

# AIS RW.E2

## Protokoły i algorytmy zarządzania energią oraz monitorowania (AIS.PA)

2020.06.26

© 2019 Instytut Nauki i Techniki Stipendium. All rights reserved.

### Spis treści

1. Wstęp .....	3
2. Algorytmy .....	4
2.1. Prognozowanie zużycia energii .....	4
2.1.1. Szeregi czasowe zapotrzebowania na energię elektryczną .....	4
2.1.2. Teoretyczne podstawy budowy modeli prognostycznych .....	5
2.1.3. Sztuczne sieci neuronowe .....	7
2.1.4. Drzewa regresyjne .....	8
2.1.5. Regresyjne lasy losowe .....	10
2.1.6. Regresja wektorów wspierających .....	11
2.1.7. Regresja k-najbliższych sąsiadów .....	13
2.2. Miary oceny jakości prognozy .....	13
2.3. Budowa wektora charakterystyk dla prognoz .....	15
2.4. Opracowanie podejścia prognostycznego opartego na algorytmach uczenia maszynowego .....	18
2.5. Uwagi implementacyjne dotyczące opracowanego podejścia .....	21
2.6. Wykorzystywane metody porównawcze .....	21
2.7. Pozostałe rozpatrywane metody prognostyczne .....	23
2.8. Wyniki eksperymentów .....	23

**Fundusze Europejskie**  
Program Regionalny**Unia Europejska**  
Europejski Fundusz  
Rozwoju Regionalnego

2.9.	Optymalizacja autonomicznego układu zasilania .....	26
2.9.1.	Układ zasilania .....	26
2.9.2.	Układ zasilania generatorami fotowoltaicznymi .....	27
2.9.3.	Układ zasilania turbinami wiatrowymi .....	39
2.9.4.	Analiza optymalizacji systemu .....	47
2.10.	Zaprojektowane wagi dla funkcjonalności przejścia dla pieszych. ....	48
3.	Praktyczne użycie protokołów i algorytmów .....	49
3.1.	Przykład poszukiwania danych do budowy modelu. ....	50
3.2.	Budowa modeli uczenia maszynowego .....	50
3.3.	Doskonalenie modeli .....	52
4.	Protokoły .....	56
4.1.	Wybrane rozwiązanie .....	58
4.2.	Przegląd warstw protokołu .....	59
4.3.	Transmisja danych .....	59
4.3.1.	Urządzenia końcowe .....	61
4.3.2.	Bramy .....	64
4.3.3.	Serwer sieciowy .....	64
4.3.4.	Serwer aplikacji .....	65
4.4.	Bezpieczeństwo .....	65

## 1. Wstęp

Niniejszy dokument stanowi protokoły i algorytmy zarządzania energią oraz monitorowania. Istotą zagadnienia jest opracowanie prototypowych algorytmów i protokołów, które powinny pozwolić uzyskać wydajne obliczeniowo i energetycznie metody adaptacji wszystkich parametrów autonomicznego sterownika, ze szczególnym uwzględnieniem dostosowywania tych parametrów przez inteligentne algorytmy. Z uwagi na wymaganą minimalizację zużycia energii przez elementy systemu AIS istotnymi aspektami jest minimalizacja złożoności obliczeniowej oraz minimalizacja ilości przesyłanych danych. W przypadku zastosowań sterownika AIS.AS na przejściach dla pieszych istotą zagadnienia technologicznego w zakresie opracowania prototypu protokołów monitorowania i zarządzania jest implementacja wydajnych i bezpiecznych algorytmów na potrzeby sterowania działaniem infrastruktury przejścia dla pieszych oraz skutecznego i wiarygodnego monitorowania efektywności oświetlenia oraz zbierania danych środowiskowych.

## 2. Algorytmy

Zadaniem inteligentnych algorytmów monitorowania jest prognozowanie i optymalizacja sposobu wykorzystania dostępnego zapasu energii tak, aby funkcje systemu działały nieprzerwanie. Na przykład w sytuacji niskiego poziomu naładowania akumulatorów i jednocześnie niekorzystnych prognoz pogody, najbardziej energochłonne funkcje systemu, czyli komunikacja z centrum przetwarzania danych, mogą być uruchamiane rzadziej lub w ograniczonym zakresie (przesłane zostaną dane z monitorowania z dokładnością godzinową zamiast standardowej minutowej). Zagadnieniem omawianym w tym rozdziale jest badanie algorytmów służących do prognozowania zużycia energii, a także optymalizacji wykorzystania budżetu energetycznego przez poszczególne elementy osprzętu. Analiza opiera się na algorytmach bilansujących i optymalizujących zużycie energii pod kątem operacyjności komponentów końcowych oraz nieprzerwanego działania całego systemu. Algorytmy nie mogą stanowić większego obciążenia energetycznego niż oszczędności, które mogą przynieść. Dzięki tym algorytmom możliwe będzie rozważne zarządzanie systemem i jego komponentami końcowymi w tym sterowanie funkcjonalnościami tak by zapewnić operacyjność priorytetowych funkcji systemu w przypadku deficytów energii elektrycznej.

### 2.1. Prognozowanie zużycia energii

Głównym celem rozdziału jest identyfikacja efektywnych metod prognostycznych, połączona z opracowaniem skutecznego podejścia do procesu krótkoterminowego prognozowania zapotrzebowania na energię elektryczną, na poziomie pojedynczego punktu usytuowania autonomicznego sterownika. Zapotrzebowanie na energię elektryczną na tym poziomie agregacji danych, determinowane jest m.in. przez warunki środowiskowe a także zużycie energii, związane z użytkowaniem określonego punktu przejścia dla pieszych. Ze względu na dużą zmienność badanego zjawiska, zaproponowane metody opierają się o algorytmy uczenia maszynowego, które są w stanie wychwycić zależności pomiędzy występującymi zjawiskami, a zapotrzebowaniem całkowitym w danej chwili. Skuteczność opracowanych metod prognostycznych, sprawdzono dla przypadku zapotrzebowania:

- godzinowego z horyzontem prognostycznym wynoszącym 24 godziny,
- godzinowego z horyzontem prognostycznym wynoszącym jedną godzinę,

W każdym rozpatrywanym przypadku, przeanalizowano ponadto jakość wypracowanych metod prognostycznych.

Dla przejrzystości pracy, przed przystąpieniem do omawiania procesu opracowywania modeli prognostycznych oraz prezentacji otrzymanych wyników, przedstawione zostały aspekty teoretyczne, związane z wykorzystanymi algorytmami uczenia maszynowego.

#### 2.1.1. Szeregi czasowe zapotrzebowania na energię elektryczną

Szeregiem czasowym nazywa się zbiór wartości cechy w uporządkowanych chronologicznie różnych przedziałach czasu. Oznaczając przez  $g$  ( $g = 1, 2, \dots, n$ ) przedziały czasu, w których obserwowano wartości pewnej zmiennej,

a przez  $z_g$  wyniki obserwacji, szereg czasowy zapisać można jako zbiór  $\{z_1, \dots, z_n\}$ . Analiza statystyczna szeregów czasowych ma na celu wykrycie prawidłowości, jakim podlega badany proces i jest podstawą prognozowania przebiegu tego zjawiska w przyszłości. Wykorzystuje się w niej metody rachunku prawdopodobieństwa i statystyki matematycznej przyjmując, że zaobserwowany szereg czasowy zapotrzebowania na energię elektryczną jest jedną z wielu możliwych realizacji pewnego dyskretnego procesu stochastycznego  $\{Z_g\}$ , określonego dla wszystkich  $g$  całkowitych i dodatnich. Jeżeli zatem ustalimy  $g$ , to  $Z_g$  jest zmienną losową, której realizacją jest element  $z_g$ , zaobserwowanego szeregu czasowego. O procesie tym zwykle zakłada się, że jest stacjonarny (co najmniej w szerszym sensie) oraz ergodyczny. Pierwsze z założeń gwarantuje, że mechanizm generujący proces stochastyczny jest niezmienny w czasie, tak, iż ani postać przyjętego modelu, ani jego parametry nie podlegają zmianom w czasie. Drugie założenie zapewnia, że wartości procesu stochastycznego stosunkowo odległe w czasie są nieskorelowane lub bardzo słabo skorelowane. Do celów prognostycznych wygodnie jest przyjąć, że proces stochastyczny jest wypadkową działania pewnego procesu deterministycznego, związanego z działaniem tzw. przyczyn głównych oraz innego, mającego pewne szczególne własności, procesu stochastycznego, związanego z działaniem tzw. przyczyn przypadkowych.

W związku z tym, w analizowanych szeregach czasowych wyróżnia się dwie składowe:

- składową systematyczną związaną z procesem deterministycznym oraz
- składową przypadkową (zwaną też składnikiem losowym lub wahaniami przypadkowymi), związaną z procesem stochastycznym o szczególnych własnościach.

Składowa systematyczna może wystąpić w postaci trendu, stałego (średniego) poziomu zmiennej prognozowanej, składowej okresowej (periodycznej), która z kolei może wystąpić w postaci wahań cyklicznych lub wahań sezonowych. Trend jest długookresową skłonnością do jednokierunkowych zmian (wzrostu lub spadku) wartości badanej zmiennej. Jest rozpatrywany jako konsekwencja działania stałego zestawu czynników na prognozowane zjawisko. Stały (średni) poziom oznacza brak trendu i oscylowanie wartości badanej zmiennej wokół pewnego stałego poziomu. Wahania cykliczne są długookresowymi rytmicznymi wahaniami wartości badanej zmiennej, wokół trendu lub stałego (średniego) poziomu zmiennej. Natomiast wahania sezonowe są zmianami wartości prognozowanej zmiennej wokół trendu lub stałego średniego poziomu zmiennej, powtarzającymi się mniej więcej w tych samych rozmiarach co pewien okres (w przybliżeniu stały), nieprzekraczający jednego roku.

Proces wyodrębniania poszczególnych składowych szeregu czasowego nazywa się dekompozycją szeregu czasowego. W praktyce dekompozycję szeregu czasowego na poszczególne składowe, przeprowadza się budując modele szeregu czasowego. W zależności od przyjętych założeń, co do wpływu poszczególnych składowych i ich wzajemnych relacji oraz sposobu określania parametrów, modele te mogą mieć różną postać.

### 2.1.2. Teoretyczne podstawy budowy modeli prognostycznych

Metody uczenia maszynowego można podzielić na metody nadzorowane i nienadzorowane. W metodach nienadzorowanych nie istnieje zdefiniowana zmienna objaśniana, którą można modelować w podanym zakresie.

Większość algorytmów to jednak metody nadzorowane, co oznacza, że dana jest zmienna objaśniana ze znanymi wartościami i wiele zmiennych objaśniających. Algorytm może wtedy nauczyć się, które wartości zmiennej objaśnianej są powiązane z wartościami zmiennych objaśniających.

Zadanie regresji nadzorowanej polega na aproksymacji funkcji odwzorowującej przykłady  $x \in X = \mathbb{R}^p, j = 1, \dots, p$ , (wartości zmiennych niezależnych) na zbiór liczb rzeczywistych. Zadanie polega zatem na określeniu pewnej funkcji  $f(x)$ :

$$f(x): X \rightarrow \mathbb{R}$$

Funkcję aproksymacyjną tworzy się na podstawie próby  $(x_g, z_g)$ , (gdzie  $g = 1, \dots, n$ ,  $x_g$  jest wektorem zmiennych objaśniających oraz  $z_g$  jest rzeczywistą wartością zmiennej objaśnianej tj. zapotrzebowania na energię elektryczną), zwanej potocznie próbą uczącą  $U$ . Utworzona na podstawie próby uczącej funkcja  $f(x_g)$ , pozwala każdemu zaobserwowanemu wektorowi  $x_g$  przypisać domniemaną wartość ze zbioru  $\mathbb{R}$ .

Dokładność nauczonego modelu, jego sprawność czy wydajność, mierzyć można dokładnością przewidywania dla nowych, nieużywanych w procesie uczenia danych, które zawarte są w tzw. próbie walidacyjnej  $W$ . Próba walidacyjna jest fragmentem zbioru danych, w którym wartości zmiennej objaśnianej są chwilowo ukryte dla tymczasowego modelu, który wykonuje predykcję obiektów zgodnie ze wzorcami i strukturą poznaną na próbie uczącej  $U$ . Efektywność aproksymacji sprawdzana jest poprzez porównanie wartości teoretycznych (wyznaczonych na podstawie modelu) z rzeczywistymi wartościami zmiennej objaśnianej. Tymczasowy model następnie jest tak dopasowywany, aby zminimalizować wartość współczynnika błędów na próbie walidacyjnej.

Z podejściem takim, związany jest problem polegający na tym, że zbiór walidacyjny odgrywa kluczową rolę przy wyborze ostatecznego modelu, co oznacza, że w rzeczywistości uczestniczy on w procesie uczenia. Aby zwiększyć poziom zaufania do ostatecznego modelu, zwykle (jeśli tylko wielkość zbioru danych na to pozwala), ze zbioru danych wydziela się dodatkowo trzeci zbiór przypadków, tak zwany zbiór testowy  $T$ . Ostateczna postać modelu (nauczonego za pomocą zbioru uczącego i sprawdzonego za pomocą zbioru walidacyjnego) jest dodatkowo testowana za pomocą zbioru testowego. Takie dodatkowe sprawdzenie praktykuje się po to, aby upewnić się, że rezultaty uzyskane dla zbioru uczącego i zbioru walidacyjnego są zgodne z rzeczywistością, a nie są tylko mechanicznym wytworem procedury uczenia.

Przedstawione postępowanie określa się mianem generalizacji wiedzy i jest jednym z podstawowych zagadnień przy uczeniu modeli. Model nadmiernie dopasowany do danych, którymi był uczony, nie będzie miał dobrych wyników dla nowych danych. Przeuczenie modelu powstaje, gdy tymczasowy model na próbie walidacyjnej, stara się uwzględnić każdy możliwy trend lub strukturę z próby uczącej. W teorii, zawsze można tak wyuczyć dostatecznie skomplikowany model, by osiągnąć zerowy błąd dla danych uczących. Jednak rzecz polega na tym, by otrzymać model, który będzie dobrze przewidywał zmienną objaśnianą dla nowych przypadków, tam gdzie znane są jedynie wartości zmiennych objaśniających.

Istnieje ścisły związek między nadmiernym dopasowaniem do danych uczących i słabą generalizacją wiedzy zawartej w danych uczących. Dlatego ucząc każdy model trzeba mieć na uwadze takie elementy jak dokładność i generalizacja wiedzy.

### 2.1.3. Sztuczne sieci neuronowe

Sztuczne sieci neuronowe (ang. artificial neural networks) to matematyczne obiekty w postaci równań lub układów równań, zwykle nieliniowych, służące analizie oraz przetwarzaniu danych. Celem sieci neuronowych jest przekształcenie danych wejściowych w dane wyjściowe o określonej charakterystyce lub taka modyfikacja układów równań, by z ich struktury i parametrów odczytać użyteczną informację. Na gruncie statystycznym wybrane typy sieci neuronowych mogą być interpretowane w ogólnych kategoriach regresji nieliniowej.

W badaniach związanych z prognozowaniem w elektroenergetyce, najczęściej wykorzystuje się wielowarstwowe, jednokierunkowe sztuczne sieci neuronowe bez sprzężeń zwrotnych. Sieci neuronowe typu wielowarstwowego perceptronu (ang. Multilayer Perceptron – MLP) są jednym z najpopularniejszych typów nadzorowanych sieci neuronowych. Powszechność ich stosowania wynika m.in. z uniwersalnych zdolności aproksymujących.

Dla przykładu, sieć MLP (3, 4, 1) oznacza sieć neuronową o trzech wejściach, czterech neuronach w warstwie ukrytej oraz jednym neuronie w warstwie wyjściowej. Ogólnie trzywarstwowa sieć neuronowa MLP (P, M, K) opisana jest wyrażeniem:

$$f(x_g, w) = h_2(W_2[h_1(W_1x_g + b_1)] + b_2)$$

gdzie  $x_g = (x_1, \dots, x_p)^T$  reprezentuje dane wejściowe,  $W_1$  jest macierzą wag warstwy pierwszej o wymiarach  $M \times P$ ,  $W_2$  jest macierzą wag warstwy drugiej o wymiarach  $K \times M$ ,  $h_i(\mathbf{u})$  oraz  $b_i$  są odpowiednio nieliniowościami (funkcjami aktywacji neuronów) oraz wartościami stałymi w kolejnych warstwach.

Jako funkcję aktywacji  $h$ , która decyduje o właściwościach działania (uczenia się i rozpoznawania) neuronu, w pracy zastosowano funkcję logistyczną postaci:

$$h = \frac{1}{1 + e^{-x}}$$

W zagadnieniach praktycznych stosuje się ponadto funkcje liniową, bipolarną, unipolarną, tangensu hiperbolicznego, wykładniczą oraz softmax.

Uczenie sieci neuronowej z nauczycielem polega na poszukiwaniu takich parametrów sieci, aby minimalizować błąd między wartościami pożądanymi  $z_g$ , a otrzymywanymi na wyjściu sieci  $p_g = \{f(x_g, \mathbf{w}): g = 1, \dots, n\}$ . Najczęściej minimalizowaną funkcją błędu jest suma kwadratów różnic pomiędzy rzeczywistą wartością zmiennej objaśnianej a jej wartością teoretyczną wyznaczoną przez model, przy ustalonych wartościach wektora wag synaptycznych:

$$E(\mathbf{w}) = \frac{1}{2} \sum_{k=1}^K \|e^{(k)}\|^2 = \frac{1}{2} \sum_{k=1}^K \|(\sum_{g=1}^n (f(x_g^{(k)}, \mathbf{w}) - z_g^{(k)}))\|^2$$

gdzie  $k$  jest numerem próbki uczącej  $x_g^{(k)}, z_g^{(k)}$  jest odpowiednim dla tej próbki wektorem wzorców, a  $K$  liczbą epok uczenia sieci neuronowej.

Proces uczenia sieci neuronowej polega na iteracyjnej modyfikacji wartości wektora wag synaptycznych  $\mathbf{w}$  (wszystkie wagi ustawiane są w jednym wektorze), w iteracji  $k + 1$ :

$$w_{k+1} = w_k + \eta_k p_k$$

gdzie  $p_k$  oznacza kierunek minimalizacji funkcji  $E(\mathbf{w})$  oraz  $\eta$  jest wielkością błędu uczenia.

Do najpopularniejszych metod optymalizacyjnych należą bez wątpienia metody gradientowe, których podstawą działania jest znajomość gradientu funkcji. Opierają się one na rozwinięciu w szereg Taylora funkcji celu  $E(\mathbf{w})$ , w najbliższym sąsiedztwie znanego punktu  $\mathbf{w}$ :

$$p_k = -[H(w_k)]^{-1} g(w_k)$$

gdzie  $\mathbf{g}$  i  $\mathbf{H}$  oznaczają odpowiednio gradient oraz hesjan ostatniego znanego rozwiązania  $w_k$ .

Powyższy wzór stanowiący istotę newtonowskiego algorytmu optymalizacji, jest zależnością czysto teoretyczną, gdyż wymaga zapewnienia dodatniej określoności hesjanu w każdym kroku, co w przypadku ogólnym jest trudne do spełnienia. Z tego powodu, w praktycznych implementacjach algorytmu rezygnuje się z dokładnego wyznaczenia hesjanu  $\mathbf{H}(w_k)$ , a w zamian stosuje się jego przybliżenie  $\mathbf{G}(w_k)$ . Jedną z najpopularniejszych metod uczenia sieci neuronowych, jest algorytm zmiennej metryki. W metodzie tej, w każdym kroku modyfikuje się hesjan (lub jego odwrotność) z kroku poprzedniego o pewną poprawkę. Jeżeli przez  $c_k$  i  $r_k$  oznaczy się odpowiednio przyrosty wektora  $\mathbf{w}$  oraz gradientu  $\mathbf{g}$  w dwu kolejnych krokach iteracyjnych,  $c_k = w_k - w_{k-1}$ ,  $r_k = g_k - g_{k-1}$ , a przez  $V_k$  macierz odwrotną przybliżonego Hesjanu:

$$V_k = [G(w_k)]^{-1}, V_{k-1} = [G(w_{k-1})]^{-1}$$

to zgodnie z najefektywniejszą formułą Broydena – Fletchera – Goldfarba – Shanno (BFGS), proces uaktualnienia wartości macierzy  $V_k$  opisuje się zależnością rekurencyjną:

$$V_k = V_{k-1} + \left( 1 + \frac{r_k^T V_{k-1} r_k}{c_k^T r_k} \right) \frac{c_k c_k^T}{c_k^T r_k} - \frac{c_k r_k^T V_{k-1} + V_{k-1} r_k c_k^T}{c_k^T r_k}$$

Jako wartość startową przyjmuje się zwykle wartość  $V_k = 1$ , a pierwsza iteracja przeprowadzana jest zgodnie z algorytmem największego spadku. Algorytm zmiennej metryki charakteryzuje się dość szybką zbieżnością. Jego wadą jest stosunkowo duża złożoność obliczeniowa, związana z koniecznością wyznaczenia w każdym kroku  $n^2$  elementów hesjanu.

#### 2.1.4. Drzewa regresyjne

Drzewa regresyjne (ang. regression trees) należą do popularnych narzędzi maszynowego uczenia indukcyjnego. Aproksymują one funkcje docelowe w sposób dyskretny i reprezentują je w strukturze drzewiastej lub alternatywnie w zbiorze reguł decyzyjnych. Charakterystyczny dla drzew decyzyjnych jest podział



wielowymiarowej przestrzeni cech na rozłączne segmenty  $R_k$ , w ramach których dopasowywany jest następnie prosty model aproksymujący lokalnie funkcję docelową wewnątrz rozpatrywanego segmentu.

W pracy wykorzystane zostało drzewo regresyjne w wersji CART (ang. Classification and Regression Trees). Drzewo to składa się z węzłów pośrednich, w których wykonuje się testy na zmiennych objaśniających, węzłów końcowych (liści) przechowujących wartości zmiennej zależnej  $z_g$  (wyrażonych jako wartość średnia obiektów należących do danego segmentu) oraz gałęzi łączących węzły. Omawiany rodzaj modelu jest drzewem binarnym, co oznacza, że z każdego węzła pośredniego wychodzą dwie gałęzie (potomkowie). W procesie konstrukcji drzewa w węzłach pośrednich zachodzi podział zbioru przykładów na dwa podzbiory: przykładów pozytywnych, które spełniają test przypisany do węzła oraz przykładów negatywnych, które nie spełniają tego testu. Dąży się do tego, aby wartości funkcji docelowej przykładów w tych podziorach miały jak najmniejszą wariancję:

$$(s)^2(P) = \frac{1}{N_k} \sum_{k \in R_k} (z_k - \bar{z}_{R_k})^2$$

gdzie:  $N_k$  – liczba obserwacji znajdujących się w segmencie  $R_k$ ,  $\bar{z}_{R_k}$  – wartość średnia funkcji docelowej przykładów w zbiorze  $R_k$ .

Wariancja określona wzorem powyższym jest miarą niejednorodności węzła (ang. impurity). Podział zbioru przykładów w węźle na dwa podzbiory w wyniku wykonania testu jest uzasadniony, jeśli prowadzi do zmniejszenia niejednorodności, czyli:

$$N_P (s)^2(R_{kP}) + N_N (s)^2(R_{kN}) < (s)^2(R_k)$$

gdzie:  $(R_k)$ ,  $(R_{kP})$ ,  $(R_{kN})$  są odpowiednio zbiorami indeksów przykładów w węźle nadrzędnym (rodzicu), oraz w węzłach podrzędnych, do których docierają odpowiednio przykłady pozytywne i negatywne, ponadto  $N_P$  i  $N_N$  są licznymi przykładów pozytywnych i negatywnych w zbiorze:

$$R_k : N_P = |R_{kP}| / |R_k|, N_N = |R_{kN}| / |R_k|$$

W przypadku zmiennych ciągłych kryterium podziału zbioru  $R_k$  w węźle jest test nierównościowy postaci:

$$T(x) = \begin{cases} P, & x_i > \theta_i \\ N, & x_i \leq \theta_i \end{cases}$$

gdzie  $\theta_i$  jest wartością progową. Zmienną do testu oraz wartość progową dobiera się stosując algorytm zachłanny. W każdym węźle rozważa się każdą zmienną objaśniającą i wszystkie możliwości podziału zbioru  $R_k$  na dwa podzbiory  $R_{kP}$  i  $R_{kN}$  przy użyciu tej zmiennej. Wymaga to posortowania zbioru  $R_k$  względem wartości analizowanej zmiennej  $x^{(p)}$  i przyjęcia jako wartości progowych środków każdego z przedziałów wyznaczonych przez kolejne wartości tej zmiennej. Liczba możliwych podziałów w danym węźle jest równa  $T-1$ , gdzie  $T$  jest liczbą unikalnych wartości, którą przyjmuje rozpatrywana zmienna niezależna.

Jeżeli zmienna  $x^{(p)}$  ma charakter jakościowy (nominalny lub porządkowy), liczba możliwych podziałów dla zmiennej porządkowej wynosi 2 oraz  $2^{T-1}$  dla zmiennej nominalnej, gdzie  $T$  jest liczbą kategorii, którą przyjmuje rozpatrywana zmienna objaśniająca. Na szczęście bez utraty ogólności dla zagadnienia regresji oraz klasyfikacji

binarnej z miarą niejednorodności węzła wykorzystującą indeks Giniego lub współczynnik entropii, liczbę przeszukiwanych podziałów można zredukować do T-1. Wymaga to posortowania zbioru kategorii względem wartości średnich badanej zmiennej zależnej  $z_g$  a następnie rozpatrywania binarnych podziałów jak w przypadku zmiennych ciągłych.

Należy zauważyć, że omawiane podziały prowadzą do podziałów lokalnych, pozwalających znaleźć najlepszy test w danym węźle, w kontekście wcześniejszych wyborów testów do węzłów nadrzędnych i wobec nieznanych testów w węzłach podrzędnych. Drzewo tak zbudowane, niekoniecznie jest optymalne globalnie.

