



Narodowe Centrum  
Badań i Rozwoju



Ministerstwo  
Spraw Wewnętrznych  
i Administracji

Badanie opiera się na wynikach pracy zespołu Executive Education Center,  
**DOB-BIO10 / 19/02/2020 - Opracowanie nowoczesnego modelu zarządzania pacjentem w stanie zagrożenia życia w oparciu o samouczącą się algorytmizację procesów decyzyjnych i analizę danych z procesów terapeutycznych**

### 3.01.01

Zupełnie nowy model działania SOR z wykorzystaniem  
algorytmów decyzyjnych i innych niż dotychczas  
technologii



Executive Education  
Center Sp. z o.o.

Tel. 22 829 64 23,  
[rkoszewski@exedu.eu](mailto:rkoszewski@exedu.eu)

[exedu.eu](http://exedu.eu)



Politechnika  
Śląska

Politechnika Śląska

Tel. 32 237 10 51  
[sekr\\_ia@polsl.pl](mailto:sekr_ia@polsl.pl)

[polsl.pl](http://polsl.pl)



VRTechnology Sp. z o.o.

Tel. 668 661 822  
[office@vrtechnology.pl](mailto:office@vrtechnology.pl)

[vrtechnology.pl](http://vrtechnology.pl)



RICOH Polska Sp. z o.o.

Tel. 22 256 15 55  
[ricoh@ricoh.pl](mailto:ricoh@ricoh.pl)

[ricoh.pl](http://ricoh.pl)

## Spis treści

Wstęp .....	3
1. Moduł prognozowania .....	6
METODOLOGIA .....	6
WYNIKI .....	8
Model addytywny .....	8
Model sieci neuronowej .....	16
Wygładzanie wykładnicze .....	17
WNIOSKI .....	23
Kod R .....	25
Dynamiczny model addytywny .....	25
Stacjonarny model addytywny .....	30
Dynamiczny model sieci neuronowej .....	32
Stacjonarny model sieci neuronowej .....	34
Dynamiczny model wygładzania wykładniczego .....	35
Stacjonarny model wygładzania wykładniczego .....	37
2. Wymiarowanie dla zasobów stałych .....	38
3. Wymiarowanie pod kątem zasobów zmiennych (personel) .....	47
4. Optymalizacja zmian .....	51
5. Optymalizacja przydziału poszczególnych pracowników do dostępnych zmian .....	55

## Wstęp

Wszystkie szpitale dążą do zapewnienia wysokiej jakości opieki w nagłych wypadkach. SOR jest przykładem systemu produkcyjnego. Wielu pacjentów jest jednocześnie poddawanych wielu procesom. Aby uzyskać opiekę wysokiej jakości, procesy te muszą być skuteczne. Mówiąc żargonem menedżerskim, procesy powinny być „wydajne”. Aby procesy były wydajne, wydajność musi być na tyle duża, aby sprostać zapotrzebowaniu. W każdym systemie, gdy równowaga między zapotrzebowaniem a wydajnością jest prawidłowa, wyniki powinny być dobre. Jeśli zapotrzebowanie wzrasta, system zazwyczaj jest w stanie na nie zareagować, a wyniki utrzymują się na normalnym poziomie. Jednak gdy zapotrzebowanie staje się zbyt duże, konieczne staje się znalezienie kompromisu między konkurującymi ze sobą celami („strefa radzenia sobie”). W takiej sytuacji wyniki mogą ulec pogorszeniu, ponieważ pracownicy będą pracować zbyt szybko lub pod wpływem stresu, aby nadążyć. Zbyt długie przebywanie w strefie radzenia sobie może mieć negatywny wpływ na morale i funkcjonowanie personelu. Mówiąc prościej, jeśli będziemy zbyt mocno obciążymy SOR, systemy zawiodą, a szkody poniesie zarówno personel, jak i pacjenci.

SOR nie jest systemem odizolowanym, tylko ściśle powiązany z decyzjami dotyczącymi wydajności i polityki podejmowanymi w innych częściach systemu opieki zdrowotnej, dlatego też można uznać, że zatłoczenie na oddziale ratunkowym jest wskaźnikiem ukazującym funkcjonowanie systemu opieki zdrowotnej. Kluczem do znalezienia trwałego rozwiązania dla SOR-ów jest zmiana polityki (finansowania) w zakresie opieki zdrowotnej, tak aby umożliwić szpitalom utrzymanie liczby pacjentów hospitalizowanych poniżej krytycznego progu oraz ustanowienie zasad zapewniających właściwe zrównoważenie liczby przyjęć z SOR-ów i przyjęć planowych.

Gdy liczba pacjentów na oddziale szpitalnym zbliża się do 85%, SOR musi zacząć przyjmować pacjentów, tzn. wykorzystuje swoje łóżka do zatrzymywania pacjentów, którzy nie powinni już dłużej na nim przebywać, lecz powinni zostać przyjęci na oddziały szpitalne. Zależy to oczywiście od liczby przyjęć w danym szpitalu, ale jest to dobra zasada praktyczna.

Istnieje kilka przyczyn zatłoczenia na SOR-ach, w tym jest to m. in. niedostępność placówek zajmujących się pacjentem po wypisie ze szpitala (ośrodków rehabilitacyjnych, specjalistycznych ośrodków opieki, domów opieki,...), co prowadzi do długotrwałego obłożenia szpitala, niedoboru personelu medycznego itp. Problemy te można sklasyfikować na różnych poziomach:

Na **poziomie systemu opieki zdrowotnej**. Często zdarza się, że struktura finansowa sprzyja zatłoczeniu szpitali, ponieważ aby uzyskać wystarczający zwrot kosztów szpitale muszą utrzymywać liczbę pacjentów hospitalizowanych przekraczającą możliwości efektywnego działania. Ponadto, ze względu na wprowadzenie nowych, mniej agresywnych technologii, szpitale zmniejszyły liczbę łóżek, a tym samym swoją zdolność do absorpcji gwałtownych wzrostów zapotrzebowania. Brak podstawowej opieki zdrowotnej, opieki psychiatrycznej, rehabilitacji i opieki społecznej (lub opieka niewystarczająca) również powoduje, że do szpitali trafia zbyt wielu pacjentów, powodując przepełnienie pacjentami, którzy nie powinni być w nich leczeni.

