Ocena algorytmów predykcji zużycia energii

Stanisław Czekalski

Politechnika Poznańska, Wydział Informatyki, Instytut Informatyki

Projekt wykonano w ramach przedmiotu Hurtownie Danych i Przetwarzanie Analityczne.

Celem projektu było zbadanie algorytmów predykcji energii elektrycznej. Dokonano analizy istniejących rozwiązań i na jej podstawie wybrano dwa algorytmy które posłużyły do budowy modeli predykcyjnych, które następnie przetestowano i oceniono według przyjętych kryteriów jakości i wydajności predykcji.

Spis treści:

1. Dane
2. Wybór algorytmów
3. Implementacja
4. Testy pierwszego algorytmu
5. Testy drugiego algorytmu
6. Analiza wyników

1. Dane

Dane dostarczone zostały przez firmę Kogeneracja Zachód. Zawarto w nich informacje o zużyciu energii elektrycznej oraz czynnikach zewnętrznych, które na nią wpłynęły. Otrzymano 112 plików w formacie csv, każdy zawierający pomiary z jednego miernika. Zmierzono 30648 wartości w godzinnych odstępach w przedziale czasu od 1 września 2015 do 28 lutego 2019, przy czym każda z nich oznacza energię zużytą w tym odstępie wyrażoną w kWh.
Czynniki zewnętrzne opisano 5 atrybutami liczbowymi:

- temperaturą zewnętrzną [°C]
- szybkością wiatru [m/s]
- wilgotnością [%]
- zachmurzeniem [oktan]
- długością dnia [h],

oraz dwoma kategorycznymi:

- typem dnia, przyjmującym 4 wartości {(poniedziałek, wtorek, środa, czwartek): 1; piątek: 2; sobota: 3; niedziela: 4}
- porą roku {wiosna: 1, lato: 2, jesień: 3, zima: 4}

przy czym wszystkie pięć atrybutów liczbowych opisujących pogodę podano tylko dla okresu od 1 stycznia 2016 do 31 grudnia 2017.

Pomiary zużycia energii dokonano przy użyciu dwóch typów czujników (ciepłomierze CO i CW). Połączono je w jeden sumaryczny przebieg czasowy. W ramach jednego okna czasowego zsumowano zużycie energii zmierzone każdym z czujników. Informacje o pogodzie nie wymagały żadnej transformacji, były takie same w każdym pliku. Uzyskano w ten sposób jeden plik ze zbiorem danych.

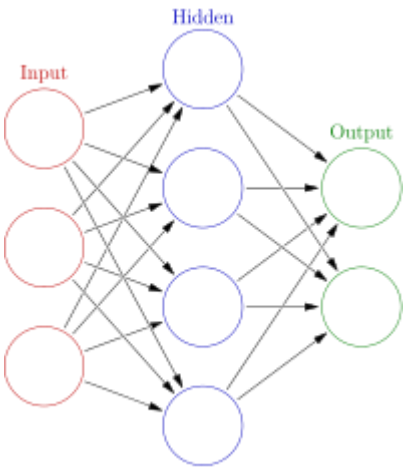
2. Wybór algorytmów

Jako pierwszy model predykcyjny wybrano model SARIMA (ang. Seasonal Autoregressive Moving Average), będący rozszerzeniem klasycznego modelu ARIMA, uwzględniającym składową sezonowości. Predykcja wartości w danej chwili czasu odbywa się na podstawie korelacji z wartościami w chwilach poprzednich. Do budowy i testowania modelu wykorzystano dane o zużyciu energii podczas całego mierzonego okresu, ich znaczna ilość sugerowała możliwość uzyskania dokładnej predykcji szeregu czasowego.

Model składa się z trzech członów:

- S(P, D, Q)m – składowa sezonowości, gdzie P to rząd opóźnień sezonowych typu AR, D różnicowanie składowej sezonowej, Q – rząd opóźnień sezonowych typu MA, m – liczba okresów w sezonie
- AR(p) – rząd (liczba opóźnień) modelu autokorelacyjnego
- I(d) – stopień różnicowania (ang. differencing), czyli liczba ile razy od danej została odjęta poprzednia próbka, wykonywane w celu eliminacj niestacjonarności szeregu czasowego

Drugim wybranym modelem jest sieć neuronowa. Zdjęcie (źródło: Wikipedia) przedstawia architekturę trójwarstwowej sieci neuronowej, z jedną warstwą ukrytą. Sieci neuronowe są zaawansowanym modelem uczenia maszynowego, potrafiącym skutecznie aproksymować złożone funkcje. Z tego powodu często stosuje się je w praktyce, gdy wymagana jest dobra



jakość predykcji.

