* in (July 2012), 2022–2026. ISBN: 978-1-4673-2580-6.
85. Liu, F. T., Ting, K. M. & Zhou, Z. Isolation Forest, 413–422. ISSN: 2374-8486 (Dec. 2008).
86. Cook, R. D. Cook's Distance (ed Lovric, M.) 301–302 (2011).

87. Jolliffe, I. T. *Principal Component Analysis and Factor Analysis* 115–128. ISBN: 978-1-4757-1904-8 (Springer New York, New York, NY, 1986).
88. Van der Maaten, L. & Hinton, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research* **9** (2008).
89. Hinton, G. E. & Zemel, R. S. in *Advances in Neural Information Processing Systems 6* (eds Cowan, J. D., Tesauro, G. & Alspector, J.) 3–10 (Morgan-Kaufmann, 1994).
90. Jain, A. *et al.* The Materials Project: A materials genome approach to accelerating materials innovation. *APL Materials* **1,** 011002. ISSN: 2166532X (2013).
91. *https://materialsproject.org*
92. *https://omdb.mathub.io/*
93. *https://cmr.fysik.dtu.dk/c2db/c2db.html*
94. *http://anant.mrc.iisc.ac.in/*
95. *http://organiccrystalbandgaps.org/index.php*
96. *https://www.materialscloud.org*
97. Rumelhart, D. E., Hinton, G. E. & Williams, R. J. Learning Representations by Back-propagating Errors. *Nature* **323,** 533–536 (1986).
98. Hamilton, W., Ying, Z. & Leskovec, J. *Inductive Representation Learning on Large Graphs* in *Advances in Neural Information Processing Systems* (eds Guyon, I. *et al.*) **30** (Curran Associates, Inc., 2017).
99. Rupp, M., Tkatchenko, A., Müller, K.-R. & von Lilienfeld, O. A. Fast and Accurate Modeling of Molecular Atomization Energies with Machine Learning. *Phys. Rev. Lett.* **108,** 058301 (Jan. 2012).
100. Faber, F., Lindmaa, A., von Lilienfeld, A. & Armiento, R. Crystal Structure Representations for Machine Learning Models of Formation Energies. *International Journal of Quantum Chemistry* **115** (Mar. 2015).
101. Behler, J. Atom-centered symmetry functions for constructing high - dimensional neural network potentials. *The Journal of chemical physics* **134,** 074106 (Feb. 2011).
102. Gastegger, M., Schwiedrzik, L., Bittermann, M., Berzsenyi, F. & Marquetand, P. WACSF - Weighted Atom-Centered Symmetry Functions as Descriptors in Machine Learning Potentials. *The Journal of Chemical Physics* **148** (Dec. 2017).
103. Batra, R. *et al.* General Atomic Neighborhood Fingerprint for Machine Learning-Based Methods. *The Journal of Physical Chemistry C* **123,** 15859–15866 (2019).
104. Huo, H. & Rupp, M. *Unified Representation of Molecules and Crystals for Machine Learning* 2017. arXiv: `1704.06439 [physics.chem-ph]`.
105. De, S., Bartok, A., Csányi, G. & Ceriotti, M. Comparing molecules and solids across structural and alchemical space. *Phys. Chem. Chem. Phys.* **18** (Dec. 2015).
106. Ewald, P. P. Die Berechnung optischer und elektrostatischer Gitterpotentiale. *Annalen der Physik* **369,** 253–287 (1921).

107. Jäger, M., Morooka, E., Canova, F., Himanen, L. & Foster, A. Machine learning hydrogen adsorption on nanoclusters through structural descriptors. *npj Computational Materials* **4** (Dec. 2018).
108. Bartok, A., Kondor, R. & Csányi, G. On representing chemical environments. *Physical Review B* **87** (Sept. 2012).
109. Zhang, Y. & Ling, C. A strategy to apply machine learning to small datasets in materials science. *npj Computational Mathematics* **4,** 25 (2018).
110. Freund, Y. & Schapire, R. E. A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting. *J. Comput. Syst. Sci.* **55,** 119–139. ISSN: 0022-0000 (Aug. 1997).
111. Breiman, L. Bagging Predictors. *Machine Learning* **24,** 123–140 (1996).
112. Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. SMOTE: Synthetic Minority Over-sampling Technique. *J. Artif. Intell. Res. (JAIR)* **16,** 321–357 (Jan. 2002).
113. Brodersen, K. H., Ong, C. S., Stephan, K. & Buhmann, J. The Balanced Accuracy and Its Posterior Distribution. *Pattern Recognition, International Conference on* **0,** 3121–3124 (Aug. 2010).
114. Vovk, V. *Kernel Ridge Regression* (eds Schölkopf, B., Luo, Z. & Vovk, V.) 105–116 ("Springer Berlin Heidelberg", Berlin, Heidelberg, 2013).
115. Saxena, S. *Handbook of Boron Nanostructures* ISBN: 978-9-814-61394-1 (crc press, 2016).
116. Tian, Y. *et al.* Inorganic Boron-Based Nanostructures: Synthesis, Optoelectronic Properties, and Prospective Applications. *10* **9,** 538 (2019).
117. Patel, R., Chou, T. & Iqbal, Z. Synthesis of Boron Nanowires, Nanotubes, and Nanosheets. *Journal of Nanomaterials* **2015,** 1–7 (Jan. 2015).
118. Boustani, I. & Quandt, A. Nanotubules of bare boron clusters:Ab initioand density functional study. *Europhysics Letters (EPL)* **39,** 527–532 (1997).
119. GindulytÄ—, A., Lipscomb, W. N. & Massa, L. Proposed Boron Nanotubes. *Inorganic Chemistry* **37,** 6544–6545 (1998).
120. Emin, D. Icosahedral boron-rich solids. *Physics Today* **40,** 55–62 (1987).
121. Cui, Y. & Lieber, C. M. Functional nanoscale electronic devices assembled using silicon nanowire building blocks. *science* **291,** 851–853 (2001).
122. Osada, M. & Sasaki, T. Two-dimensional dielectric nanosheets: novel nanoelectronics from nanocrystal building blocks. *Advanced Materials* **24,** 210–228 (2012).
123. Oganov, A. & Solozhenko, V. Boron: a hunt for superhard polymorphs. *Journal of Superhard Materials* **31,** 285 (2009).
124. Albert, B. & Hillebrecht, H. Boron: elementary challenge for experimenters and theoreticians. *Angewandte Chemie International Edition* **48,** 8640–8668 (2009).
125. Okada, J. T. *et al.* Visualizing the mixed bonding properties of liquid boron with high-resolution X-Ray compton scattering. *Physical review letters* **114,** 177401 (2015).

