Prace Instytutu Mechaniki Górotworu PAN
Tom 26, nr 1-4, Marzec-Grudzień 2024, s. 69-75

Uczenie maszynowe do predykcji szeregu czasowego przebiegu krzywej dyfuzji

Mateusz Zaręba

Akademia Górniczo-Hutnicza im. Stanisława Staszica w Krakowie,
Wydział Geologii, Geofizyki i Ochrony Środowiska, al. Mickiewicza 30, 30-059 Kraków

Marta Skiba

Instytut Mechaniki Górotworu Polskiej Akademii Nauk, ul. Władysława Reymonta 27, 30-059 Kraków

Streszczenie

W pracy wykorzystano metody uczenia maszynowego do predykcji szeregu czasowego krzywej dyfuzji. Krzywe dyfuzji zostały wygenerowane w języku Python przy użyciu biblioteki NumPy. Dane zostały sformatowane jako szereg czasowy, co umożliwiło efektywne trenowanie i testowanie modeli predykcyjnych. Do modelowania dynamiki procesu dyfuzji wykorzystano szeroki zestaw metod obejmujących modele statystyczne, sieci neuronowe oraz podejścia regresyjne. Przeprowadzona analiza modeli wskazuje na istotne różnice w ich skuteczności. Modele regresyjne wykazały najwyższą precyzję prognozowania, osiągając najniższe wartości błędów, co sugeruje ich większą zdolność do uchwycenia charakterystyki analizowanego zjawiska. Z kolei modele szeregów czasowych, takie jak ARIMA i TCN, charakteryzowały się wyższymi błędami, co może wynikać z trudności w odwzorowaniu dynamiki procesu.

Słowa kluczowe: efektywny współczynnik dyfuzji (*De*); uczenie maszynowe; szereg czasowy; model regresyjny

1. Wstęp

Zagrożenie metanowe związane jest z występowaniem metanu w górotworze i jego uwalnianiem się w wyniku prowadzonej działalności górniczej. Pomimo znaczącego postępu w rozpoznawaniu i zwalczaniu zagrożenia metanowego, obserwuje się jego narastanie w obszarach górniczych wielu kopalń. Jest to związane ze zwiększaniem głębokości prowadzenia eksploatacji, coraz większą metanonośnością pokładów oraz ciśnieniem złożowym gazów. Obecność metanu odpowiada także za występowanie zjawisk gazogeodynamicznych, wśród których największe zagrożenie związane jest z wyrzutami metanu i skał. Te niebezpieczne zjawiska występują najczęściej w węglu ze stref przyuskokowych, odmienionym strukturalnie (Lama i Bodziony, 1998; Godyń i Kožušníková, 2019).

Podstawę oceny stanu wspomnianych zagrożeń w pokładach węgla kamiennego stanowi analiza parametrów sorpcyjnych węgla. Do określenia właściwości układu węgiel – gaz w warunkach laboratoryjnych wykorzystuje się przede wszystkim dwa parametry, tj. pojemność sorpcyjną *a* oraz efektywny współczynnik dyfuzji *De*. Stanowią one uzupełnienie badań dołowych w zakresie rozpoznania zagrożenia metanowego i wyrzutowego w kopalniach. Pojemność sorpcyjna określa zdolność pokładów do akumulacji gazu, natomiast efektywny współczynnik dyfuzji decyduje o tempie emisji gazu z pokładów. Wśród metod stosowanych do pomiarów sorpcji gazu istotne znaczenie mają metody grawimetryczne (Zhang i in., 2013), gdzie ilość zasorbowanego gazu wyznaczana jest w sposób bezpośredni, na podstawie pomiaru przyrostu masy badanego sorbentu, po wprowadzeniu do układu gazowego sorbatu, przy zachowaniu stałego ciśnienia oraz temperatury. Metody te posiadają szereg zalet (Levine, 1992; Saghafi i in., 2007), jednak z uwagi na ich czasochłonność, związaną z osiągnięciem przez układ równowagi sorpcyjnej, a także wysoki koszt komercyjnie dostępnych urządzeń grawimetrycznych, ich wykorzystanie do bieżącej prognozy zagrożenia wyrzutami gazów i skał jest znacznie ograniczone.