3. Implementacja

Projekt wykonano przy użyciu języka programowania Python, wykorzystując biblioteki Scikit-lear, Pandas, Numpy, Matplotlib oraz Glob. Oferują one implementację wielu popularnych algorytmów uczenia maszynowego oraz umożliwiają pracę z danymi: ich reprezentację w strukturach, transformację oraz wizualizację.

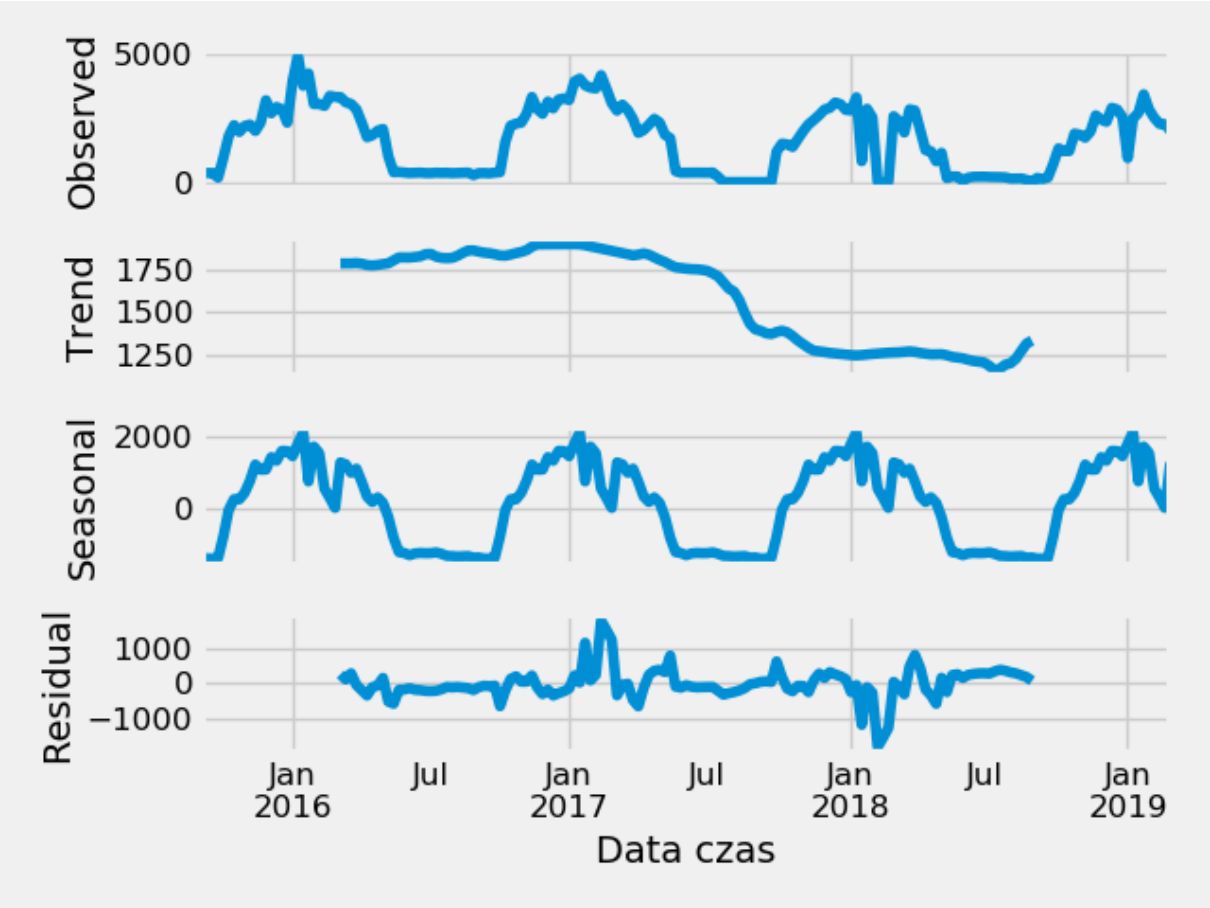
4. Testy pierwszego algorytmu

W pierwszej kolejności zaimplementowano model SARIMA, ponieważ parametry opisujące opóźnienia mogą również zostać wykorzystane przy budowie sieci neuronowej.
Model SARIMA do predykcji wykorzystuje tylko próbki z poprzednich chwil czasu.W pierwszym kroku należało zmniejszyć granularność danych. Zsumowano zużycie energii w ramach okna czasowego. W pierwszej kolejności przebadano agregację pomiarów w ciągu jednego dnia.
Następnie konieczne było czy badany szereg czasowy jest stacjonarny. W tym celu użyto testu Augemnted Dickey-Fuller. Dla języka Python jego implementacja znajduje się w pakiecie statsmodels.tsa.stattools.