126. Kondo, T. Recent progress in boron nanomaterials. *Science and Technology of Advanced Materials* **18,** 780–804 (2017).
127. Ordanyan, S., Vikhman, S., Nesmelov, D., Danilovich, D. & Panteleev, I. *Nonoxide high-melting point compounds as materials for extreme conditions* in *Advances in Science and Technology* **89** (2014), 47–56.
128. Zhang, H. *et al.* Nanostructured LaB6 field emitter with lowest apical work function. *Nano letters* **10,** 3539–3544 (2010).
129. Zhang, H., Zhang, Q., Tang, J. & Qin, L.-C. Single-crystalline LaB6 nanowires. *Journal of the American Chemical Society* **127,** 2862–2863 (2005).
130. Chen, C.-H., Aizawa, T., Iyi, N., Sato, A. & Otani, S. Structural refinement and thermal expansion of hexaborides. *Journal of Alloys and Compounds* **366,** L6–L8 (2004).
131. Kimura, S., Nanba, T., Tomikawa, M., Kunii, S. & Kasuya, T. Electronic structure of rare-earth hexaborides. *Physical Review B* **46,** 12196 (1992).
132. Peysson, Y. *et al.* Thermal properties of CeB6 and LaB6. *Journal of Magnetism and Magnetic Materials* **47,** 63–65 (1985).
133. Xia, Y. *et al.* One-dimensional nanostructures: synthesis, characterization, and applications. *Advanced materials* **15,** 353–389 (2003).
134. Xia, Y. *et al.* One-dimensional nanostructures: synthesis, characterization, and applications. *Advanced materials* **15,** 353–389 (2003).
135. Boustani, I. & Quandt, A. Nanotubules of bare boron clusters:Ab initioand density functional study. *Europhysics Letters (EPL)* **39,** 527–532 (Sept. 1997).
136. Ciuparu, D., Klie, R. F., Zhu, Y. & Pfefferle, L. Synthesis of Pure Boron Single-Wall Nanotubes. *The Journal of Physical Chemistry B* **108,** 3967–3969 (2004).
137. Wang, J., Prof, Y. & Prof, Y.-C. A New Class of Boron Nanotube. *ChemPhysChem* **10,** 3119–3121 (Dec. 2009).
138. Yang, X., Ding, Y. & Ni, J. Ab initio prediction of stable boron sheets and boron nanotubes: Structure, stability, and electronic properties. *Physical Review B* **77,** 41402– (Jan. 2008).
139. Wu, X. *et al.* Two-Dimensional Boron Monolayer Sheets. *ACS Nano* **6,** 7443–7453 (2012).
140. Yin, J. *et al.* Boron Nitride Nanostructures: Fabrication, Functionalization and Applications. *Small (Weinheim an der Bergstrasse, Germany)* **12** (Apr. 2016).
141. Beiss, P., Ruthardt, R. & Warlimont, H. Powder Metallurgy Data. Refractory, Hard and Intermetallic Materials · 1 Introduction: Datasheet from Landolt-Börnstein - Group VIII Advanced Materials and Technologies · Volume 2A2: "Powder Metallurgy Data. Refractory, Hard and Intermetallic Materials".
142. Rubio, A., Corkill, J. L. & Cohen, M. L. Theory of graphitic boron nitride nanotubes. *Phys. Rev. B* **49,** 5081–5084 (7 Feb. 1994).
143. Chopra, N. G. *et al.* Boron Nitride Nanotubes. *Science* **269,** 966–967 (1995).

144. Kim, J. H., Pham, T. V., Hwang, J. H., Kim, C. S. & Kim, M. J. Boron nitride nanotubes: synthesis and applications. *Nano Convergence* **5,** 17. ISSN: 2196-5404 (June 2018).
145. Carroccio, A. *et al.* Searching for wheat plants with low toxicity in celiac disease: Between direct toxicity and immunologic activation. *Digestive and liver disease : official journal of the Italian Society of Gastroenterology and the Italian Association for the Study of the Liver* **43,** 34–9 (Jan. 2011).
146. Shakerzadeh, E. in *Boron Nitride Nanotubes in Nanomedicine* (eds Ciofani, G. & Mattoli, V.) 59–77 (William Andrew Publishing, Boston, 2016). ISBN: 978-0-323-38945-7.
147. Severino, P. *et al.* in *Nanobiomaterials in Cancer Therapy* (ed Grumezescu, A. M.) 91–115 (William Andrew Publishing, 2016). ISBN: 978-0-323-42863-7.
148. Hopwood, C. *et al.* Genetic and Environmental Influences on Personality Trait Stability and Growth During the Transition to Adulthood: A Three-Wave Longitudinal Study. *Journal of personality and social psychology* **100,** 545–56 (Mar. 2011).
149. Li, X. & Golberg, D. in *Boron Nitride Nanotubes in Nanomedicine* (eds Ciofani, G. & Mattoli, V.) 79–94 (William Andrew Publishing, Boston, 2016). ISBN: 978-0-323-38945-7.
150. Verger, L. *et al.* Overview of the synthesis of MXenes and other ultrathin 2D transition metal carbides and nitrides. *Current Opinion in Solid State and Materials Science* (Mar. 2019).
151. Naguib, M. *et al.* Two-Dimensional Transition Metal Carbides. *ACS Nano* **6.** PMID: 22279971, 1322–1331 (2012).
152. Naguib, M. *et al.* New Two-Dimensional Niobium and Vanadium Carbides as Promising Materials for Li-Ion Batteries. *Journal of the American Chemical Society* **135,** 15966–15969 (2013).
153. Khazaei, M. *et al.* Novel Electronic and Magnetic Properties of Two - Dimensional Transition Metal Carbides and Nitrides. *Advanced Functional Materials* **23,** 2185–2192 (2013).
154. Anasori, B., Lukatskaya, M. R. & Gogotsi, Y. 2D metal carbides and nitrides (MXenes) for energy storage. *Nature Reviews Materials* **2** (2017).
155. Hong Ng, V. M. *et al.* Recent progress in layered transition metal carbides and/or nitrides (MXenes) and their composites: synthesis and applications. *J. Mater. Chem. A* **5,** 3039–3068 (7 2017).
156. Jian-Feng, Z., Hui-Yang, C. & Hong-Bing, W. Research Progress of Novel Two-dimensional Material MXene. *Journal of Inorganic Materials* **32,** 561 (June 2017).
157. Rasool, K. *et al.* Antibacterial Activity of Ti3C2Tx MXene. *ACS Nano* **10,** 3674–3684 (2016).
158. Arabi Shamsabadi, A., Sharifian Gh., M., Anasori, B. & Soroush, M. Antimicrobial Mode-of-Action of Colloidal Ti3C2Tx MXene Nanosheets. *ACS Sustainable Chemistry & Engineering* **6,** 16586–16596 (2018).

159. A. Jastrzębska, A. *et al.* Biological Activity and Bio-Sorption Properties of the Ti2C Studied by Means of Zeta Potential and SEM. *Int. J. Electrochem. Sci.* **12,** 2159–2172 (2017).
160. Rozmysłowska-Wojciechowska, A. *et al.* Influence of modification of Ti3C2 MXene with ceramic oxide and noble metal nanoparticles on its antimicrobial properties and ecotoxicity towards selected algae and higher plants. *RSC Adv.* **9,** 4092–4105 (8 2019).
161. Jastrzębska, A. M. *et al.* The Atomic Structure of Ti2C and Ti3C2 MXenes is Responsible for Their Antibacterial Activity Toward E. coli Bacteria. *Journal of Materials Engineering and Performance* **28,** 1272–1277 (2018).
162. Soundiraraju, B. & George, B. K. Two-Dimensional Titanium Nitride (Ti2N) MXene: Synthesis, Characterization, and Potential Application as Surface-Enhanced Raman Scattering Substrate. *ACS Nano* **11,** 8892–8900 (2017).
163. Jastrzębska, A. *et al.* In vitro studies on cytotoxicity of delaminated Ti3C2 MXene. *Journal of Hazardous Materials* **339,** 1–8. ISSN: 0304-3894 (2017).
164. Xue, Q. *et al.* Photoluminescent Ti3C2 MXene Quantum Dots for Multicolor Cellular Imaging. *Advanced Materials* **29,** 1604847 (2017).
165. Yu, X. *et al.* Fluorine-free preparation of titanium carbide MXene quantum dots with high near-infrared photothermal performances for cancer therapy. *Nanoscale* **9,** 17859–17864 (45 2017).
166. Zhou, L. *et al.* Titanium carbide (Ti3C2Tx) MXene: A novel precursor to amphiphilic carbide-derived graphene quantum dots for fluorescent ink, light-emitting composite and bioimaging. *Carbon* **118,** 50–57 (2017).
167. Lin, H., Wang, X., Yu, L., Chen, Y. & Shi, J. Two-Dimensional Ultrathin MXene Ceramic Nanosheets for Photothermal Conversion. *Nano Letters* **17.** PMID: 28026960, 384–391 (2017).
168. Dai, C. *et al.* Biocompatible 2D Titanium Carbide (MXenes) Composite Nanosheets for pH-Responsive MRI-Guided Tumor Hyperthermia. *Chemistry of Materials* **29,** 8637–8652 (2017).
169. Liu, G. *et al.* Surface Modified Ti3C2 MXene Nanosheets for Tumor Targeting Photothermal/Photodynamic/Chemo Synergistic Therapy. *ACS Applied Materials & Interfaces* **9,** 40077–40086 (2017).
170. Chen, X. *et al.* Ratiometric photoluminescence sensing based on Ti3C2 MXene quantum dots as an intracellular pH sensor. *Nanoscale* **10,** 1111–1118 (3 2018).
171. Han, X. *et al.* 2D Ultrathin MXene-Based Drug-Delivery Nanoplatform for Synergistic Photothermal Ablation and Chemotherapy of Cancer. *Advanced Healthcare Materials* **7,** 1701394 (2018).
172. Hussein, E. A. *et al.* Plasmonic MXene-based nanocomposites exhibiting photothermal therapeutic effects with lower acute toxicity than pure MXene. *International Journal of Nanomedicine* **14,** 4529–4539 (2019).