Niektóre czynniki są związane z **decyzjami podejmowanymi przez szpitale**. Na przykład brak koordynacji działań kierownictwa i ustalenia priorytetów wśród różnych służb i oddziałów zmniejsza prawdopodobieństwo powodzenia każdego zamierzonego rozwiązania. Ponadto, szpitale nie są zorganizowane tak, aby zaspokajać zapotrzebowanie operacyjne przez 24 godziny na dobę, 7 dni w tygodniu, co powoduje niepotrzebne obciążenie SOR-ów, które są dostępne przez całą dobę, przez cały tydzień. Podobnie, brak możliwości skorzystania z opieki podstawowej i opieki poza zwykłymi

godzinami pracy prowadzi do potencjalnego pogorszenia ostrych i przewlekłych stanów chorobowych, a w konsekwencji do częstszych wizyt na SOR i hospitalizacji. Wreszcie, przepisy dotyczące Covid-19 wymagają znacznych ograniczeń przestrzeni na SOR-ach, przepływu pacjentów, zabraniają stawiania łóżek w korytarzach itp.

Pewne inne czynniki na **SOR-ach mają charakter wewnętrzny** i ograniczają ich przepustowość. Na przykład większa różnorodność pacjentów, która wymaga więcej czasu i środków, większa dostępność czasochłonnych technologii oraz częstsze stosowanie badań obrazowych (USG, TK i MRI) wydłużają czas oceny pacjentów. SOR cierpi również z powodu problemów spowodowanych opóźnieniami w pracy laboratoriów, radiologii i konsultantów. SOR w CSK ucierpiał również z powodu niedoboru i rotacji personelu medycznego, a wypalenie zawodowe wśród personelu pracującego na tym oddziale jest wysokie. Niedobór powoduje niemożność zapewnienia wszystkich dostępnych łóżek lub nieefektywne działania.

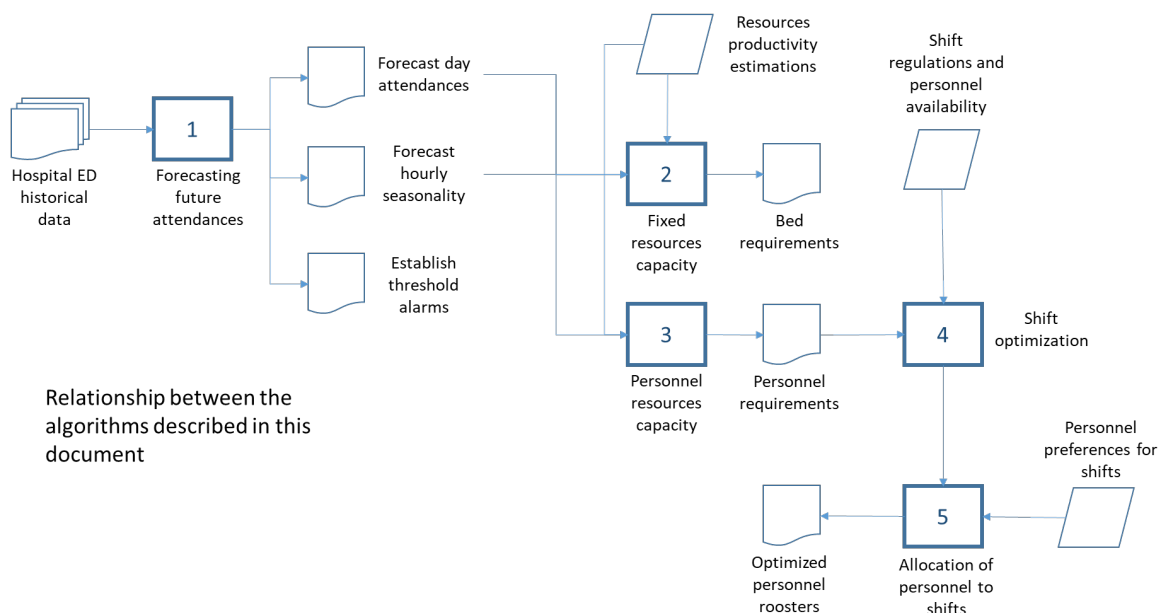
Ponadto, inne czynniki mogą być związane z **obsługą pacjentów opuszczających SOR** przez inne służby. Wśród nich można wymienić blokady dostępu, w przypadku których duża liczba pacjentów w szpitalu lub niewydolność operacyjna uniemożliwiają udostępnienie łóżek. Do tego dochodzi nieefektywny proces przekazywania pacjentów z SOR-u do oddziałów szpitalnych, gdy pielęgniarki szpitalne mogą nie przyjmować zgłoszeń w kluczowych porach dnia lub gdy są zajęte, albo gdy wypisy pacjentów szpitalnych często następują bardzo późno w ciągu dnia/wczesnym wieczorem, gdy SOR jest już wypełniony. Jednym z najbardziej prawdopodobnych problemów jest jednak nieformalne rezerwowanie (blokowanie) łóżek w placówkach szpitalnych, gdy dostępne łóżka są „zachowywane” dla określonych typów pacjentów.

Jak widać, działania, które proponujemy w SOR-ach, przy pomocy zastosowania algorytmów optymalizacyjnych, tylko wskażą niektóre wąskie gardła i poprawią ich wykorzystanie, jednak nie rozwiążą problemów, których przyczyny leżą poza Szpitalnymi Oddziałami Ratunkowymi.

Podobnie jak wiele innych szpitali, CSK powinno uruchamiać plany działania w przypadku wystąpienia znacznego zatłoczenia na SOR-ze (np. odwoływać planowe przyjęcia i operacje oraz zapobiegać przeniesieniom). Jeśli takie plany awaryjne są wdrażane dopiero po fakcie, a działania są opóźnione, ich wpływ jest ograniczony, a skuteczność co najwyżej umiarkowana. Modele dynamiczne omówione w innych dokumentach mogą być bardzo skuteczne w wywoływaniu wymaganej reakcji w odpowiednim czasie.