Drzewo regresyjne najczęściej buduje się według schematu zstępującego. Zaczynając od węzła startowego (korzenia), w którym rozważa się cały zbiór uczący, przemieszcza się do kolejnych węzłów w wyniku podziału przykładów na podzbiory  $R_{kP}$  i  $R_{kN}$ . Wielkość drzewa uzależniona jest od złożoności rozpatrywanego problemu. Wielkością drzewa, która implikuje stopień jego dopasowania do danych uczących można sterować, np. za pomocą parametru  $c_p$ . Parametr ten wskazuje algorytmowi o ile powinien się zwiększyć ogólny współczynnik determinacji modelu w każdym kroku, aby mogło dojść do podziału danego węzła decyzyjnego. Inny sposób polega na utworzeniu rozgałęzienia w węźle tylko wtedy, jeśli zanieczyszczenie ulegnie dostatecznej redukcji, w przeciwnym przypadku tworzy się liść. Można też zbudować drzewo maksymalne, a potem przyciąć je do odpowiedniego rozmiaru. Kryteria przycinania (ang. pruning) biorą pod uwagę jakość dopasowania i złożoność drzewa mierzoną np. liczbą liści. Jeszcze inny sposób ustalania struktury drzewa opiera się na walidacji krzyżowej.

### 2.1.5. Regresyjne lasy losowe

Algorytm lasów losowych (ang. random forest) jest techniką rozwiązującą problemy związane z regresyjnymi drzewami decyzyjnymi. Ogólna idea lasów losowych polega na stworzeniu wielu różnych drzew regresyjnych, stąd w nazwie słowo „las”. W ten sposób ograniczone zostaje zjawisko niestabilności prognoz. Słowo „losowe” bierze swój początek w tym, że każde drzewo jest tworzone na losowej próbie  $n$  obserwacji pobieranych ze zwracaniem ze zbioru uczącego  $U$  (próba bootstrapowa). Drugi element losowości polega na tym, że w przeciwieństwie do regresyjnych drzew losowych, gdzie każdy kolejny podział jest wyłaniany na podstawie wszystkich zmiennych objaśniających, w algorytmie lasów losowych, najlepszy podział jest określany w oparciu o próbę losową zmiennych. Standardowym zaleceniem dla problemów regresyjnych jest losowanie  $p/3$  zmiennych objaśniających. Istotna jest również uwaga, że każde z drzew składowych lasu losowego nie podlega przycinaniu. Prognozowanie na podstawie modelu lasu losowego polega na określeniu prognoz dla każdego drzewa wchodzącego w skład lasu oraz wyznaczeniu średniej arytmetycznej tych prognoz indywidualnych, jako prognozy całego modelu.

Najważniejszymi zaletami lasów losowych jest ich empirycznie wykazana efektywność oraz mała liczba parametrów sterujących ich działaniem. Należy jedynie zdecydować ile drzew ma wchodzić w skład lasu oraz ile powinna wynosić minimalna liczba obserwacji w każdym liściu drzewa. Jest to istotna zaleta w porównaniu do sieci neuronowych czy regresyjnej metody wektorów nośnych, gdzie liczba parametrów (a tym samym liczba

możliwych wariantów ich ustawień) jest duża. Optymalne wartości opisanych powyżej parametrów metody lasów losowych, mogą być określone na zbiorze uczącym z wykorzystaniem techniki sprawdzianów krzyżowych.

Interesującą obserwacją jest to, że nawet znaczna liczba drzew oraz niska minimalna liczba obserwacji w liściach drzew nie prowadzi do dużego pogorszenia się zdolności prognostycznych lasów losowych. Stwierdzenie to łagodzi objawy związane z konsekwencjami stosowania nieoptymalizowanych wartości parametrów sterujących. Kolejną pozytywną cechą lasów losowych jest wbudowany mechanizm określania ważności poszczególnych zmiennych objaśniających, umożliwiając stworzenie wykresu ważności zmiennych niezależnych.

Algorytm lasów losowych może się nie sprawdzić w przypadku dużej liczby zmiennych objaśniających, z których tylko niewielka liczba jest znacząca dla prognozowania. Sytuacja ta powoduje, że w momencie, gdy następuje losowanie ich podzbioru może się okazać, że nie została wylosowana żadna ważna zmienna objaśniająca.

### 2.1.6. Regresja wektorów wspierających

Choć maszyny wektorów wspierających (ang. support vector machines) pierwotnie były przeznaczone tylko do rozwiązywania zadania klasyfikacji można je także zastosować do zadania analizy regresji, co zostało zaproponowane pierwszy raz w 1997 roku, jako algorytm  $\varepsilon$ -SVR (ang. Support Vector Regression). Podobnie jak w przypadku wcześniej omawianych metod uczenia maszynowego, celem algorytmu jest przybliżenie wartości rzeczywistej  $z_g$  pewną funkcją aproksymacyjną  $f(x_g)$ .

W tym celu, budowana jest optymalna hiperpłaszczyzna, która otacza margines, na zewnątrz którego znajdują się wektory podpierające. Do wyznaczenia tego marginesu potrzebna jest funkcja błędu, np. postaci:

$$|z_g - f(x_g)|_{\varepsilon} = \begin{cases} 0, & |z_g - f(x_g)| \leq \varepsilon \\ |z_g - f(x_g)| - \varepsilon, & |z_g - f(x_g)| > \varepsilon \end{cases}$$

Wartość  $\varepsilon$  definiuje margines, wewnątrz którego wartość funkcji błędu wynosi zero. Minimalizowany błąd wygląda następująco:

$$\left( \min_w \left( \frac{1}{2} \|w\|^2 + C \sum_{g=1}^n |z_g - f(x_g)|_{\varepsilon} \right) \right)$$

W przedstawionej formule, można wyróżnić dwie składowe funkcji celu:

$\frac{1}{2} \|w\|^2$  – odpowiadająca za minimalizowanie szerokości marginesu,

$\sum_{g=1}^n |z_g - f(x_g)|_{\varepsilon}$  – odpowiadająca za minimalizowanie sumy odległości od marginesu obiektów leżących poza tym marginesem.

Parametr  $C$  odpowiada za określenie wagi, jaką przypisuje się poszczególnym składowym.

W szczególności mała wartość  $C$  powodować będzie wzrost szerokości marginesu (większe uśrednianie wartości), zaś duże wartości  $C$  jego zawężanie (lepsze dopasowanie do obiektów).

Tak zdefiniowane zadanie przekształca się następnie do nieco innej postaci. W tym celu wprowadza się dwie dodatkowe zmienne, określające wartość funkcji błędu dla danych odstających (czyli tych, które nie mieszczą w obszarze wyznaczonym przez margines błędu):

$$\xi = (z_g - f(x_g)) - \varepsilon \text{ dla } z_g > f(x_g) + \varepsilon$$

$$\xi^* = (f(x_g) - z_g) - \varepsilon \text{ dla } z_g < f(x_g) - \varepsilon$$

W przypadku obiektów znajdujących się wewnątrz wyznaczonego obszaru, wartość funkcji błędu wynosi zero, czyli uznaje się, że jeśli spełniona jest nierówność  $|z_g - f(x_g)| \leq \varepsilon$ , to wartość funkcji w określonym punkcie  $x_g$  (czyli dla danego obiektu  $x_g$ ) została określona poprawnie.

Minimalizowany błąd po uwzględnieniu nowych zmiennych wygląda następująco:

$$\left( \min_{w, \xi, \xi^*} \left( \frac{1}{2} \|w\|^2 + C \left( \sum_{g=1}^n \xi + \sum_{g=1}^n \xi^* \right) \right) \right)$$

Tak jak w poprzednim zapisie, także i w tym przypadku równania widać, że celem optymalizacji jest minimalizowanie szerokości marginesu oraz minimalizowanie sumy odległości od marginesu obiektów leżących poza nim. Oczywiście im węższy margines, tym więcej obiektów znajdzie się poza nim (rośnie wartość drugiej składowej funkcji celu). Po przekształceniach, które można znaleźć np. w pracy Schölkopf i Smola z 2002, zadanie przekształcone do postaci dualnej Lagrange'a przyjmuje postać:

$$\max_{\alpha, \alpha^*} L_d(\alpha, \alpha^*) = -\varepsilon \sum_{g=1}^n (\alpha_g^* + \alpha_g) + \sum_{g=1}^n (\alpha_i^* + \alpha_i) z_g$$

$$-\frac{1}{2} \sum_{g,h=1}^n (\alpha_g^* + \alpha_g)(\alpha_h^* + \alpha_h) K(x_g + x_h)$$

przy ograniczeniach

$$\sum_{g=1}^n \alpha_g = \sum_{g=1}^n \alpha_g^*$$

$$\bigvee_{g=1}^n 0 \leq \alpha_g \leq C$$

$$\bigvee_{g=1}^n 0 \leq \alpha_g^* \leq C$$

gdzie  $K(x_g, x_h)$  jest funkcją jądrową (ang. kernel function) zdefiniowaną wzorem:

$$K(x_g, x_h) = \langle x_g, x_h \rangle$$

Ostatecznie funkcja regresji przyjmuje postać:

$$f(x_g) = \sum_{g=1}^n (\alpha_g^* + \alpha_g) K(x_g, x_h) + b$$

### 2.1.7. Regresja k-najbliższych sąsiadów

Estymator regresji k-najbliższych sąsiadów (k-nn, ang. k-nearest neighbors regression) jest metodą nieparametryczną, co oznacza, że nie czyni się założeń odnośnie modelu generującego dane. Jego główną zaletą jest prostota konstrukcji i niska złożoność obliczeniowa.

Prognozę wartości zmiennej objaśnianej  $z_g$  metodą k-nn na podstawie wektora zmiennych objaśniających  $x_g$  wyznacza się jako:

$$z_g = \frac{\sum_{h=1}^k z_h I(x_g, x_h)}{k}$$

gdzie:

$$I(x_g, x_h) = \begin{cases} 1, & \text{gdy } x_h \text{ jest jednym z } k \text{ najbliższych sąsiadów } x_g \\ 0, & \text{w przeciwnym wypadku} \end{cases}$$

natomiast  $x_h$  jest jednym z k-najbliższych sąsiadów  $x_g$ , w przypadku gdy odległość  $d(x_g, x_h)$  należy do k najmniejszych odległości pomiędzy obserwacjami ze zbioru  $X$ , a  $x$ . Najczęściej wykorzystywaną odległością jest odległość Euklidesa zdefiniowana wzorem:

$$d(x_g, x_h) = (\sum_{j=1}^p (x_g^j - x_h^j)^2)^{0,5}$$

W przypadku zastosowania odległości Euklidesa złożoność obliczeniowa regresji k-nn wynosi  $O(knp)$ , gdzie  $k$  to liczba sąsiadów,  $n$  to liczba obserwacji w zbiorze  $X$ , natomiast  $p$ , to liczba zmiennych objaśniających. Regresja k-nn nie jest metodą odporną, jednak w przypadku zbiorów danych o skomplikowanej strukturze (np. w przypadku zależności nieliniowych) regresja k-nn dużo lepiej może zostać dopasowana do danych, niż metody zakładające zależności liniowe (regresja wyznaczona klasyczną metodą najmniejszych kwadratów).

## 2.2. Miary oceny jakości prognozy

Do wstępnych warunków, które uzasadniają wykonanie prognozy należą: znajomość oszacowań parametrów modelu, w którym zmienna prognozowana pełni rolę zmiennej objaśniającej, stabilność modelu w czasie (nie ulegają zmianom postać analityczna i wartości parametrów strukturalnych) lub następują zmiany powolne i regularne, które można z dostateczną dokładnością przewidzieć. Rozkład składnika losowego modelu jest stacjonarny lub zmiany tego rozkładu są powolne i na tyle regularne, że umożliwiają odpowiednio dokładne ekstrapolacje, znane są wartości zmiennych objaśniających w okresie prognostycznym, dopuszczalna jest ekstrapolacja modelu poza obszar zmienności zmiennych objaśniających zaobserwowanych w próbie, na podstawie której oszacowano model. W praktyce najczęściej stosuje się prognozy punktowe, rzadziej prognozy przedziałowe.

Jakość prognozy weryfikuje się wykonując: prognozę „ex post” (trafność prognozy) – prognoza została wykonana po upływie czasu, na który była ustalana lub prognozę „ex ante” (dokładność prognozy) – prognoza została wykonana przed upływem czasu, na który była ustalana. Błąd prognozy traktuje się jako wielkość losową. Do najczęściej stosowanych miar trafności prognozy m.in. należą:

---

Projekt pt. „Prace rozwojowe oraz testy w warunkach rzeczywistych autonomicznego i inteligentnego sterownika” nr Umowy z Województwem Łódzkim, w imieniu którego działa Centrum Obsługi Przedsiębiorcy: **RPLD.01.02.02-10-0006/18-00** realizowany w ramach Poddziałania I.2.2 Regionalnego Programu Operacyjnego Województwa Łódzkiego na lata 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego.

- Błąd absolutny (ang. Error):

$$E_g = p_g - z_g$$

gdzie  $z_g$  jest wartością rzeczywistą w okresie  $g$ , a  $p_g$  jest wartością przewidywaną w okresie  $g$ .

- Błąd względny (ang. Relative Error):

$$RE_g = \frac{p_g - z_g}{z_g}$$

- Średni absolutny błąd procentowy (ang. Mean Absolute Percentage Error):

$$MAPE = \frac{1}{n} \sum_{g=1}^n |RE_g|$$

- Średni błąd kwadratowy (ang. Mean Squared Error):

$$MSE = \frac{1}{n} \sum_{g=1}^n E_g^2$$

W pracy ponadto zostały zastosowane dwie dodatkowe miary zdefiniowane w następujący sposób:

- Jakość (ang. Accuracy) – która określa, ile prawidłowych wskazań uzyskuje model w danym horyzoncie prognostycznym. W tym celu konieczne jest zdefiniowanie prawidłowej prognozy jako wartości znajdującej się w pewnym zakresie procentowym w stosunku do rzeczywistego zapotrzebowania. Dla przykładu, przy zapotrzebowaniu równym 0,1 kWh, 15% przedział poprawności wynosiłby 0,085–0,115 kWh, w wyniku czego, prognoza wynosząca 0,2 kWh zostanie uznana za złą. W praktyce jednak, taka wartość prognozy byłoby dopuszczalna. Aby przezwyciężyć tę fałszywą utratę dokładności, można zdefiniować dwa zakresy, w których prognoza będzie uznawana za poprawną. Dla zapotrzebowania mniejszego od 1 kWh dopuszczalny będzie błąd w zakresie  $\pm 0,15$  kWh od wartości rzeczywistej, natomiast dla zapotrzebowania większego niż 1 kWh, dopuszczalny błąd prognozy powinien zawierać się w przedziale nie większym niż 15%. Dokładność prognozy dla godziny  $t$  jest zdefiniowana jako:

$$Acc = \sum 1\{z_g > 1 \& |z_g - p_g| < p_g * 0,15 + \sum 1\{z_g > 1 \& |z_g - p_g| < 0,15$$

- r-MAPE (ang. resistant-Mean Absolute Percentage Error) – M-estymator parametru  $\mu$  wykorzystujący estymator największej wiarygodności jest zdefiniowany jako rozwiązanie  $\theta$  dla:

$$\sum_{g=1}^n \rho \left( \frac{\left| \frac{p_g - z_g}{z_g} \right| - \theta}{\sigma} \right) = \min_{\theta}$$

lub

$$\sum_{g=1}^n \varphi \left( \frac{\left| \frac{p_g - z_g}{z_g} \right| - \theta}{\sigma} \right) = 0$$

gdzie  $\varphi = \rho'$ , oraz  $\sigma$  jest parametrem skali. Dla określonej pozytywnej stałej  $k$ , estymator Hubera (1981) jest zdefiniowany następującą funkcją:

$$\varphi(k) = \begin{cases} k & z_g > k \\ z_g & -k \leq z_g \leq k \\ -k & z_g < -k \end{cases}$$

gdzie  $k$  jest dostrajającym parametrem determinującym odporność, ustawionym na wartość 1,5. W rzeczywistości parametr skali  $\sigma$  jest nieznanym, więc estymuje się jego wartość za pomocą odpornego estymatora zdefiniowanego:

$$\text{MAD} = \text{mediana}(|z_g - \text{mediana}(z_g)|)$$

W zagadnieniach stosuje się także błędy: średni błąd bezwzględny, średni absolutny błąd względny, średni kwadratowy błąd względny, pierwiastek średniego kwadratowego błędów, pierwiastek, suma kwadratów błędów bezwzględnych, maksymalny absolutny błąd procentowy oraz współczynnik korelacji liniowej Pearsona.

Błędy „ex ante” są miarą dopuszczalności prognoz. Tego typu błędy mogą być tylko szacowane, a sposób szacowania zależy od metody prognostycznej. Dopuszczalność prognozy wyznacza się w tym samym czasie, w którym wyznacza się prognozę. Wiele metod nie pozwala na wyznaczenie błędów „ex ante” (metody średniej ruchomej, metody prognozowania przez analogię, nieliniowe i niesprowadzalne do liniowych modele regresji).

### 2.3. Budowa wektora charakterystyk dla prognoz

W prezentowanych w tym rozdziale opisach badań, główna uwaga skupiona została na prognozowaniu zapotrzebowania na energię elektryczną dla pojedynczego przejścia dla pieszych o horyzoncie jednej godziny oraz 24 godzin do przodu. W tym celu, skonstruowany został wektor zmiennych niezależnych (atrybutów) przedstawiony w Tabeli 1. Atrybuty te, wyznaczone zostały empirycznie na podstawie szeregów czasowych godzinowego zapotrzebowania na energię elektryczną, temperatury, wilgotności, oraz informacji kalendarzowych.

Zapotrzebowanie na energię elektryczną zmienia się w czasie w zależności od pory dnia (cykle dzienne), dnia tygodnia (cykle tygodniowe), dnia miesiąca (cykle miesięczne), pory roku (cykle sezonowe) oraz dni nietypowych np. świąt. Z tego też względu, wyznaczone zostały zmienne jakościowe przyjmujące wartości 0 lub 1 (w tym zmienne związane z godziną, oraz zmienne związane z dniem miesiąca, zmienne związane z dniem tygodnia i miesiącem, a także zmienne wskazujące dzień świąteczny oraz zachód słońca). Cecha informująca o dniu świątecznym, związana była ze wszystkimi świątami i dniami wolnymi od pracy. Zapotrzebowanie determinowane jest ponadto przez poziom oświetlenia oraz poziom nasłonecznienia, co związane jest bezpośrednio ze wschodem i zachodem słońca. Zmienna wskazująca zachód słońca przyjmowała wartość 1, jeżeli część rozpatrywanej godziny, w której odnotowany był stan zachodu, stanowił więcej niż połowę. Taki sposób przygotowania danych powoduje, że funkcja regresji w zależności od wystąpienia danej kombinacji cech jakościowych połączonych z pozostałymi zmiennymi objaśniającymi, może wskazywać mniejszą lub większą wartość prognozy badanego zjawiska.

Podstawowymi zmiennymi branymi pod uwagę w procesie prognozowania są informacje pochodzące bezpośrednio z szeregu czasowego. Cechy powstałe poprzez dekompozycję szeregu czasowego określają m.in. minimalne, maksymalne oraz rzeczywiste zapotrzebowanie w określonych przedziałach czasowych (nawet do 35 dni wstecz).

Nr zmiennej	Opis	Formuła
1-24	Binarna informacja wskazująca rozpatrywaną godzinę	$G_i, i = 1, \dots, 24$
25-55	Binarna informacja wskazująca rozpatrywany dzień miesiąca	$D_i, i = 1, \dots, 31$
56-62	Binarna informacja wskazująca rozpatrywany dzień tygodnia	$T_i, i = 1, \dots, 7$
63-74	Binarna informacja wskazująca rozpatrywany miesiąc w roku	$M_i, i = 1, \dots, 12$
75	Binarna informacja wskazująca dzień świąteczny	S
76	Binarna informacja wskazująca zachód słońca	N
77-100	Zapotrzebowanie w poprzednich 24 godzinach	$Z_{g-i}, i = 1, \dots, 24$
101-104	Minimalne zużycie w poprzednich 3, 6, 12, 24 godzinach	$\min\{Z_{g-1}, \dots, Z_{g-i}\}, i = 3, 6, 12, 24$
105-108	Maksymalne zużycie w poprzednich 3, 6, 12, 24 godzinach	$\max\{Z_{g-1}, \dots, Z_{g-i}\}, i = 3, 6, 12, 24$
109-114	Zapotrzebowanie o tej samej godzinie w poprzednim tygodniu	$Z_{g,d-i}, i = 2, \dots, 7$
115-118	Zapotrzebowanie o tej samej godzinie w poprzednim miesiącu	$Z_{g,d-i}, i = 14, 21, 28, 35$
119-122	Średnia temperatura w wybranym okresie poprzedniego tygodnia	$\text{avg}\{T_{g-1}, \dots, T_{g-i[+1]}\}, i = 1, 3, 6, 12, 24$
123-128	Temperatura o tej samej godzinie w poprzednim tygodniu	$T_{g,d-i}, i = 2, \dots, 7$
129-152	Prognozowana temperatura o tej samej godzinie w dniu kolejnym	$T_{g+i}, i = 1, \dots, 24$
153-177	Temperatura w poprzednich 24 godzinach	$T_{g-i}, i = 1, \dots, 24$



178-181	Średnia temperatura w wybranym okresie poprzedniego miesiąca	$\text{avg}\{T_{g,d-1}, \dots, T_{g,d-i[+1]}\}, i = 7, 14, 21, 28, 35$
182-185	Średnia wilgotność w wybranym okresie poprzedniego tygodnia	$\text{avg}\{H_{g-1}, \dots, H_{g-i[+1]}\}, i = 1, 3, 6, 12, 24$
186-192	Wilgotność o tej samej godzinie w poprzednim tygodniu	$H_{g,d-i}, i = 2, \dots, 7$
193-216	Prognozowana wilgotność o tej samej godzinie w dniu kolejnym	$H_{g+i}, i = 1, \dots, 24$
217-240	Wilgotność w poprzednich 24 godzinach	$H_{g-i}, i = 1, \dots, 24$
241-244	Średnia wilgotność w wybranym okresie poprzedniego miesiąca	$\text{avg}\{H_{g,d-1}, \dots, H_{g,d-i[+1]}\}, i = 7, 14, 21, 28, 35$
245-248	Średnia prędkość wiatru w wybranym okresie poprzedniego tygodnia	$\text{avg}\{W_{g-1}, \dots, W_{g-i[+1]}\}, i = 1, 3, 6, 12, 24$
249-254	Prędkość wiatru o tej samej godzinie w poprzednim tygodniu	$W_{g,d-i}, i = 2, \dots, 7$
255-278	Prognozowana prędkość wiatru o tej samej godzinie w kolejnym dniu	$W_{g+i}, i = 1, \dots, 24$
279-302	Prędkość wiatru w poprzednich 24 godzinach	$W_{g-i}, i = 1, \dots, 24$
303-306	Średnia prędkość wiatru w wybranym okresie poprzedniego miesiąca	$\text{avg}\{W_{g,d-1}, \dots, W_{g,d-i[+1]}\}, i = 7, 14, 21, 28, 35$
307-310	Średnie nasłonecznienie w wybranym okresie poprzedniego tygodnia	$\text{avg}\{N_{g-1}, \dots, N_{g-i[+1]}\}, i = 1, 3, 6, 12, 24$
311-317	Nasłonecznienie o tej samej godzinie w poprzednim tygodniu	$N_{g,d-i}, i = 2, \dots, 7$
318-341	Prognozowane nasłonecznienie o tej samej godzinie w kolejnym dniu	$N_{g+i}, i = 1, \dots, 24$
342-365	Nasłonecznienie w poprzednich 24 godzinach	$N_{g-i}, i = 1, \dots, 24$
366-369	Średnie nasłonecznienie w wybranym okresie poprzedniego miesiąca	$\text{avg}\{N_{g,d-1}, \dots, N_{g,d-i[+1]}\}, i = 7, 14, 21, 28, 35$
370-393	Ilość uruchomień przejścia przez pieszych w poprzednich 24 godzinach	$U_{g-i}, i = 1, \dots, 24$

394-399	Ilość uruchomień przejścia przez pieszych o tej samej godzinie w poprzednim tygodniu	$N_{g,d-i}, i = 2, \dots, 7$
400-403	Ilość uruchomień przejścia przez pieszych o tej samej godzinie w poprzednim miesiącu	$N_{g,d-i}, i = 14, 21, 28, 35$

Jednym z założeń uczenia nadzorowanego, jest brak występowania współliniowości zmiennych objaśniających. Wprowadzając do modelu silnie liniowo zależne ze sobą zmienne, wprowadza się do modelu (przy każdej zmiennej) małą bądź zerową unikalną część wyjaśnienia zmiennej objaśnianej. W zależności od zastosowanej metody, jedna ze zmiennych niezależnych silnie powiązanych, straci swoją moc przewidywania zmiennej zależnej na rzecz innej zmiennej niezależnej, z racji występowania silnego związku pomiędzy nimi. Występowanie współliniowości może powodować zafałszowanie wartości parametrów modelu np. może powodować ich przewartościowanie, niedowartościowanie, zmianę znaku parametru z ujemnego na dodatni i na odwrót. Z tego też względu, podczas procesu uczenia modeli prognostycznych na danych godzinowych, z każdej z grup zmiennych jakościowych, usunięta została jedna zmienna, związana z ostatnią wartością (poziomem) pierwotnej zmiennej niezależnej. Ogólnie, liczba zmiennych niezależnych rozpatrywanych w modelu (utworzonych w oparciu o zmienną jakościową) wynosi  $c-1$  (gdzie  $c$  jest liczbą kategorii zmiennej jakościowej).

## 2.4. Opracowanie podejścia prognostycznego opartego na algorytmach uczenia maszynowego

Punktem wyjścia w całym procesie uczenia nadzorowanego jest podział całego szeregu czasowego na trzy części, które odpowiadają odpowiednio próbie uczącej, walidacyjnej oraz testowej.

Głównymi kryteriami branymi pod uwagę w procesie uczenia modeli prognostycznych, jest uzyskanie przede wszystkim dobrej generalizacji wiedzy, przy możliwie jak najmniejszym błędzie. Najczęściej wykorzystywaną miarą jakości prognozy w zagadnieniach elektroenergetycznych jest błąd MAPE. Z tego też względu, w procesie poszukiwania najlepszych parametrów dla wszystkich modeli zbudowanych za pomocą algorytmów uczenia maszynowego, minimalizowana była funkcja postaci:

$$f(MAPE_U, MAPE_W) = \frac{1}{2} |MAPE_U - MAPE_W| + \frac{1}{2} MAPE_W$$

gdzie  $MAPE_U$  oraz  $MAPE_W$ , są odpowiednio błędami dla zbioru uczącego oraz walidacyjnego.

Tak zdefiniowana funkcja celu zapewnia stabilność modelu w czasie oraz posiadanie jak najmniejszego błędu na obserwacjach jawnie nie biorących udziału w procesie uczenia tj. w przyszłości dobre przewidywanie dla nowych danych.