173. Tang, W. *et al.* Multifunctional Two-Dimensional Core–Shell MXene@Gold Nanocomposites for Enhanced Photo–Radio Combined Therapy in the Second Biological Window. *ACS Nano* **13,** 284–294 (2019).
174. Ren, C. E. *et al.* Charge- and Size-Selective Ion Sieving Through Ti3C2Tx MXene Membranes. *The Journal of Physical Chemistry Letters* **6,** 4026–4031 (2015).
175. Rakhi, R. B., Nayak, P., Xia, C. & Alshareef, H. N. Novel amperometric glucose biosensor based on MXene nanocomposite. *Scientific Reports* **6,** 36422 (2016).
176. Shahzad, A. *et al.* Two-Dimensional Ti3C2Tx MXene Nanosheets for Efficient Copper Removal from Water. *ACS Sustainable Chemistry & Engineering* **5,** 11481–11488 (2017).
177. Wu, L. *et al.* 2D transition metal carbide MXene as a robust biosensing platform for enzyme immobilization and ultrasensitive detection of phenol. *Biosensors and Bioelectronics* **107,** 69–75. ISSN: 0956-5663 (2018).
178. Arabi Shamsabadi, A., Sharifian Gh., M., Anasori, B. & Soroush, M. Antimicrobial Mode-of-Action of Colloidal Ti3C2Tx MXene Nanosheets. *ACS Sustainable Chemistry & Engineering* **6,** 16586–16596 (2018).
179. Kumar, S., Lei, Y., Alshareef, N. H., Quevedo-Lopez, M. & Salama, K. N. Biofunctionalized two-dimensional Ti3C2 MXenes for ultrasensitive detection of cancer biomarker. *Biosensors and Bioelectronics* **121,** 243–249. ISSN: 0956-5663 (2018).
180. Zheng, J., Wang, B., Ding, A., Weng, B. & Chen, J. Synthesis of MXene/DNA/Pd/Pt nanocomposite for sensitive detection of dopamine. *Journal of Electroanalytical Chemistry* **816,** 189–194. ISSN: 1572-6657 (2018).
181. Pandey, R. P. *et al.* Ultrahigh-flux and fouling-resistant membranes based on layered silver/MXene (Ti3C2Tx) nanosheets. *J. Mater. Chem. A* **6,** 3522–3533 (8 2018).
182. Zhao, X. *et al.* Antioxidants Unlock Shelf-Stable Ti3C2Tx (MXene) Nanosheet Dispersions. *Matter* **1,** 513 (2019).
183. Natu, V. *et al.* Edge Capping of 2D-MXene Sheets with Polyanionic Salts To Mitigate Oxidation in Aqueous Colloidal Suspensions. *Angewandte Chemie International Edition* **58,** 12655–12660 (2019).
184. Zhu, J., Tang, Y., Yang, C., Wang, F. & Cao, M. Composites of TiO2 Nanoparticles Deposited on Ti3C2 MXene Nanosheets with Enhanced Electrochemical Performance. *Journal of The Electrochemical Society* **163,** A785–A791 (2016).
185. Zhang, C. J. *et al.* Oxidation Stability of Colloidal Two-Dimensional Titanium Carbides (MXenes). *Chemistry of Materials* **29,** 4848–4856 (2017).
186. Dai, C. *et al.* Two-Dimensional Tantalum Carbide (MXenes) Composite Nanosheets for Multiple Imaging-Guided Photothermal Tumor Ablation. *ACS Nano* **11,** 12696–12712 (2017).
187. Lin, H., Wang, Y., Gao, S., Chen, Y. & Shi, J. Theranostic 2D Tantalum Carbide (MXene). *Advanced Materials* **30,** 1703284 (2018).

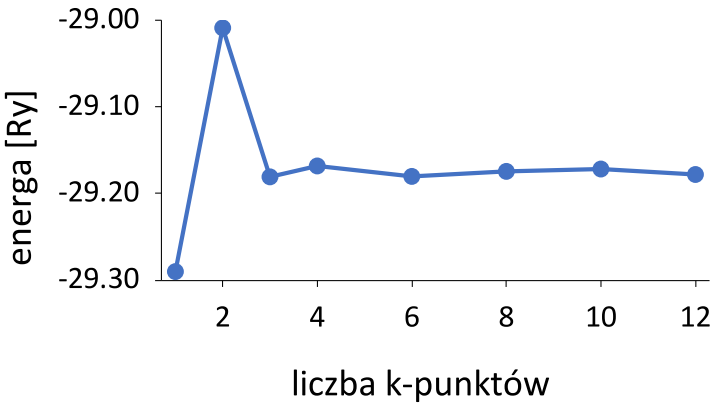
188. Liu, Z. *et al.* 2D Superparamagnetic Tantalum Carbide Composite MXenes for Efficient Breast-Cancer Theranostics. *Theranostics* **8,** 1648–1664 (2018).
189. Lin, H., Gao, S., Dai, C., Chen, Y. & Shi, J. A Two-Dimensional Biodegradable Niobium Carbide (MXene) for Photothermal Tumor Eradication in NIR-I and NIR-II Biowindows. *Journal of the American Chemical Society* **139,** 16235–16247 (2017).
190. Han, X. *et al.* Therapeutic mesopore construction on 2D Nb2C MXenes for targeted and enhanced chemo-photothermal cancer therapy in NIR-II biowindow. *Theranostics* **8,** 4491–4508 (2018).
191. Mashtalir, O., Lukatskaya, M. R., Zhao, M.-Q., Barsoum, M. W. & Gogotsi, Y. Amine-Assisted Delamination of Nb2C MXene for Li-Ion Energy Storage Devices. *Advanced Materials* **27,** 3501–3506 (2015).
192. Peng, C. *et al.* A hydrothermal etching route to synthesis of 2D MXene (Ti3C2, Nb2C): Enhanced exfoliation and improved adsorption performance. *Ceramics International* **44,** 18886–18893. ISSN: 0272-8842 (2018).
193. Szuplewska, A. *et al.* 2D Ti2C (MXene) as a novel highly efficient and selective agent for photothermal therapy. *Materials Science and Engineering: C* **98,** 874–886. ISSN: 0928-4931 (2019).
194. Feng, W. *et al.* Ultrathin Molybdenum Carbide MXene with Fast Biodegradability for Highly Efficient Theory-Oriented Photonic Tumor Hyperthermia. *Advanced Functional Materials* **29,** 1901942 (2019).
195. Tao, Q. *et al.* Two-dimensional Mo1.33C MXene with divacancy ordering prepared from parent 3D laminate with in-plane chemical ordering. *Nature Communications* **8** (2017).
196. F.Liu *et al.* Preparation of High-Purity V2C MXene and Electrochemical Properties as Li-Ion Batteries. *J. Electrochem. Soc.* **164,** 709–A713 (2017).
197. Urbankowski, P. *et al.* Synthesis of two-dimensional titanium nitride Ti4N3 (MXene). *Nanoscale* **8,** 11385–11391 (22 2016).
198. Piliszek, R. *et al.* MDFS - MultiDimensional Feature Selection. arXiv: `1811.00631 [stat.ML]` (2018).
199. Niedzielewski, K., Marchwiany, M., Piliszek, R., Michalewicz, M. & Rudnicki, W. Multidimensional Feature Selection and High Performance ParalleX. *SN Computer Science* **1,** 40 (Oct. 2019).
200. Schölkopf, B., Williamson, R., Smola, A., Shawe-Taylor, J. & Platt, J. *Support Vector Method for Novelty Detection* in. **12** (Jan. 1999), 582–588.
201. Liu, F. T., Ting, K. M. & Zhou, Z.-H. *Isolation Forest* in *2008 Eighth IEEE International Conference on Data Mining* (2008), 413–422.
202. Rousseeuw, P. & Driessen, K. A Fast Algorithm for the Minimum Covariance Determinant Estimator. *Technometrics* **41,** 212–223 (Aug. 1999).
203. Chapman, J., Batra, R. & Ramprasad, R. Machine learning models for the prediction of energy, forces, and stresses for Platinum. *Computational Materials Science* **174,** 109483. ISSN: 0927-0256 (2020).