Rezultatem tego modułu projektu jest zestaw algorytmów opracowanych specjalnie z myślą o osiągnięciu następującego celu: umożliwienie przeprowadzania okresowych analiz i prognoz zapotrzebowania oraz wynikających z nich obliczeń wydajności oddziału SOR, tak aby jego stała wydajność (stanowiska, łóżka, sprzęt...) była wystarczająca do pokrycia zapotrzebowania w szczytowych okresach w nadchodzących latach oraz aby można było dostosować wydajność personelu do wymaganego poziomu usług z maksymalną skutecznością.

Aby osiągnąć ten cel, opracowaliśmy kilka algorytmów, które w zintegrowany sposób dostarczają wymaganych wyników. Poniższy diagram ilustruje zależności między pięcioma algorytmami:



Relationship between the  
algorithms described in this  
document

Relationship between the algorithms described in this document	Zależności między algorytmami opisanymi w tym dokumencie
Hospital ED historical data	Dane historyczne dotyczące szpitalnego oddziału SOR
Forecasting future attendances	Prognozowanie przyjęć w przyszłości
Forecast day attendances	Prognozowanie liczby przyjęć w skali doby
Forecast hourly seasonality	Prognozowanie godzinowej sezonowości
Establish threshold alarms	Ustalenie alarmów progowych
Resources productivity estimations	Oszacowanie wydajności zasobów
Shift regulations and personnel availability	Przepisy dotyczące zmian i dostępność personelu
Fixed resources capacity	Wydajność zasobów stałych
Bed requirements	Wymagania dotyczące łóżek
Shift optimization	Optymalizacja zmian
Personnel	Wydajność
resources capacity	zasobów kadrowych
Personnel requirements	Wymagania dotyczące personelu
Optimized personnel rosters	Zoptymalizowane grafiki pracowników
Allocation of personnel to shifts	Przydzielanie pracowników do zmian
Personnel preferences for shifts	Preferencje pracowników dotyczące zmian

Prognozowanie zapotrzebowania i zarządzanie wydajnością to skomplikowane zadania dla służb ratownictwa medycznego ze względu na niepewność, złożone zależności i duże narażenie na kontakt z innymi ludźmi. Opublikowane badania nie wykazują zintegrowanych rozwiązań dla tych zadań. Dlatego celem naszych badań jest przedstawienie wyników analizy danych z oddziału SOR dużego szpitala, aby pokazać możliwość rutynowego zastosowania zintegrowanego prognozowania i zarządzania wydajnością przy zastosowaniu zaawansowanych narzędzi do badań nad działaniami do danych szpitala CSK MSWiA.

Na podstawie wstępnego algorytmu prognozowania opracowano i wdrożono układ logiczny do zarządzania wydajnością szpitala. Układ ten obejmuje porównanie prognozowanego zapotrzebowania z dostępnymi zasobami medycznymi oraz model symulacji stochastycznej dynamiki systemu w celu oceny wyników różnych konfiguracji obiektów i zasobów. Układ logiczny dostarcza również menedżerom szpitali narzędzia decyzyjnego do określania liczby i rozmieszczenia zasobów medycznych na oddziałach ratunkowych na podstawie analizy kosztów i korzyści zasobów oraz umożliwia poprawę jakości usług. Takie wyniki pomagają w przydzielaniu lekarzy do różnych stanowisk,

określaniu harmonogramu ich pracy i uwzględnianiu dodatkowych lekarzy. Badanie to przyczyniło się do opracowania zintegrowanego rozwiązania zaprojektowanego w celu wdrożenia wyżej wymienionej układu logicznego. Rozwiązanie to łączy prognozowanie, symulację na potrzeby zarządzania wydajnością, projektowanie procesów i wsparcie informatyczne, ułatwiając praktyczne, rutynowe stosowanie złożonych modeli. Integracja wyraźnie uwzględnia rozwiązanie, które ma także zdolność adaptacji, i umożliwia łatwe korzystanie z niego w zmieniających się warunkach.

## 1. Moduł prognozowania

Przepełnienie szpitalnych oddziałów ratunkowych (SOR-ów) stało się istotnym problemem prowadzącym do potencjalnego wzrostu kosztów opieki zdrowotnej, powodującym zwiększony poziom stresu wśród personelu i pacjentów na SOR-ach, a co najważniejsze - wpływającym negatywnie na wyniki leczenia pacjentów. Jednym z aspektów tego problemu jest trudność w przewidywaniu czasu i nasilenia przepełnienia. Możliwość przewidywania nasilenia przepełnienia, zwłaszcza z dokładnością co do godziny, mogłaby znacznie usprawnić pracę na SOR-ze. W tym celu oceniamy, jak różne modele sprawdzają się w prognozowaniu liczby pacjentów przybywających na SOR.

### METODOLOGIA

Wybraliśmy trzy matematyczne modele prognostyczne w oparciu o wcześniej opracowane przez nas testy, odniesienia znalezione w literaturze oraz dostępność i możliwość obliczenia ich za pomocą programu R Project do obliczeń statystycznych. Poprzednie testy przeprowadzono na podstawie danych dotyczących przyjęć na oddziały ratunkowe w Hospital Clinic de Barcelona od stycznia 2017 r. do grudnia 2018 r.

Te trzy modele to model addytywny, model sieci neuronowej i model wygładzania wykładniczego.

Model addytywny definiuje się w następujący sposób:

$$Dem_i = a + b \times i + \sum_{j=1}^{12} c_j \times m_j + \sum_{k=1}^7 d_k \times w_k + \varepsilon$$

gdzie:

- $a$  jest stałą
- $b$  to tendencja z dnia  $i$
- $c(j)$  oznacza czynnik z miesiąca  $j$
- $m(j)$  jest zmienną fikcyjną równą 1, w przypadku miesiąca  $j$ , i 0 w pozostałych przypadkach
- $d(k)$  oznacza współczynnik dnia powszedniego  $k$
- $w(k)$  jest zmienną fikcyjną równą 1, w przypadku dnia powszedniego, i 0 w pozostałych przypadkach

Głównymi zaletami modelu addytywnego są jego prostota, zrozumienie przez specjalistów niebędących matematykami oraz łatwość obliczania za pomocą technik regresji liniowej. Współczynniki  $a$ ,  $b$ ,  $c(j)$  i  $d(k)$  uzyskane w wyniku regresji liniowej są poddawane testom statystycznym w celu wyłonienia tych, które są różne od zera, a zatem reprezentatywne statystycznie i mogą być wykorzystane w modelu.