Podczas procesu uczenia wykorzystano następujące ustawienia parametrów algorytmów.

### Sztuczne sieci neuronowe

Sztuczne sieci neuronowe w przeciwieństwie do innych algorytmów uczenia maszynowego wymagają specjalnego przygotowania danych wejściowych. Dlatego też, wektor atrybutów wejściowych dla zmiennych na skali ilorazowej, znormalizowany został poprzez standaryzację, natomiast dla zmiennych na skali nominalnej (dychotomicznej) poprzez przekształcenie wartości przyjmujących wartość 0 na wartość -1. Zmienna objaśniana normalizowana jest poprzez unitaryzację zerową, względem zakresu wartości dla zbioru uczącego, walidacyjnego oraz testowego. Ostatecznie, w celu wykonania prognozy, proces przekształcania zmiennej zależnej jest odwracany.

Algorytmem uczenia sieci neuronowej jest algorytm zmiennej metryki (BFGS). Minimalizowaną funkcją w procesie uczenia jest suma kwadratów reszt. Sieć posiada jedną warstwę wejściową z liczbą neuronów równą liczbie zmiennych objaśniających oraz jeden neuron w warstwie wyjściowej. W warstwie ukrytej sprawdzana jest różna ilość neuronów znajdująca się w następującym zbiorze {10, 15, 20, 25, 30, 35, 40, 45, 50}. Do aktywacji wszystkich neuronów znajdujących się w sieci, wykorzystana została funkcja logistyczna. Czynniki regularyzacyjny wprowadzający karę za zbyt duże wagi w sieci (zbytne dopasowanie do danych uczących), ustawiony był odpowiednio na 0,01, 0,1 lub 0,5. Sumarycznie, za każdym razem uczonych było 27 sieci neuronowych o różnych parametrach (liczba neuronów w warstwie ukrytej pomnożona przez liczbę wartości sprawdzanych kar). W celu przewyższenia przeuczenia sieci neuronowej, po każdej zakończonej iteracji uczącej (liczba epok uczących wynosiła 50 iteracji), sprawdzana była wartość miary zdefiniowanej w powyższym równaniu. Zachowywany był następnie stan sieci (wartości wag synaptycznych), minimalizujący powyższą miarę. Na zakończenie, spośród 27 nauczonych sieci neuronowych, wybierana była sieć charakteryzująca się najmniejszym błędem.

### **Regresja k-najbliższych sąsiadów**

Aby polepszyć działanie algorytmu regresji k-najbliższych sąsiadów, zastosowano normalizację zmiennych objaśniających (standaryzacja dla zmiennych ilościowych oraz dla zmiennych binarnych przekształcenie wartości przyjmujących wartość 0 na wartość -1). Jej zastosowanie powoduje, że wszystkie wymiary, dla których liczona jest odległość euklidesowa posiadają jednakową istotność. W przeciwnym przypadku mogłoby dojść do sytuacji, w której pojedynczy wymiar zdominowałby inne wymiary.

Model regresji k-najbliższych sąsiadów uczony za pomocą funkcji knnreg. Liczba sąsiadów, na podstawie których wyznaczana była wartość prognozy, sprawdzana była dla wartości znajdujących się w zbiorze: {5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 250, 300}. Dla każdej z wymienionych wartości, wyznaczany był błąd MAPE dla zbioru uczącego i walidacyjnego. Za optymalną liczbę sąsiadów, a co za tym idzie, ostateczną postać modelu, uznawana była wartość minimalizująca powyższy wzór.

### **Drzewa regresyjne**

W tym przypadku do uczenia wykorzystano drzewa regresyjne z implementacją algorytmu CART. Minimalizowanym kryterium w procesie podziału wielowymiarowej przestrzeni cech, jest rozproszenie (wariancja) wartości wokół wartości średniej zmiennej zależnej, dla obserwacji znajdujących się w danym węźle

(liściu), tj. na każdym etapie podziału węzła wybierana była ta zmienna oraz jej konkretna wartość, która minimalizowała sumę kwadratów reszt. Minimalna liczba obserwacji, która musiała się znajdować w węźle aby rozpatrywany był ewentualny podział, wynosiła 20. Każdy liść musiał posiadać minimum 6 obserwacji, w przeciwnym wypadku węzeł, z którego powstał liść nie był dzielony.

Zamiast przycinania drzewa na końcu działania algorytmu, zastosowano przycinanie w trakcie wzrostu. Ogólnie, podejście takie powoduje przerwanie procesu tworzenia nowych węzłów z podziałami, gdy kolejne podziały dają niewielki przyrost w trafności predykcyjnej. Parametr  $cp$  ustawiony od wartości 0 do wartości 0,1 (inkrementacja wynosząca 0,001), wskazywał algorytmowi, o ile powinien zwiększyć się ogólny współczynnik determinacji modelu, aby mogło dojść do podziału danego węzła decyzyjnego. Drzewo budowane było maksymalnie do głębokości wynoszącej 30 poziomów. Spośród 1000 przetestowanych drzew regresyjnych, ostatecznie wybierana była ta struktura, która charakteryzowała się najmniejszym błędem na zbiorze walidacyjnym przy jednoczesnej jak najlepszej stabilności (zgodnie z powyższym wzorem).

### **Regresyjne lasy losowe**

Za każdym razem przed przystąpieniem do uczenia drzewa, losowane jest ze zwracaniem z  $n$ -elementowej próby uczącej  $n$  wektorów obserwacji. Na podstawie takiej pseudo-próby, do której trafiało w przybliżeniu 63% różnych przypadków, tworzone było następnie drzewo regresyjne CART. Każde drzewo było budowane do swoich maksymalnych rozmiarów (bez przycinania wstępującego ani zstępującego), przy założeniu, że minimalna liczba obserwacji w liściu jest nie mniejsza niż 5 obserwacji. Przed przystąpieniem do podziału każdego drzewa, losowany był pewien podzbiór zmiennych objaśniających. Liczba losowanych zmiennych do podziału wynosiła od 10 do maksymalnej liczby zmiennych niezależnych, otrzymując ostatecznie 12 różnych podzbiorów o innej liczebności. Liczba drzew w lesie losowym wynosiła 500. Wyznaczenie ostatecznej wartości prognozy, ustalana była nie poprzez standardowe uśrednianie po wszystkich drzewach, w których dana obserwacja uczestniczyła w procesie uczenia, lecz poprzez zastosowanie odpornego estymatora Hubera.

Na zakończenie, spośród 12 zbudowanych struktur lasów losowych, wybierany był model związany z najmniejszym błędem, zgodnie z powyższym wzorem.

### **Regresja wektorów nośnych**

Do konstrukcji modelu regresji wektorów nośnych w wersji  $\varepsilon$ -SVR, wykorzystano implementację algorytmu SMO (ang. Sequential minimal optimization for solving the quadratic programming problem). Wykorzystywaną funkcją jądrową, była funkcja liniowa. Wartość parametru  $\varepsilon$  definiująca szerokość marginesu, wewnątrz którego wartość funkcji błędu wynosi zero, ustawiona została na wartości ze zbioru {0,01, 0,05, 0,1, 0,25, 0,5, 0,75, 1}. Parametr regularyzacyjny  $C$ , sterujący karą za przekroczenie wektora nośnego (jest to kompromis między dokładnością dopasowania a zdolnością uogólniania), ustawiony był odpowiednio na wartości ze zbioru {0,0001, 0,0005, 0,001, 0,005, 0,01, 0,05, 0,1, 0,25, 0,5, 0,75, 1}. Ostatecznie, tak jak we wszystkich poprzednich przypadkach, opierając się na wynikach 77 zbudowanych modeli, wybierany był ten model, który minimalizował funkcję określoną powyższym wzorem.

## 2.5. Uwagi implementacyjne dotyczące opracowanego podejścia

Przedstawione wartości przetestowanych parametrów dla każdego algorytmu, wynikały z chęci przetestowania jak największego wachlarza możliwych ustawień, mając jednocześnie na uwadze ich znaczenie teoretyczne, logiczne jak i praktyczne. Zakres przyjętych wartości został ograniczony, ze względu na wykryte podczas fazy symulacji numerycznych prawidłowości, przedstawione poniżej:

- dobór zmiennych objaśniających do modelu poprzez m.in. eliminację nieskorelowanych zmiennych niezależnych ze zmienną zależną, heurystyczne przeszukiwanie zbioru zmiennych z wykorzystaniem algorytmów genetycznych, tworzenie nowych zmiennych niezależnych za pomocą technik redukcji wymiaru (analiza składowych głównych) wpływa negatywnie na ostateczną jakość prognozy,
- wykorzystanie zmiennych jakościowych w swojej pierwotnej postaci (np. zmienna wyróżniająca godzinę, dzień miesiąca czy też dzień tygodnia), nie przekładało się na poprawę wyników prognozy,
- większa ilość neuronów w warstwie ukrytej sieci neuronowej nie prowadziła do poprawy wyników predykcji, natomiast znacząco wydłużała czas związany z adaptacją wektora wag synaptycznych,
- w tego typu problemie liczba iteracji wynosząca 50, była wystarczająca. Sieci zazwyczaj już po ok. 10–20 epokach uczenia wykazywały nadmierne dopasowanie do danych,
- kara za zbyt duże wagi w sieci neuronowej większa od wartości 0,5, nie prowadziła do poprawy wyników,
- ze względów obliczeniowych (pamięciowych), maksymalna liczba najbliższych sąsiadów wynosiła 300,
- przy wartości parametru  $cp$  powyżej 0,1 drzewo regresyjne generowało co najwyżej jeden podział (korzeń i dwa liście),
- zastosowanie uśredniania jako metody wyznaczania ostatecznej wartości prognozy w lasach losowych, prowadziło do gorszych rezultatów. Było to szczególnie widoczne dla danych godzinowych, gdzie ilość obserwacji jak i rozrzut prognozy były duże,
- zastosowanie jądra gaussowskiego niezależnie od jego szerokości w regresji wektorów nośnych, prowadziło do uzyskania stałych prognoz, a co za tym idzie, względnie gorszych rezultatów,
- zastosowanie jądra wielomianowego w regresji wektorów nośnych, prowadziło do zbytelnego przeuczenia modelu względem zbioru uczącego. Było to szczególnie widoczne wraz ze wzrostem stopnia wielomianu.

## 2.6. Wykorzystywane metody porównawcze

### Prognoza naiwna

Pierwszą z wykorzystywanych metod porównawczych była prognoza naiwna. Za wartość prognozowaną uznawano odpowiednio:

- dla danych godzinowych oraz horyzontu prognostycznego wynoszącego 24 godziny, wartość z dnia poprzedniego tej samej godziny,

- dla danych godzinowych oraz horyzontu progностycznego wynoszącego jedną godzinę, wartość z godziny poprzedniej,

### Prognoza losowa

Drugą z wykorzystywanych metod porównawczych jest prognoza losowa. W pierwszym kroku zaobserwowane wartości zapotrzebowania na energię, pogrupowane zostały względem godzin w skali dnia (prognoza godzinowa) lub względem dnia tygodnia (prognoza dobowa). Dla każdej grupy zapotrzebowań, wyznaczone zostały dystrybuanty empiryczne. W kolejnym kroku losowana była wartość z rozkładu równomiernego z przedziału od 0 do 1 (losowana była wartość  $p$  – prawdopodobieństwo). Wartość ta służyła następnie do oszacowania kwantyla dystrybuanty empirycznej (ostateczna wartość prognozy), poprzez ważone uśrednianie kolejnych statystyk porządkowych  $z_g$  (funkcja quantile):

$$Q_p = (1 - \gamma)z_g + \gamma z_{g+1}$$

gdzie  $\gamma = np + m - g$ , ponadto  $n$  jest liczbą obserwacji,  $g = \text{floor}(np + m)$  oraz  $m = 1 - p$ .

### Model szeregu czasowego

Trzecią z wykorzystywanych metod porównawczych była prognoza wykorzystująca model ARIMA(p,d,q) (dane dobowe) lub sezonowy model ARIMA(p,d,q)(P,D,Q) (dane godzinowe). Modele estymowane były za pomocą funkcji dokonującej automatycznej identyfikacji i estymacji postaci modelu, minimalizując kryterium informacyjne Akaikego postaci:

$$AIC = -2\log(L) + 2(p+q+P+Q+k)$$

gdzie  $L$  jest maksymalizowaną funkcją wiarygodności oraz  $k = 1$ , jeżeli wyraz wolny modelu nie równa się zero (w przeciwnym wypadku  $k = 0$ ).

W przypadku danych godzinowych maksymalne wartości opóźnień wynosiły:  $p = 12$ ,  $q = 12$ ,  $P = 2$ ,  $Q = 2$ , natomiast do estymacji modeli dla danych dobowych maksymalne wartości opóźnień ustawione zostały na:  $p = 14$ ,  $q = 14$ . Rząd integracji każdego szacowanego modelu testowany był za pomocą testu KPSS (Kwiatkowski – Phillips – Schmidt – Shin) sprawdzającego hipotezę zerową o stacjonarności szeregu czasowego. Za ostateczną wartość prognozy uznawano odpowiednio:

- dla danych godzinowych oraz horyzontu progностycznego wynoszącego 24 godziny, 24-tą wartość prognozy,
- dla danych godzinowych oraz horyzontu progностycznego wynoszącego jedną godzinę, pierwszą wartość prognozy,

### Regresja krokowa

Ostatnią z wykorzystywanych metod porównawczych jest liniowa regresja krokowa, wykorzystująca funkcję lm w dwóch wersjach. Metoda wprowadzania postępującego jest najprostszą z procedur budowania modelu. Zakłada ona kolejne (krokowe) dołączanie do listy zmiennych objaśniających uwzględnionych w modelu, tych

zmiennych, które mają najistotniejszy wpływ na zmienną zależną. Wprowadzanie zmiennych do modelu określone było za pomocą wartości krytycznych dla statystyki F, badającej statystyczną istotność zestawu parametrów strukturalnych danego modelu liniowego. Wartość krytyczna dla wprowadzania zmiennych do modelu musiała przekraczać wartość krytyczną przy 5% poziomie istotności. Procedura kończyła swoje działanie w momencie wykorzystania wszystkich zmiennych objaśniających lub w sytuacji braku statystycznej istotności dla wprowadzenia jakiegokolwiek dodatkowej zmiennej.

Drugą metodą krokowego budowania modeli była procedura eliminacji wstecznej. Zakłada ona kolejne (krokowe) usuwanie z modelu zbudowanego ze wszystkich potencjalnych zmiennych tych spośród nich, które w danym kroku mają najmniej istotny wpływ na zmienną zależną. Usuwanie zmiennych z modelu odbywało się za pomocą wartości krytycznych dla statystyki F. Procedura kończyła swoje działanie w sytuacji braku statystycznej istotności dla usuwania jakiegokolwiek dodatkowej zmiennej przy 5% poziomie istotności.

## 2.7. Pozostałe rozpatrywane metody prognostyczne

W badaniach jako metody porównawcze oraz rozwinięcia metod opartych o algorytmy uczenia maszynowego sprawdzono następujące metody prognostyczne:

- sieci neuronowe posiadające dwie warstwy ukryte oraz uczone algorytmami typu propagacja wsteczna i gradienty sprzężone,
- samoorganizujące się sieci neuronowe w wersji nadzorowanej (bdk, Bi-Directional Kohonen map),
- krokowa regresja logistyczna, model opracowywany był poprzez transformację logitową rzeczywistego zapotrzebowania na energię, a następnie szacowanie linowego modelu regresyjnego,
- regularyzowana regresja liniowa z selekcją zmiennych (LASSO, ang. Least Absolute Shrinkage and Selection Operator),
- wielozmienna regresja adaptacyjna z użyciem funkcji sklepanych (MARS, ang. Multivariate Adaptive Regression Splines),
- rozmyte systemy regułowe oparte o funkcję: WM – Wang i Mendel oraz FIR.DM – (ang. fuzzy inference rules by descent method).

Przetestowane metody wykazywały się zróżnicowaną jakością prognostyczną. Charakteryzowały się one zazwyczaj lepszymi wynikami niż klasyczne metody prognostyczne (wymienione powyżej metody porównawcze tj. prognoza naiwna, prognoza losowa, model szeregu czasowego oraz regresja krokowa). Nie stanowiły jednak konkurencji dla najlepszych metod prognostycznych, takich jak sztuczne sieci neuronowe oraz regresja wektorów nośnych.

## 2.8. Wyniki eksperymentów

Mając na uwadze czytelność prezentowanych wyników, ustalone zostały następujące oznaczenia:

- Z-24, Z-1 (prognoza naiwna),
- Fg (prognoza losowa),

- L\_f (krokowa postępująca regresja liniowa),
- L\_b (krokowa wsteczna regresja liniowa),
- ARIMA (model szeregu czasowego),
- KNN (regresja k-najbliższych sąsiadów),
- RPART (drzewa regresyjne),
- RF (regresyjne lasy losowe),
- NNET (sztuczne sieci neuronowe),
- SVR (regresja wektorów nośnych).

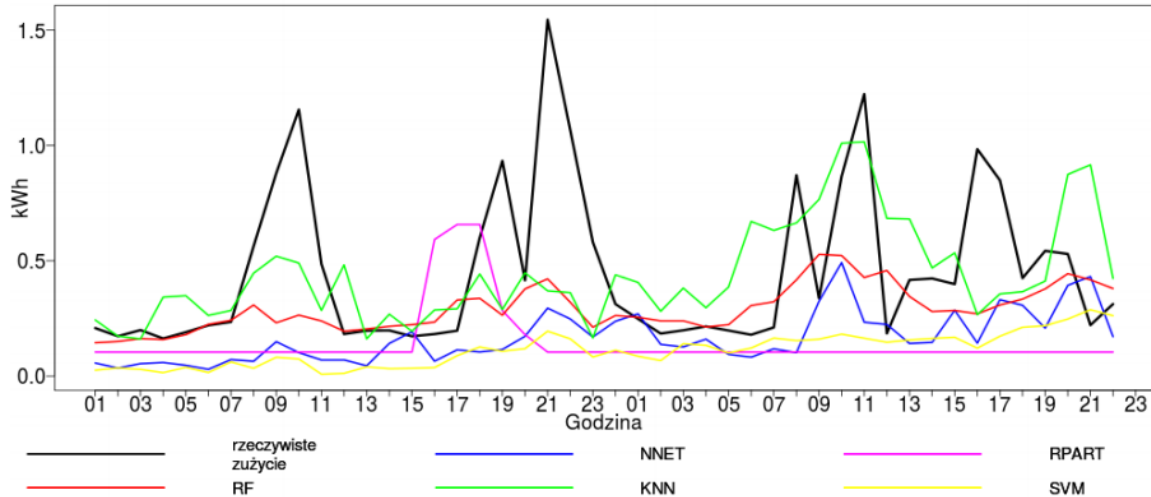
Wyniki prognoz dla modeli z horyzontem prognostycznym wynoszącym 24 godziny, zaprezentowane są w poniższej tabeli. Dla zbioru testowego średni absolutny błąd procentowy oraz odporny średni absolutny błąd procentowy dla RPART wynoszą odpowiednio 77,49% i 68,90%, dla NNET po 58,31%, dla SVR 72,43% i 72,83%, dla RF 42,77% i 40,35% oraz dla KNN 67,04% i 54,63%. W niektórych przypadkach, zaobserwować można mniejsze wartości odpornych błędów procentowych, mogących świadczyć o występowaniu prognoz w dużym stopniu niedokładnych, które zaburzać mogą oczekiwaną wartość zwykłego błędu procentowego. Sytuacja ta szczególnie dotyczy metod porównawczych. Miara Accuracy, która mierzy ile poprawnych wskazań dokonał model, wynosi: dla regresji k-najbliższych sąsiadów 45,65%, dla drzewa regresyjnego 77,49%, dla regresyjnych lasów losowych 42,77%, dla sieci neuronowych 47,92% oraz dla regresji wektorów wspierających 33,33%. Błąd średni kwadratowy, informujący jak bardzo wartość prognozy różni się od wartości rzeczywistej wynosi odpowiednio: dla RPART 0,23, dla NNET 0,19, dla SVR 0,22, dla RF 0,17 oraz 0,13 dla KNN.

Model	MAPE (%)	r_MAPE (%)	Acc (%)	MSE	MAPE (%)	r_MAPE (%)	Acc (%)	MSE	MAPE (%)	r_MAPE (%)	Acc (%)	MSE
	Próba ucząca				Próba walidacyjna				Próba testowa			
Z <sub>24</sub>	117,33	61,62	55,51	0,24	85,55	57,20	57,85	0,16	77,87	45,59	47,92	0,20
F <sub>g</sub>	152,30	74,05	53,13	0,25	70,62	54,12	57,02	0,12	132,87	66,68	45,83	0,29
ARIMA	79,97	71,86	64,58	0,10	75,78	69,50	47,11	0,09	56,05	56,02	50,00	0,13
L_f	99,39	89,47	48,07	0,10	90,28	87,05	27,27	0,11	136,17	130,18	8,33	0,37
L_b	103,64	93,98	47,47	0,11	78,31	73,73	42,15	0,10	59,29	59,38	47,92	0,15
KNN	73,33	57,63	61,76	0,09	75,12	71,00	40,50	0,07	67,04	54,63	45,65	0,13
RPART	59,22	42,98	67,41	0,07	57,51	51,05	61,98	0,11	77,49	68,90	47,92	0,23
RF	1,10	0,97	100,00	0,00	33,83	33,22	72,73	0,07	42,77	40,35	58,33	0,17
NNET	52,84	44,00	73,51	0,03	45,00	43,13	67,77	0,08	58,31	58,31	50,00	0,19
SVR	47,21	45,38	72,17	0,14	44,06	44,06	64,46	0,10	72,43	72,83	33,33	0,22

Rozpatrując każdy z wydzielonych podzbiorów (uczący, walidacyjny, testowy) oraz posługując się błędem MAPE, stwierdzić można, że opracowane modele uczenia maszynowego wykazują się dobrą stabilnością. Tylko lasy losowe, ze względu na swoją naturę (modele składowe budowane są do maksymalnych rozmiarów), wykazują się zbytym dopasowaniem do zbioru uczącego.

W celu porównania w formie graficznej jakości proponowanych metod prognostycznych, utworzono poniższy rysunek.





Rysunek przedstawia rezultaty prognoz dla dwóch dni testowych dla metod wykorzystujących algorytmy uczenia maszynowego. W przypadku podstawowej wersji modeli prognostycznych, zaobserwować można, iż linie obrazujące wartości prognozowane, przebiegają zazwyczaj w gładszy sposób (oprócz regresji k-najbliższych sąsiadów) oraz charakteryzują się niedoszacowaniem w stosunku do wartości rzeczywistych.

Wyniki prognoz dla modeli podstawowych z horyzontem prognostycznym wynoszącym jedną godzinę, zaprezentowano w poniższej tabeli. W przypadku zbioru testowego najmniejszą wartość średniego absolutnego błędu procentowego oraz odpornego średniego absolutnego błędu procentowego, odnotowano dla metody naiwnej wykorzystującej jako prognozę wartość z godziny poprzedniej. Zauważyć należy, że generowane błędy dla tego rodzaju prognoz, w przypadku modeli opartych o algorytmy uczenia maszynowego, są mniejsze niż dla prognoz z horyzontem dobowym.

Model	Próba ucząca				Próba walidacyjna				Próba testowa			
	MAPE (%)	r_MAPE (%)	Acc (%)	MSE	MAPE (%)	r_MAPE (%)	Acc (%)	MSE	MAPE (%)	r_MAPE (%)	Acc (%)	MSE
Z <sub>-1</sub>	51,10	33,22	70,83	0,14	46,36	33,94	64,46	0,07	33,11	24,79	71,74	0,11
F <sub>g</sub>	144,98	71,04	54,61	0,25	80,79	60,48	47,11	0,18	145,20	54,43	69,57	0,25
ARIMA	78,74	71,40	65,03	0,10	85,51	58,86	53,72	0,16	129,39	103,67	34,78	0,12
L <sub>f</sub>	93,63	80,23	53,72	0,09	114,01	80,55	40,50	0,14	68,95	57,59	58,70	0,08
L <sub>b</sub>	92,53	76,14	54,76	0,10	72,04	62,98	48,76	0,10	53,06	46,96	69,57	0,08
KNN	67,70	55,45	59,23	0,09	69,25	59,36	48,76	0,12	53,42	48,14	63,04	0,08
RPART	80,06	70,85	52,08	0,10	80,49	67,95	47,11	0,12	63,42	63,14	65,22	0,08
RF	0,85	0,64	100,00	0,00	50,24	42,79	58,68	0,09	43,41	39,49	73,91	0,07
NNET	44,58	44,51	67,71	0,17	42,21	42,21	57,85	0,16	40,42	40,42	67,39	0,11
SVR	49,40	45,88	70,53	0,12	52,63	50,21	52,89	0,13	44,13	44,13	73,91	0,09

Modelem charakteryzującym się najmniejszym błędem MAPE oraz najlepszą stabilnością, okazały się sztuczne sieci neuronowe (40,42%). Najmniejszy odporny błąd MAPE oraz największą wartość jakości Acc uzyskały regresyjne lasy losowe (odpowiednio 39,49% oraz 73,91%). W odniesieniu do modeli opartych o algorytmy

uczenia maszynowego, nie występują w tym przypadku drastyczne różnice pomiędzy błędem MAPE a r-MAPE, świadczące o występowaniu prognoz w dużym stopniu niedokładnych.

## 2.9. Optymalizacja autonomicznego układu zasilania

Dzięki coraz powszechniejszemu zastosowaniu systemów zasilania energią odnawialną z zastosowaniem generatorów fotowoltaicznych (PV) i turbin wiatrowych (TW) jest możliwość zaimplementowania tych układów do zasilania w sposób autonomiczny. Bardzo istotnym czynnikiem wpływającym na efektywność wielosystemowych układów zasilania jest odpowiednie zbilansowanie całego systemu. Opisanie zostały analizy dokonane na podstawie rzeczywistych danych pomiarowych z zastosowaniem współpracujących układów turbin wiatrowych, paneli fotowoltaicznych oraz magazynu wyprodukowanej energii elektrycznej.