1.1. Uczenie maszynowe

Uczenie maszynowe (ang. *ML – Machine Learning*) to dziedzina sztucznej inteligencji (ang. *AI – Artificial Intelligence*), która koncentruje się na opracowywaniu algorytmów zdolnych do samodzielnego uczenia się na podstawie różnorodnych danych (Bishop, 2006). Może być wykorzystywane zarówno w badaniach naukowych, jak i w zastosowaniach praktycznych w biznesie, przemyśle czy medycynie. Jednym z głównych celów stosowania uczenia maszynowego jest predykcja przyszłych wartości, co pozwala na lepsze zrozumienie badanych zjawisk, optymalizację procesów eksperymentalnych oraz modelowanie przyszłych scenariuszy. Dzięki temu możliwe jest np. skuteczniejsze planowanie działań w teraźniejszości. Drugim istotnym zastosowaniem jest analiza wzorców w danych, szczególnie w sytuacjach, gdy wielowymiarowość i złożoność czynników utrudniają ich interpretację za pomocą klasycznych metod statystycznych. Algorytmy ML potrafią wykrywać ukryte zależności i struktury, które mogłyby umknąć człowiekowi. Kolejnym obszarem, nieco bardziej odległym od tradycyjnych badań naukowych, jest podejmowanie decyzji na podstawie analizy danych (Alpaydin, 2006). W takich zastosowaniach algorytmy uczenia maszynowego pełnią funkcję quasi-kognitywną – analizują dostępne informacje i wybierają najbardziej optymalne działania w celu osiągnięcia zamierzonego efektu. Jest to kluczowe np. w systemach autonomicznych, finansach czy dynamicznych systemach zarządzania produkcją. Uczenie maszynowe stale się rozwija, poszerzając zakres zastosowań i stając się nieodzownym narzędziem w wielu dziedzinach nauki i technologii (Goodfellow i in., 2016, Sarker i in., 2022).

W ramach uczenia maszynowego wyróżniamy trzy typy uczenia – uczenie nienadzorowane, uczenie nadzorowane oraz uczenie ze wzmocnieniem.

1.1.1. Uczenie nadzorowane

Uczenie nadzorowane (ang. *supervised learning*) polega na budowie modeli predykcyjnych na podstawie zbioru danych, w którym każda obserwacja składa się z wektora cech (zmienne niezależne) oraz odpowiadającej mu wartości docelowej (zmienna zależna) (van Engelen i in., 2020).

Niech:

- $X = \{x_1, x_2, ..., x_n\}$ oznacza zbiór zmiennych niezależnych (cech), gdzie $x_i \in \mathbb{R}^d$ to wektor cech dla i-tej obserwacji.
- $Y = \{y_1, y_2, ..., y_n\}$ oznacza zbiór wartości docelowych (etykiet), gdzie $y_i \in \mathbb{R}$ dla regresji lub $y_i \in \{c_1, c_2, ..., c_k\}$ dla klasyfikacji.
- $M = (X, Y)$ to pełny zbiór danych zawierający n obserwacji.
- M dzielony jest na zbiór treningowy $M_{TR} \subset M$ oraz zbiór testowy $M_{TE} \subset M$, gdzie $M_{TR} \cup \mathrm{M_{TE}} = M$ i $M_{TR} \cap M_{TE} = \varnothing$.

Celem uczenia nadzorowanego jest znalezienie funkcji odwzorowującej $f: X \rightarrow Y$ w taki sposób, aby minimalizować funkcję straty $L(y,\hat{y})$, mierzącą różnicę między rzeczywistą wartością y a przewidywaną $\hat{y} = f(x)$. Przykładowe funkcje straty to (Goodfellow i in., 2016):

1. Dla problemu regresji średni błąd kwadratowy (MSE):

$$MSE = L\left(y,\hat{y}\right) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - \widehat{y_i}\right)^2 \tag{1}$$

2. Dla problemu klasyfikacji może to być entropia krzyżowa:

$$L\left(y,\hat{y}\right) = -\sum_{i=1}^{n} y_i \log \widehat{y_i} \tag{2}$$

Proces uczenia polega na optymalizacji parametrów modelu θ poprzez minimalizację $L(y,\hat{y})$ przy użyciu algorytmów takich jak np. spadek gradientu:

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta L\left(y,\hat{y}\right) \tag{3}$$

gdzie η to współczynnik uczenia (ang. *learning rate*).