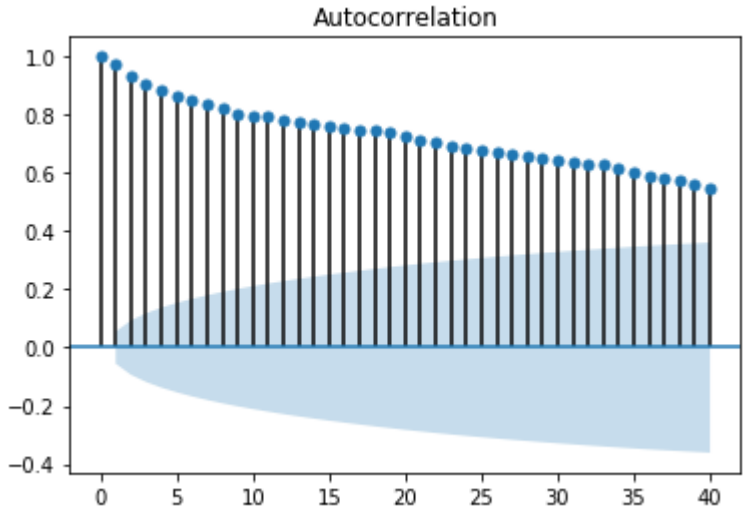
```
ADF_Stationarity_Test(df.Energia, printResults = True)
```

```
Augmented Dickey-Fuller Test Results:
ADF Test Statistic         -5.287666
P-Value                     0.000006
# Lags Used                50.000000
# Observations Used     30597.000000
Critical Value (1%)        -3.430564
Critical Value (5%)        -2.861634
Critical Value (10%)       -2.566820
dtype: float64
Is times series stationary? True
```

Uzyskano wynik pozytywny, badany szereg jest stacjonarny przy badaniu opóźnienia 50 dniowego, nie trzeba więc go różnicować. Poniższy wykres przedstawia dekompozycję szeregu czasowego na składowe. Z wykresu wynika że konieczne będzie różnicowanie składowej sezonowej.

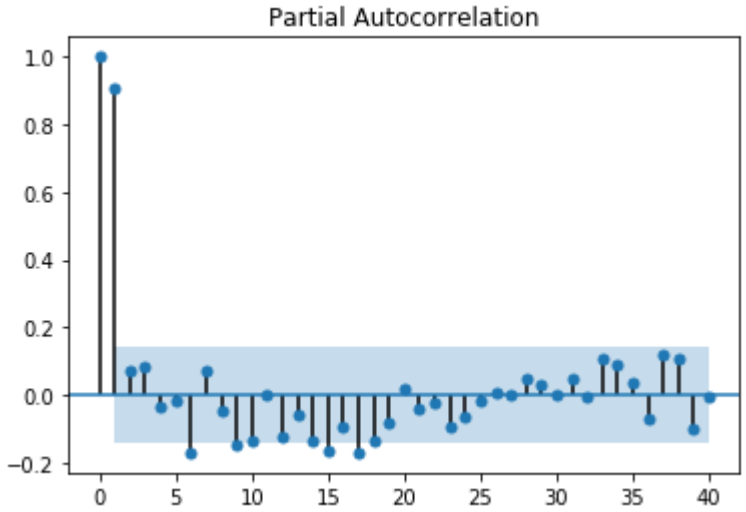


W celu doboru parametrów modelu SARIMA posłużono się wykresami autokorelacji acf i częściowej autokorelacji pacf.