Model sieci neuronowej jest tworzony przy użyciu funkcji `nnetar` pakietu `forecast` w R. Oblicza ona sieci neuronowe typu feed-forward z pojedynczą warstwą ukrytą i opóźnionymi danymi wejściowymi do prognozowania jednowariantowych szeregów czasowych.

Do sieci neuronowej typu feed-forward dopasowuje się opóźnione wartości  $y$  jako dane wejściowe i pojedynczą warstwą ukrytą z węzłami dot. wielkości. Dane wejściowe dotyczą opóźnień od 1 do  $p$  oraz opóźnień od  $m$  do  $mP$ , gdzie  $m$ =częstotliwość( $y$ ). Jeśli podano  $xreg$ , jego kolumny są również używane jako dane wejściowe. Jeśli w  $y$  lub  $xreg$  brakuje wartości, odpowiadające im wiersze (oraz wszystkie inne, które zależą od nich jako opóźnienia) są pomijane w dopasowaniu. Dopasowuje się łącznie kilka sieci, każdą z losowymi wagami początkowymi. Są one następnie uśredniane przy obliczaniu prognoz. Sieć jest trenowana do prognozowania jednoetapowego. Prognozy wieloetapowe są obliczane rekurencyjnie. Więcej informacji można znaleźć na stronie <https://otexts.com/fpp3/nnetar.html>

Model wygładzania wykładniczego jest konstruowany za pomocą funkcji `es` pakietu `smooth` w R. Umożliwia ona skonstruowanie modelu wygładzania wykładniczego (znanego również jako ETS), poprzez wybieranie najbardziej odpowiedni spośród 30 możliwych. Wykorzystuje on algorytm rozgałęzień i ograniczeń do utworzenia puli modeli do sprawdzenia, a następnie konstruuje model o najniższym kryterium informacyjnym. Więcej szczegółów można znaleźć w pracy Hyndman et al.(2008) oraz na stronie <https://forecasting.svetunkov.ru/en/category/r-en/smooth/es-function/>

Dlatego w przypadku każdego modelu zastosowaliśmy tę samą procedurę. Obliczyliśmy dzienną liczbę przyjęć na oddział ratunkowy w okresie od stycznia 2017 r. do grudnia 2019 r. na podstawie danych dostarczonych przez Hospital Clinic de Barcelona. Na podstawie tych danych obliczyliśmy dwie wersje modelu: dynamiczną i stacjonarną.

Wersja dynamiczna jest obliczana dla okresu 12 miesięcy w celu utworzenia prognoz dla kolejnych dwóch miesięcy. Przedział czasowy jest przesuwany o jeden miesiąc do przodu, aby ponownie obliczyć model i jego prognozy. Pierwszy przedział czasowy obejmuje okres od 01.01.2017 r. do 31.12.2018 r. i służy do prognozowania dziennych przyjęć dla stycznia i lutego 2019 r. Kolejny przedział czasowy obejmuje okres od 01.02.2017 r. do 31.01.2019 r. i służy do prognozowania dziennych przyjęć dla lutego i marca 2019 r. Ostatni obejmuje okres od 01.11.2017 r. do 31.10.2019 r. i dotyczy prognoz dla listopada i grudnia 2019 r. W ten sposób otrzymujemy 11 zestawów dwumiesięcznych prognoz dziennych.

Dla każdego zestawu 11 dwumiesięcznych prognoz dziennych obliczamy dwa kluczowe wskaźniki wydajności: średni błąd kwadratowy (ang. mean square error, MSE) i średni bezwzględny błąd procentowy (ang. mean absolute percentage error, MAPE). Jako błąd, czyli przewidywana wartość dzienna minus wartość rzeczywista, MSE oblicza się jako średnią kwadratów błędów. Jako bezwzględny błąd procentowy, czyli wartość bezwzględną błędu podzieloną przez wartość rzeczywistą, MAPE oblicza się jako średnią bezwzględnych błędów procentowych. W ten sposób uzyskujemy 11 wartości MSE i MAPE, wykorzystując je do obliczenia ich średnich.

Wersja stacjonarna modelu jest obliczana z przedziałem czasowym 12 miesięcy w celu prognozowania kolejnych 12 miesięcy. Przedział czasowy obejmuje okres od 01.01.2017 r. do 31.12.2018 r. i służy do prognozowania dziennych przyjęć w 2019 r. Obliczamy MSE i MAPE na podstawie prognoz modelu i dziennych przyjęć na rok 2019.

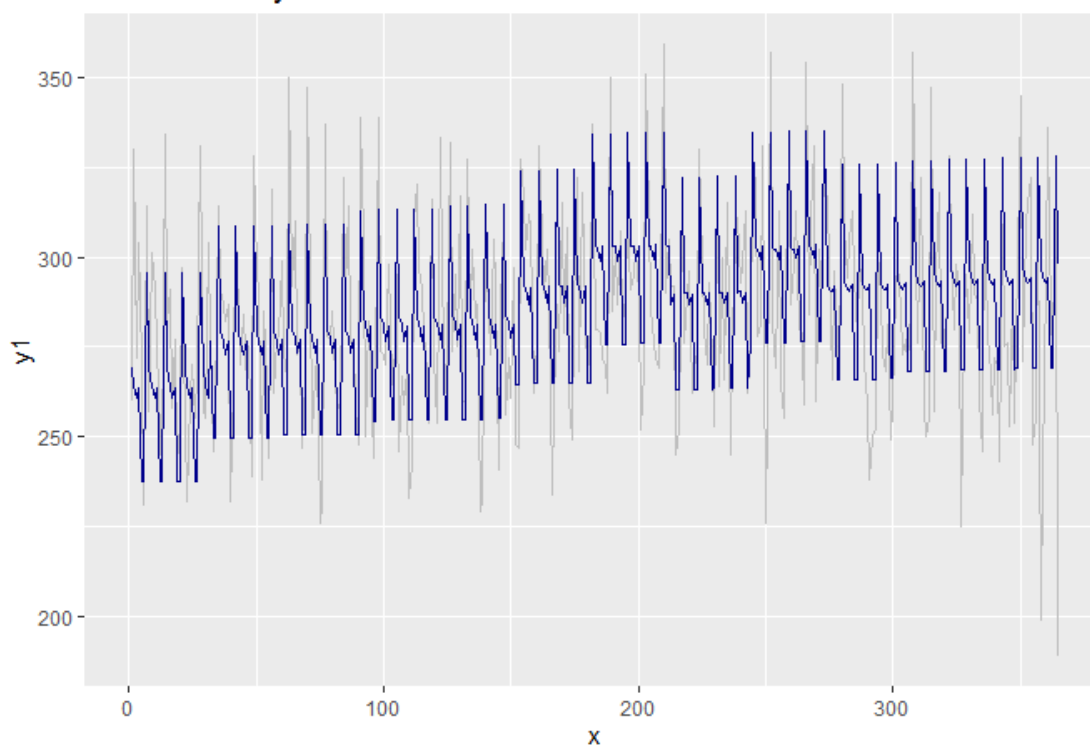
Modele i ich wersje porównuje się za pomocą wskaźników MSE i MAPE, uznając za lepsze te modele i wersje, które mają mniejsze wartości MSE i MAPE.