Uczenie nadzorowane znajduje zastosowanie w wielu dziedzinach:

- Klasyfikacja – *y* przyjmuje wartości dyskretne, np. rozpoznawanie obrazów, detekcja oszustw.
- Regresja – *y* jest liczbą rzeczywistą, np. prognozowanie cen nieruchomości, analiza ryzyka kredytowego.

1.1.2. Uczenie nienadzorowane

Uczenie nienadzorowane (ang. *unsupervised learning*) to klasa algorytmów uczenia maszynowego, w której model analizuje dane bez wcześniejszych etykiet i samodzielnie odkrywa ukryte wzorce oraz struktury w zbiorze danych (Zaręba i in., 2023) W przeciwieństwie do uczenia nadzorowanego, gdzie dostępne są oznaczone przykłady (pary wejście-wyjście), w uczeniu nienadzorowanym algorytmy same poszukują zależności i regularności w danych wejściowych *X*. Niech $X = \{x_1, x_2, ..., x_n\}$, gdzie $x_i \in \mathbb{R}^d$, oznacza zbiór danych składający się z *n* obiektów o *d* cechach, a $f: X \rightarrow Y$ to funkcja odwzorowująca zbiór wejściowy na nową reprezentację *Z*, która pokazuje nieznane dotąd zależności. Celem uczenia nienadzorowanego jest znalezienie struktury w danych, którą można wyrazić jako: $Z = f(X)$, gdzie *Z* to nowa reprezentacja danych, np. w postaci grup lub zredukowanej liczby wymiarów.

2. Materiały i metody

2.1. Model teoretyczny

W celu wykonania testów modeli uczenia maszynowego na kontrolowanym zbiorze danych proces dyfuzji został zamodelowany przy użyciu analitycznego rozwiązania równania dyfuzji dla idealnych warunków brzegowych. Równanie opisujące zmianę masy w czasie ma postać:

$$\frac{M(t)}{M_\infty} = 1 - \frac{6}{\pi^2}\sum_{1}^{\infty}\exp\left(-\frac{n^2\pi^2 D_e t}{R^2}\right) \tag{4}$$

gdzie $M(\infty)$ to asymptotyczna wartość masy, *D* to współczynnik dyfuzji, *R* to uśredniony promień cząstek, a *t* to czas.

2.2. Symulacja teoretycznych przebiegów krzywych dyfuzji

Aby wygenerować syntetyczne krzywe dyfuzji, wybrane parametry były losowane w pętli w realistycznych zakresach oddając kontrolowany, ale zbliżony do rzeczywistego przebieg. Przyjęto $M(\infty)$ jako zmienną losową o rozkładzie normalnym ze średnią 2,1 cm^3/g i odchyleniem standardowym 0,6 cm^3/g. Współczynnik dyfuzji *D*, będący kluczowym parametrem dla prezentowanego rozwiązania został zdefiniowany jako wartość z rozkładu w zakresie od $2\times10^{\wedge-8}$ do $2\times10^{\wedge-10}$ cm^2/s. Dla promienia cząstek założono stałą wartość równą 0,011157 cm.

Na podstawie tych parametrów obliczono krzywe dyfuzji, numerycznie sumując kolejne wyrazy szeregu w analitycznym rozwiązaniu w pętli dla 1000 krzywych. Początkowe wyrazy szeregu zostały następnie ograniczone do 150. Ograniczenie to miało na celu zachowanie równowagi między dokładnością obliczeń a efektywnością obliczeniową, jednocześnie unikając nadmiernych błędów numerycznych dla małych czasów.