204. Ward, L. *et al.* Including crystal structure attributes in machine learning models of formation energies via Voronoi tessellations. *Physical Review B* **96** (July 2017).
205. Montavon, G. *et al.* Learning Invariant Representations of Molecules for Atomization Energy Prediction. *Advances in Neural Information Processing Systems* **1,** 449–457 (Jan. 2012).
206. Collins, C., Gordon, G., von Lilienfeld, A. & Yaron, D. Constant Size Molecular Descriptors For Use With Machine Learning (Jan. 2017).
207. Schütt, K. *et al.* How to represent crystal structures for machine learning: Towards fast prediction of electronic properties. *Physical Review B* **89** (July 2013).
208. Ward, L. *et al.* Machine learning prediction of accurate atomization energies of organic molecules from low-fidelity quantum chemical calculations. *MRS Communications* **9,** 1–9 (Aug. 2019).
209. Nyshadham, C. *et al.* Machine-learned multi-system surrogate models for materials prediction. *npj Computational Materials* **5** (Dec. 2019).
210. Stuke, A. *et al.* Chemical diversity in molecular orbital energy predictions with kernel ridge regression. *The Journal of Chemical Physics* **150,** 204121 (2019).
211. Li, L. *et al.* Understanding Machine-learned Density Functionals. *International Journal of Quantum Chemistry* **116** (Apr. 2014).
212. Behler, J. Perspective: Machine learning potentials for atomistic simulations. *The Journal of Chemical Physics* **145,** 170901 (Nov. 2016).
213. Barker, J., Bulin, J., Hamaekers, J. & Mathias, S. Localized Coulomb Descriptors for the Gaussian Approximation Potential (Nov. 2016).
214. Butler, K., Davies, D., Cartwright, H., Isayev, O. & Walsh, A. Machine learning for molecular and materials science. *Nature* **559** (July 2018).
215. https://www.python.org/.
216. https://numpy.org/.
217. https://pandas.pydata.org/.
218. http://www.creativyst.com/Doc/Articles/CSV/CSV01.html.
219. https://scikit-learn.org/stable/.
220. https://wiki.fysik.dtu.dk/ase/index.html.
221. Himanen, L. *et al.* DScribe: Library of descriptors for machine learning in materials science. *Computer Physics Communications* **247,** 106949. ISSN: 0010-4655 (2020).
222. https://singroup.github.io/dscribe/index.html.
223. DalcĂn, L., Paz, R. & Storti, M. MPI for Python. *Journal of Parallel and Distributed Computing* **65,** 1108–1115. ISSN: 0743-7315 (2005).
224. https://www.mpi-forum.org/.
225. Giannozzi, P. *et al.* QUANTUM ESPRESSO: a modular and open-source software project for quantum simulations of materials (Jan. 2009).

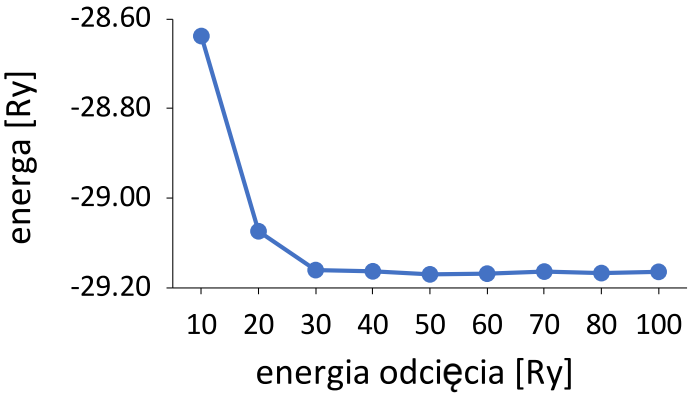
Dodatek A

Dobór optymalnych parametrów dla kodu Quantum Espresso

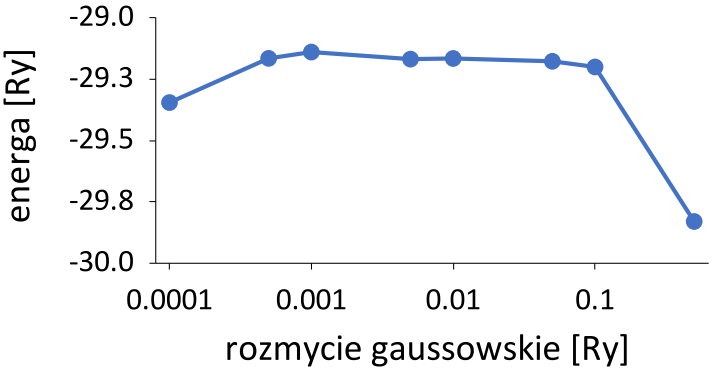
Na wykresach A.1, A.2, A.3 oraz A.4 przedstawiono wyniki uzyskane podczas doboru parametrów obliczeń. Na ich podstawie podczas obliczeń użyto parametrów; liczba punktów k 6, energia odcięcia 40 Ry, rozmycie gaussowskie 0.01 oraz stałą sieci dla obszaru próżni 35 Å.



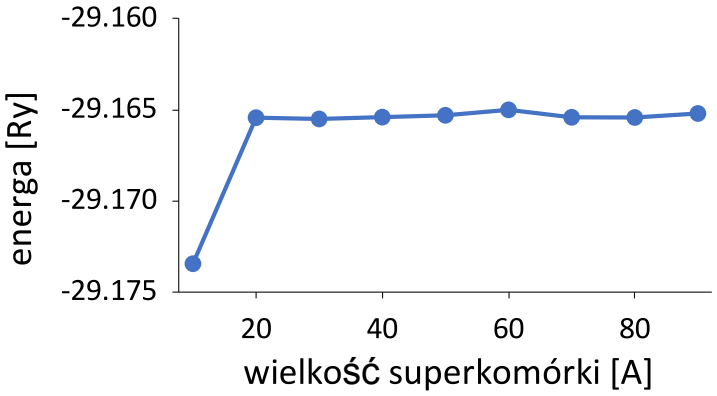
Rysunek A.1: Skalowalność energii nanorurki boru w funkcji liczby k-punktów.



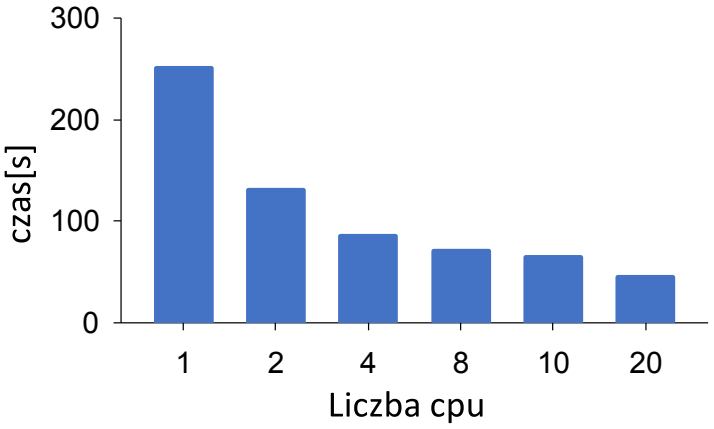
Rysunek A.2: Skalowalność energii nanorurki boru w funkcji odcięcia energii dla rozkładu funkcji falowych na fale płaskie.



Rysunek A.3: Skalowalność energii nanorurki boru w funkcji rozmycia gaussowskiego.



Rysunek A.4: Skalowalność energii nanorurki boru w funkcji wielkości superkomórki (stałej sieci).