Istotnym problemem źródeł generowania energii słonecznej oraz wiatrowej jest ich zmienność w czasie oraz w przypadku turbin wiatrowych duża stochastyczność i zmienność. W celu rozwiązania omawianego problemu należy do hybrydowego systemu fotowoltaiczno-wiatrowego zaimplementować system gromadzenia nadwyżek produkowanej energii. Prowadzone badania wykazują znaczący postęp w dziedzinie prognozowania uzysku energetycznego zarówno dla fotowoltaiki, jak i turbin wiatrowych. Jednak sama wiedza, że w przeciągu następných godzin będzie dostępna określona wielkość energii elektrycznej nie jest wystarczająca w wypadku układów autonomicznych, ponieważ funkcją celu jest pokrycie całkowitego występującego zapotrzebowania na energię w badanych źródłach. Badania wskazują w szczególności na częściową komplementarność źródeł słonecznych i wiatrowych. Połączenie koncepcji magazynowania energii oraz wykorzystania czasowej komplementarności różnych źródeł energii stało się przyczynkiem do rozwoju tak zwanych układów hybrydowych i skojarzonych. Obecne kierunki prac koncentrują się głównie na wielokryterialnej optymalizacji parametrów takich układów.

### 2.9.1. Układ zasilania

W analizowanym układzie można wyróżnić dwa źródła energii jakimi są, instalacja fotowoltaiczna (PV) i turbina wiatrowa (TW). Ze względu na mogące wystąpić rozbieżności w podaży i popytu energii, układ został wyposażony w magazyn energii. Podstawowym zadaniem obliczeniowym, związanym z ustaleniem wartości zmodyfikowanej funkcji celu w rozpatrywanym zadaniu optymalizacji struktury elektrowni solarno-wiatrowej, jest wyznaczenie energii generowanej w układzie w okresie jednego roku. W przypadku posiadania danych pomiarowych prędkości wiatru oraz gęstości mocy promieniowania słonecznego, energię elektryczną generowaną w omawianym układzie hybrydowym, wyznaczamy z zależności:

$$A_e = \sum_{n=1}^{N_1-1} \frac{P_1(u_{w(n+1)}) + P(V(n))}{2} \Delta t_w + =$$

$$\sum_{n=1}^{N_1-1} \frac{P_2(u_{w(n+1)}) + P(E(n))}{2} \Delta t_{PV}$$

$\Delta t_{PV}$  - krok czasowy pomiarów gęstości mocy promieniowania  $\Delta t_{tw}$  - krok czasowy pomiarów gęstości mocy energii wiatrowej,  $E_{(n)}$  - wartość gęstości mocy promieniowania,  $P(E_{(n)})$  – moc paneli PV dla próbki pomiarowej n. W zależności uwzględniono uśrednioną wartość mocy turbiny wiatrowej  $P_1$  oraz paneli PV  $P_2$ , odpowiednio w okresach  $\Delta t_w$  i  $\Delta t_{PV}$ , wynikającą z zastosowania liniowej aproksymacji zmian mocy w czasie.

## 2.9.2. Układ zasilania generatorami fotowoltaicznymi

Rodzaj zastosowanych ogniw fotowoltaicznych w dużym stopniu decyduje o nominalnej mocy systemu zasilania, wymiary decydują o zajętej powierzchni oraz masie elektrowni PV. Dlatego ważnym czynnikiem mającym wpływ na skuteczną realizację projektu jest wybór odpowiednich generatorów fotowoltaicznych w celu uzyskania wymaganej mocy elektrycznej koniecznej dla zasilania instalacji i ładowania akumulatora.

Energię promieniowania słonecznego obliczono z zależności:

$$A_e^s = J_{PV} = \sum_{n=1}^{12} P_2(E_{(n)Avg})T_{(n)Sun}$$

gdzie:  $E_{(n)Avg}$  - średnia miesięczna gęstość mocy promieniowania,  $P_2(E_{(n)Avg})$  - moc panelu odpowiadająca gęstości,  $(E_{(n)Avg})T_{(n)Sun}$  - średni czas nasłonecznienia dla miesiąca n.

### 2.9.2.1. Usłonecznienie na wybranym terenie w roku 2018

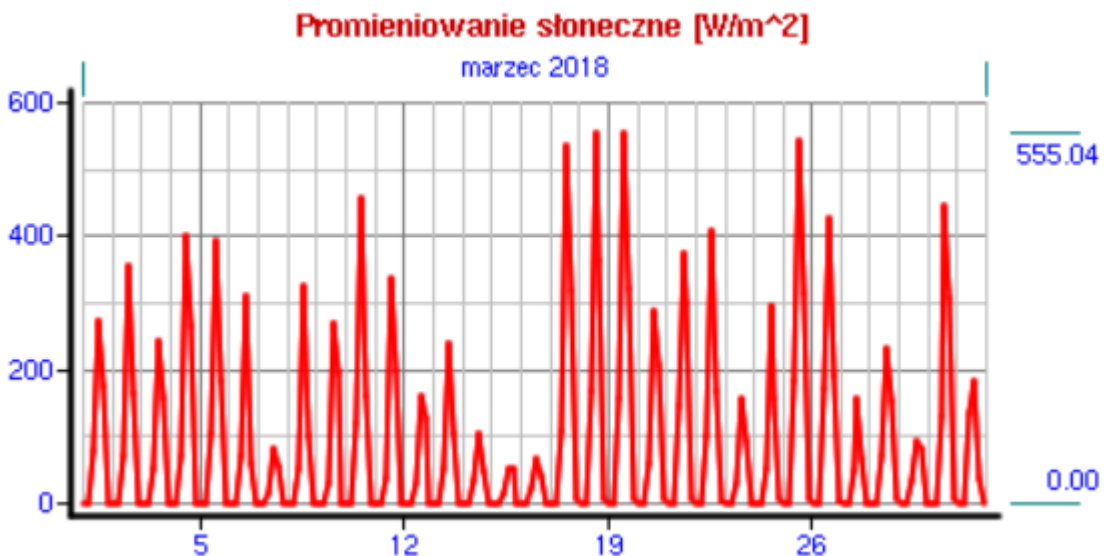
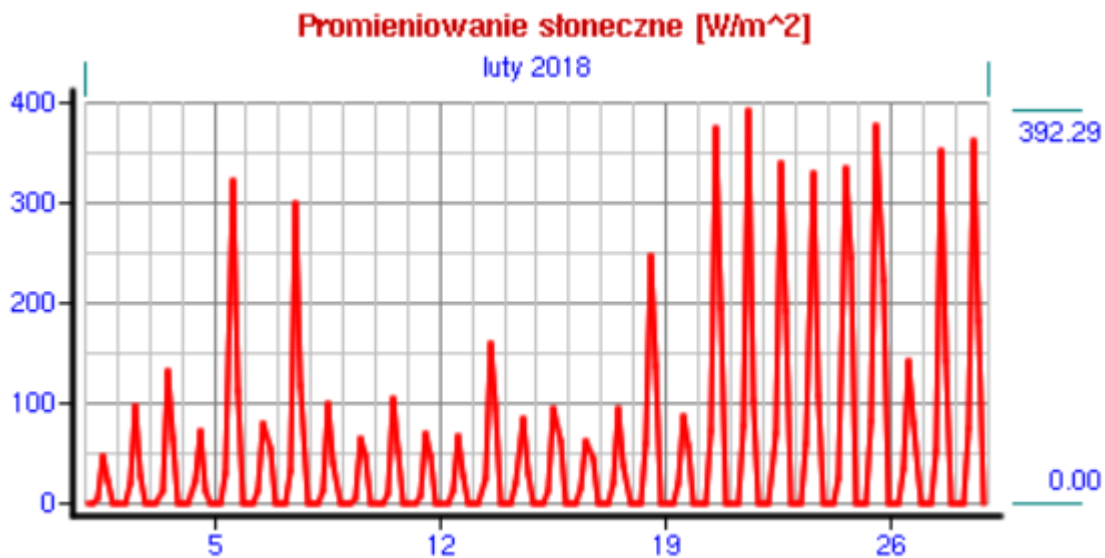
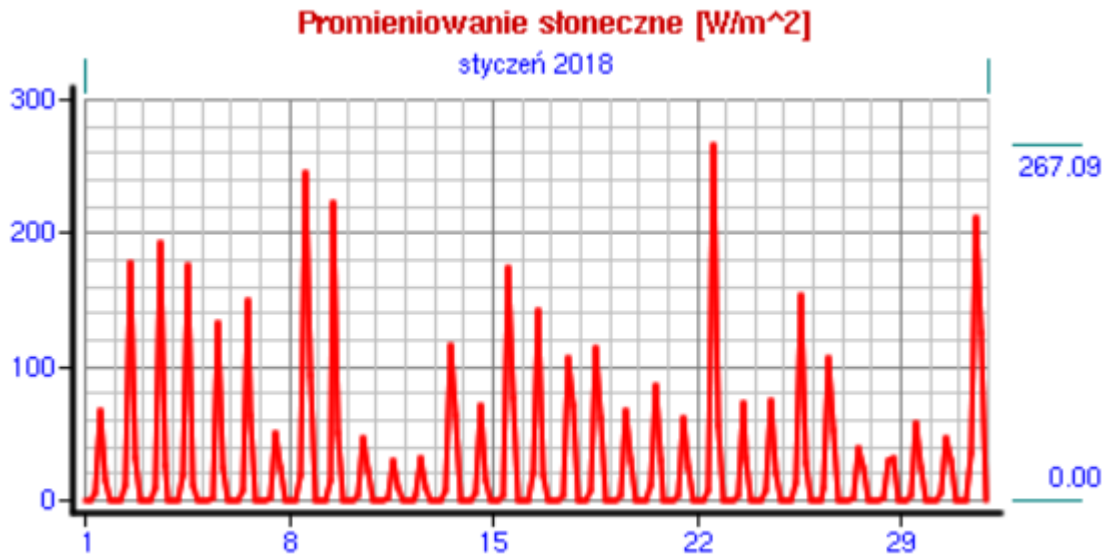
Usłonecznienie jest definiowane, jako liczba godzin słonecznych, czas podany w godzinach, podczas którego na powierzchnię Ziemi padają bezpośrednio promienie słoneczne. Jest to parametr opisujący głównie warunki pogodowe a nie zasoby energii słonecznej. Wykorzystuje się go jednak w energetyce słonecznej do szacowania warunków pracy instalacji.

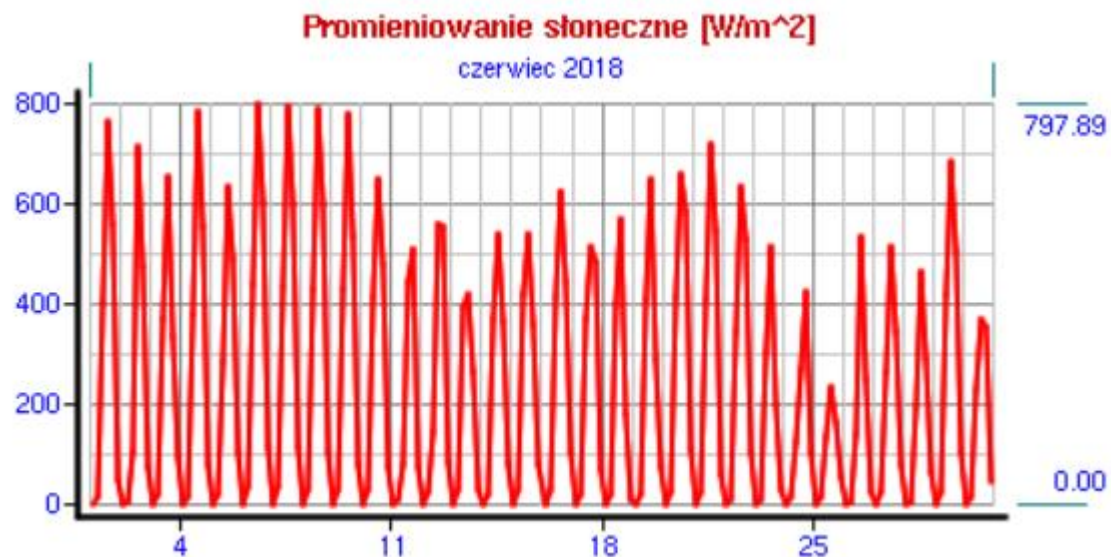
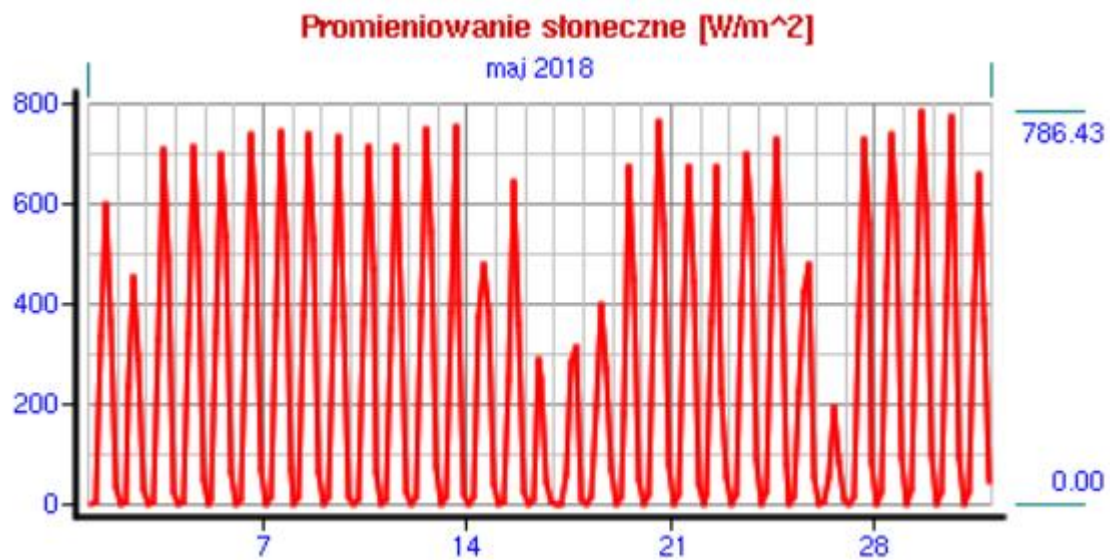
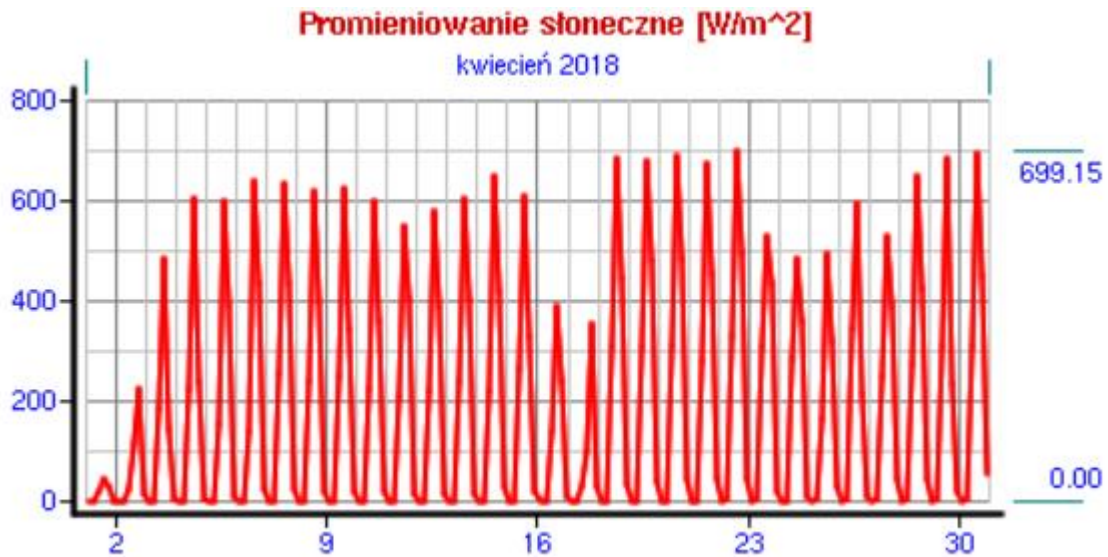
Rok	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	I-XII
2018	37.3	55.7	95.5	238.8	285.2	215.5	184.3	245.1	180.4	131.5	65.6	14.8	1749.7

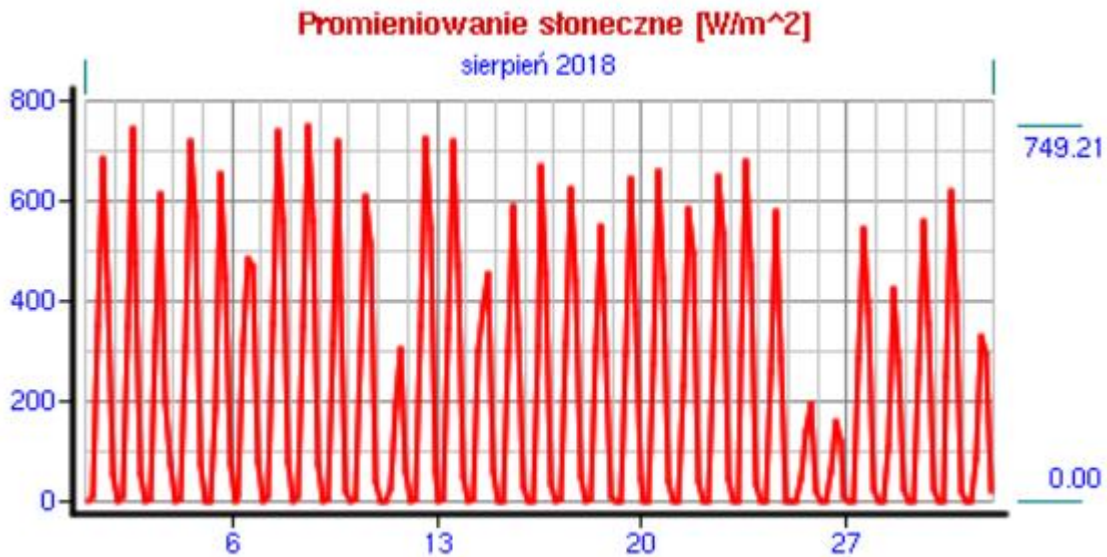
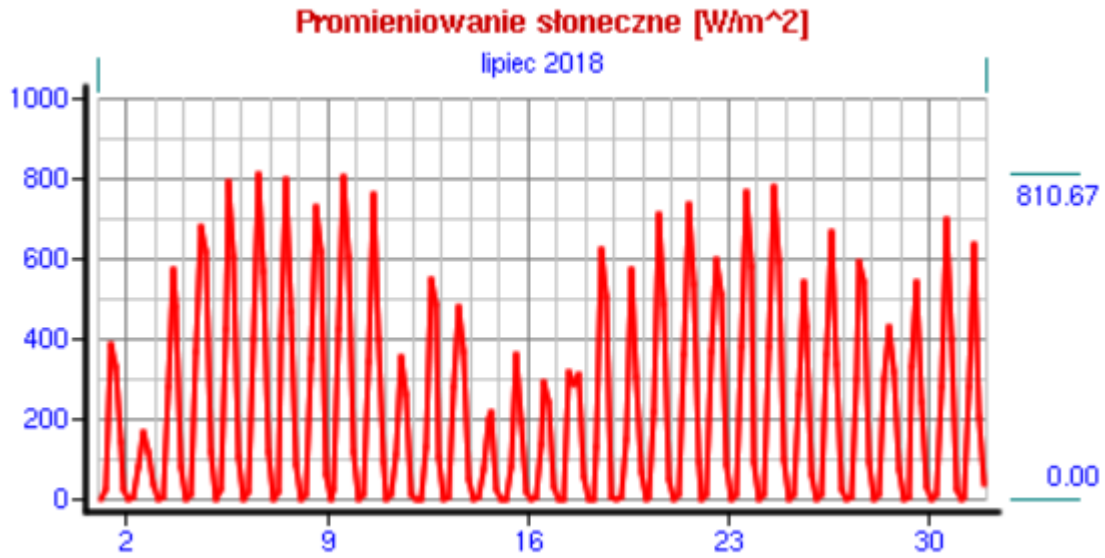
Źródłem pochodzenia danych jest Instytut Meteorologii i Gospodarki Wodnej - Państwowy Instytut Badawczy. Dane Instytutu Meteorologii i Gospodarki Wodnej - Państwowego Instytutu Badawczego zostały przetworzone.

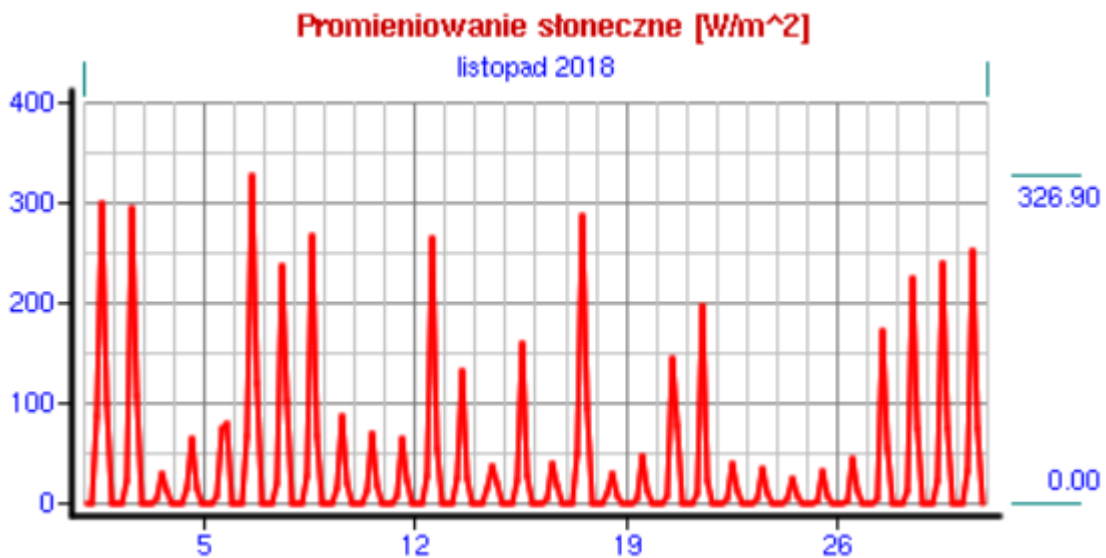
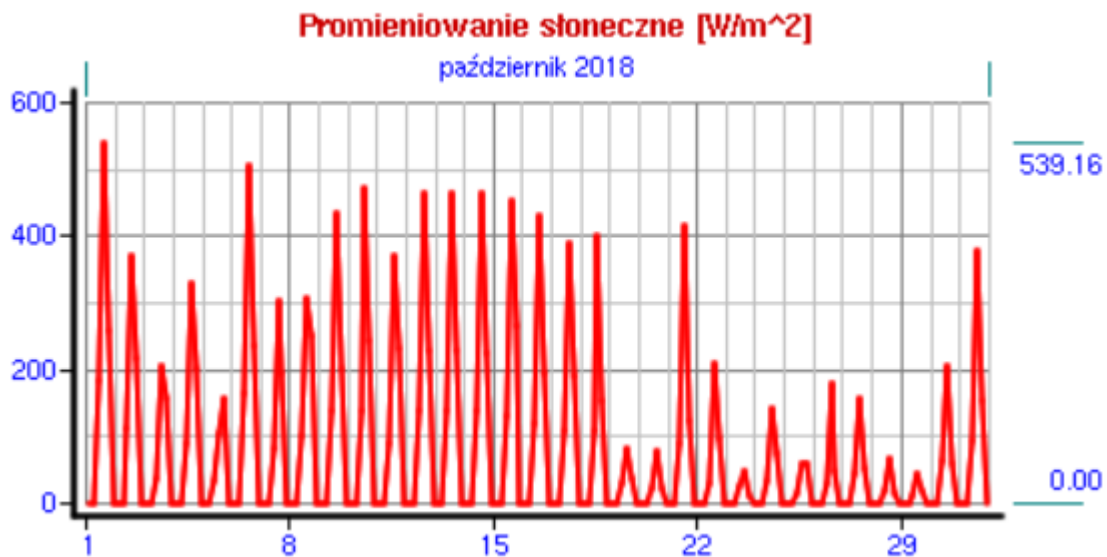
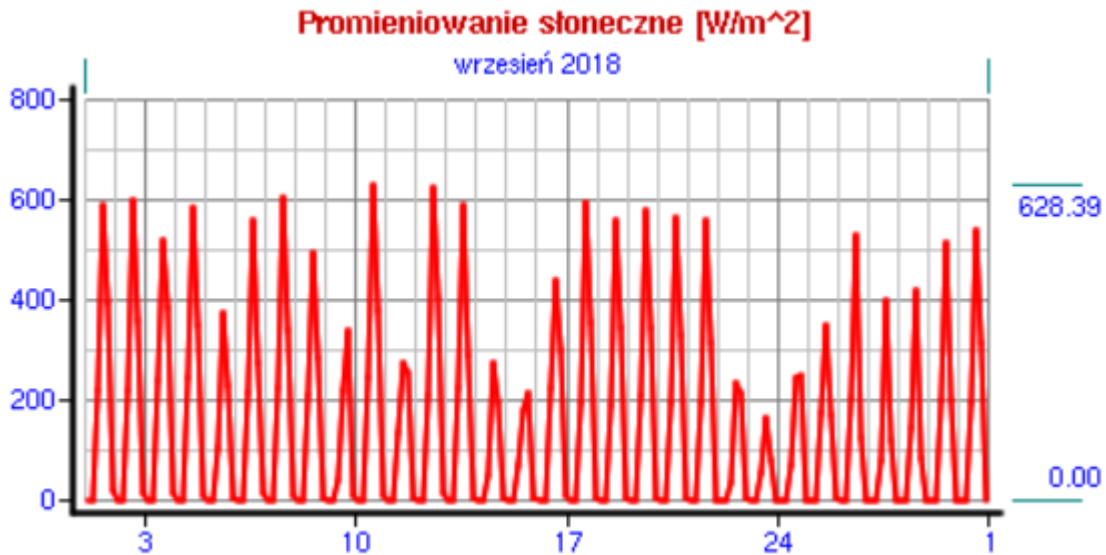
### 2.9.2.2. Dane (miesięczne) promieniowania słonecznego na wybranym terenie w roku 2018

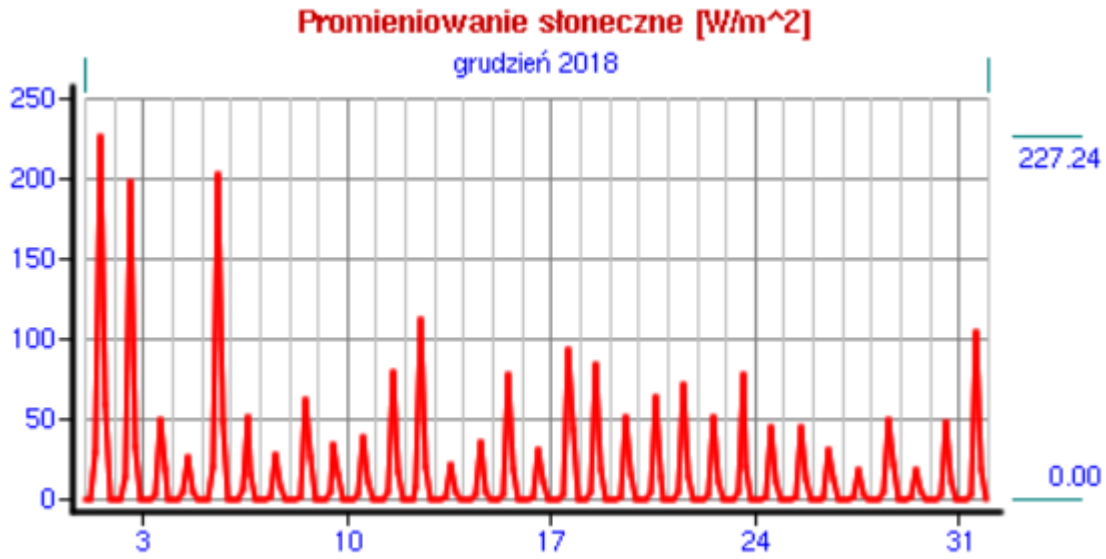
Natężenie promieniowania słonecznego to chwilowa wartość gęstości mocy promieniowania słonecznego padającego w ciągu jednej sekundy na powierzchnię jednego m<sup>2</sup>, prostopadłą do kierunku promieniowania. Wartość ta podawana jest zazwyczaj w [W/m<sup>2</sup>] lub [kW/m<sup>2</sup>].