Krzywe dyfuzji zostały wygenerowane w języku Python przy użyciu biblioteki NumPy. Funkcja *wygeneruj_krzywe_dyfuzji()* pozwala obliczyć na siatce czasowej obejmującej zadaną przez użytkownika liczbę kroków w zakresie zdefiniowanym przez użytkownika – domyślnie od 0 do *n* jednostek czasu. Symulację powtórzono dla 500 losowych zestawów parametrów, tworząc różnorodny zbiór danych. Schemat działania algorytmu funkcji *wygenruj_krzywe_dyfuzji()* wygląda następująco:

1. Na podstawie danych wprowadzonych przez użytkownika definiowana jest siatka czasowa do symulacji, poprzez co utworzono liniowy wektor czasu od 0 do *n*, zawierający m punktów.
2. Dla każdego zestawu losowo wybranych parametrów funkcja *wygeneruj_krzywe_dyfuzji()* generuje przebieg krzywej.
3. Wektory czasu oraz odpowiadające im krzywe dyfuzji wraz z zestawem wylosowanych parametrów zapisywane są w formie tablicowej do dalszego przetwarzania.

2.3. Uczenie maszynowe do predykcji szeregu czasowego przebiegu krzywej dyfuzji

Dane zostały sformatowane jako szereg czasowy za pomocą biblioteki *Darts* (Herzen i in., 2022), co umożliwiło efektywne trenowanie i testowanie modeli predykcyjnych. Stworzono obiekt typu *TimeSeries*, w którym wartości $M(t)$ przypisano do kolumny Y, a indeksy czasowe do specjalnej kolumny *sample_index*. Zbiór danych podzielono na część treningową oraz testową zachowując nierówny stosunek wielkości zbiorów – treningowy zawierał większą część zbioru danych, a testowy mniejszą. Podziału dokonano zachowując chronologię czasową co znaczy, że próbki w zbiorze testowym związane są z końcowymi krokami czasowymi.

Do modelowania dynamiki procesu dyfuzji wykorzystano szeroki zestaw metod, obejmujących modele statystyczne, sieci neuronowe oraz podejścia regresyjne:

1) Model ARIMA (ang. *Autoregressive Integrated Moving Average*) – klasyczna, często stosowana metoda do analizy szeregów czasowych (Box i Jenkins, 1976). Składa się z trzech głównych komponentów:
   - Autoregresyjnego – wykorzystuje zależność między obserwacjami w serii czasowej poprzez ich wcześniejsze wartości,
   - Integrancyjnego – eliminuje trend w danych poprzez różnicowanie,
   - Średniej ruchomej – modeluje zależność między bieżącą wartością a błędami wcześniejszych predykcji.

   Model ARIMA jest szczególnie skuteczny dla danych, które wykazują liniowe zależności i stabilną strukturę.
2) Model regresyjny oparty na opóźnieniach – w tym przypadku zastosowano model regresji, który uwzględnia 12 poprzednich wartości czasowych jako zmienne wejściowe. Podejście to pozwala modelowi na wychwycenie zależności między przeszłymi obserwacjami a przyszłymi wartościami. Jest to przykład klasycznej metody statystycznej, która dobrze sprawdza się w sytuacjach, gdy istnieją silne relacje między kolejnymi próbkami.
3) Model TCN (ang. *Temporal Convolutional Network*) – nowoczesna architektura neuronowa wykorzystująca konwolucje do analizy szeregów czasowych (Herzen i in., 2022). W porównaniu do tradycyjnych rekurencyjnych sieci neuronowych (RNN) model TCN oferuje:
   - równoległe przetwarzanie danych (co zwiększa wydajność obliczeniową),
   - zdolność do uchwycenia długoterminowych zależności,
   - stabilność w procesie uczenia dzięki zastosowaniu normalizacji wag.

W analizie wykorzystano TCN z minimalnym oknem wejściowym wynoszącym 10 próbek oraz oknem prognozowania równym 8, przy liczbie epok uczenia równej 50. W celu usunięcia komponentu okresowego otrzymanego w ramach predykcji modelem TCN, wykonano dodatkowo wygładzania krzywej w różnych oknach i oznaczono wyniki jako CNS.