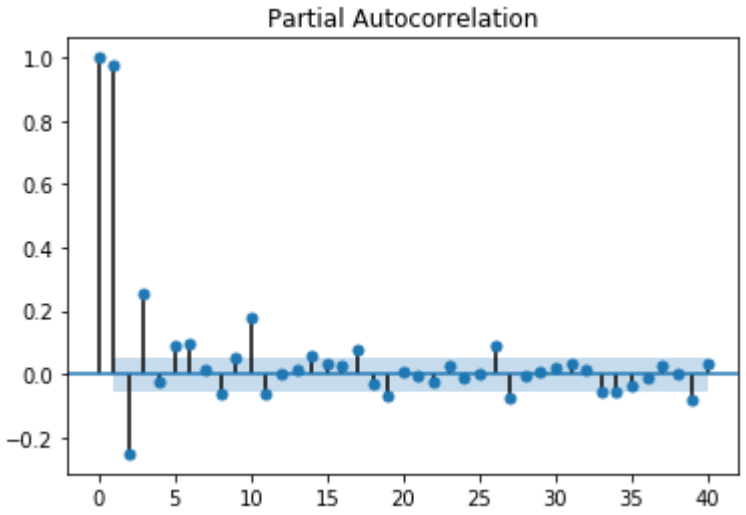
Wykres autokorelacji wskazuje na istotną zależność wartości badanej próbki od próbek poprzednich, z wykresu częściowej autokorelacji odczytujemy jednak, że wynika to z propagacji korelacji, i znaczący wpływ mają trzy przeszłe próbki. Obserwujemy dodatkowo możliwy wpływ próbki sprzed 10 dni, jej wykorzystanie może jednak zbyt skomplikować model. Ustalono więc następujące parametry modelu:

- SARIMA(p=3, d=0, q=0)(P=1, D=1, Q=0)m=365

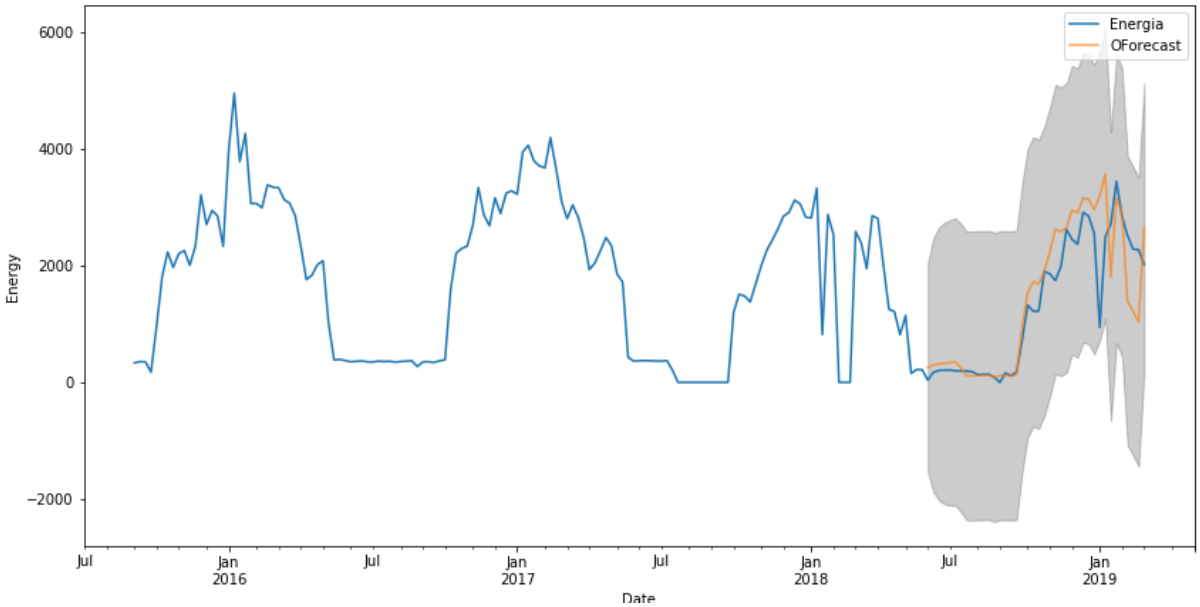


Jako zbiór treningowy wykorzystano próbki do dnia 3 czerwca 2018, dokonano dopasowania modelu do danych.