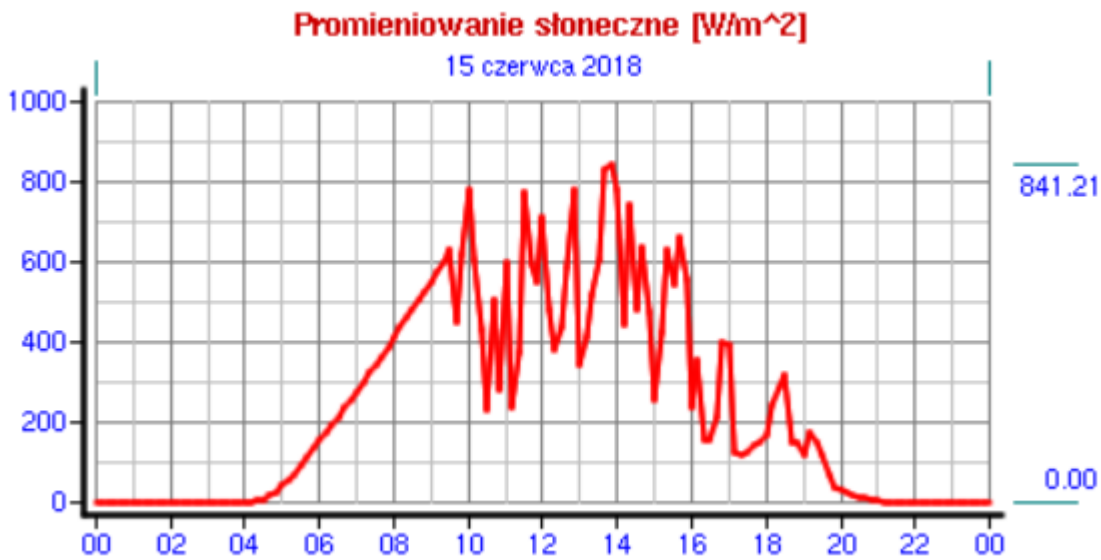






2.9.2.3. Dane (rozkład dobowy) promieniowania słonecznego na wybranym terenie w wybranych (reprezentacyjnych) dniach roku 2018



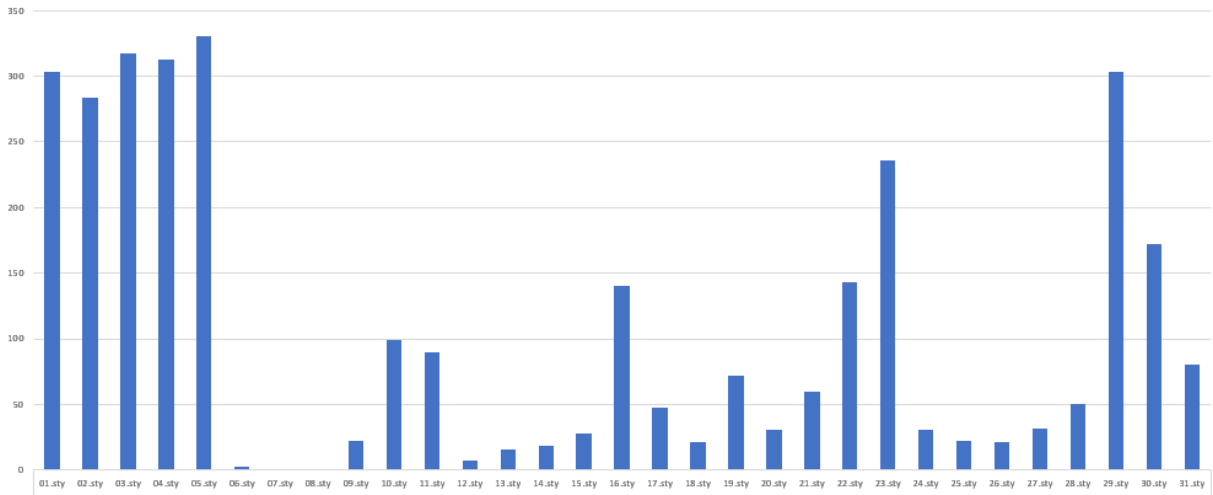




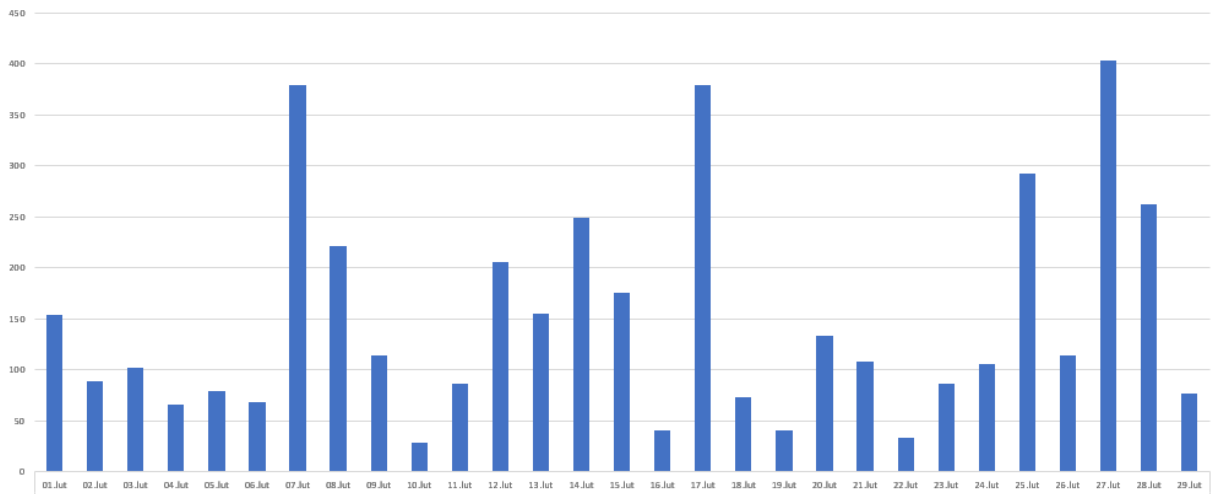
#### 2.9.2.4. Dane (miesięczne) generowania energii przez panel słoneczny na wybranym terenie w roku 2016

Informacje wskazujące dobową sumę generowanej energii w ciągu miesiąca przez panel słoneczny zgodny z założeniami projektowymi.

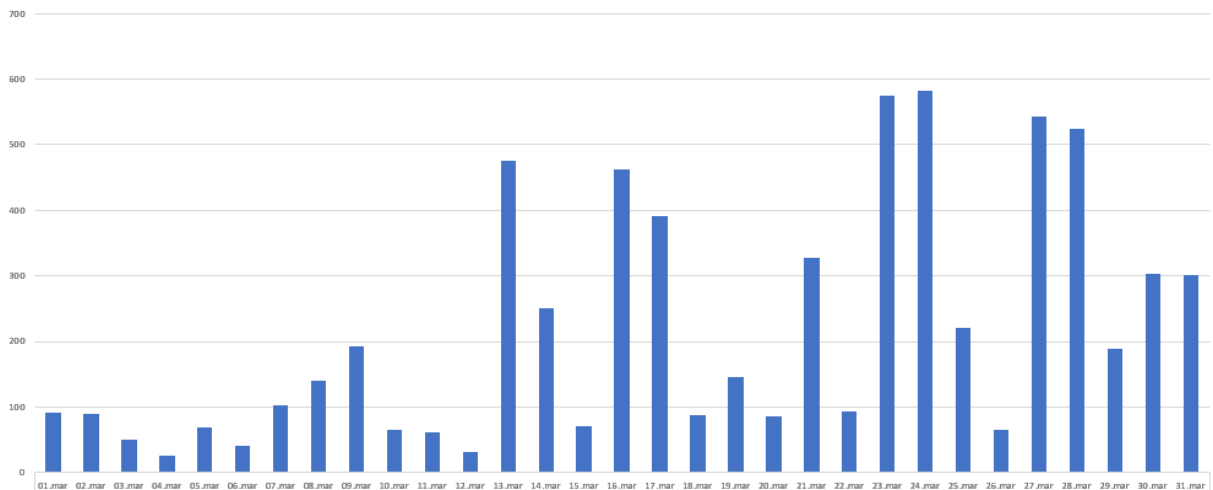
Styczeń 2016



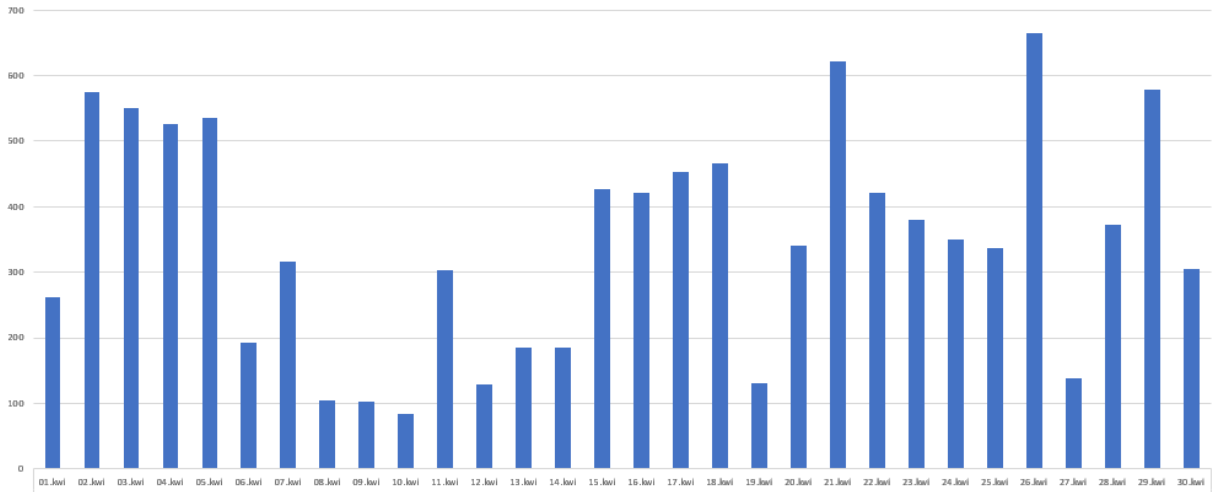
Luty 2016:



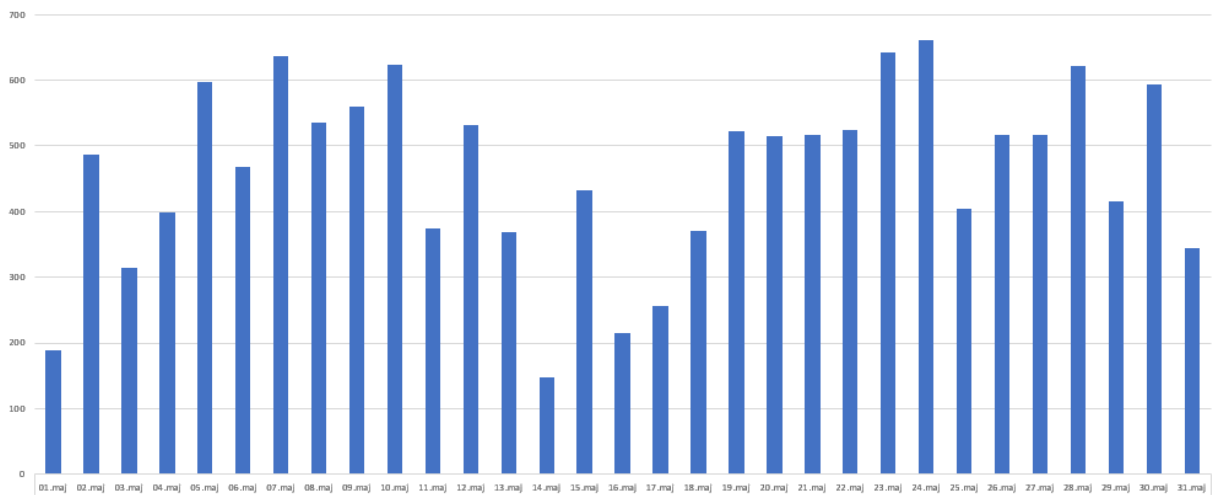
Marzec 2016:



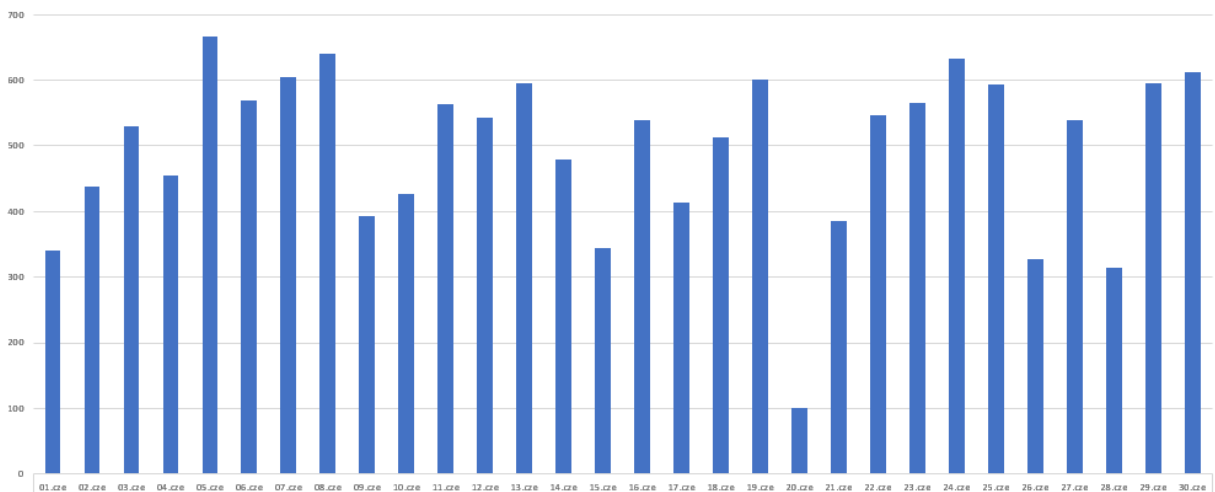
Kwiecień 2016:



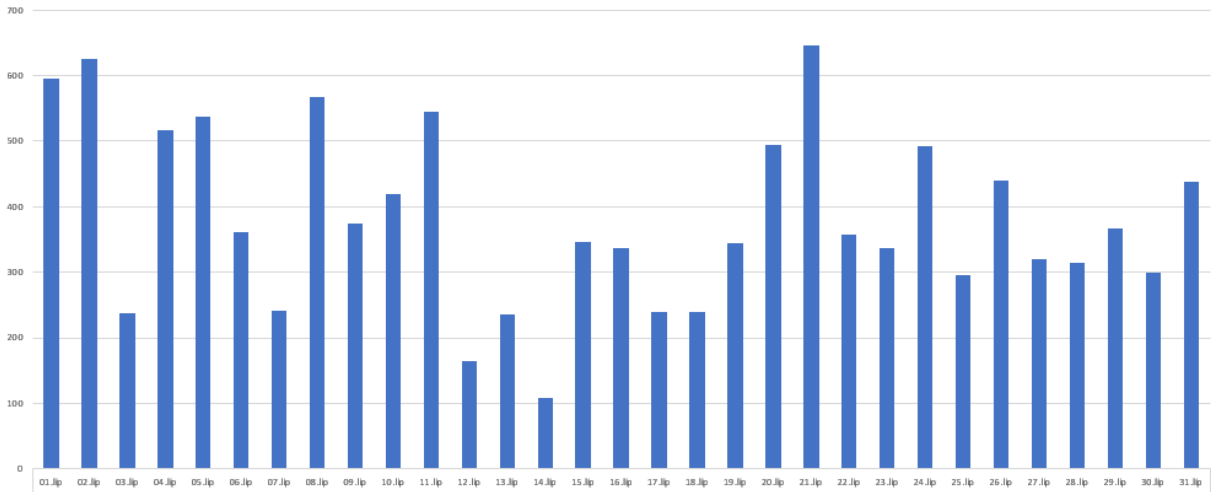
Maj 2016:



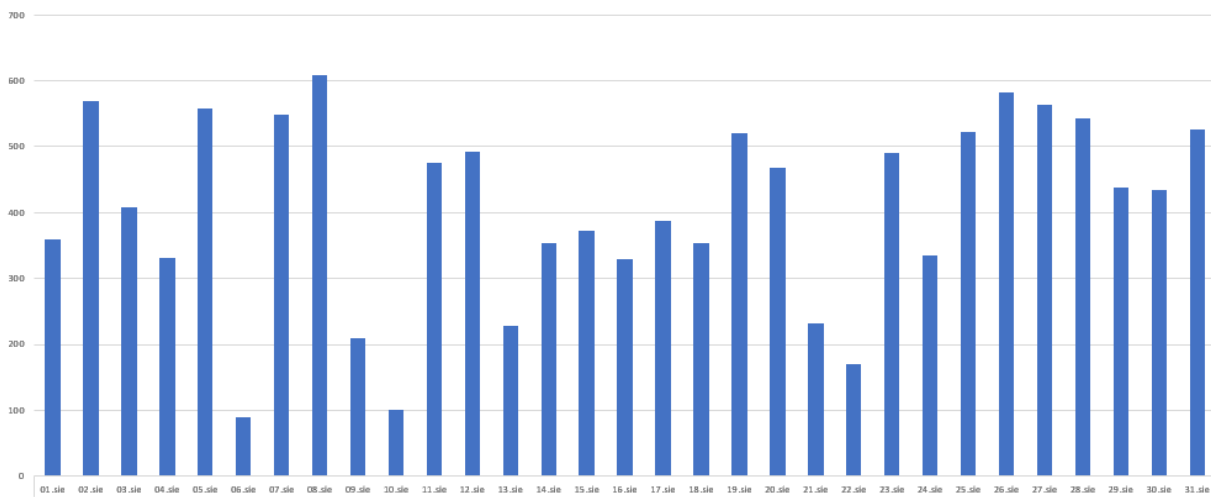
Czerwiec 2016:



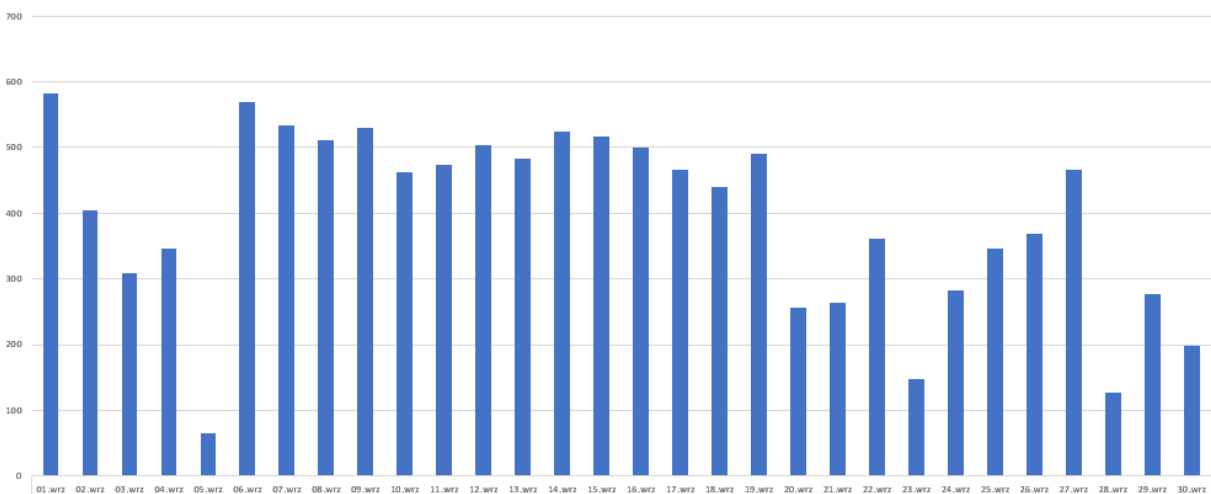
Lipiec 2016:



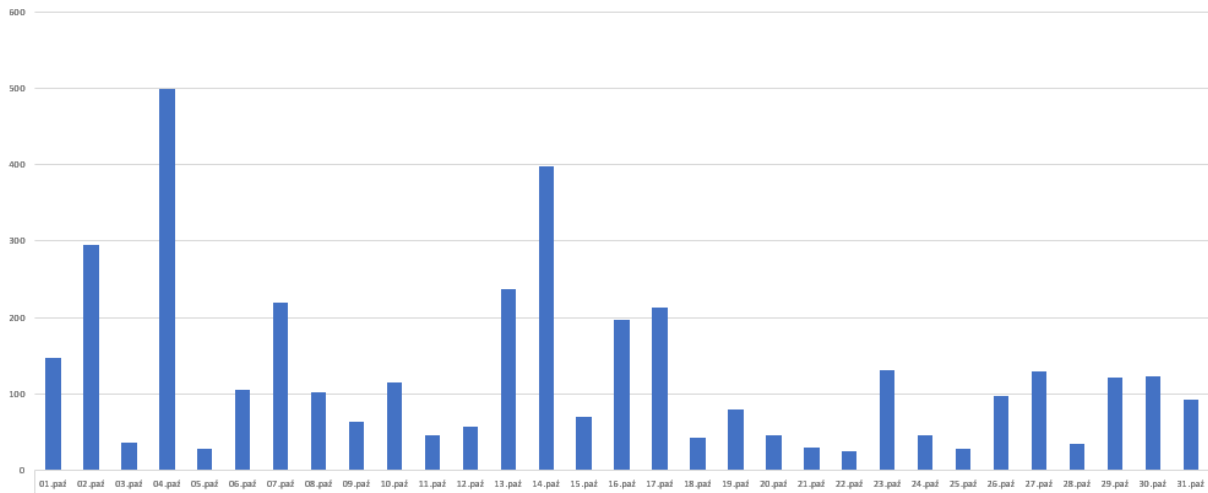
Sierpień 2016:



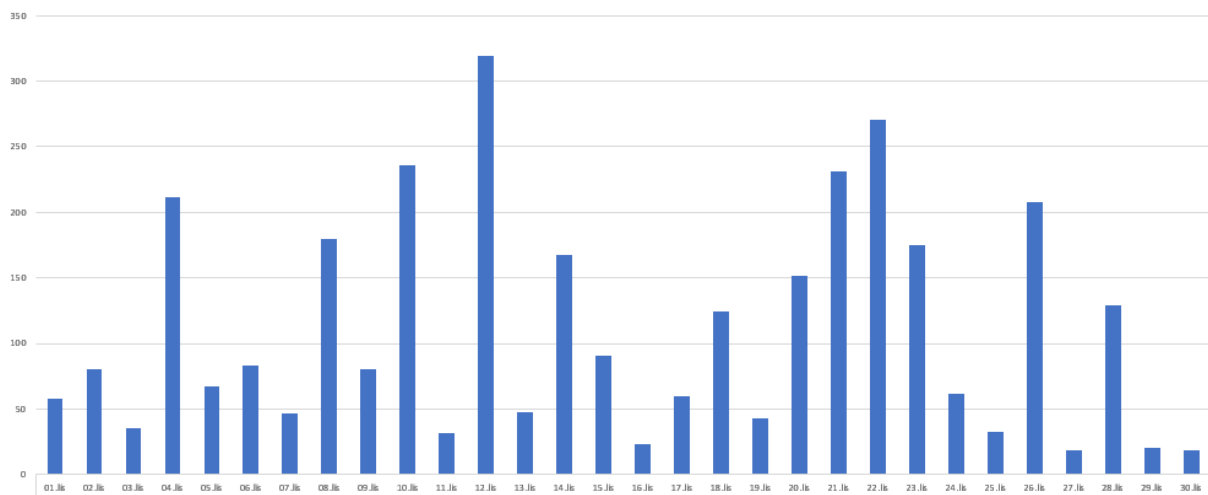
Wrzesień 2016:



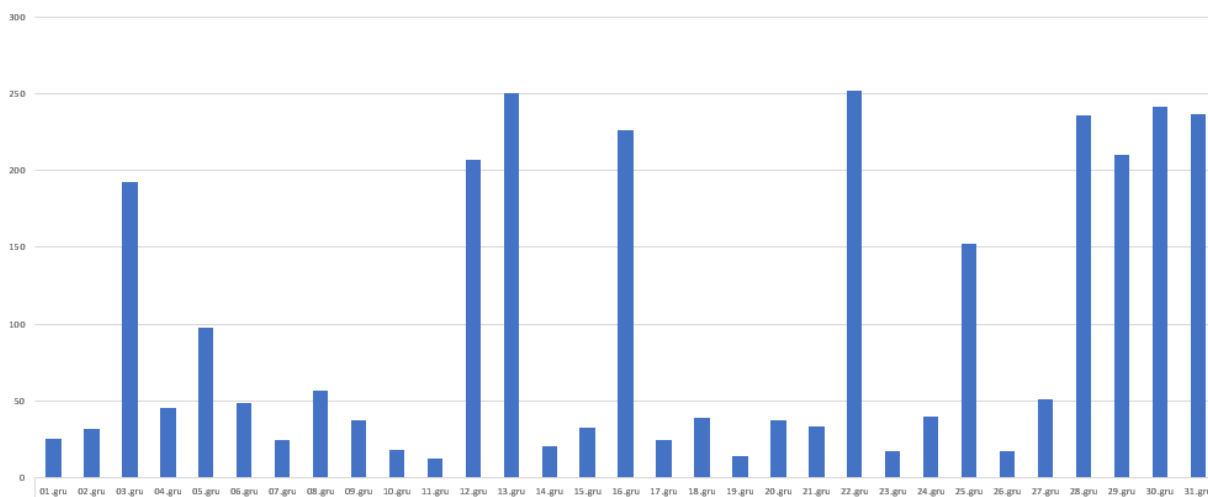
Październik 2016:



Listopad 2016:



Grudzień 2016:



### 2.9.3. Układ zasilania turbinami wiatrowymi

Ze względu na ograniczoną dostępność promieniowania słonecznego w okresach jesienno-zimowych, należy uzupełnić układ hybrydowy o generatory wiatrowe. Wybierając generator wiatrowy należy zwrócić uwagę

Projekt pt. „Prace rozwojowe oraz testy w warunkach rzeczywistych autonomicznego i inteligentnego sterownika” nr Umowy z Województwem Łódzkim, w imieniu którego działa Centrum Obsługi Przedsiębiorcy: **RPLD.01.02.02-10-0006/18-00** realizowany w ramach Poddziałania I.2.2 Regionalnego Programu Operacyjnego Województwa Łódzkiego na lata 2014-2020 współfinansowanego ze środków Europejskiego Funduszu Rozwoju Regionalnego.

głównie na wydajność z jaką elektrownia będzie pracowała najczęściej, czyli przy sile wiatru charakterystycznej dla konkretnego obszaru Polski.