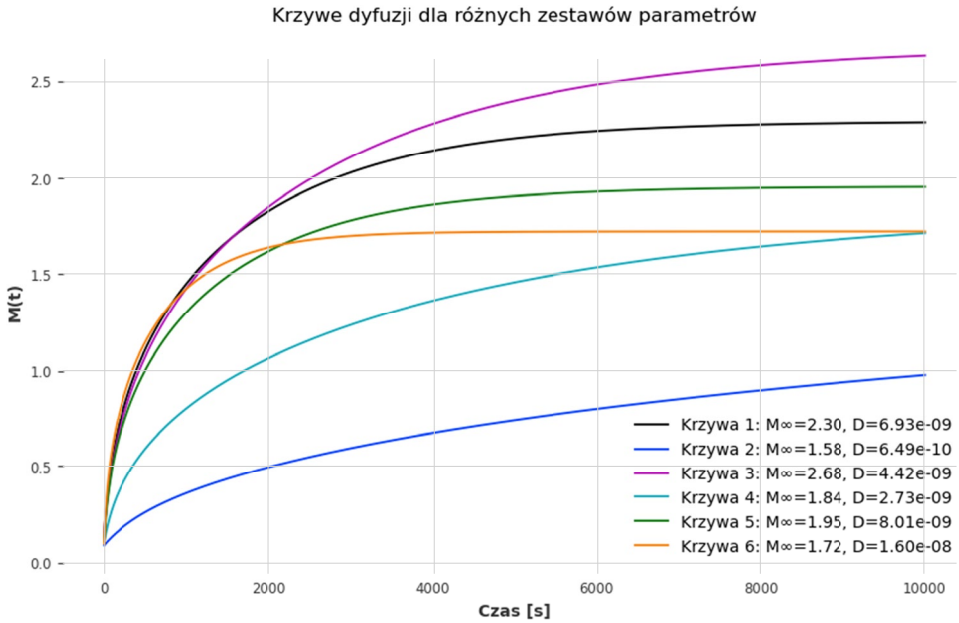
W celi ewaluacji modeli posłużono się trzema metrykami błędów (Hyndma i Koehler, 2006):

a. MAE (ang. *Mean Absolute Error*) – Średni błąd bezwzględny - mierzy średnią różnicę między rzeczywistymi a przewidywanymi wartościami, ignorując znak błędu. Wynik jest w tych samych jednostkach co dane.
b. RMSE (ang. *Root Mean Squared Error*) – Pierwiastek z błędu średniokwadratowego – mierzy średnie odchylenie przewidywań od rzeczywistych wartości, ale dodatkowo podnosi błędy do kwadratu przed ich uśrednieniem. To oznacza, że większe błędy mają większy wpływ na końcowy wynik.
c. MAPE (ang. *Mean Absolute Percentage Error*) – Średni procentowy błąd bezwzględny – mierzy średni procentowy błąd przewidywań względem rzeczywistych wartości. Daje wynik w procentach, co pozwala na łatwiejsze porównanie różnych modeli i zbiorów danych.