Okazało się to jednak niemożliwe, granuralność danych ustalona na jeden dzień okazało się zbyt złożona obliczeniowo dla modelu SARIMA. W tej sytuacji postanowiono wykorzystać pomiar tygodniowy. Wykres acf dla tego przypadku wygląda bardzo podobnie, natomiast wykres pacf wskazuje na użycie jednego opóźnienia modelu AR.

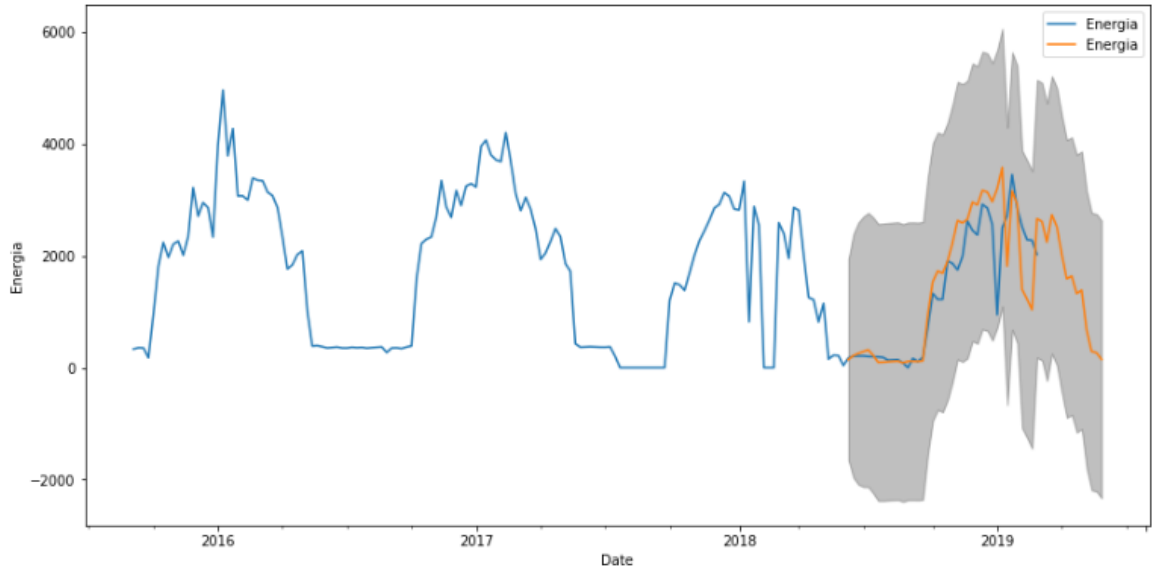


W fazie testowania uzyskano następujący wynik:



Ogólny trend został prawidłowo uchwycony, dokładność predykcji jednak mogłaby być większa.

Model SARIMA daje możliwość prognozowania przyszłych wartości energii, uzyskano następujący wykres. Prognoza jest prawdopodobna.