### 2.9.3.1. Średnie prędkości wiatru na wybranym terenie w roku 2018

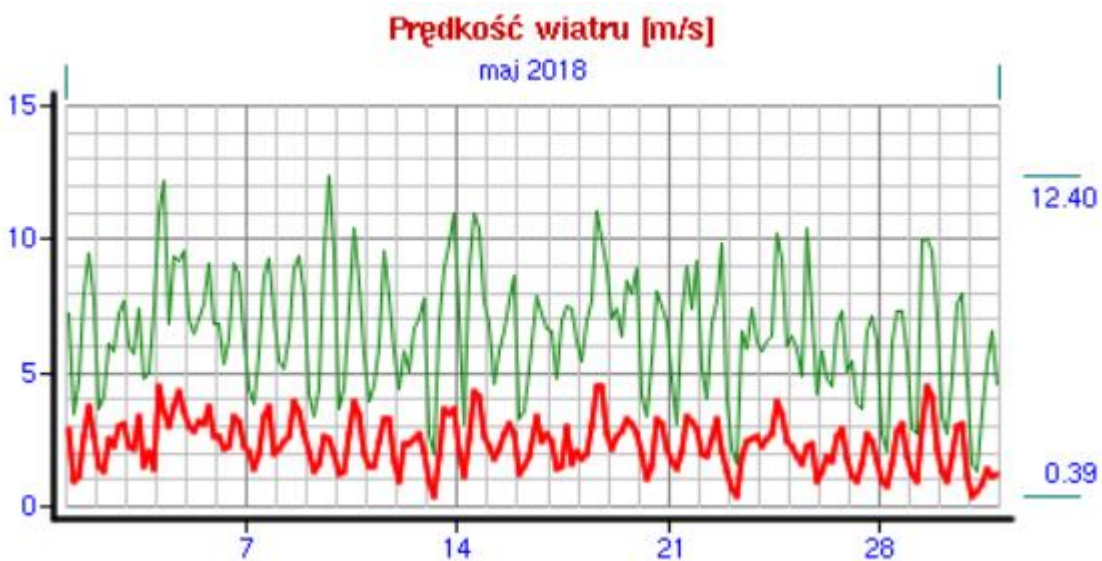
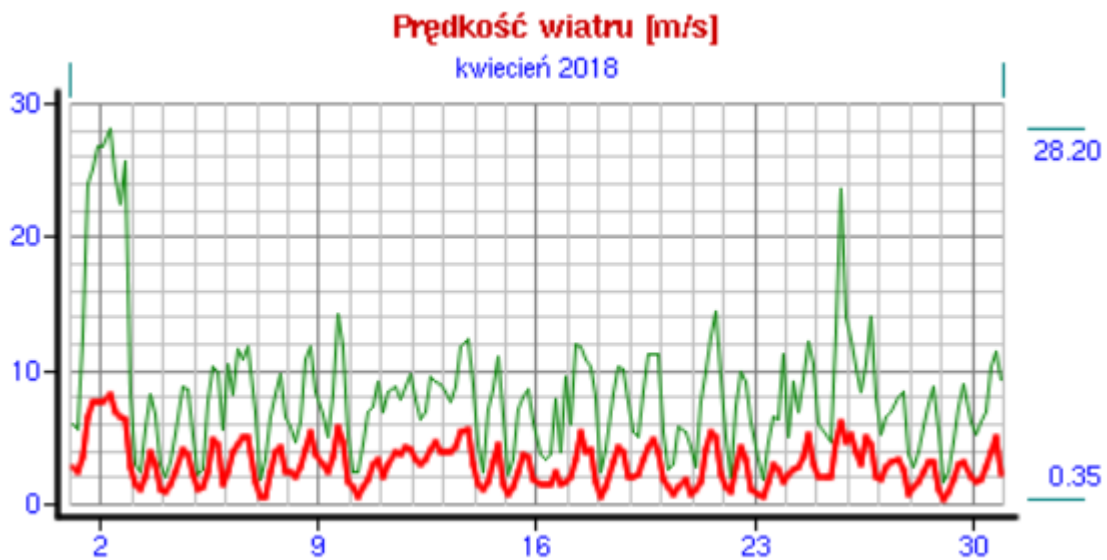
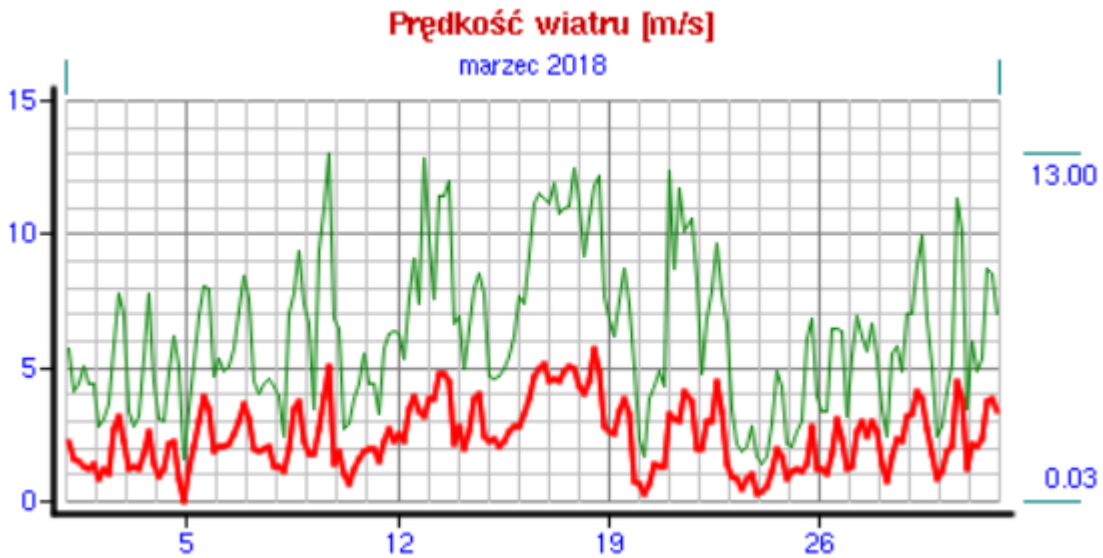
Rok	I	II	III	IV	V	VI	VII	VIII	IX	X	XI	XII	I-XII
2018	3.9	3.1	3.4	3.9	3.7	3.5	3.5	2.7	2.9	3.1	3.6	3.6	3.4

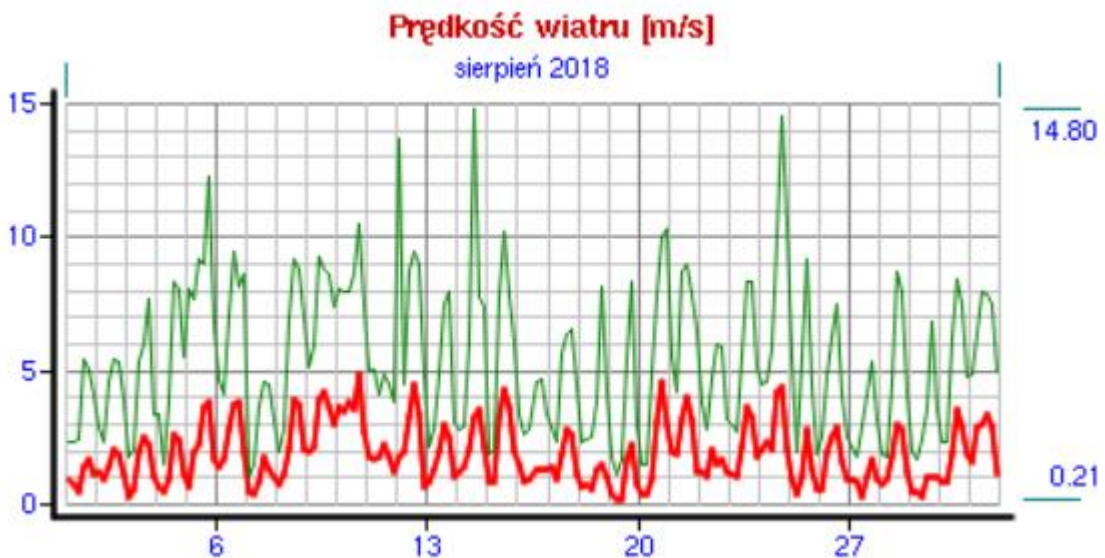
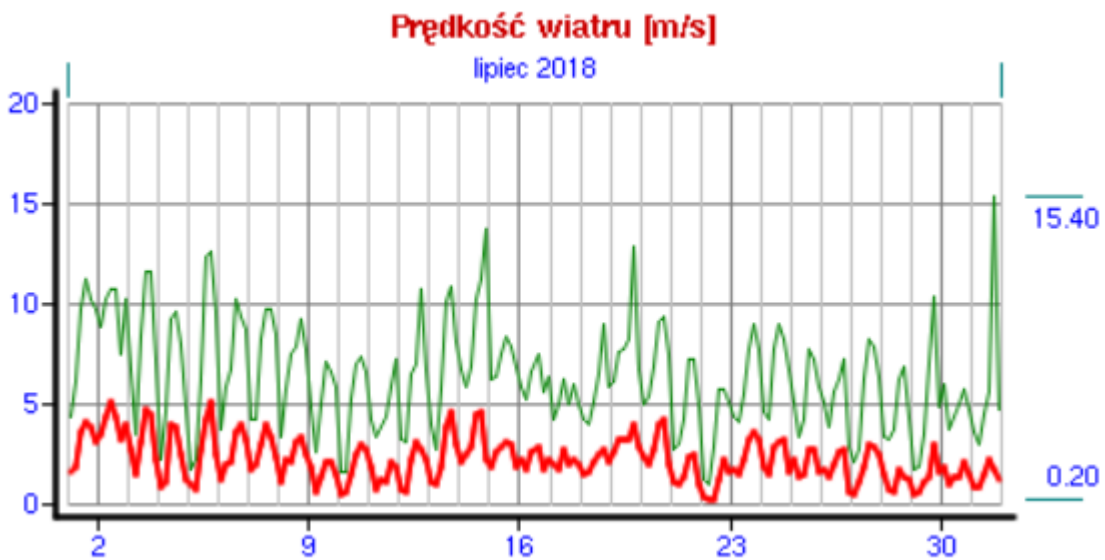
### 2.9.3.2. Dane (miesięczne) średnich prędkości wiatru na wybranym terenie w roku 2018

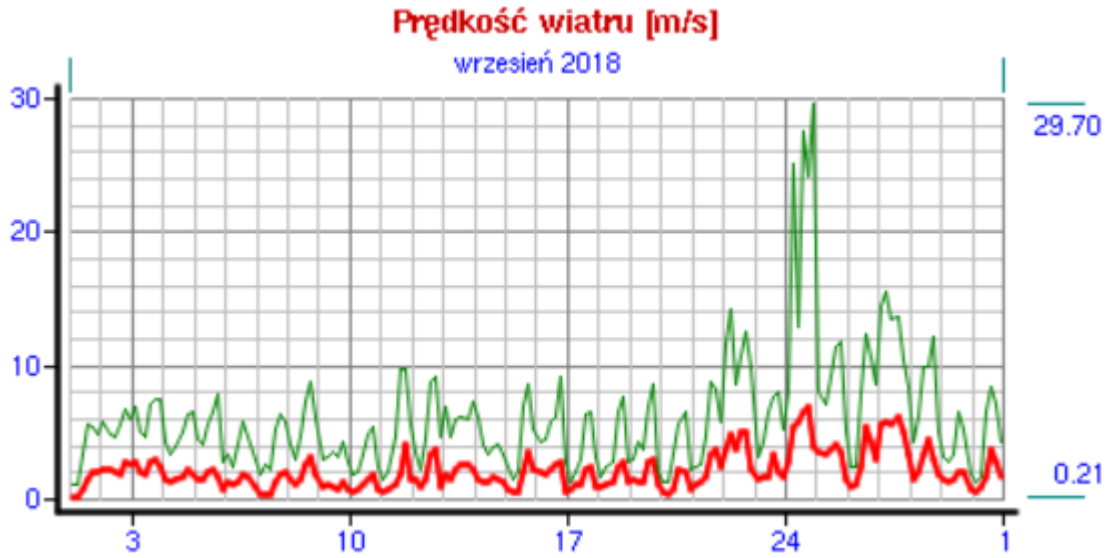
Kolor czerwony na poniższych wykresach wskazuje prędkość wiatru, zaś kolorem zielonym oznaczono prędkość wiatru w porzywie.





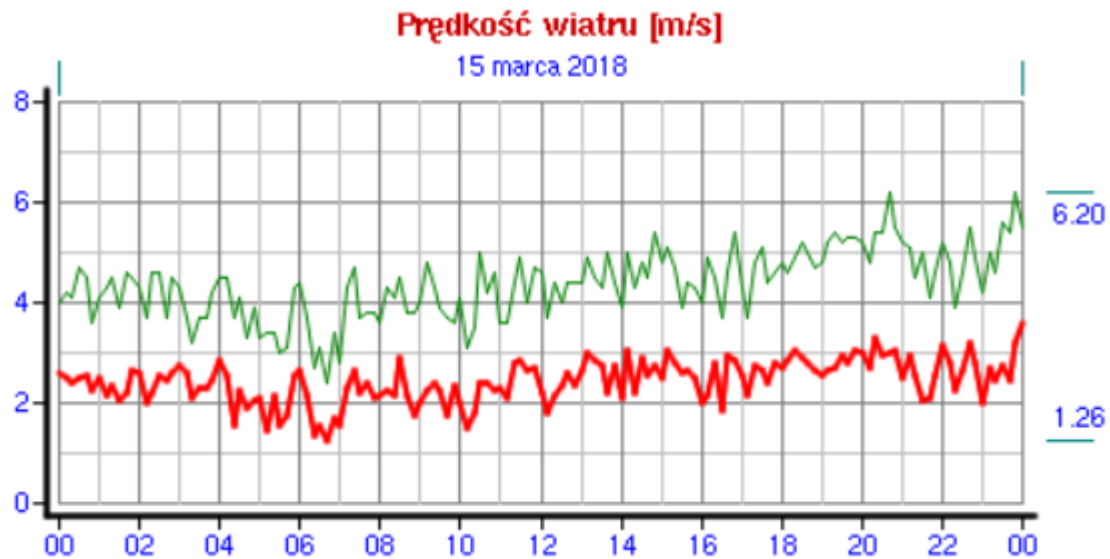


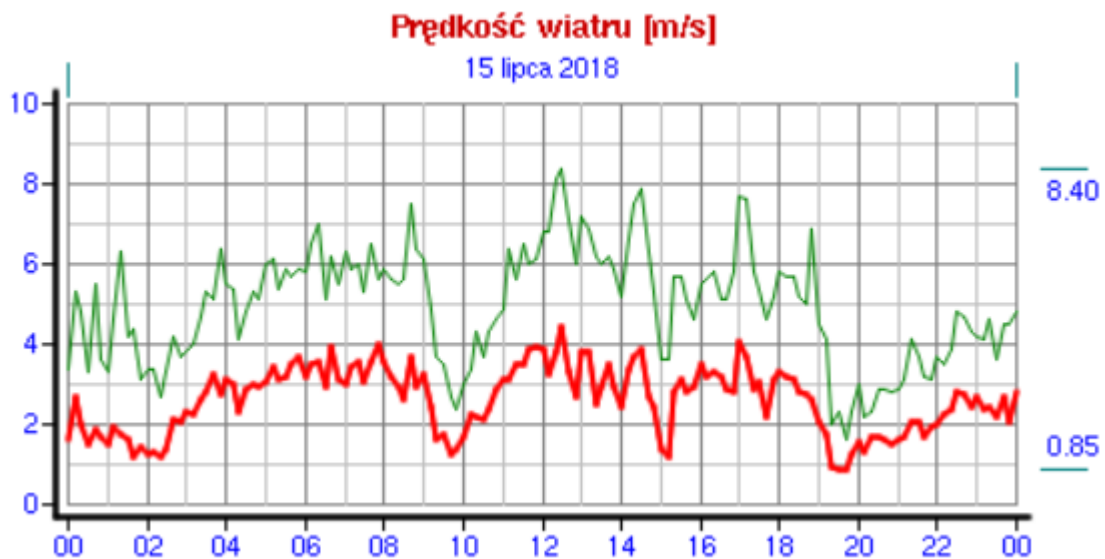






2.9.3.3. Dane (rozkład dobowy) średnich prędkości wiatru na wybranym terenie w wybranych (reprezentacyjnych) dniach roku 2018







#### 2.9.4. Analiza optymalizacji systemu

Roczna ilość generowanej energii elektrycznej dla ustalonej parametrów elektrowni wiatrowej została wyznaczona w oparciu o średnioroczną prędkość wiatru dla danej lokalizacji geograficznej, rozkład gęstości prawdopodobieństwa Weibulla oraz krzywe mocy zastosowanych turbin wiatrowych. W przypadku generatorów PV została wyznaczona wielkość generowanej energii w okresie roku w oparciu o rozkład średniej gęstości mocy promieniowania dla poszczególnych miesięcy w badanej lokalizacji geograficznej oraz charakterystyk prądowo-napięciowych zastosowanych paneli PV. Pokrywanie zapotrzebowania na energię elektryczną wyłącznie w oparciu o pobór energii z magazynu energii występuje tylko w godzinach, gdy teoretycznie nie może być generowana energia z instalacji fotowoltaicznej i generatorów wiatrowych. Należy tak dobrać moc znamionową instalacji fotowoltaicznej ( $P^{PV}$ ) oraz turbinę wiatrową o mocy znamionowej ( $P^{TW}$ ) by ich suma wymiany energii z akumulatorem była minimalna lub zerowa, a jednocześnie prawdopodobieństwo wystąpienia deficytu energii było mniejsze niż dopuszczalne ( $LOLP^{max}$ ). Dla tak zdefiniowanego problemu, możliwe jest zastosowanie

zależności oraz równań przedstawionych na potrzeby modeli symulacyjnych, na tej podstawie model optymalizacyjny przyjmuje następującą postać:

funkcja celu wynosi:

$$\min Z = \sum_{i=1}^m \sum_{j=1}^n E_{ij}^D + \sum_{i=1}^m \sum_{j=1}^n E_{ij}^N$$

$E_{ij}^D$  – deficyt energii

$E_{ij}^N$  – nadwyżka energii

przy ograniczeniu:

$$\frac{\sum_{i=1}^m \sum_{j=1}^n E_{ij}^D}{\sum_{i=1}^m \sum_{j=1}^n E_{ij}^Z} \leq LOLP^{max}$$

$LOLP^{max}$  – prawdopodobieństwo wystąpienia deficytu energii,

$E_{ij}^Z$  – zapotrzebowanie na energię dnia i w chwili, generowane przez odbiory.

Ze względu na zmienną naturę generacji wiatrowej oraz słonecznej, należy oczekiwać, iż sumaryczny uzysk instalacji PV oraz turbiny wiatrowej bardzo rzadko będzie wynosił dokładnie moc znamionową, sytuacja taka jest jednak w rzeczywistości dopuszczalna i możliwa. W związku z tym przyjęto założenie, o nieprzekraczaniu przez sumaryczną moc zainstalowaną w PV oraz TW maksymalnej zdolności do przyjmowania energii przez akumulator. Zmienność zapotrzebowania na energię jest kluczowym czynnikiem wpływającym na planowanie pracy hybrydowego systemu PV-TW. Jak już wspomniano, w wypadku zapotrzebowania na energię, można wyróżnić pewne charakterystyczne jej przebiegi w ujęciu rocznym, tygodniowym, dobowym oraz godzinnym, a tym samym istotną rolę zaczyna odgrywać dostępność odnawialnych źródeł energii w szczególności z perspektywy cyklu dobowego oraz rocznego.

## 2.10. Zaprojektowane wagi dla funkcjonalności przejścia dla pieszych.

- I. Waga 1 krytyczna:
  - a. Sterowanie oświetleniem znaków D-6 i A-16
- II. Waga 2:
  - a. Sterowanie sekcjami oświetlenia głównego przejścia dla pieszych
  - b. Sterowanie natężeniem oświetlenia (0–100%)
  - c. Sygnalizacja stanu systemu (LED)
  - d. Pomiar pozyskanej energii (moc, prąd, napięcie)
  - e. Optymalizacja zużycia energii na poziomie lokalnym sterownika i jego komponentów.
  - f. Awaryjna komunikacja - Komunikat o awarii lub przewidywanym rozładowaniu akumulatora
  - g. Awaryjna komunikacja – Zdalna konfiguracja
- III. Waga 3:
  - a. Komunikacja krótkiego zasięgu



- b. Przycisk wyzwalania oświetlenia
- IV. Waga 4:
- a. Sterowanie oświetleniem chodnika
  - b. Odbieranie i przetwarzanie sygnałów z czujników:
    - i. Pomiar zużytej energii
    - ii. Czujnik oświetlenia
    - iii. Czujnik temperatury
    - iv. Detektor ruchu PIR
- V. Waga 5:
- a. Komunikacja z centrum monitorowania (wstrzymanie, leniwa transmisja, normalna transmisja).  
Dotyczy to zarówno wysyłania jak i odbierania danych, komunikatów, procedur.
  - b. Optymalizacja i prognozowanie zużycia energii na poziomie centrum monitorowania.
  - c. Odbieranie i przetwarzanie sygnałów z czujników :
    - i. Detektor pojazdów/osób
    - ii. Stacja klimatyczna:
      - 1. nasłonecznienie,
      - 2. ciśnienie,
      - 3. siła i kierunek wiatru,
      - 4. poziom hałasu,
      - 5. temperatura,
      - 6. opady,
      - 7. widoczność,
      - 8. zanieczyszczenia powietrza / smog.

### 3. Praktyczne użycie protokołów i algorytmów

Proces budowy modeli predykujących odpowiedzialnych za predykcję określonych wartości jest złożony. Podzielić go można na dwa etapy, pierwszym z nich jest przygotowanie danych, a drugim budowa i ewentualne ich doskonalenie.

Dane z kolei dzielą się na te które są danymi **zależnymi** – takich których konieczne jest predykowanie oraz danymi **niezależnymi**, które w skrócie stanowią dane, w których jesteśmy w posiadaniu i to dzięki nim jesteśmy w stanie „przewidzieć” dane zależne.

Krótko mówiąc istnieje pewna korelacja posiadanymi danymi niezależnymi, a danymi zależnymi. Najprostszym przykładem takiej korelacji jest związek pomiędzy godzinami w dniu robocze, a samochodowym ruchem ulicznym w miastach. Daną zależną w tym przypadku stanowi wielkość ruchu ulicznego, a dane niezależne mające na nią wpływ to godzina (w ciągu doby) oraz dzień tygodnia, nietrudno jednak sobie wyobrazić że na ruch uliczny mają wpływ także inne czynniki takie jak pogoda (w ciepłe, słoneczne dni ludzie chętniej podróżują do pracy np. rowerem) , dni wolne od pracy w postaci świąt itd.

Inną grupę stanowią czynniki losowe, które bardzo trudno przewidzieć, a często mają one bardzo znaczący wpływ na dane zależne. Przykładem takiego wpływu może być epidemia i związane z nią ogólne zmniejszenie ruchu na ulicach. Między innymi ze względu na takie losowe zdarzenia, żaden model przewidujący nie jest w stanie

zapewnić stuprocentowej dokładności.

### 3.1. Przykład poszukiwania danych do budowy modelu.

Analizując historyczne dane jesteśmy w stanie stwierdzić, które z nich są danymi niezależnymi mającymi wpływ na dane które chcemy przewidywać. Dobrym sposobem na przeprowadzenie takiego wywiadu jest utworzenie wykresu z posiadanych danych i wspólne ich naniesienie na wykres. Jeśli tak utworzone krzywe zachowują względnie taki sam odstęp od siebie przez całą długość osi czasu (czy innej odpowiedniej dla nich miary), lub przeciwnie przy spadkach jednej z nich następuje wzrost drugiej w takim przypadku dana ta może być uznana jako kandydata do zmiennej niezależnej, na której będzie opierał się model uczenia maszynowego.



Przykładem takiego wykresu jest zależność temperatury powietrza w st. Celsjusza i nasłonecznienia wyrażonego w  $W/m^2$ . Na osi odciętych natomiast znajduje się tydzień (roku). Dane są pogrupowane tak aby jednej wartości odpowiadała średnia z całego tygodnia. W tym przypadku temperaturę można uznać za kandydata do zmiennej niezależnej, która pozwoli na zbudowanie modelu na jej podstawie.

### 3.2. Budowa modeli uczenia maszynowego

Proces tworzeniem modelu na wcześniej przygotowanych danych polega na ich zaczytaniu do pamięci oraz podziale – na wcześniej wspomniane dane zależne i niezależne. Przy budowie modeli uczenia maszynowego można użyć *Pythona* jako języka programowania wraz z biblioteką *scikit-learn*, do obsługi w tym języku najlepiej użyć środowiska deweloperskiego *Jupyter Notebook*.