## WYNIKI

### Model addytywny

#### Dynamiczny

ADDITIVE daily model as union of 11 models



ADDITIVE daily model as union of 11 models

ADDYTYWNY model dzienny jako połączenie 11 modeli

```
## #####
## Model 1
## Start date: 2017-01-01 End date: 2019-01-01
## Checkup dates
## Start date: 2019-01-01 End date: 2019-03-01
##
##
##      &nbsp;      Estimate   Std. Error   t value   Pr(>|t|)
## :-----: :-----: :-----: :-----:
## ***(Intercept)** 250.2      1.606      155.8      0
##      **m1**      -12.62     2.919      -4.324     1.745e-05
##      **m2**      -8.832     3.054      -2.892     0.003941
##      **m7**       11.48     2.919       3.934     9.159e-05
##      **m9**       7.358     2.962       2.484     0.0132
##      **w1**       57.95     2.59      22.38     1.346e-84
##      **w2**       31.36     2.598      12.07     1.105e-30
##      **w3**       26.28     2.598      10.11     1.438e-22
##      **w4**       23.32     2.598       8.974     2.459e-18
##      **w5**       26.2      2.598      10.08     1.854e-22
##
```



```
##
## -----
## Observations      Residual Std. Error    $R^2$      Adjusted $R^2$
## -----
##           730              21.65          0.4457      0.4387
## -----
##
## Table: Fitting linear model: n ~ m1 + m2 + m7 + m9 + w1 + w2 + w3 + w4
+ w5
##
## #####
## Model 2
## Start date: 2017-02-01 End date: 2019-02-01
## Checkup dates
## Start date: 2019-02-01 End date: 2019-04-01
##
##
## |      &nbsp;      | Estimate | Std. Error | t value | Pr(>|t|) |
## | :-----: | :-----: | :-----: | :-----: | :-----: |
## | ***(Intercept)** | 249.6 | 1.537 | 162.4 | 0 |
## | **m7** | 11.96 | 2.899 | 4.124 | 4.151e-05 |
## | **w1** | 58.9 | 2.622 | 22.47 | 3.413e-85 |
## | **w2** | 29.83 | 2.622 | 11.38 | 1.036e-27 |
## | **w3** | 26.97 | 2.614 | 10.32 | 2.215e-23 |
## | **w4** | 23.43 | 2.614 | 8.963 | 2.66e-18 |
## | **w5** | 26.73 | 2.622 | 10.2 | 6.684e-23 |
##
##
## -----
## Observations      Residual Std. Error    $R^2$      Adjusted $R^2$
## -----
##           730              21.83          0.43      0.4253
## -----
##
## Table: Fitting linear model: n ~ m7 + w1 + w2 + w3 + w4 + w5
##
## #####
## Model 3
## Start date: 2017-03-01 End date: 2019-03-01
## Checkup dates
## Start date: 2019-03-01 End date: 2019-05-01
##
##
## |      &nbsp;      | Estimate | Std. Error | t value | Pr(>|t|) |
## | :-----: | :-----: | :-----: | :-----: | :-----: |
## | ***(Intercept)** | 250.4 | 1.529 | 163.8 | 0 |
## | **m7** | 11.36 | 2.885 | 3.938 | 8.999e-05 |
## | **w1** | 58.72 | 2.609 | 22.51 | 1.904e-85 |
## | **w2** | 29.29 | 2.609 | 11.23 | 4.438e-27 |
## | **w3** | 26.97 | 2.601 | 10.37 | 1.367e-23 |
## | **w4** | 22.73 | 2.601 | 8.742 | 1.59e-17 |
## | **w5** | 26.79 | 2.609 | 10.27 | 3.435e-23 |
##
##
```

```
##
## -----
## Observations      Residual Std. Error    $R^2$      Adjusted $R^2$
## -----
##           730              21.72          0.4299      0.4252
## -----
##
## Table: Fitting linear model: n ~ m7 + w1 + w2 + w3 + w4 + w5
##
## #####
## Model 4
## Start date: 2017-04-01 End date: 2019-04-01
## Checkup dates
## Start date: 2019-04-01 End date: 2019-06-01
##
##
##      &nbsp;      | Estimate | Std. Error | t value | Pr(>|t|) |
## :-----: :-----: :-----: :-----: :-----:
## ***(Intercept)**      | 246.7 | 2.099 | 117.5 | 0 |
## **day**      | 0.009436 | 0.00381 | 2.476 | 0.0135 |
## **m7**      | 12.6 | 2.893 | 4.354 | 1.53e-05 |
## **m9**      | 7.882 | 2.919 | 2.7 | 0.007092 |
## **w1**      | 58.56 | 2.585 | 22.66 | 3.118e-86 |
## **w2**      | 28.97 | 2.585 | 11.21 | 5.493e-27 |
## **w3**      | 26.88 | 2.585 | 10.4 | 1.089e-23 |
## **w4**      | 22.03 | 2.585 | 8.523 | 9.063e-17 |
## **w5**      | 26.14 | 2.585 | 10.11 | 1.424e-22 |
##
##
## -----
## Observations      Residual Std. Error    $R^2$      Adjusted $R^2$
## -----
##           730              21.55          0.4381      0.4319
## -----
##
## Table: Fitting linear model: n ~ day + m7 + m9 + w1 + w2 + w3 + w4 + w5
##
## #####
## Model 5
## Start date: 2017-05-01 End date: 2019-05-01
## Checkup dates
## Start date: 2019-05-01 End date: 2019-07-01
##
##
##      &nbsp;      | Estimate | Std. Error | t value | Pr(>|t|) |
## :-----: :-----: :-----: :-----: :-----:
## ***(Intercept)**      | 245.9 | 2.214 | 111.1 | 0 |
## **day**      | 0.01039 | 0.003917 | 2.652 | 0.008171 |
## **m6**      | 6.819 | 2.988 | 2.282 | 0.0228 |
## **m7**      | 13.23 | 2.923 | 4.525 | 7.057e-06 |
## **m9**      | 8.472 | 2.93 | 2.891 | 0.003953 |
## **w1**      | 59.48 | 2.566 | 23.18 | 3.012e-89 |
## **w2**      | 28.72 | 2.566 | 11.19 | 6.328e-27 |
```