3. Wyniki

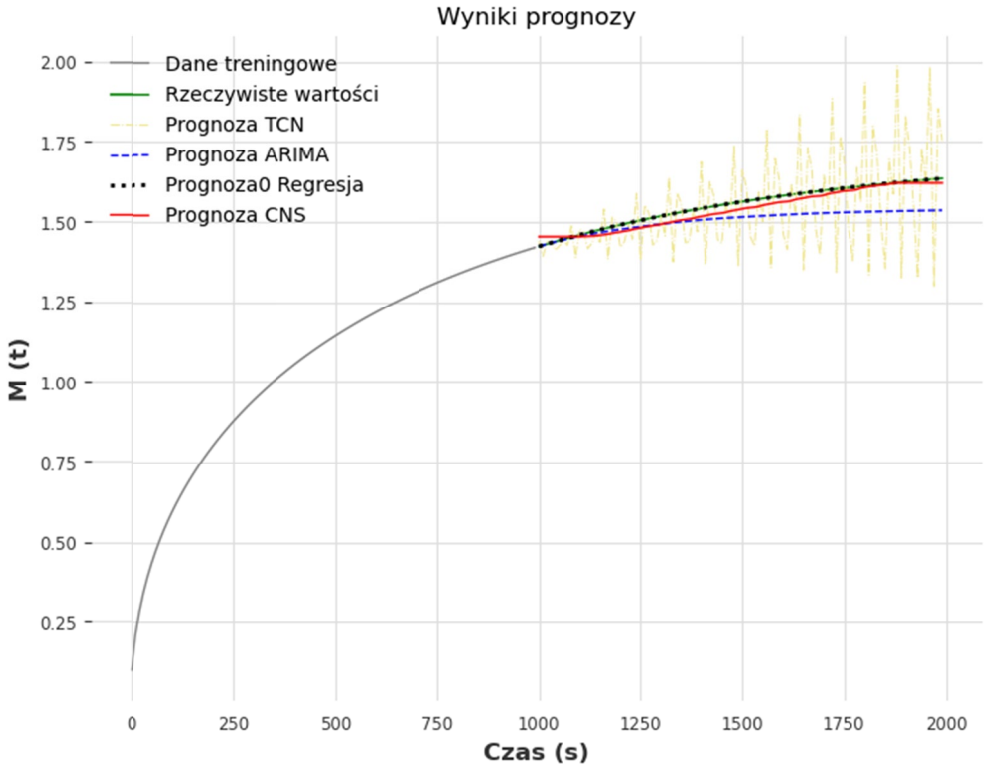
Rysunek 1 przedstawia losowo wybrane teoretyczne przebiegi procesu dyfuzji, wygenerowane na podstawie zbioru symulacji numerycznych (ZNIPRU). Krzywe ilustrują wpływ parametrów $M(\infty)$ i D na dynamikę transportu masy, gdzie wyższe wartości D prowadzą do szybszego osiągania stanów równowagi,

a większe $M(\infty)$ oznaczają wyższy poziom docelowy transportowanej substancji. W kontekście predykcji momentu wypłaszczenia kluczowe są przedziały czasowe w okolicach 2000 sekund, szczególnie dla krzywych 1, 3, 5 i 6, gdzie tempo zmian zaczyna znacząco maleć.



Rys. 1. Krzywe dyfuzji dla różnych wartości parametrów M(∞) oraz D losowo wybrane ze zbioru teoretycznych krzywych

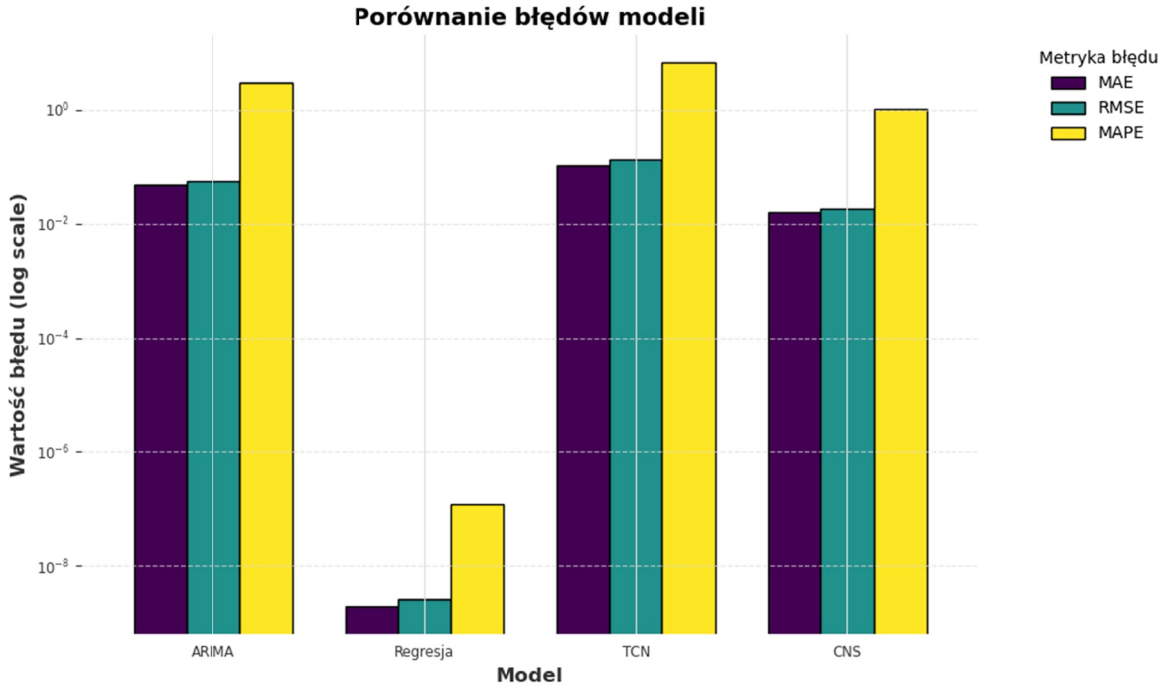
Rysunek 2 przedstawia wyniki predykcji dla procesu dyfuzji przy użyciu różnych modeli predykcyjnych. Oś pozioma oznacza czas (s), a oś pionowa wartość M(t). Szara linia – reprezentuje dane treningowe, czyli rzeczywiste wartości użyte do nauki modeli, ale nie brane pod uwagę w przypadku oceny jakości predykcji. Zielona linia reprezentuje rzeczywiste wartości w obszarze testowym, służące do oceny dokładności