Poniżej przedstawiono zrzut ekranu z pierwszej komórki pokazujący sposób zaczytania danych, zaokrąglenie wartości nasłonecznienia do wielokrotności liczby 100 oraz prezentuje pierwsze dziesięć wartości dla których nasłonecznienie przyjmuje wartości różne od zera.

```
In [3]: import numpy as np
import pandas as pd
from sklearn.metrics import accuracy_score, confusion_matrix
from sklearn.ensemble import RandomForestClassifier
from sklearn import svm, tree
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier
from sklearn import linear_model

df = pd.read_csv('NaslonecznienieML.csv')
df = df.drop(columns=['dataczaspomiaru', 'dataczaspomiaru.1', 'monthofyear', 'dayofweek', 'dayofyear'])
df.naslonecznienie = (df.naslonecznienie/100).astype(int)*100
df[df.naslonecznienie != 0].head(10)
```

```
Out[3]:
```

	naslonecznienie	temperatura	hour	weekofyear
59	200	-1.9	11	1
60	100	-1.8	12	1
61	100	-1.7	13	1
62	200	-2.0	14	1
63	100	-2.7	15	1
108	200	2.3	12	1
109	700	2.4	13	1
110	200	2.1	14	1
152	400	-11.7	8	2
153	1200	-9.0	9	2

Tak przygotowane dane mogą posłużyć do budowania modeli na ich podstawie. Ze względu na to iż istnieje wiele różnych algorytmów uczenia maszynowego wybranie jednego (najlepszego), polegać będzie na znalezieniu takiego, który cechował się będzie najniższym współczynnikiem błędów podczas testowania.

Jeden z miarodajnych współczynników błędów to opisany wcześniej średni błąd kwadratowy – *mean squared error* (MSE). Przetestowane algorytmy to odpowiednio:

- Regresja k-najbliższych sąsiadów (*KNN*)
- Klasyfikacja wektorów nośnych (*SVC*)
- Drzewo regresyjne (*Tree*)
- Lasy losowe (*Forest*)
- Regresja wektorów nośnych (*SVR*)

W celu odpowiedniego przetestowania należy otworzyć każdy z obiektów reprezentujących te algorytmy, następnie wytrenować i obliczyć błąd. W celu posiadania punktu odniesienia całość danych podzielona będzie na cztery części. Dzielanie to realizuje się za pomocą wbudowanej metody w bibliotekę *scikit-learn*:

```
In [12]: X = df.drop(columns=['naslonecznienie'])
y = df.naslonecznienie.astype(np.int64)

X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3, random_state=40)
```

Metoda ta jest odpowiedzialna za losowy podział danych na odpowiednie części, jej argumenty to:

- *X* – obiekt zawierający wszystkie dane niezależne
- *y* – obiekt zawierający wszystkie dane zależne
- *test\_size* – wartość która po wymnożeniu przez 100 da procentową wartość jaką stanowią dane testowe (w tym przypadku stosunek danych testowych do uczących wynosi 30/70%)
- *random\_state* – wartość liczbowa odpowiadająca temu w jaki sposób mają zostać podzielone dane, inaczej mówiąc gdybyśmy nie podali tego argumentu to używając tej metody wiele razy na identycznym zbiorze danych otrzymamy za każdym razem różnie podzielone dane, jeżeli wywołamy z taką samą liczbą otrzymamy dokładnie taki sam podział

Zwrócone przez tę metodę obiekty zawierają dane testowe oraz dane do trenowania modeli, obiekty zawierające w nazwie odpowiednio *test* oraz *train*. Dodatkowo te dwie grupy są podzielone na obiekty zawierające dane zależne i niezależne, odpowiednie nazwy rozpoczynające się od *X* oraz *y*.

Tak powstałe obiekty posłużą do wytrenowania obiektów opartych o wszystkie wymienione wcześniej algorytmy. Do tego celu posłużymy się listą tych obiektów, którego nazwa to *classifiers* po której następnie będzie przebiegać iteracja dla trenowania i wyświetlania wielkości błędów.

```
In [24]: classifiers = list()

model1knn = knn = KNeighborsClassifier()
classifiers.append(model1knn)

model2svc = svm.SVC()
classifiers.append(model2svc)

model3tree = tree.DecisionTreeClassifier()
classifiers.append(model3tree)

model4random = RandomForestClassifier()
classifiers.append(model4random)

model5svr = svm.SVR()
classifiers.append(model5svr)
```

Trenowanie i obliczanie błędów dla poszczególnych modeli:

```
In [73]: from sklearn.metrics import mean_squared_error

MSEErrors = list()
for clf in classifiers:
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    MSEErrors.append(mean_squared_error(y_test, y_pred))
print(MSEErrors)

[21288.290867449185, 23252.218723160608, 25247.638133409677, 23853.421127970225, 23100.490302133945]
```

Powyższy fragment zrzutu ekranu przedstawia sposób przeprowadzenia wyliczenia błędu jakim cechuje się konkretny model, kolejność wyników odpowiada wcześniej wymienianym algorytmom. Jak widać najlepszym (o najniższym współczynniku błędu) okazał się algorytm k-najbliższych sąsiadów, a najmniej dopasowanym model oparty o drzewa decyzyjne.

### 3.3. Doskonalenie modeli

Udoskonalenie modeli odbywa się z wykorzystaniem połączenia obu powyższych podpunktów. Doskonalenie, przebiega w dokładnie taki sam sposób jak budowanie opisane budowanie modeli, ale poprzez dodawanie i odejmowanie innych kandydatów do zmiennych niezależnych (otrzymanych zgodnie z algorytmem z punktu o poszukiwaniu danych) i tak na przykład poprzez dodanie do zmiennych niezależnych smogu pomogło obniżyć najlepszy osiągnięty wyniki spośród błędów do wartości poniżej 20 000, jednak pozostałe współczynniki błędów wzrosły, prezentują się one następująco:

```
In [91]: from sklearn.metrics import mean_squared_error

MSEErrors = list()
for clf in classifiers:
    clf.fit(X_train, y_train)
    y_pred = clf.predict(X_test)
    MSEErrors.append(mean_squared_error(y_test, y_pred))
print(MSEErrors)

[24013.29095637099, 25065.010112684195, 23279.399017624964, 19219.878647789657, 29388.16888129996]
```

Ostatecznie najniższe wartości współczynniki błędów spośród wszystkich przebadanych możliwości uzyskano dla danych zależnych obejmujących następującą listę:

- temperatura powietrza [°C]
- wilgotność względna powietrza [%]
- tydzień roku [liczba]
- godzina w ciągu doby [liczba]
- zachmurzenie [oktany]

Najniższe wyniki jakie udało się uzyskać są ponad dwukrotnie niższe niż przy pierwszej próbie – obejmującej jedynie temperaturę, tydzień w ciągu roku oraz godzinę doby. Co ciekawe pomimo wspomnianego wcześniej poprawienia wartości dane obejmujące zanieczyszczenie powietrza w postaci smogu ostatecznie nie znajdują się w danych niezależnych użytych do trenowania modeli. Wyniki dla podanych wyżej danych niezależnych prezentują się następująco:

```
In [29]: from sklearn.metrics import mean_squared_error

MSEErrors = list()
for clf in classifiers:
    clf.fit(X_train, y_train)
    y_pred= clf.predict(X_test)
    acc.append(mean_squared_error(y_test, y_pred))
print(MSEErrors)

Out[29]: [12948.754652161466,
20498.139135413683,
15419.410249069568,
10655.596908101918,
18385.244864364457]
```

Inną miarą jakiej użyto w projekcie jest porównanie oczekiwanych wartości z tymi przewidywanymi. W tym wypadku jest to wartość procentowa mówiąca o tym w ilu przypadkach wartości przewidywane pokrywają się z rzeczywistymi, jest ona głównie używana do sprawdzania modeli opartych o algorytmy klasyfikacji jednak ze względu na to iż przewidywane przez nas wartości nasłonecznienia zostały odpowiednio zaokrąglone na początku działań możemy zastosować tę miarę również w tym przypadku.

Dla przetestowanych 3493 danych otrzymane wartości prezentują się następująco:

Nazwa modelu: ilość poprawnych	Poprawność:
przewidywane_svr: 2412	69,05%
przewidywane_forest: 2959	84,71%
przewidywane_tree: 2790	79,87%
przewidywane_svc: 2739	78,41%
przewidywane_knn: 2919	83,57%

Inne miary to:

- procent fałszywych pozytywnych wyników
- procent fałszywych negatywnych wyników

O fałszywym pozytywnym wyniku mówimy w takim przypadku, gdy w rzeczywistości nasłonecznienie wynosiło 0, a przewidywane wartości były dodatnie, oznacza to że sterownik będzie oczekiwał iż akumulator będzie ładowany energią pozyskiwaną z energii słonecznej, a w rzeczywistości tak się nie stanie.

Znalezienie liczby odpowiadającej fałszywym pozytywnym oraz fałszywym negatywnym wyników polegało na zgromadzeniu wszystkich (rzeczywistych i predykowanych wartości) w jednym obiekcie i iterację po nim. Przykład dodawania przewidzianych wartości:

```
In [91]: model5svr.fit(X_train, y_train)
y_pred= (model5svr.predict(X_test)/100).astype(int)*100

df_predictions = pd.DataFrame({"rzeczywiste": y_test, "przewidywane_svr": y_pred})
```

Po takim dodaniu wartości powstaje obiekt którego pierwsze 15 rzędów wygląda następująco:

```
In [114]: df_predictions.head(15)
Out[114]:
```

	index	rzeczywiste	przewidywane_svr	przewidywane_forest	przewidywane_tree	przewidywane_svc	przewidywane_knn
0	7115	300	0	300	300	0	300
1	7651	0	0	0	0	0	0
2	1801	0	0	0	0	0	0
3	6411	0	0	0	0	0	0
4	1556	0	0	0	0	0	0
5	6631	0	0	0	0	0	0
6	3026	0	0	0	0	0	0
7	2709	0	200	0	0	0	0
8	9377	0	0	0	0	0	0
9	5842	400	200	400	400	300	400
10	1472	0	0	0	0	0	0
11	1350	0	0	0	0	0	0
12	1244	0	0	0	0	0	0
13	7062	0	0	0	0	0	0
14	1328	200	0	200	100	0	100

Są to wymieszane historyczne wartości nasłonecznienia z odpowiadającym im przewidywaniom dla wszystkich omawianych algorytmów.

Poniższa ilustracja przedstawia sposób osiągnięcia liczby fałszywych wyników.

```
In [109]: # obliczanie fałszywych pozytywnych wyników dla poszczególnych modeli

real_neg_value = 0

false_positive_SVR = 0
false_positive_Forest = 0
false_positive_Tree = 0
false_positive_SVC = 0
false_positive_KNN = 0

false_negative_SVR = 0
false_negative_Forest = 0
false_negative_Tree = 0
false_negative_SVC = 0
false_negative_KNN = 0

for row in df_predictions.itertuples():
    if row[2] == 0:
        real_neg_value = real_neg_value+1
        if row[3] != 0:
            false_positive_SVR = false_positive_SVR+1
        if row[4] != 0:
            false_positive_Forest = false_positive_Forest+1
        if row[5] != 0:
            false_positive_Tree = false_positive_Tree+1
        if row[6] != 0:
            false_positive_SVC = false_positive_SVC+1
        if row[7] != 0:
            false_positive_KNN = false_positive_KNN+1
    elif row[2] != 0:
        if row[3] == 0:
            false_negative_SVR = false_negative_SVR+1
        if row[4] == 0:
            false_negative_Forest = false_negative_Forest+1
        if row[5] == 0:
            false_negative_Tree = false_negative_Tree+1
        if row[6] == 0:
            false_negative_SVC = false_negative_SVC+1
        if row[7] == 0:
            false_negative_KNN = false_negative_KNN+1

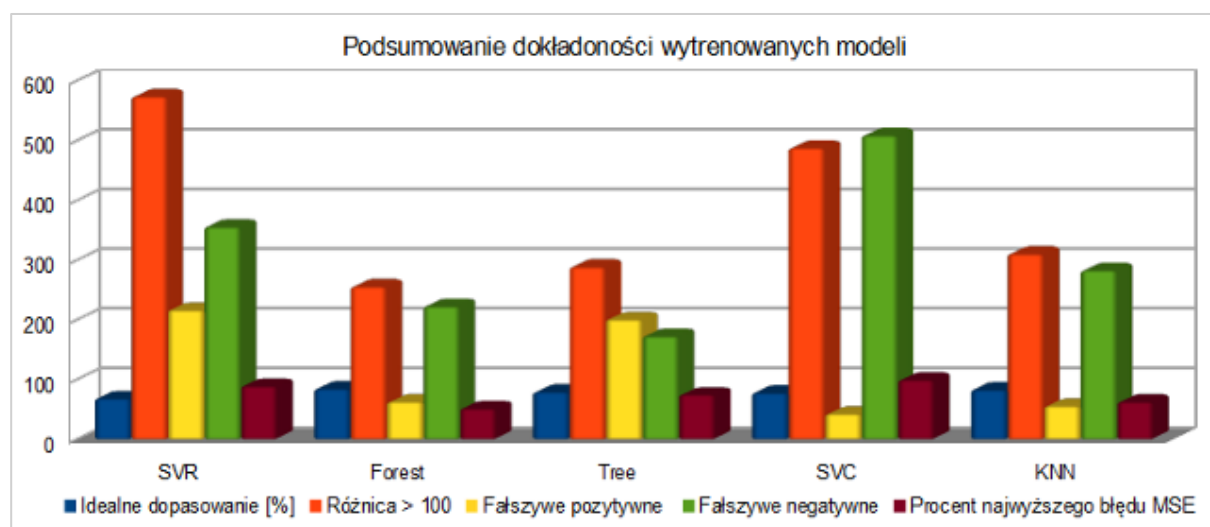
print('Liczba wszystkich danych rzeczywistych danych wynoszących 0: ' + str(real_neg_value))
print('-----')
print("Fałszywie pozytywne wyniki dla SVR:\t" + str(false_positive_SVR))
print("Fałszywie pozytywne wyniki dla Forest:\t" + str(false_positive_Forest))
print("Fałszywie pozytywne wyniki dla Tree:\t" + str(false_positive_Tree))
print("Fałszywie pozytywne wyniki dla SVC:\t" + str(false_positive_SVC))
print("Fałszywie pozytywne wyniki dla KNN:\t" + str(false_positive_KNN))
print('-----')
print("Fałszywie negatywne wyniki dla SVR:\t" + str(false_negative_SVR))
print("Fałszywie negatywne wyniki dla Forest:\t" + str(false_negative_Forest))
print("Fałszywie negatywne wyniki dla Tree:\t" + str(false_negative_Tree))
print("Fałszywie negatywne wyniki dla SVC:\t" + str(false_negative_SVC))
print("Fałszywie negatywne wyniki dla KNN:\t" + str(false_negative_KNN))

Liczba wszystkich danych rzeczywistych danych wynoszących 0: 2546
-----
Fałszywie pozytywne wyniki dla SVR:      217
Fałszywie pozytywne wyniki dla Forest:   63
Fałszywie pozytywne wyniki dla Tree:     201
Fałszywie pozytywne wyniki dla SVC:      43
Fałszywie pozytywne wyniki dla KNN:      56
-----
Fałszywie negatywne wyniki dla SVR:      356
Fałszywie negatywne wyniki dla Forest:   223
Fałszywie negatywne wyniki dla Tree:     173
Fałszywie negatywne wyniki dla SVC:      509
Fałszywie negatywne wyniki dla KNN:     283
```

Analogiczny sposób sprawdzono w ilu przypadkach różnica pomiędzy wartościami wyniosła więcej niż 100 W/m<sup>2</sup>.

Tabelaryczne i graficzne zestawienia otrzymanych wyników prezentowane są poniżej:

SVR				
Liczba wszystkich	Idealne dopasowanie	Różnica > 100	Fałszywe pozytywne	Fałszywe negatywne
3493	2412	574	217	356
Forest				
Liczba wszystkich	Idealne dopasowanie	Różnica > 100	Fałszywe pozytywne	Fałszywe negatywne
3493	2967	256	63	223
Tree				
Liczba wszystkich	Idealne dopasowanie	Różnica > 100	Fałszywe pozytywne	Fałszywe negatywne
3493	2788	289	201	173
SVC				
Liczba wszystkich	Idealne dopasowanie	Różnica > 100	Fałszywe pozytywne	Fałszywe negatywne
3493	2739	488	43	509
KNN				
Liczba wszystkich	Idealne dopasowanie	Różnica > 100	Fałszywe pozytywne	Fałszywe negatywne
3493	2919	311	56	283



Podsumowując przykład można stwierdzić iż najlepiej przewidyującym modelem jest ten wykorzystujący algorytm lasów losowych, który to cechował się najniższym współczynnikiem średniego błęd kwadratowego, najwyższą ilością idealnych dopasowań oraz współczynnikiem fałszywie przewidywanych pozytywnych wartości poniżej 2%.

## 4. Protokoły

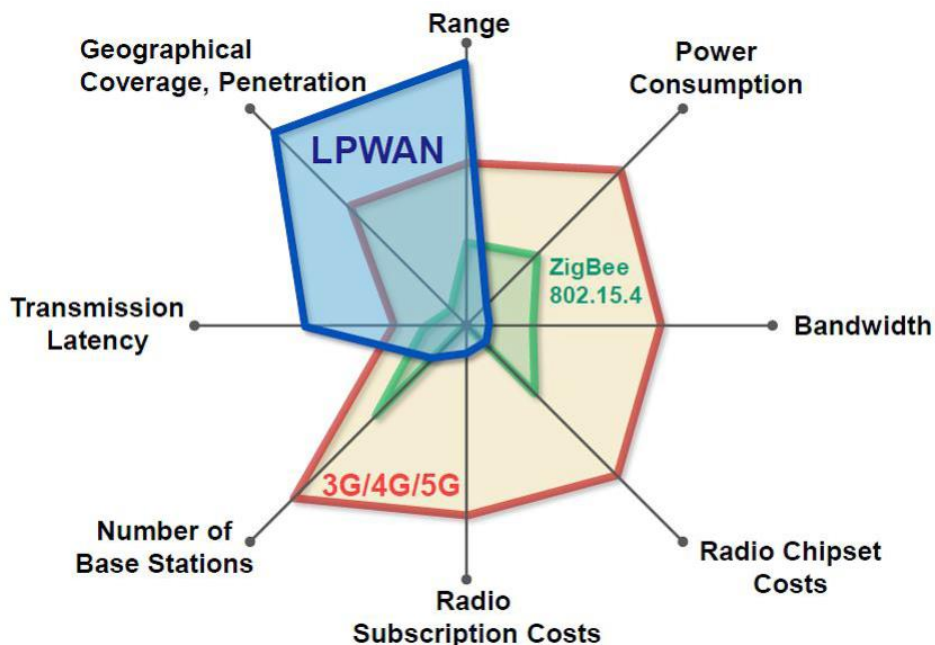
Wybór oszczędnego energetycznie protokołu transmisji danych w głównej mierze polegał na porównaniu stosowanych niskoenergetycznych technologii komunikacji radiowej w oparciu o kryteria, istotne z punktu widzenia wymagań projektowych.

- **Niezawodność** - w przypadku usług operatorów sieci komórkowych jest ona duża, lecz wiąże się z większym kosztem usługi. Dodatkowo usługa jest standardowa, a więc dostępna na obszarze np. całego kraju. Z drugiej strony, jeśli parametry niezawodności i prędkości transmisji nie są priorytetowe, to ponosząc mniejsze nakłady finansowe, można zastosować sieć LoRaWAN, uzyskując porównywalny zasięg komunikacji.
- **Czas życia baterii i pobór prądu w trybie uśpienia** – parametry wpływające na czas pracy urządzenia bez konieczności serwisowania. W przypadku urządzeń badanych w zagadnieniu pobór prądu



z baterii jest większy, jednak czas reakcji urządzeń końcowych krótszy i prędkość transmisji większa w porównaniu z LoRaWAN.

- **Pokrycie sieci i zasięg** – w przypadku urządzeń i sieci pokrycie dla chmury projektowej będzie takie samo, jak dla sieci LTE (te same stacje bazowe), przy czym wymagane jest uruchomienie obsługi urządzeń przez operatora sieci LTE na danym obszarze. Sieci prywatne oparte o pasmo ISM mogą mieć lepsze pokrycie i zasięg na terenach wiejskich, natomiast w terenie zurbanizowanym parametry są porównywalne do sieci LTE. Oba parametry zależą od mocy wyjściowej, gdzie dla LTE wynosi ona 20 do 23 dBm (100 do 200 mW), a dla pasma 868 MHz 14 dBm, (25 mW). Zatem pomimo kilkukrotnie mniejszej mocy sieci LoRa dorównują zasięgiem i pokryciem sieciom CloT. Jest to ich duża zaleta w przypadku budowania prywatnej niezależnej sieci dalekiego zasięgu.
- **Lokalizacja** – zależnie od przeznaczenia, sieć CloT może być oparta o istniejące stacje bazowe telefonii komórkowej, zatem koszty uruchomienia nowej usługi w istniejącej infrastrukturze nie są duże. Natomiast może to być ograniczenie w porównaniu np. z technologią LoRaWAN, gdzie rozmieszczenie urządzeń, stanowiących bramy, w terenie może być dostosowywane do konkretnych wymagań projektu.
- **Koszt sieci** – obejmuje koszty urządzeń końcowych, stacji bazowych, bramek, opłaty za użytkowanie aplikacji chmurowych na zasadach komercyjnych, abonamenty sieci LTE. Przygotowanie całkowicie prywatnych sieci, np. LoRaWAN, również wymaga poniesienia kosztów na zbudowanie własnej warstwy serwerów sieciowych oraz serwerów aplikacyjnych.



Rys. Porównanie najistotniejszych kryteriów użytkowych bezprzewodowych technologii komunikacyjnych.

## 4.1. Wybrane rozwiązanie

**LoRaWAN** (ang. Long Range Wide Area Network) jest koncepcją rozległej komunikacji z wykorzystaniem Internetu i LoRa dla potrzeb Internetu Przedmiotów (IoT). LoRaWAN to protokół MAC (*Medium Access Protocol* – protokół dostępu do łącza). To radiowy protokół komunikacyjny, który pozwala łączyć się z Internetem urządzeniom IoT wyposażonym w łącze radiowe. Połączenie nie jest realizowane wprost, ale za pomocą specjalnych stacji bazowych nazywanych koncentratorami. Bardzo ważną cechą standardu jest możliwość uzyskania relatywnie dużego zasięgu liczonego w kilometrach przy bardzo ograniczonej mocy nadawania (20 dBm) i co za tym idzie - przy małym poborze prądu. LoRaWAN bazuje na otwartym standardzie i wykorzystuje pasmo ISM - w Europie jest to 868 MHz. To ma daleko idące konsekwencje, bo umożliwia tworzenie tanich sieci bez konieczności uzyskiwania administracyjnych zgód i ponoszenia opłat za wykorzystanie pasma.

Protokół LoRaWAN został opracowany pod kątem dużej wydajności, dalekiego zasięgu i małego poboru mocy dla urządzeń Internetu Rzeczy. Korzysta z zalet technologii LoRa i optymalizuje zużycie energii oraz obsługuje mechanizmy optymalizacji ruchu pomiędzy węzłami. Protokół ten jest w pełni dwukierunkowy, co pozwala na niezawodny przesył informacji (dostępność potwierdzeń).

Oprogramowanie stosu LoRaWAN oparto o model warstwowy składający się z warstw: aplikacji, MAC i PHY. Warstwa aplikacji zawiera funkcje i procedury użytkownika, które wykorzystują możliwości warstwy MAC do realizowania przesyłania danych (nadawania i odbierania) przez łącze radiowe LoRa. Warstwa MAC jest pomostem pomiędzy warstwą aplikacji wykonującą funkcje związane z pracą czujnika i warstwą fizyczną PHY. W niej jest organizowana transmisja danych właściwa dla danej klasy urządzeń (A, B, lub C). Warstwa fizyczna PHY to układ transceivera wysyłający i odbierający dane radiowe w kanale radiowym. Zawiera modulator, demodulator CSS, oraz tory nadajnika i odbiornika wraz z układami konfiguracyjnymi i układami sterowania .

**Niskie zużycie energii przez urządzenia końcowe.** Urządzenia nie muszą się synchronizować z bramami i wysyłają dane gdy spełnione są założone warunki. W pozostałym czasie pozostają w stanie uśpienia charakteryzującym się minimalnym poborem prądu.

**Dwukierunkowa komunikacja.** Urządzenie końcowe może nie tylko przysyłać dane do bramki, ale też odbierać je, zgodnie z konfiguracją. Konfigurację tę można zmieniać z poziomu aplikacji.

**Pojemność sieci. Skalowalność.** LoRaWAN zostało zaprojektowane pod kątem dużych sieci IoT, w których tysiące urządzeń może być podłączone nawet do niewielkiej liczby bramek. Te bramki mogą nasłuchiwać na wielu kanałach i przetwarzać liczne wiadomości w tym samym czasie.

**Wybór prędkości transmisji.** Kluczową cechą LoRaWAN jest szybkość, z jaką dane mogą być wysyłane. Przewidziano różne prędkości transmisji, których można użyć do przesyłu danych. Określa się je mianem SF (*Spreading Factors*). W efekcie wolniejsza transmisja pozwala na bardziej niezawodny przesył informacji na większe odległości. Łatwo sobie zwizualizować tę zależność. Wystarczy wyobrazić sobie, że chce się porozmawiać z osobą, która stoi bardzo blisko. Można mówić do niej całkiem szybko, a ona i tak będzie w stanie wszystko

usłyszeć. Jeśli chce się natomiast porozmawiać z kimś kto stoi daleko, trzeba będzie wolniej mówić, by być zrozumiałym. Ta zasada została zaszyta w protokole LoRaWAN.