Rysunek A.5: Czas obliczeń przy użyciu kodu Quantum Espresso w funkcji liczby procesorów.

Dodatek B

MSE przewidywania energii całkowitej i kohezji nanorurek borowych

Tabela B.1: Wartość metryki MSE przewidywania energii całkowitej dla nanorurek borowych obliczona na **zbiorze treningowym** oraz dla **testowania krzyżowego (CV)**. Oznaczenia modeli są opisane w tabeli 5.2

	zbiór treningowy			CV		
model	ESM	CM	SM	ESM	CM	SM
regRF	0,71	0,84	1,19	1,66	1,89	3,26
XRT	0,00	0,00	0,00	1,76	2,06	3,46
regLIN	6,69	2,50	2,96	6,57	1,98	3,26
regLIN-L1	6,77	2,55	3,04	6,65	2,10	3,36
regLIN-L2	6,69	2,50	2,96	6,55	1,98	3,26
SVM-lin	13,82	2,50	2,96	14,02	1,98	3,26
SVM-rbf	13,00	8,47	10,80	26,80	15,09	21,47
SVM-sig	0,00	0,11	0,11	29,54	25,96	27,93
Ada-lin	7,34	3,23	5,79	7,08	2,68	6,36
Ada-tree	0,08	0,16	0,16	1,51	1,76	3,20
KRR-lin	14,12	1,92	3,02	14,22	1,78	3,55
KRR-rbf	14,01	8,42	11,01	26,70	15,19	20,03

Tabela B.2: Wartość metryki MSE przewidywania energii kohezji dla nanorurek borowych obliczona podczas **zbiorze treningowym** oraz dla **testowania krzyżowego (CV)**. Oznaczenia modeli są opisane w tabeli 5.2

	uczenie			CV		
model	ESM	CM	SM	ESM	CM	SM
refRF	1,37	1,41	1,91	3,57	3,72	5,18
XRT	0,00	0,00	0,00	3,77	3,96	5,32
XGBoost	1,47	1,92	2,70	3,62	3,84	5,19
regLIN	4,05	3,59	4,63	4,75	3,99	5,55
RegLIN-L1	4,23	3,75	4,82	4,49	3,84	5,12
RegLIN-L2	4,07	3,59	4,63	4,72	3,98	5,54
SVM-lin	4,58	3,77	4,87	4,98	4,00	5,31
SVM-rbf	6,05	5,97	6,02	6,50	6,41	6,49
SVM-sig	0,01	0,01	0,01	12,71	15,11	13,77
Ada-lin	9,59	4,31	10,19	14,90	9,10	12,43
Ada-tree	0,24	0,27	0,50	3,63	3,70	5,24
KRR-lin	4,40	3,59	4,63	5,08	3,98	5,54
KRR-rbf	3,32	3,68	3,33	6,42	6,54	6,47

Dodatek C

MSE przewidywania energii całkowitej i kohezji nanorurek azotku-borowych

Tabela C.1: Wartość metryki MSE obliczona na **zbiorze treningowym** oraz dla **testowania krzyżowego (CV)** podczas przewidywania energii całkowitej nanorurki azotku-boru. Oznaczenia modeli są opisane w tabeli 5.2

	zbiór treningowy			CV		
model	ESM	CM	SM	ESM	CM	SM
regRF	2,10	1,03	1,66	6,18	2,94	4,64
XRT	0	0	0	6,24	2,47	4,33
XGB	4,25	1,73	2,86	6,32	2,60	4,11
regLIN	14,28	3,11	4,18	15,68	3,29	4,40
regLIN-L1	10,47	3,06	4,21	11,23	3,15	4,32
regLIN-L2	10,44	3,10	4,18	11,23	3,29	4,40
SVM-lin	25,13	3,05	4,07	26,74	3,23	4,29
SVM-rbf	23,85	18,54	20,74	50,74	36,91	41,28
SVM-sig	0,00	0,23	0,16	53,12	43,55	49,82
Ada-lin	14,81	3,76	6,38	13,83	4,16	5,74
Ada-tree	0,16	0,16	0,20	6,22	2,72	4,43
KRR-lin	24,94	3,15	4,37	26,44	3,24	4,20
KRR-rbf	23,65	18,44	20,64	49,94	26,91	40,73

Tabela C.2: Wartość metryki MSE obliczona na **zbiorze treningowym** oraz dla **testowania krzyżowego (CV)** podczas przewidywania energii kohezji nanorurki azotku-boru. Oznaczenia modeli są opisane w tabeli 5.2

	zbiór treningowy			CV		
model	ESM	CM	SM	ESM	CM	SM
refRF	2,11	1,73	2,79	5,77	4,75	7,38
XRT	0,00	0,00	0,00	6,01	4,59	8,37
XGBoost	1,32	1,45	2,50	5,97	4,96	7,39
regLIN	6,00	4,12	6,14	10,12	6,13	8,71
RegLIN-L1	5,76	4,43	6,41	7,19	5,20	7,50
RegLIN-L2	5,61	4,12	6,14	8,05	6,12	8,70
SVM-lin	6,60	4,58	6,78	8,70	6,15	8,72
SVM-rbf	10,64	10,35	10,54	11,10	10,84	11,02
SVM-sig	1,00	0,96	0,67	14,56	12,98	17,65
Ada-lin	5,77	5,17	7,98	8,58	8,41	11,57
Ada-tree	0,31	0,41	0,51	6,13	5,08	7,70
KRR-lin	6,07	4,13	6,19	8,87	6,16	8,62
KRR-rbf	5,61	5,64	5,69	11,16	11,19	11,27

Dodatek D

Opis zmiennych w zbiorze doświadczalnym MXenes

Tabela D.1: Szczegółowy opis zmiennych w zbiorze doświadczalnym użytych do budowania modelu dla MXenes

Nazwa zmiennej	zakres wartości
Modyfikacje powierzchni	PVP, SP,$MNie_x$+SP, HA, Fe_xO_y+SP, PEI, PEG, DNA+Pt+Pd, Ag, L-ACC, PAS CTAC+PEG+SiO_2+c(RGDyC)+SiO_2, Au+Fe_3O_4, Au+PEG, PLL, APTES+CEA, PVA, Au
Rozmiar boczny	od kilku do kilkuset nm, od kilku do kilkuset μm
Grubość	od kilku do kilkuset nm
Środek trawiący	HF, LiF+HCl, LiF+HCl+$AlCl_3$, NaF+HCl, KF+HCl, KF+LiF+NaF,
Środek rozwarstwiający	ultradźwięki, DMF+wysokie ciśnienie+wysoka temperatura, wysokie ciśnienie+wysoka temperatura, TBAOH, TBAOH+ultradźwięki, TPAOH, TMAOH, DMSO+ultradźwięki, brak
Węgiel (C) na powierzchni	Tak, Nie
Tlen (O) na powierzchni	Tak, Nie
Fluor (F) na powierzchni	Tak, Nie
Aluminium (Al) na powierzchni	Tak, Nie
Tytan (Ti) na powierzchni	Tak, Nie
Azot (N) na powierzchni	Tak, Nie
Chlor (Cl) na powierzchni	Tak, Nie
Krzem (Si) na powierzchni	Tak, Nie
Lit (Li) na powierzchni	Tak, Nie
Inne pierwiastki na powierzchni	Tak, Nie
Obecność M_xO_y	Tak, Nie
Toksyczność	Tak, Nie

Skróty użyte w tabeli D.1:
SP - Fosfolipid sojowy (ang. Soybean phospholipid);

PVP – poliwinylopirolidon (ang. polyvinylpyrrolidone);
HA - kwas hialuronowy (ang. hyaluronic acid);
PEI – polietylenoimina (ang. polyethylene imine);
PEG – glikol polietylenowy (ang. polyethylene glycol);
PVA – alkohol poliwinylowy (ang. polyvinyl alcohol);
NMP - N-metylo-2-pirolidon (ang. N-methyl-2-pyrrolidone);
TMAOH - wodorotlenek tetrametyloamoniowy (ang. tetramethylammonium hydroxide);
TBAOH - wodorotlenek tetrabutyloamoniowy (ang. tetrabuthylammonium hydroxide);
TPAOH - wodorotlenek tetrapropyloamoniowy (ang. tetrapropylammonium hydroxide);
DMF - dimetyloformamid (ang. dimethylformamide);
DMSO – dimetylosulfotlenek (ang. dimethyl sulfoxide);
CTAC - chlorek cetan trimetyloamoniowy (ang. cetanecyltrimethylammonium chloride);
c(RGDyC) - cykliczny pentapeptyd arginina-glicyno-asparaginowy (ang. cyclic arginine-glycine-aspartic pentapeptide);
PLL – poli-L-lizyna (ang. poly-L-lysine);
APTES - (3-aminopropylo)trietoksysilan (ang. (3-aminopropyl)triethoxysilane);
CEA - antygen karcynoembrionalny (ang. carcinoembryonic antigen);
L-ACC – kwas L-askorbinowy (ang. L-ascorbic acid);
PAS – sole polianionowe (ang. polyanionic salts).