```
##      **w3**      27.1      2.574      10.53      3.367e-24
##      **w4**      22.32     2.574      8.671      2.823e-17
##      **w5**      26.21     2.574      10.18      7.642e-23
##
##
## -----
##      Observations      Residual Std. Error      $R^2$      Adjusted $R^2$
## -----
##           730              21.43              0.4479              0.441
## -----
##
## Table: Fitting linear model: n ~ day + m6 + m7 + m9 + w1 + w2 + w3 + w4
+ w5
##
## #####
## Model 6
## Start date: 2017-06-01 End date: 2019-06-01
## Checkup dates
## Start date: 2019-06-01 End date: 2019-08-01
##
##
##      &nbsp;      Estimate      Std. Error      t value      Pr(>|t|)
## :-----: :-----: :-----: :-----: :-----:
##      ***(Intercept)**      245.1          2.26          108.4          0
##      **day**      0.01382      0.004014          3.442      0.0006099
##      **m6**      7.419          3.022          2.455      0.01433
##      **m7**      13.76          2.949          4.665      3.671e-06
##      **m9**      8.765          2.941          2.98      0.002978
##      **w1**      59.26          2.563          23.12      6.928e-89
##      **w2**      27.55          2.563          10.75      4.244e-25
##      **w3**      26.91          2.564          10.5      4.365e-24
##      **w4**      22.17          2.555          8.676      2.71e-17
##      **w5**      26.85          2.555          10.51      3.983e-24
##
##
## -----
##      Observations      Residual Std. Error      $R^2$      Adjusted $R^2$
## -----
##           730              21.34              0.4479              0.441
## -----
##
## Table: Fitting linear model: n ~ day + m6 + m7 + m9 + w1 + w2 + w3 + w4
+ w5
##
## #####
## Model 7
## Start date: 2017-07-01 End date: 2019-07-01
## Checkup dates
## Start date: 2019-07-01 End date: 2019-09-01
##
##
##      &nbsp;      Estimate      Std. Error      t value      Pr(>|t|)
## :-----: :-----: :-----: :-----: :-----:
##
```

```
## | ** (Intercept) ** | 243.9 | 2.167 | 112.6 | 0 |
## | ** day ** | 0.01909 | 0.003914 | 4.877 | 1.325e-06 |
## | ** m7 ** | 14.26 | 2.938 | 4.853 | 1.495e-06 |
## | ** m9 ** | 8.901 | 2.927 | 3.041 | 0.002442 |
## | ** w1 ** | 58.61 | 2.552 | 22.97 | 4.9e-88 |
## | ** w2 ** | 27.29 | 2.552 | 10.69 | 7.159e-25 |
## | ** w3 ** | 27.11 | 2.552 | 10.62 | 1.396e-24 |
## | ** w4 ** | 23.21 | 2.552 | 9.095 | 9.031e-19 |
## | ** w5 ** | 27.32 | 2.552 | 10.71 | 6.343e-25 |
##
##
## -----
## Observations Residual Std. Error $R^2$ Adjusted $R^2$
## -----
## 730 21.28 0.449 0.4429
## -----
##
## Table: Fitting linear model: n ~ day + m7 + m9 + w1 + w2 + w3 + w4 + w5
##
## #####
## Model 8
## Start date: 2017-08-01 End date: 2019-08-01
## Checkup dates
## Start date: 2019-08-01 End date: 2019-10-01
##
##
## | &nbsp; | Estimate | Std. Error | t value | Pr(>|t|) |
## | :-----: | :-----: | :-----: | :-----: | :-----: |
## | ** (Intercept) ** | 244.3 | 2.074 | 117.8 | 0 |
## | ** day ** | 0.01985 | 0.003896 | 5.094 | 4.47e-07 |
## | ** m7 ** | 11.06 | 2.902 | 3.81 | 0.0001509 |
## | ** m9 ** | 9.008 | 2.914 | 3.091 | 0.002068 |
## | ** w1 ** | 59.05 | 2.545 | 23.21 | 2.058e-89 |
## | ** w2 ** | 26.69 | 2.537 | 10.52 | 3.418e-24 |
## | ** w3 ** | 26.82 | 2.536 | 10.57 | 2.158e-24 |
## | ** w4 ** | 23.67 | 2.544 | 9.303 | 1.603e-19 |
## | ** w5 ** | 26.53 | 2.544 | 10.43 | 8.082e-24 |
##
##
## -----
## Observations Residual Std. Error $R^2$ Adjusted $R^2$
## -----
## 730 21.18 0.459 0.453
## -----
##
## Table: Fitting linear model: n ~ day + m7 + m9 + w1 + w2 + w3 + w4 + w5
##
## #####
## Model 9
## Start date: 2017-09-01 End date: 2019-09-01
## Checkup dates
## Start date: 2019-09-01 End date: 2019-11-01
##
```