Rys. 2. Porównanie modeli predykcyjnych w prognozowaniu procesu dyfuzji

prognoz i wyliczenia metryk błędów. Żółta przerywana linia – prognoza modelu TCN, charakteryzująca się znacznie większą wariancją niż pozostałe modele oraz czerwona linia dla modelu CNS – o znacznie lepszym dopasowaniu. Niebieska przerywana linia to prognoza modelu ARIMA natomiast czarna kropkowana linia reprezentuje prognozę modelu regresji, która jest bardzo zbliżona do wartości rzeczywistych. Badane modele znacząco różnią się jakością dopasowania.

Rysunek 3 przedstawia porównanie błędów modeli ARIMA, Regresja, TCN i CNS w prognozowaniu procesu dyfuzji, przy czym wartości błędów zaprezentowano w skali logarytmicznej. Analiza wyników wskazuje, że model regresyjny charakteryzuje się najniższymi wartościami błędu, co sugeruje jego najwyższą skuteczność w odwzorowaniu rzeczywistych wartości. Model ARIMA wykazuje wyraźnie wyższe wartości błędu, szczególnie w zakresie błędu procentowego, co może świadczyć o ograniczonej zdolności do uchwycenia dynamiki procesu. Model TCN odznacza się najwyższymi wartościami błędów, co wskazuje na jego ograniczoną przydatność w precyzyjnym prognozowaniu. Model CNS osiąga wyniki pośrednie, jednak jego skuteczność jest niższa niż w przypadku regresji. Wyniki analizy jednoznacznie wskazują, że modele regresyjne lepiej radzą sobie z predykcją badanych danych w porównaniu do metod opartych na szeregach czasowych.



Rys. 3. Wartości błędów predykcji

4. Wnioski

Przeprowadzona analiza modeli prognostycznych dla procesu dyfuzji wskazuje na istotne różnice w ich skuteczności. Modele regresyjne wykazały najwyższą precyzję prognozowania, osiągając najniższe wartości błędów, co sugeruje ich większą zdolność do uchwycenia charakterystyki analizowanego zjawiska. Z kolei modele szeregów czasowych, takie jak ARIMA i TCN, charakteryzowały się wyższymi błędami, co może wynikać z trudności w odwzorowaniu dynamiki procesu. Szczególną uwagę zwraca fakt, że dla określonych zakresów czasowych dokładność prognozy była kluczowa dla określenia momentu stabilizacji procesu, co ma istotne znaczenie praktyczne, pozwalające na zmniejszenie czasu analiz laboratoryjnych. Wnioski te wskazują na przewagę metod regresyjnych w badanym kontekście, jednak dalsze badania są niezbędne w celu optymalizacji działania modeli. Zaleca się przeprowadzenie bardziej szczegółowej analizy przestrzeni hiperparametrów oraz eksplorację innych metod predykcyjnych, które mogą lepiej uchwycić specyfikę procesu. Dodatkowo, istotnym krokiem w dalszych analizach powinno być testowanie modeli na rzeczywistych danych, co pozwoli zweryfikować ich skuteczność w warunkach eksperymentalnych i zwiększyć ich potencjalną użyteczność aplikacyjną.