Dodatek E

Energia kohezji MXenes

Tabela E.1: Wartość metryki MSE dla przewidywania energii MXenes obliczona podczas uczenia modelu. Oznaczenia modeli są opisane w tabeli 5.2

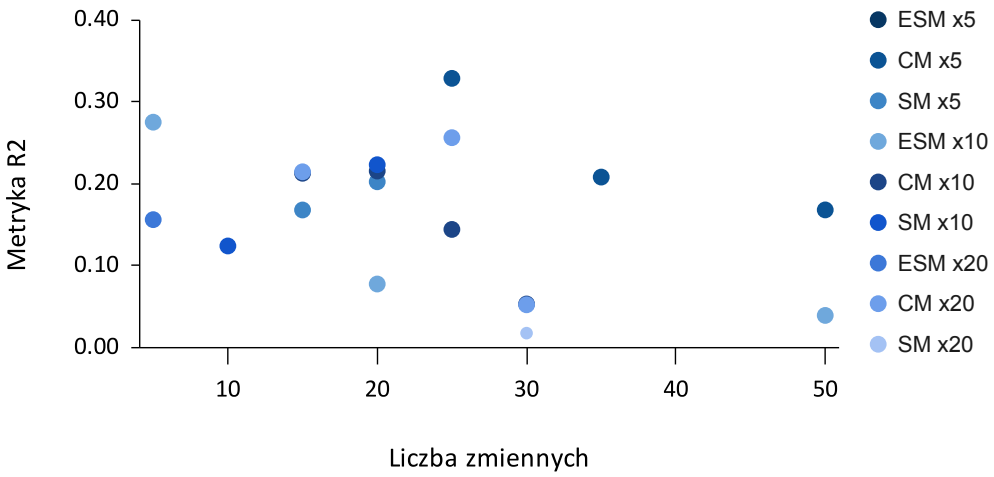
MSE	ESM	CM	SM	G2	G4	wG2	wG4	SF	wSF	AF
regRF	0,05	0,04	0,04	0,06	0,07	0,03	0,05	0,05	0,04	0,05
regLIN	0,01	0,00	0,00	0,20	0,22	0,20	0,22	0,16	0,16	0,08
regLIN-l1	0,08	0,08	0,10	0,27	0,27	0,22	0,22	0,27	0,20	0,23
regLIN-l2	0,01	0,00	0,00	0,21	0,24	0,20	0,22	0,19	0,16	0,10
regSVM-lin	0,03	2,93	3,80	0,23	0,25	0,22	0,48	0,21	0,24	0,11
regSVM-rbf	0,01	0,01	0,01	0,18	0,25	0,01	0,01	0,20	0,01	0,01
regSVM-sig	109,68	0,29	0,29	1,70	0,28	0,29	34,52	1,78	0,29	258,91
ADA-tree	0,03	0,03	0,03	0,06	0,07	0,03	0,05	0,03	0,03	0,03
ADA-LIN	0,10	0,03	0,04	0,17	0,23	0,17	0,25	0,08	0,10	0,13
XRT	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00	0,00
KRR-lin	0,01	0,00	0,00	0,26	0,43	0,21	0,40	0,23	0,16	0,11
KRR-rbf	0,18	0,18	0,18	0,20	0,25	0,17	0,18	0,21	0,18	0,18

Tabela E.2: Wartość metryki MSE obliczona dla przewidywania energii MXenes dla dziesięciokrotnego testowania krzyżowego. Wartość X oznacza wartość większą niż 1000. Oznaczenia modeli są opisane w tabeli 5.2

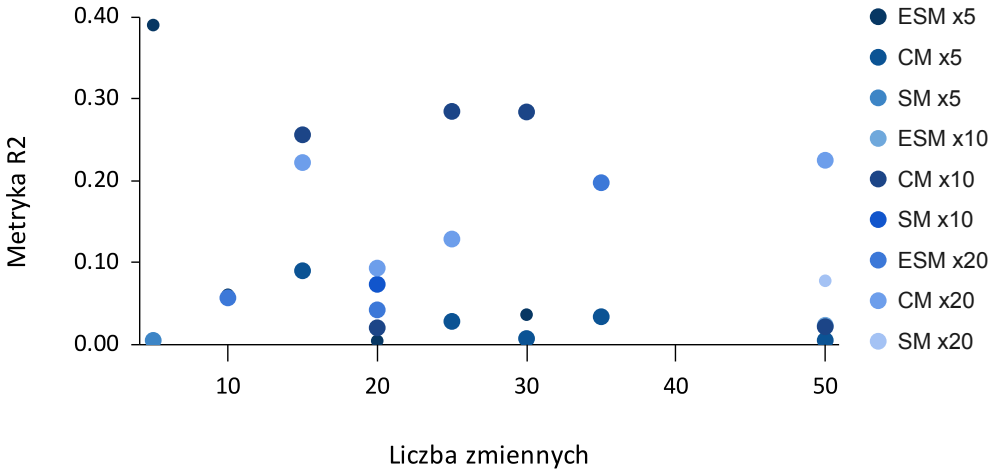
MSE	ESM	CM	SM	G2	G4	wG2	wG4	SF	wSF	AF
regRF	0,24	0,19	0,21	0,15	0,28	0,18	0,33	0,18	0,44	0,26
regLIN	X	X	X	0,39	0,27	0,50	0,27	0,47	0,94	X
regLIN-L1	0,33	0,45	0,34	0,18	0,27	0,26	0,40	0,23	0,20	0,24
regLIN-L2	1,04	0,42	0,24	0,27	0,26	0,17	0,19	0,37	0,40	0,33
regSVM-lin	1,42	0,52	0,89	0,21	0,39	0,32	0,27	0,44	0,70	0,52
regSVM-rbf	0,14	0,34	0,35	0,35	0,30	0,16	0,45	0,36	0,37	0,35
regSVM-sig	1,90	20,14	11,21	2,09	0,62	0,60	0,36	0,37	1,19	0,38
ADA-tree	0,24	0,46	0,27	0,23	0,48	0,27	0,34	0,23	0,34	0,51
ADA-LIN	0,69	0,36	0,82	0,41	0,31	0,31	0,67	0,92	0,37	0,37
XRT	0,29	0,26	0,25	0,23	0,42	0,23	0,38	0,31	0,34	0,21
KRR-lin	28,23	31,49	31,69	1,32	0,97	2,21	0,81	0,82	1,33	1,59
KRR-rbf	0,36	0,47	0,40	0,31	0,39	0,22	0,17	0,21	0,38	0,54

Dodatek F

Zależność dokładności przewidywania energii kohezji od liczby zmiennych dla MXenes