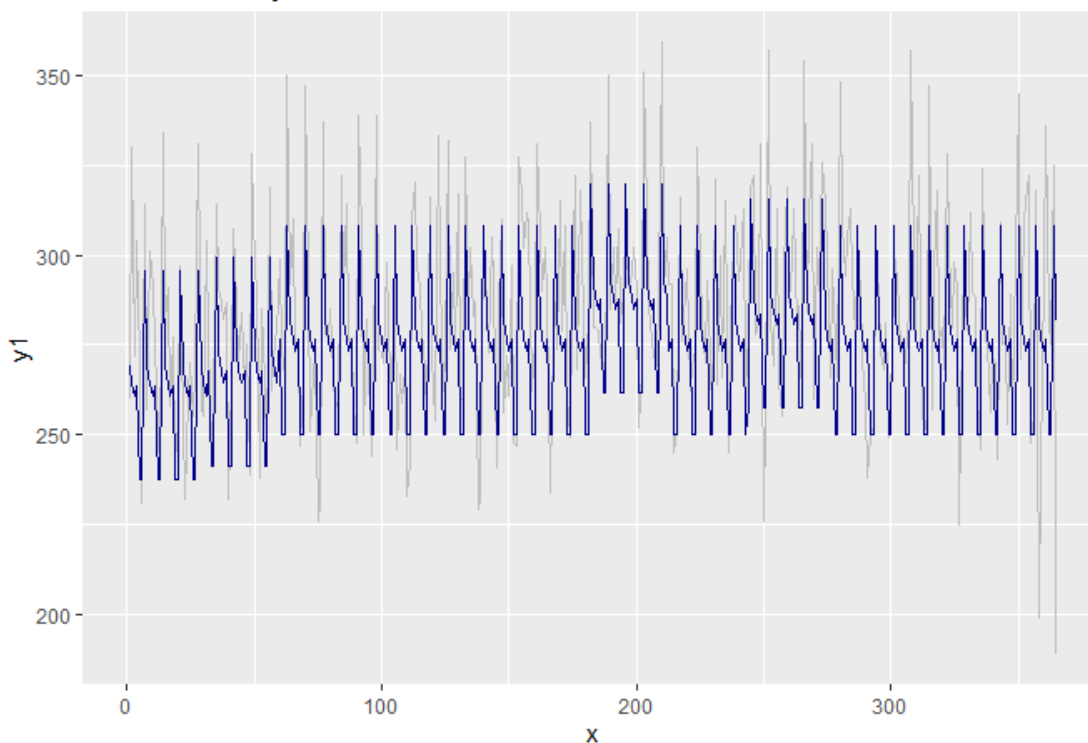
```
##
##      &nbsp;      Estimate   Std. Error   t value   Pr(>|t|)
## :-----: :-----: :-----: :-----:
##  **(Intercept)**      243.9        2.047       119.2         0
##      **day**      0.02301      0.003824        6.018      2.817e-09
##      **m7**       10.73         2.82         3.805      0.0001539
##      **m9**        9.603        2.889         3.324      0.0009324
##      **w1**       58.54         2.496        23.45      7.904e-91
##      **w2**       26.33         2.496        10.55       2.7e-24
##      **w3**       25.97         2.496        10.41      1.011e-23
##      **w4**       23.23         2.496         9.309      1.523e-19
##      **w5**       26.78         2.488        10.76      3.718e-25
##
##
## -----
##  Observations   Residual Std. Error   $R^2$   Adjusted $R^2$
## -----
##      730             20.8           0.4668     0.4609
## -----
##
## Table: Fitting linear model: n ~ day + m7 + m9 + w1 + w2 + w3 + w4 + w5
##
## #####
## Model 10
## Start date: 2017-10-01 End date: 2019-10-01
## Checkup dates
## Start date: 2019-10-01 End date: 2019-12-01
##
##
##      &nbsp;      Estimate   Std. Error   t value   Pr(>|t|)
## :-----: :-----: :-----: :-----:
##  **(Intercept)**      245.1        1.964       124.8         0
##      **day**      0.02048      0.003821         5.36      1.124e-07
##      **m7**       11.05         2.818         3.92      9.682e-05
##      **m9**       12.49         2.91         4.292      2.014e-05
##      **w1**       59.77         2.486        24.04      3.027e-94
##      **w2**       27.06         2.494        10.85      1.67e-25
##      **w3**       25.43         2.494        10.19      6.886e-23
##      **w4**       24.81         2.494         9.947      6.238e-22
##      **w5**       26.37         2.494        10.57      2.223e-24
##
##
## -----
##  Observations   Residual Std. Error   $R^2$   Adjusted $R^2$
## -----
##      730             20.78           0.4843     0.4786
## -----
##
## Table: Fitting linear model: n ~ day + m7 + m9 + w1 + w2 + w3 + w4 + w5
##
## #####
## Model 11
## Start date: 2017-11-01 End date: 2019-11-01
```

```
## Checkup dates
## Start date: 2019-11-01 End date: 2020-01-01
##
##
##      &nbsp;      | Estimate | Std. Error | t value | Pr(>|t|) |
## :-----: | :-----: | :-----: | :-----: | :-----: |
## ***(Intercept)** | 245.5 | 1.94 | 126.5 | 0 |
## **day** | 0.02174 | 0.003707 | 5.865 | 6.833e-09 |
## **m7** | 11.12 | 2.761 | 4.029 | 6.201e-05 |
## **m9** | 12.52 | 2.84 | 4.41 | 1.194e-05 |
## **w1** | 58.62 | 2.466 | 23.77 | 1.135e-92 |
## **w2** | 28.64 | 2.466 | 11.61 | 1.07e-28 |
## **w3** | 24.86 | 2.458 | 10.11 | 1.45e-22 |
## **w4** | 23.58 | 2.458 | 9.591 | 1.388e-20 |
## **w5** | 25.04 | 2.466 | 10.15 | 1.004e-22 |
##
##
## -----
## Observations      Residual Std. Error      $R^2$      Adjusted $R^2$
## -----
##      730              20.53              0.4823      0.4766
## -----
##
## Table: Fitting linear model: n ~ day + m7 + m9 + w1 + w2 + w3 + w4 + w5
```

Model	MSE	MAPE
Model 1	469	0,067
Model 2	271	0,047
Model 3	329	0,052
Model 4	347	0,055
Model 5	380	0,056
Model 6	294	0,049
Model 7	234	0,044
Model 8	353	0,055
Model 9	474	0,063
Model 10	393	0,056
Model 11	805	0,070
Average	395	0,056