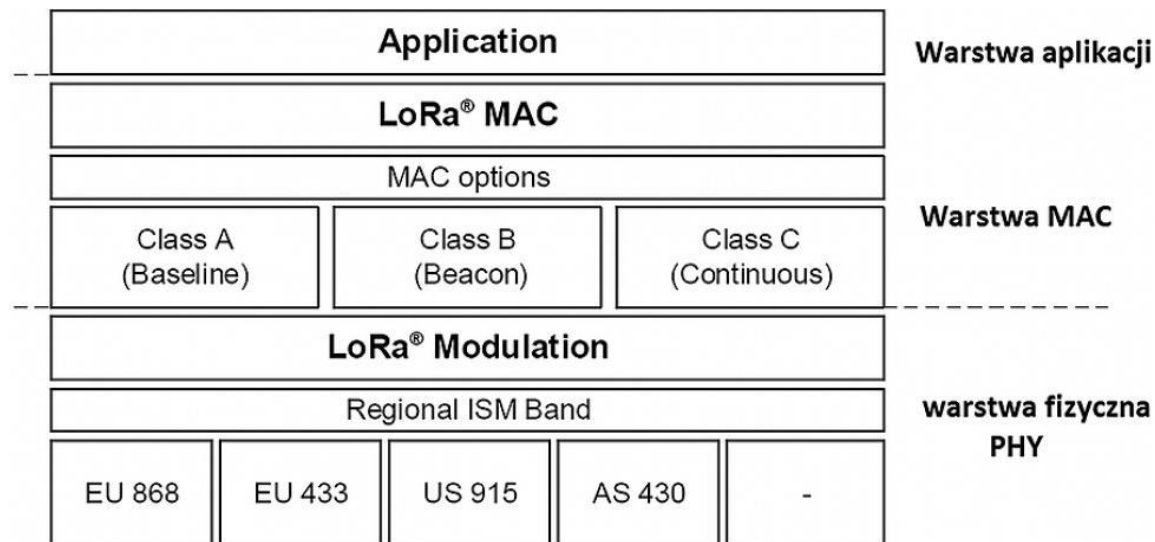
**Adaptacyjna szybkość przesyłu danych.** Wyróżniającą cechą LoRaWAN jest to, że sieć może automatycznie zoptymalizować prędkość, z jaką urządzenie przesyła dane. Funkcja *Adaptive Data Rate* (ADR – Adaptacyjna szybkość przesyłu danych) jest kluczowa dla zwiększania sumarycznej przepustowości sieci LoRaWAN. ADR pozwala na łatwe skalowanie sieci, po prostu poprzez zwiększanie liczby bramek. Dzięki nim wiele urządzeń automatycznie zmieni swój współczynnik rozproszenia (SF), a w związku z tym skróci czas potrzebny na transmisję pakietu, w efekcie pozostawiając więcej wolnego pasma dla innych urządzeń.

ADR to prosty mechanizm, który prowadzi do zmiany szybkości transmisji w oparciu o następujące zasady:

- Jeśli moc sygnału radiowego (nazywanego budżetem łącza) jest duża, szybkość transmisji może być zwiększona.
- Jeśli budżet łącza jest mały, szybkość transmisji może być obniżona.

Protokół LoRaWAN optymalizuje szybkość transmisji danych w celu zminimalizowania czasu antenowego (czasu transmisji) i zużycia energii przez urządzenia emitujące. W porównaniu ze stałą prędkością transmisji danych w innych technologiach LPWA ta optymalizacja może stukrotnie zmniejszyć średnie zużycie energii przez podłączony obiekt.

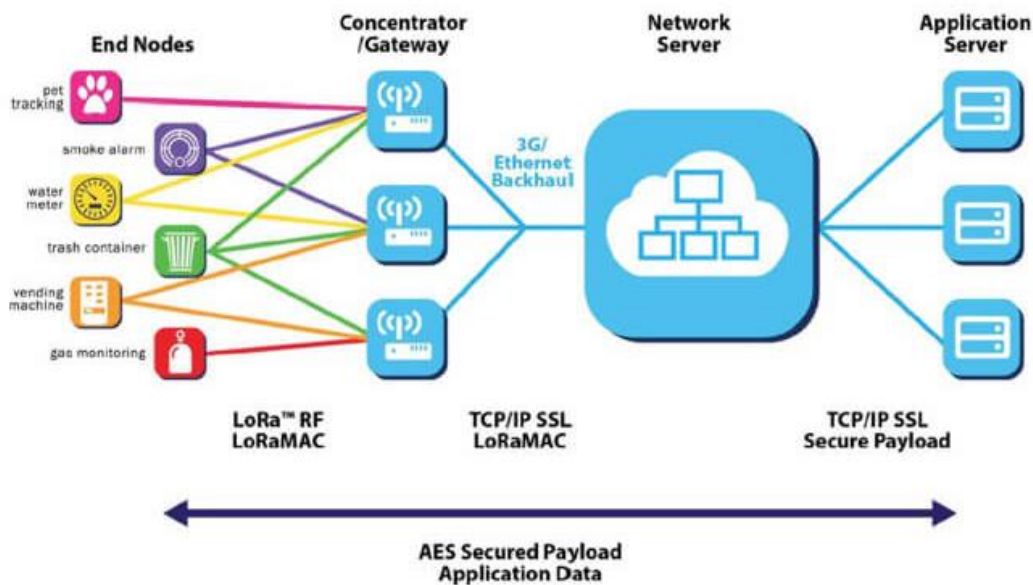
## 4.2. Przegląd warstw protokołu



## 4.3. Transmisja danych

Technologia LoRaWAN została zaprojektowana do takich zastosowań, kiedy czujnik przesyła niewielkie ilości danych kilka razy dziennie, a przez większość czasu pozostaje nieaktywny. Jest dobrze przystosowana do inteligentnych liczników, urządzeń śledzących, czujników itp. Nie jest przeznaczona do obsługi aplikacji

wymagających dużych szybkości transmisji danych, takich jak audio lub wideo. Protokół LoRaWAN można jednak wykorzystać do sterowania innymi funkcjami urządzeń bezprzewodowych.



Rys. Architektura sieci opartej na LoRaWAN.

Definicja LoRaWAN obejmuje także mechanizm szyfrowania punkt-punkt, co umożliwia tworzenie bardzo bezpiecznych połączeń. Ponadto obsługuje bezprzewodową rejestrację nowych urządzeń w sieci i nadawanie w trybie multicast (komunikacja jeden-do-wielu). Standard ten został opracowany tak, by zapewnić zgodność urządzeń pomiędzy różnymi sieciami LoRaWAN na całym świecie.

LoRaWAN składa się z czterech głównych komponentów. Są nimi:

- **Urządzenia końcowe** – obsługują określone funkcje (np. czujnikowe), mogą wykonywać obliczenia oraz wyposażone są w moduły radiowe. Urządzenia te mogą przysyłać dane do bramek oraz same otrzymywać dane. Mogą pracować z użyciem małej baterii nawet przez wiele lat, o ile tylko zostaną wprowadzone do stanu głębokiego uśpienia, celem optymalizacji poboru mocy. Wyróżnia się tu urządzenia klas A, B i C.
- **Bramki (modemy, punkty dostępu)** – odbierają dane nadawane przez węzły końcowe za pomocą LoRaWAN. Wiadomości te są często konwertowane na pakiety, które można przysyłać za pomocą tradycyjnych sieci IP. Bramka jest więc podłączona do serwera sieciowego, do którego przysyła wszystkie wiadomości.
- **Serwer sieciowy** – wszystkie te wiadomości przesyłane przez bramki są przekazywane do serwera sieciowego. Jest on odpowiedzialny przede wszystkim za:
  - przekierowywanie i przekazywanie danych do odpowiednich aplikacji,
  - zapewnianie jakości, tzn. określanie, która z bramek jest najlepsza do skierowania wiadomości przesyłanej do wybranego węzła. Zazwyczaj operacja ta bazuje na porównaniu parametru

jakości łącza, który jest obliczany na podstawie współczynników RSSI (*Receive Signal Strength Indication*)

- o oszacowanie wskaźnika siły odbieranego sygnału oraz SNR (*Signal to Noise Ratio* – stosunek sygnał/szum) poprzednio otrzymanych pakietów
- o usuwanie zduplikowanych wiadomości, jeśli zdarzyło się, że dane z węzła zostały przekazane do serwera przez więcej niż jedną bramkę
- o deszyfrowanie wiadomości przesyłanych z węzłów końcowych i szyfrowanie informacji przesyłanych do węzłów

Bramki zazwyczaj łączą się z serwerem sieciowym za pomocą szyfrowanego połączenia IP. Sieć zawiera interfejs do nadzorowania pracy i instalacji nowych bramek, pozwalając kontrolerowi sieci na zarządzanie nimi, rozwiązywanie problemów, wykrywanie usterek, monitorowanie pojawiających się alarmów itp.

- **Serwery aplikacji** – w nich realizowana jest właściwa aplikacja IoT, która wykonuje zadania na bazie danych zbieranych przez urządzenia końcowe. Serwery aplikacyjne pracują zazwyczaj w oparciu o prywatne lub publiczne chmury, które łączą się z serwerami sieciowymi LoRaWAN i wykonują konkretne zadania.

#### 4.3.1. Urządzenia końcowe

Urządzenia (węzły) końcowe to sprzęt, który obsługuje określone funkcje czujnikowe, ma nieco mocy obliczeniowej oraz moduł radiowy do konwersji zbieranych informacji na sygnał radiowy. Urządzenia te mogą przysyłać dane do bramek oraz same otrzymywać dane. Mogą pracować z użyciem małej baterii nawet przez wiele lat, o ile tylko zostaną wprowadzone do stanu uśpienia, celem optymalizacji poboru mocy.

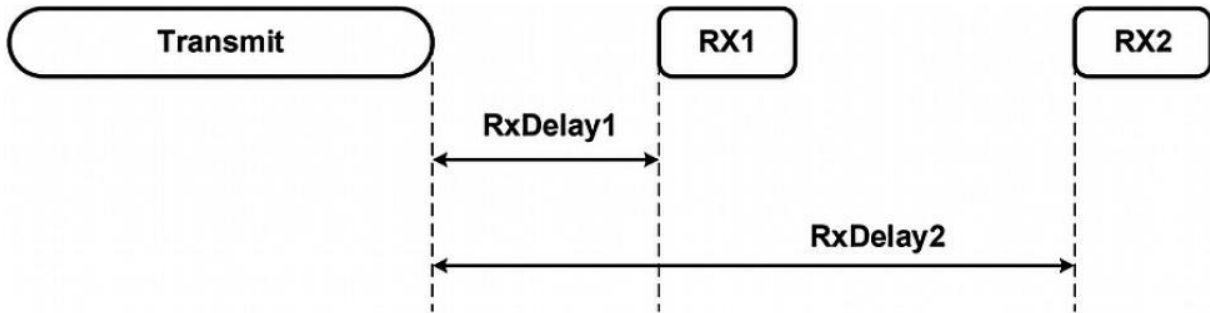
Gdy urządzenie końcowe przesyła informacje do bramki, transmisja ta kategoryzowana jest jako „Uplink”, a gdy otrzymuje dane z bramki, używa się określenia „Downlink”. Bazując na tej nomenklaturze, można wydzielić trzy rodzaje urządzeń końcowych:

- klasy A,
- klasy B,
- klasy C.

**Urządzenia klasy A (Class A)** to takie, które pobierają najmniej energii w porównaniu do pozostałych klas. Na jednej baterii urządzenia mogą pracować nawet 10 lat. Decyduje o tym fakt, że mogą odbierać informacje (downlink) tylko po wysłaniu własnych danych (uplink). Każda wymiana danych rozpoczyna się od wysłania przez urządzenie końcowe danych (uplink). Po ich wysłaniu węzeł odczeka przez zdefiniowane dwa określone czasy (okna czasowe) na dane z serwera (wiadomość downlink). Po odebraniu wiadomości z serwera urządzenie końcowe (węzeł) przechodzi do stanu głębokiego uśpienia, aby ograniczyć pobór energii przy pracy z zasilaniem

baterijnym. Zaletą klasy A jest bardzo mały pobór energii, a wadą spore opóźnienia łącza, bo serwer sieciowy nie "wie", kiedy zostanie zainicjowana transmisja

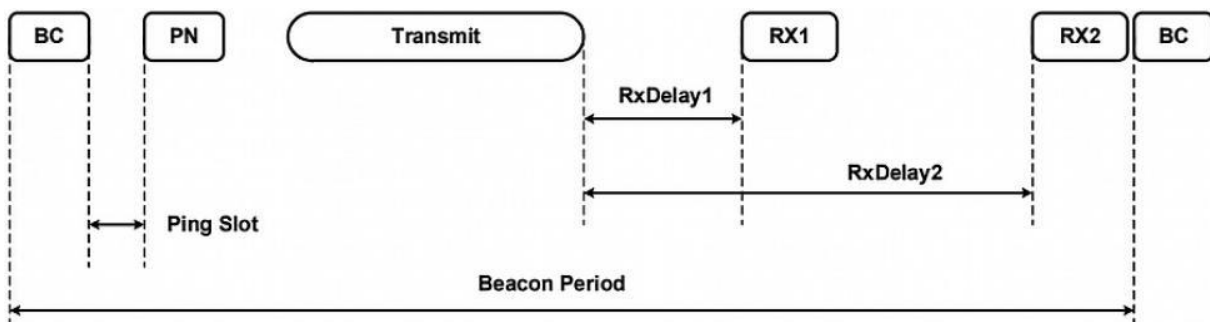
Urządzenia klasy A nadają się do przesyłania danych z określonym interwałem, np. co 15 minut, lub do przesyłania informacji w oparciu o zdarzenia – np. gdy mierzona temperatura przekroczy 21 °C lub spadnie poniżej 19 °C. Do klasy A należą czujniki zasilane bateryjnie lub urządzenia wykonawcze niewrażliwe na opóźnienia.



Rys. Wymiana informacji z urządzeniem klasy A

**Węzły klasy B (Class B)** służą głównie do sygnalizacji zmian. Urządzenia klasy B mogą przesyłać i odbierać dłuższe informacje z mniejszym opóźnieniem reakcji serwera spowodowanej losowym wysyłaniem danych (uplink) przez urządzenie klasy A. W klasie B urządzenie okresowo otwiera okno czasowe do odbierania informacji z serwera (downlink). Czas otwarcia łącza (Beacon Period) pomiędzy urządzeniem i serwerem jest ustalany przez zsynchronizowane zegary. Synchronizacja jest realizowana przez sekwencyjne wysyłanie przez serwer komendy synchronizującej odbieranej przez urządzenie. Sama transmisja odbywa się tak jak w klasie A. Zaletą klasy B jest szybszy dostęp do połączenia, bo serwer sieciowy "wie", w jakim czasie może spodziewać się transmisji z urządzenia.

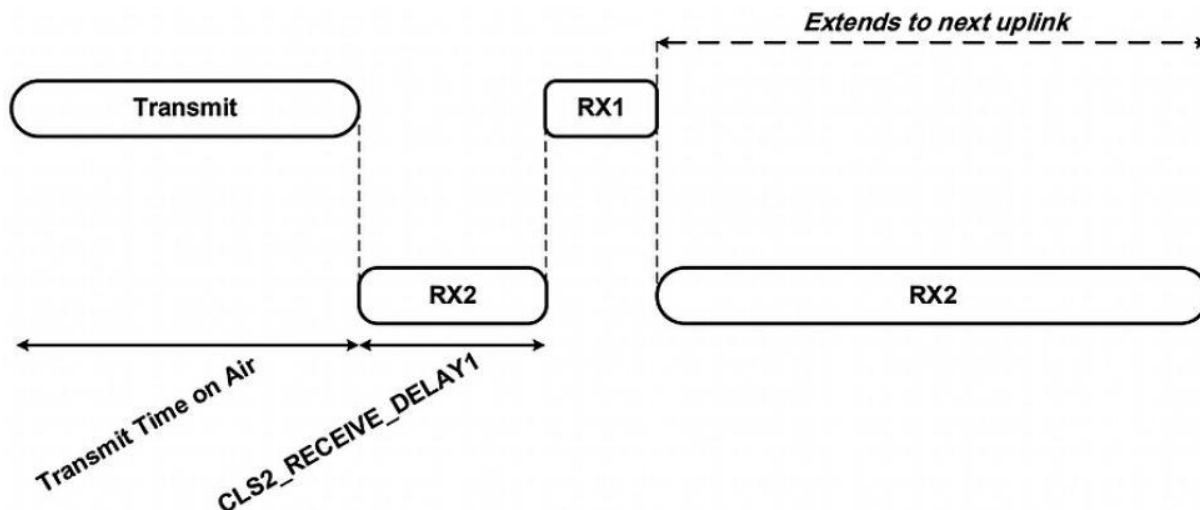
Do klasy B należą urządzenia wykonawcze zasilane bateryjnie, których czas opóźnienia ruchu w dół sieci wymaga kontroli. Działanie ich oparte jest na komunikacji podzielonej na części (sloty) synchronizowanej z sygnalizacją sieci. Węzły B mogą otrzymywać większą ilość wiadomości niż w przypadku klasy A. To pozwala zmniejszyć opóźnienia przesyłanych wiadomości, ale kosztem obniżenia efektywności energetycznej urządzenia.



Rys. Wymiana informacji z urządzeniem klasy B

**Urządzenia klasy C (Class C)** to urządzenia pracujące w otwartym kanale komunikacyjnym ciągle nasłuchując. Urządzenie klasy C może ciągle odbierać dane, jeśli samo ich nie wysyła. Klasa ta nie jest przewidziana do zasilania

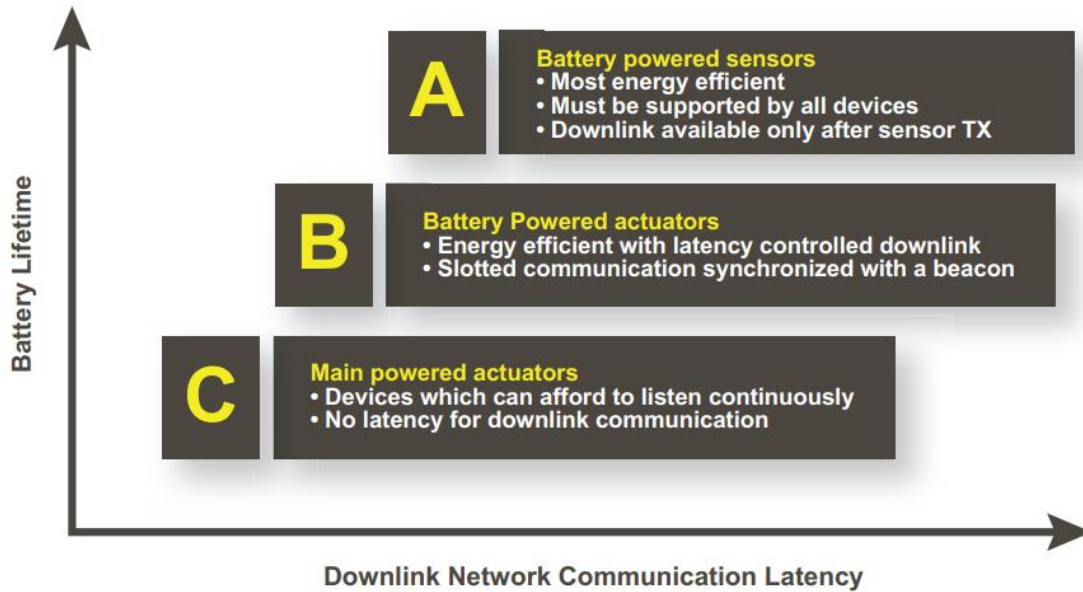
baterijnego i wymaga stałego zasilania z sieci energetycznej. W trakcie pracy jest ciągle otwierane okno czasowe. Zmniejsza to opóźnienia łącza umożliwiając przesłanie relatywnie dużej ilości danych, ale znacznie podwyższa pobór energii. Z tego względu jest to najmniej energooszczędna klasa i zazwyczaj wymaga zastosowania stałego źródła zasilania do pracy.



Rys. Wymiana informacji z urządzeniem klasy C

Węzły LoRa nie są związane z określoną bramą – transmisja może być jednocześnie odbierana przez kilka bram.

Węzły sieci są asynchroniczne i mogą transmitować dane gdy one będą gotowe do wysłania, np. gdy wystąpiło kontrolowane przez węzeł zdarzenie lub według przyjętego harmonogramu. Przyjęta zasada pracy węzłów końcowych w sieci LoRaWAN jest oszczędna dla baterii - węzły nie muszą się synchronizować co pochłania w innych sieciach sporo energii. W urządzeniach klasy A i B stosuje się głębokie uśpienie w czasie braku aktywności pozwalające na mocne obniżenie poboru prądu. Ale niezbędny jest też pewien stopień wydajności zabudowanego mikrokontrolera pozwalający na obsługę stosu, kodowanie transmisji danych i oczywiście obsługę czujnika pomiarowego. Dlatego wielu producentów stosuje 32-bitowe mikrokontrolery mające zaawansowane tryby obniżonego poboru energii.



Rys. Klasy węzła. Czas pracy baterii w funkcji akceptacji opóźnienia transmisji.

#### 4.3.2. Bramy

Bramy (modemy, punkty dostępu) to urządzenia, które odbierają dane nadawane przez węzły końcowe za pomocą LoRaWAN. Wiadomości te są często konwertowane na pakiety, które można przesyłać za pomocą tradycyjnych sieci IP. Brama jest więc podłączona do serwera sieciowego, do którego przesyła wszystkie wiadomości. Brama musi mieć zdolność odbierania dużej liczby pakietów od węzłów końcowych - cel ten osiąga się przez dobór prędkości transmisji oraz przez wykorzystanie dostępnych kanałów

Bramy są z założenia przezroczyste i mają ograniczoną moc obliczeniową. Wszystkie złożone operacje i inteligentne algorytmy są realizowane na serwerze sieciowym. W zależności od zastosowania bramy (na zewnątrz budynków) można wykorzystać takie jak np. Kerlink IoT Station i LoRiX One.

#### 4.3.3. Serwer sieciowy

Wszystkie te wiadomości przesyłane przez bramki są przekazywane do serwera sieciowego. To w nim następują bardziej skomplikowane procesy, związane z przetwarzaniem danych. Jest on odpowiedzialny przede wszystkim za:

- Przekierowywanie i przekazywanie danych do odpowiednich aplikacji.
- Określanie, która z bramek jest najlepsza do skierowania wiadomości przesyłanej do wybranego węzła. Zazwyczaj operacja ta bazuje na porównaniu parametru jakości łącza (Link Quality), który obliczany jest na podstawie współczynników RSSI (*Receive Signal Strength Indication* – wskaźnik siły odbieranego sygnału) oraz SNR (*Signal to Noise Ratio* – stosunek sygnał-szum) poprzednio otrzymanych pakietów.
- Usuwanie zduplikowanych wiadomości, jeśli zdarzyło się, że dane z węzła zostały przekazane do serwera przez więcej niż jedną bramkę.



- Deszyfrowanie wiadomości przesyłanych z węzłów końcowych i szyfrowanie informacji przesyłanych do węzłów. Bramki zazwyczaj łączą się z serwerem sieciowym za pomocą szyfrowanego połączenia IP. Sieć zazwyczaj zawiera interfejs do nadzorowania pracy i instalacji nowych bramek, pozwalając kontrolerowi sieci na zarządzanie nimi, rozwiązywanie problemów, wykrywanie usterek, monitorowanie pojawiających się alarmów itp.

#### 4.3.4. Serwer aplikacji

Serwer aplikacji to miejsce, w którym realizowana jest właściwa aplikacja IoT – taka, która robi coś sensownego na podstawie danych zbieranych przez urządzenia końcowe. Serwery aplikacyjne pracują zazwyczaj w oparciu o prywatne lub publiczne chmury, które łączą się z serwerami sieciowymi LoRaWAN i wykonują konkretne zadania. Za połączenie pomiędzy serwerem aplikacji a węzłami i bramkami LoRaWAN odpowiada serwer sieciowy.

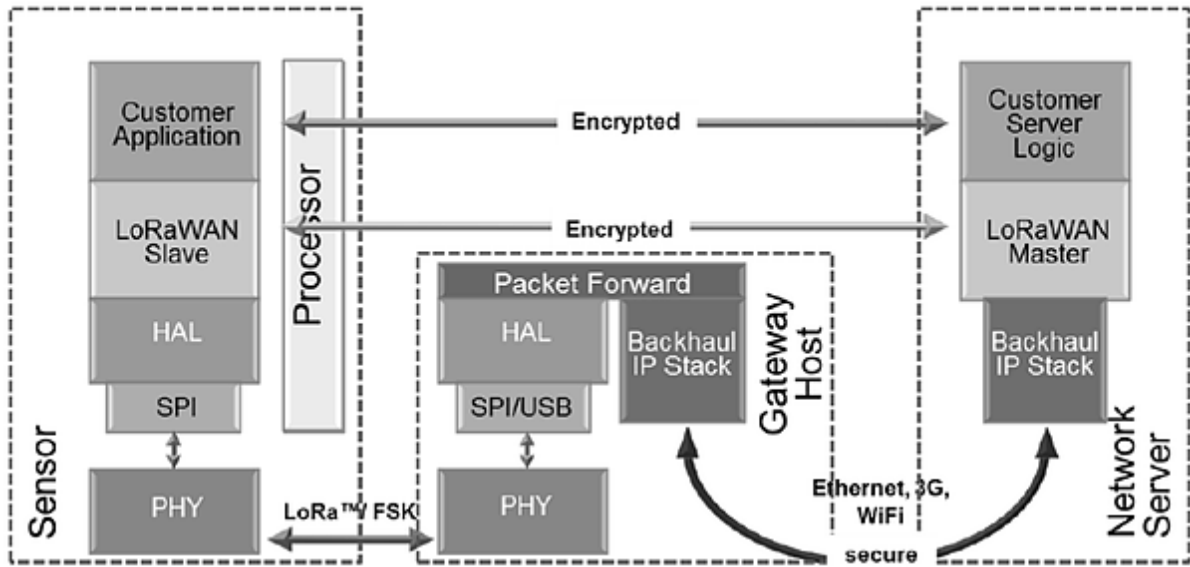
### 4.4. Bezpieczeństwo

Zapewnienie bezpieczeństwa jest niezmiernie ważne w każdej sieci LPWAN (*Low Power Wide Area Network*). LoRaWAN wykorzystuje w tym celu dwie warstwy zabezpieczeń:

- Jedna na poziomie sieci. Warstwa sieciowa jest odpowiedzialna za identyfikację węzłów, sprawdzanie czy wiadomości faktycznie pochodzą z konkretnych urządzeń i jest określana mianem sprawdzania integralności sieci. Realizuje również szyfrowanie komend MAC.
- Druga na poziomie aplikacji. W warstwie aplikacji realizowane jest szyfrowanie i deszyfrowanie właściwych, użytecznych danych.

Oba klucze są szyfrowane 128-bitowym algorytmem AES w trybie ECB (Electronic Codebook).

Zabezpieczenia na poziomie sieci pozwalają być pewnym, że dane urządzenie końcowe jest faktycznie tym, którym miało być, podczas gdy zabezpieczenia na poziomie aplikacji sprawiają, że operator sieci nie ma dostępu do danych użytkownika.



Koniec