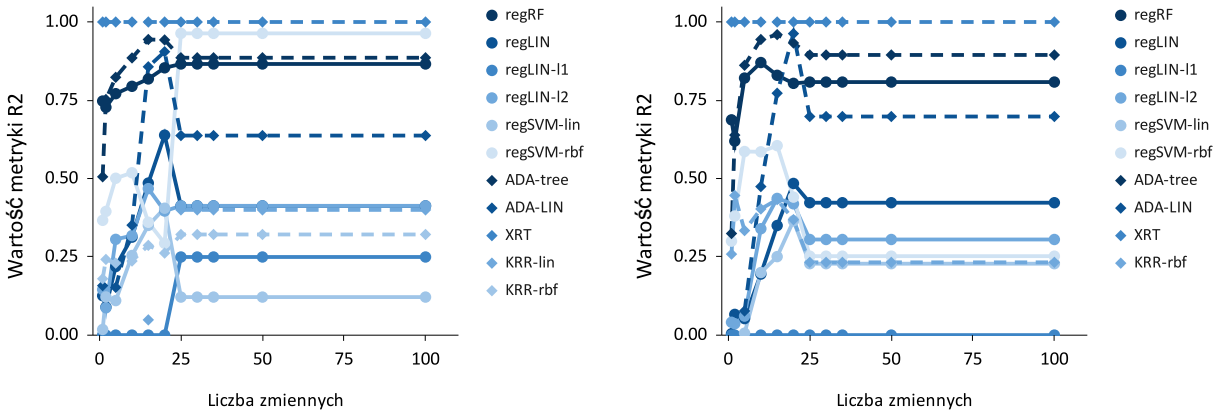


(a) szum zmiennych położeniowych



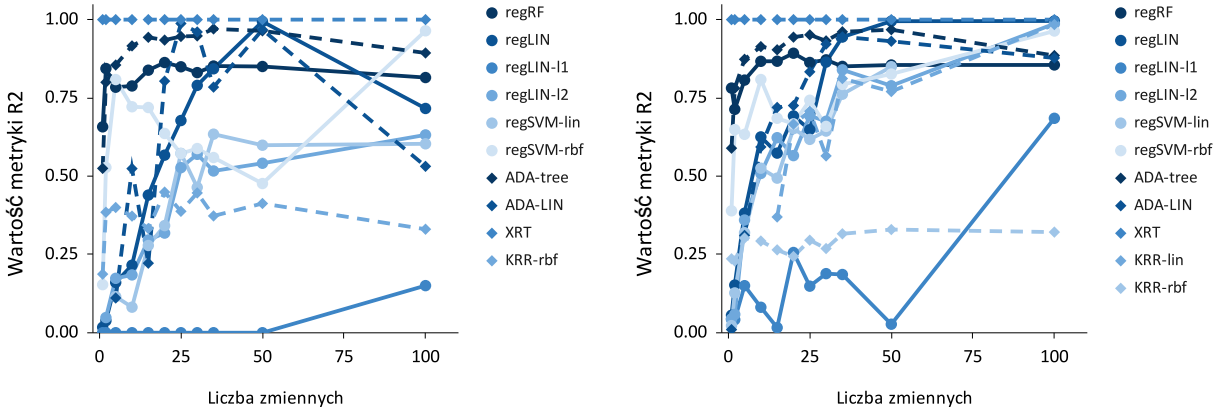
(b) szumu położeń atomów

Rysunek F.1: Wartość metryki R2 przewidywanie energii kohezji MXenes w funkcji liczby zmiennych generowanych przez PCA dla syntetycznie zwielokrotnionych zbiorów danych, poprzez dodanie szumu do (a) położeń atomów oraz do (b) zmiennych położeniowych. Oznaczenia modeli są opisane w tabeli 5.2.