5. Testy drugiego algorytmu

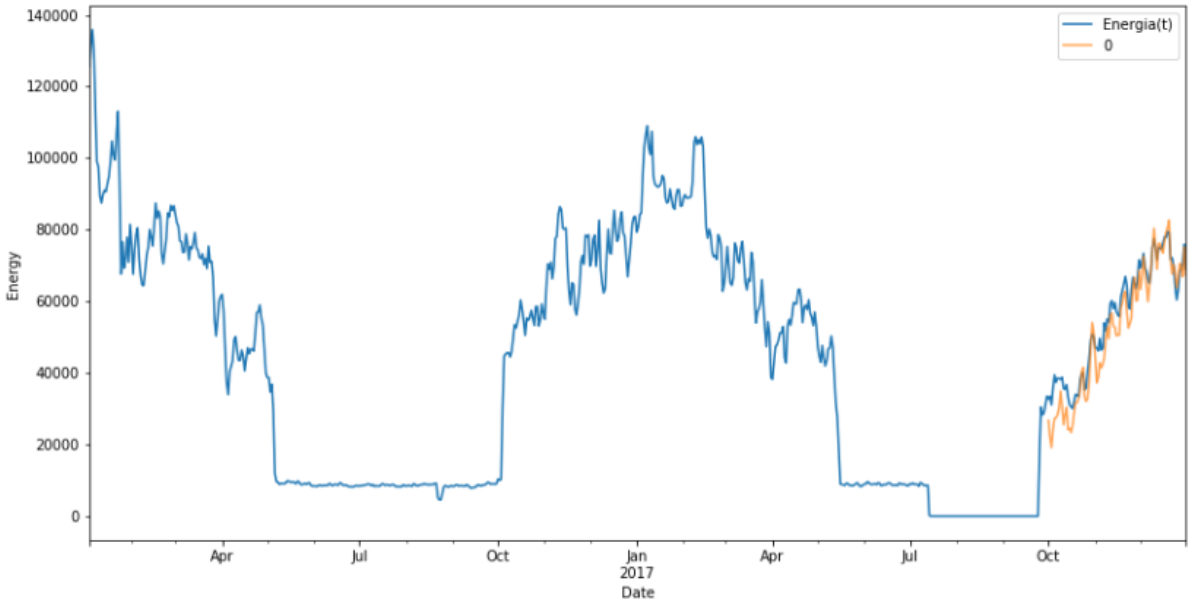
Drugim zbudowanym modelem była sieć neuronowa. Pierwszym koniecznym krokiem w jej budowie było przekształcenie szeregu czasowego do postaci problemu nadzorowanego uczenia maszynowego. W tym celu napisano funkcję dokującą takiego przekształcenia. Do danego rekordu dokładane są informacje o wartościach zużywanej energii i warunkach zewnętrznych z poprzednich chwil czasu.

Wektor wejściowy wygląda w następujący sposób. Wagi poszczególnych składowych były takie same,

$$\begin{pmatrix} \mathbf{Energia(t-3)} \\ \mathbf{Temp\_zewn(t-3)} \\ \mathbf{V\_wiatru(t-3)} \\ \mathbf{Wilg(t-3)} \\ \mathbf{Zachm(t-3)} \\ \vdots \\ \mathbf{Temp\_zewn(t)} \\ \mathbf{V\_wiatru(t)} \\ \mathbf{Wilg(t)} \\ \mathbf{Zachm(t)} \\ \mathbf{Dlug\_dnia} \\ \mathbf{Typ\_dnia} \\ \mathbf{Pora\_roku} \end{pmatrix}$$