Praca została wykonana w ramach prac statutowych realizowanych w IMG PAN w roku 2024, finansowanych przez Ministerstwo Nauki i Szkolnictwa Wyższego.

Literatura

[1] Alpaydin, E. (2014). Introduction to machine learning (3rd ed.). MIT Press.

[2] Bishop, C.M. (2006). Pattern recognition and machine learning. Springer.

[3] Box, G.E.P., Jenkins, G.M. (1976). Time series analysis: Forecasting and control (2nd ed.). Holden-Day.

[4] Godyń K., Kožušníková A. 2019: Microhardness of Coal from Near-Fault Zones in Coal Seams Threatened with Gas-Geodynamic Phenomena, Upper Silesian Coal Basin, Poland. Energies, **12** (9), 1756.

[5] Goodfellow, I., Bengio, Y., Courville, A. (2016). Deep learning. MIT Press.

[6] Herzen, J., Lässig, F., Piazzetta, S.G., Neuer, T., Tafti, L., Raille, G., Van Pottelbergh, T., Pasieka, M., Skrodzki, A., Huguenin, N., Dumonal, M., Kościsz, J., Bader, D., Gusset, F., Benheddi, M., Williamson, C., Kosinski, M., Petrik, M., & Grosch, G. (2022). Darts: User-friendly modern machine learning for time series (arXiv:2110.03224v3 [cs.LG]).

[7] Hyndman, R.J., & Koehler, A.B. (2006). Another look at measures of forecast accuracy. International Journal of Forecasting, **22** (4), 679-688.

[8] Lama R.D., Bodziony J. 1998: Management of outburst in underground coal mines, International Journal of Coal Geology 35, pp. 83-115.

[9] Levine J.R., 1992: Influences of coal composition on coal seam reservoir quality. A review. Symp. Coalbed Methane Res. Dev., Townsville.

[10] Saghafi A., Faiz M., Roberts D., 2007: CO2 storage and gas diffusivity properties of coals from Sydney Basin, Australia. International Journal of Coal Geology **70**, 240-254.

[11] Sarker, I.H. Machine Learning: Algorithms, Real-World Applications and Research Directions. SN COMPUT. SCI. **2**, 160 (2021)

[12] van Engelen, J.E., Hoos, H.H. A survey on semi-supervised learning. Mach. Learn **109**, 373-440 (2020). DOI: https://doi.org/10.1007/s10994-019-05855-6

[13] Zaręba, M., Danek, T., & Stefaniuk, M. (2023). Unsupervised Machine Learning Techniques for Improving Reservoir Interpretation Using Walkaway VSP and Sonic Log Data. Energies, **16** (1), 493.

[14] Zhang L., Ren T.-X., Aziz N., 2013: A stydy of laboratory testing and calculation methods for coal sorption isotherms. Journal of Coal Science & Engineering (China) **19**, No. 2, 193-202.

Machine learning for time series prediction of the diffusion curve

Abstract

In this paper, machine learning methods were used to predict the time series of the diffusion curve. The diffusion curves were generated in Python using the NumPy library. The data was formatted as a time series, which enabled efficient training and testing of the prediction models. A broad set of methods including statistical models, neural networks and regression approaches were used to model the dynamics of the diffusion process. The analysis of the models shows significant differences in their performance. Regression models showed the highest forecasting precision, achieving the lowest error values, suggesting their greater ability to capture the characteristics of the analyzed phenomenon. In contrast, time series models, such as ARIMA and TCN, had higher errors, which may be due to the difficulty of mapping the dynamics of the process.

Keywords: effective diffusion coefficient (*De*); machine learning; time series; regression model