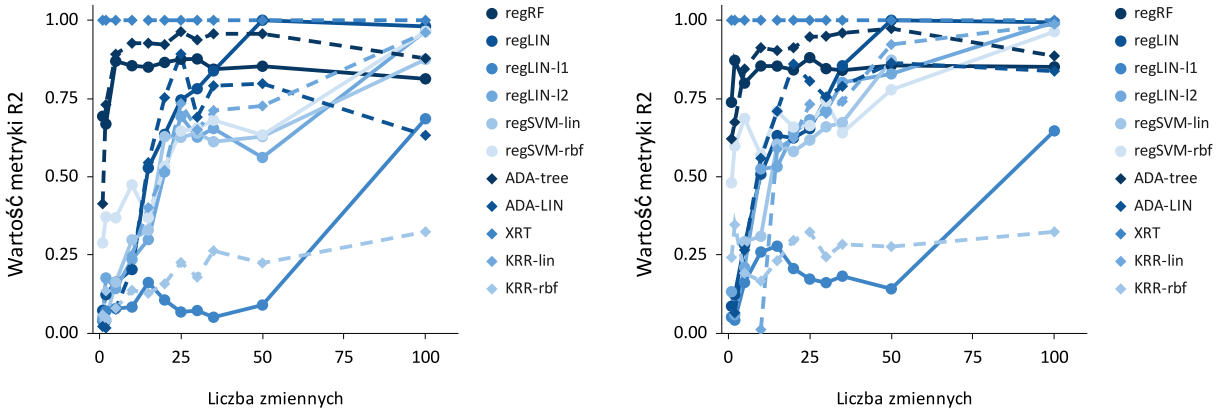
(a) wSF

(b) SF



(c) AF

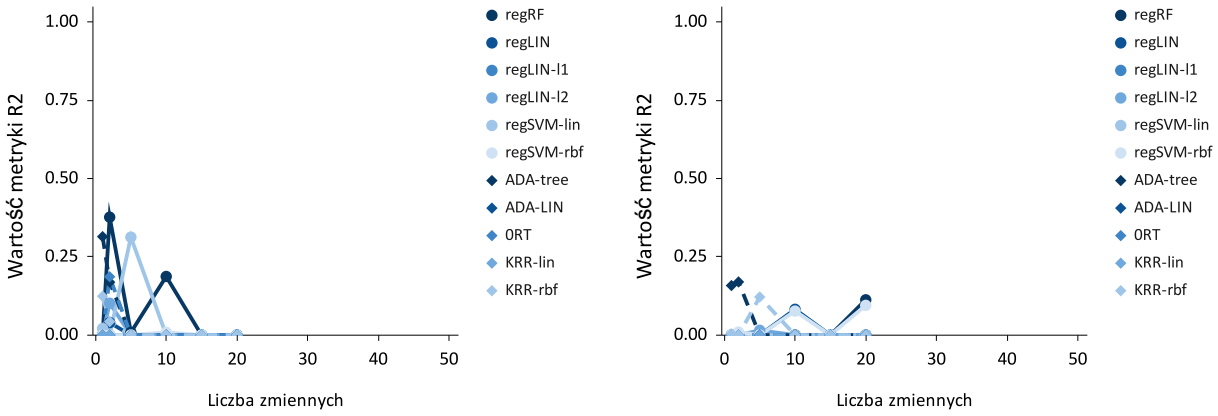
(d) CM



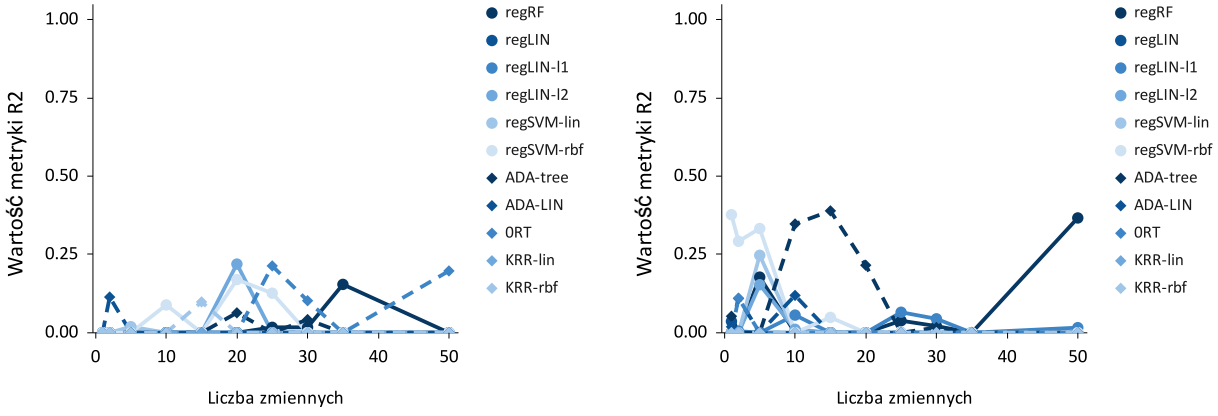
(e) ESM

(f) SM

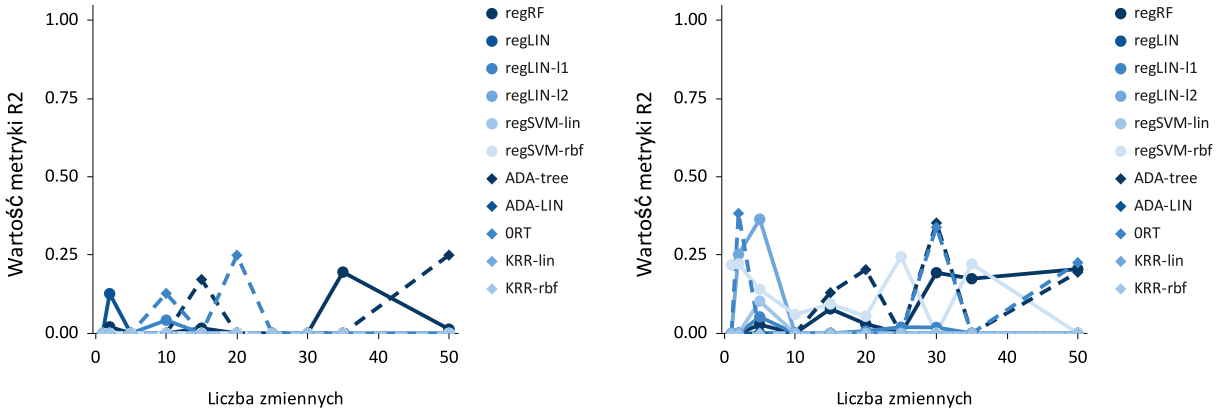
Rysunek F.2: Wartość metryki R2 przewidywanie energii kohezji MXenes w funkcji liczby zmiennych generowanych przez PCA na zbiorze treningowym dla różnych zmiennych położeniowych: (a) Ważone funkcje symetrii, (b) Funkcje symetrii, (c) Odcisk atomowy, (d) Macierz Kulombowska, (e) Macierz Ewalda, (f) Macierz sinusoid. Oznaczenia modeli są opisane w tabeli 5.2.



(a) wSF (b) SF



(c) AF (d) CM



(e) ESM (f) SM

Rysunek F.3: Wartość metryki R2 przewidywanie energii kohezji MXenes w funkcji liczby zmiennych generowanych przez PCA dla testowania krzyżowego dla różnych zmiennych położeniowych: (a) Ważone funkcje symetrii, (b) Funkcje symetrii, (c) Odcisk atomowy, (d) Macierz Kulombowska, (e) Macierz Ewalda, (f) Macierz sinusoid. Wartości ujemne zostały pominięte na wykresach. Oznaczenia modeli są opisane w tabeli 5.2.

Dodatek G

Listingi kodów

```
#załadowanie bibliotek
import pandas as pd
from sklearn import linear_model
from sklearn.model_selection import train_test_split

#wczytanie danych z pliku input.csv
data = pd.read_csv("input.csv")

#wyodrębnienie zmiennej zależnej Y
#i zmiennych niezależnych X
Y = data['y']
x = data.drop(['y'], axis=1)

#podział zbioru na podzbiór treningowy i walidacyjny
X_train, X_test, Y_train, Y_test = train_test_split(X, Y)
#budowa i uczenie modelu na podzbiorze treningowym
reg = linear_model.LinearRegression()
reg.fit(X_train, Y_train)

#przewidywanie na podzbiorze treningowym
y_out_train = reg.predict(X_train)

#przewidywanie na podzbiorze walidacyjnym
y_out_test = reg.predict(X_test)

#wypisanie wyniku
print('MSE_trein:', mean_squared_error(Y_train, y_out_train))
print('MSE_test:', mean_squared_error(Y_ttest, y_out_test))
```

Listing 1: Kod budowania i walidacji modelu liniowego

```
#załadowanie bibliotek
import pandas as pd
from sklearn import linear_model
from sklearn.model_selection import cross_val_score

#wczytanie danych z pliku input.csv
data = pd.read_csv("input.csv")

#wyodrębnienie zmiennej zależnej Y
#i zmiennych niezależnych X
Y = data['y']
x = data.drop(['y'], axis=1)

#budowa i uczenie modelu na podzbiorze treningowym
reg = linear_model.LinearRegression()

#wypisanie wyniku
print('MSE_CV:', -cross_val_score(reg, X, Y,
                    scoring='neg_mean_squared_error', cv=5))
```

Listing 2: Kod obliczania metryki R2 poprzez 5-krotnego testowania krzyżowego modelu liniowego

```
#załadowanie bibliotek
from ase.io import read
from dscribe.descriptors import CoulombMatrix

#maksymalna liczba atomów w obserwacji
N_MAX = 10
#ścieżka dostępu do pliku
root = r'INPUTY/VASP'
filePOSCAR = root+'/POSCAR'

#wczytanie położeń z pliku POSCAR
out = read(filePOSCAR)

#budowa obiektu cm
cm = CoulombMatrix(n_atoms_max=N_MAX,
                    permutation='sorted_l2')

#obliczanie wrtości elementów Coulomb Matrix
features =  cm.create(out)[0]
```

Listing 3: Kod obliczania Macierzy Kulombowskiej

```
#załadowanie bibliotek
import numpy as np
from ase.io import read

#funkcja obliczająca macierz odległości
#INPUTS:
#  xyz - położenia atomów
def dyst_ontTOall(xyz):
       dyst = np.ones( (len(xyz), len(xyz)) )*100
       for i in range(1, len(xyz)):
              dyst[i, :] = np.sqrt(np.sum(np.power(xyz -
              np.roll(xyz, i, axis=0),2),axis=1))
       return  np.transpose(dyst)

#funkcja odcięcia:
#INPUTS:
#  dyst - odległość
#  Rc - promień odcięcia
def fc(r, rc):
       out = np.zeros_like(r)
   out = 0.5*(np.cos(np.pi * r / rc) + 1)
       out[r>rc] = 0
       return out
    #wektorowa wersja fc
vfc = np.vectorize(fc)

#Symetry functions G2
#INPUTS:
#  dyst - macierz odległości pomiędzy atomami
#  a, d - parametry funkcji
#  Rc - promień odcięcia
#UWAGA:
#  wymaga wcześniejszego zdefiniowania funkcji odcięcia fc
def G2(dyst, a, d, Rc=6):
       return np.sum(np.exp(-a * np.square(dyst - d))
                            * vfc(dyst, Rc), axis=1)

#ścieżka dostępu do pliku
root = r'INPUTY/VASP'
filePOSCAR = root+'/POSCAR'

#wczytanie położeń z pliku POSCAR
out = read(filePOSCAR)
xyz = out.positions

#obliczanie macierzy odległości
dyst = dyst_ontTOall(xyz=xyz)

features5 = np.sort(G2(dyst, 1, 0.5, 6))
```

Listing 4: Kod obliczania funkcji symetrii G2

```
#załadowanie bibliotek
from mpi4py import MPI

#stworzenie komunikatora
comm = MPI.COMM_WORLD
#obliczenie numeru procesu
#oraz wielkości komunikatora
comm_rank = comm.Get_rank()
comm_size = comm.Get_size()

#lista wszystkich modeli
MODELS = ('regRF', 'regLIN', 'regLIN_L1', 'regLIN_L2',
'regSVM_lin', 'regSVM_rbf', 'regSVM_sig', 'ADA_tree',
'ADA_LIN', 'XRT')

#obliczanie liczby modeli na proces
step = int(len(MODELS) / comm_size)
if step == 0:
       exit(1)

#początkowy i końcowy indeks modeli przypadających na jeden proces
mod_name_down = step * comm_rank
mod_name_up = step * (comm_rank +1)

#słownik przechowujący wyniki wszystkich modeli
scoreALL = dict()
#pętla po wszystkich modelach rozdzielona pomiędzy procesy
for model in MODELS[y_name_down:y_name_up]:
   \dots
   scoreALL[model] = score
   \dots
#agregacja wyniku
scoreALL = comm.gather(scoreALL, root=0)
```

Listing 5: Fragment kodu zarównoleglonego MPI4py