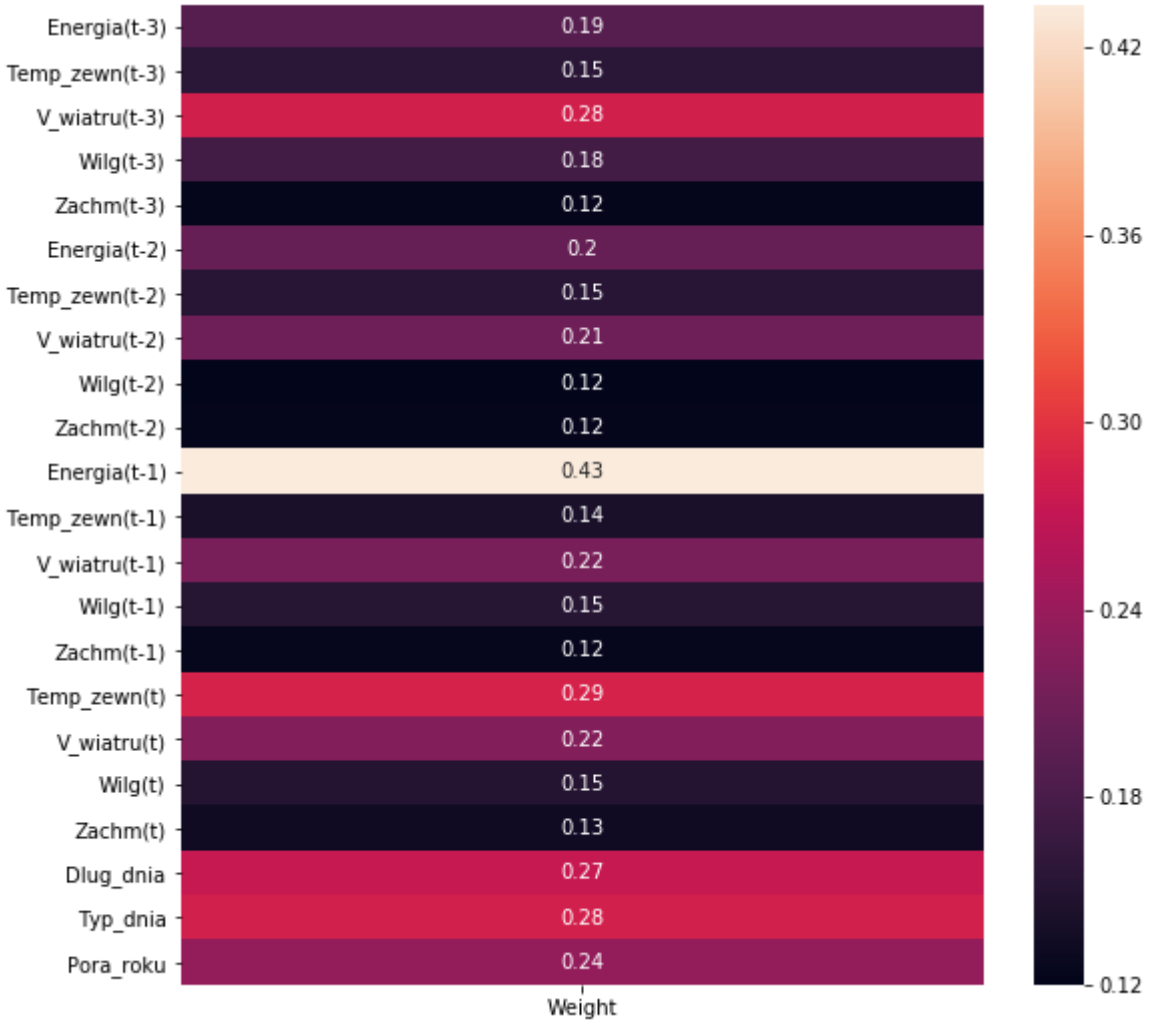
W przypadku sieci neuronowej możliwe okazało się wykorzystanie jednodniowej granularności pomiarów. Zdecydowano wykorzystać rząd opóźnień równy 3. W sieci neuronowej wykorzystano 3 warstwy ukryte, każda składająca się z 33 neuronów (1.5 raza liczba atrybutów wejściowych).

Wyniki uzyskane tą metodę nawet bez optymalizacji są już lepsze. Przeprowadzona optymalizacja dodatkowo jeszcze je poprawiła. Zastosowanie regularyzacji L2 ze współczynnikiem alfa równym 10E5 zapobiegło przeuczeniu modelu, a przeszukiwanie przestrzeni hiperparametrów learning_rate_init oraz momentum poprawiło dopasowanie modelu do danych. W rezultacie otrzymano następujący wynik.



W ostatnim kroku zbadano istotność każdej cechy wejściowej. W tym celu zsumowano wartości bezwzględne wag na połączeniach idących od danego neurona wejściowego, do neuronów warstwy ukrytej. Uzyskano wektor, którego elementy reprezentują istotność danej cechy. Zilustrowano to na wykresie typu heatmap. Wskazuje on, że największą rolę odgrywa

wartość energii z poprzedniego okna czasowego. Istotne okazały się również temperatura zewnętrzna w danym dniu, jego typ, długość. Dodatkowo istotna była prędkość wiatru sprzed trzech dni, mówiąca może o zachodzących zmianach pogody.



6. Analiza wyników

Zbudowane modele predykcje znacząco różniły się jakością predykcji i wydajnością. Sieć neuronowa okazała się dokładniejsza, dodatkowo proces uczenia trwał szybciej. Dużą poprawę wydajności uzyskano dzięki wykorzystaniu regularyzacji zapobiegającej przeuczeniu modelu. Klasyczne podejście predykcji szeregu czasowego przy użyciu modelu SARIMA okazało się w przypadku badanego problemy niewystarczające. Stosunkowo nowa metoda wykorzystująca uczenie głębokiej sieci neuronowej sprawdziła się znacznie lepsza. Analiza autokorelacji okazała się bardzo pomocna przy wyborze cech. Zastosowane narzędzia umożliwiły szczegółową analizę danych oraz dobre strojenie modelu. Porównanie dokładności predykcji obu modeli przedstawia się następująco:

	SARIMA	Sieć Neuronowa
RMSE	91064.53	5296.42
R2	0.78	0.93