Stacjonarny

### ADDITIVE daily model 2017-2018 checked in 2019



ADDITIVE daily model 2017-2018 checked in 2019

ADDYTYWNY model dzienny 2017-2018 sprawdzony w 2019 r.

```
##
##      &nbsp;      Estimate   Std. Error   t value   Pr(>|t|)
## :-----: :-----: :-----: :-----:
## ** (Intercept) **    250.2      1.606     155.8      0
##      **m1**          -12.62     2.919     -4.324    1.745e-05
##      **m2**          -8.832     3.054     -2.892    0.003941
##      **m7**           11.48     2.919      3.934    9.159e-05
##      **m9**           7.358     2.962      2.484     0.0132
##      **w1**           57.95      2.59     22.38    1.346e-84
##      **w2**           31.36      2.598     12.07    1.105e-30
##      **w3**           26.28      2.598     10.11    1.438e-22
##      **w4**           23.32      2.598      8.974    2.459e-18
##      **w5**           26.2       2.598     10.08    1.854e-22
##
##
## -----
## Observations   Residual Std. Error   $R^2$   Adjusted $R^2$
## -----
##      730             21.65           0.4457   0.4387
## -----
##
## Table: Fitting linear model: n ~ m1 + m2 + m7 + m9 + w1 + w2 + w3 + w4 + w5
```

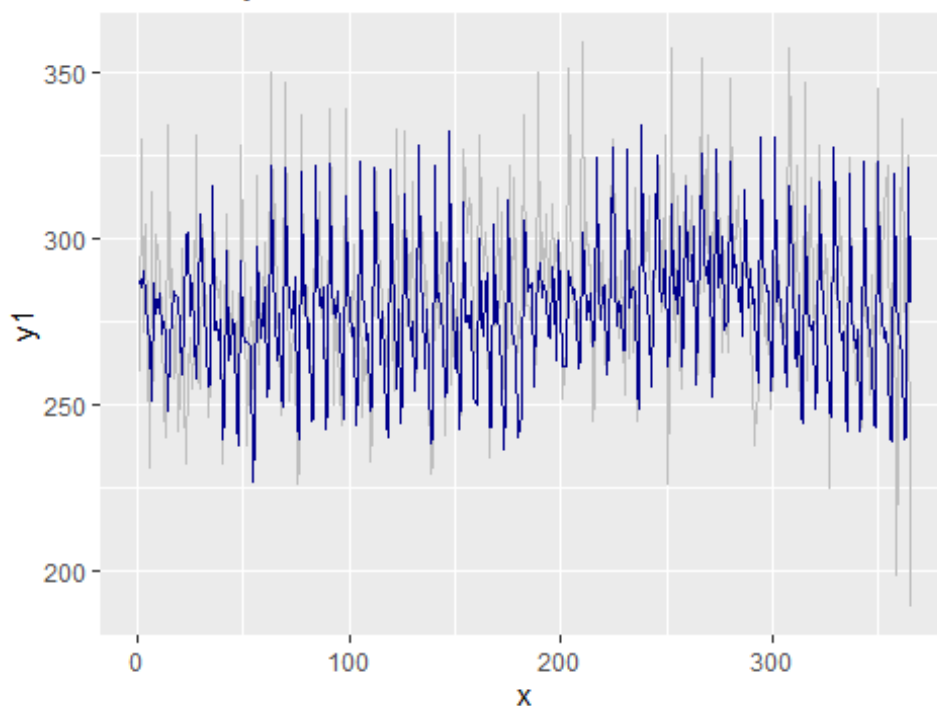


## MSE: 500.9011 MAPE 0.06412862

Model sieci neuronowej

Dynamiczny

ANN daily model as union of 11 models



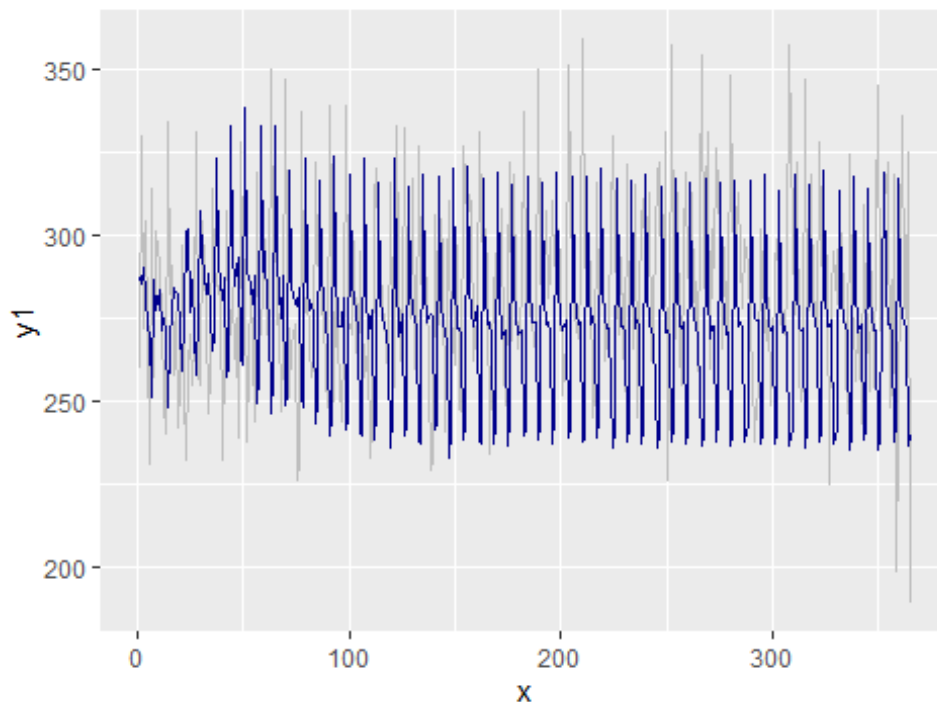
ANN daily model as union of 11 models

Dzienny model ANN jako połączenie 11 modeli

##	Model	MSE	MAPE
##	:-----:	:-----:	:-----:
##	Model 1	1.374	0,109
##	Model 2	382	0,058
##	Model 3	310	0,052
##	Model 4	390	0,059
##	Model 5	609	0,072
##	Model 6	610	0,078
##	Model 7	473	0,060
##	Model 8	525	0,066
##	Model 9	524	0,063
##	Model 10	421	0,061
##	Model 11	944	0,086
##	Average	597	0,07

## Stacjonarny

### ANN daily model 2017-2018 checked in 2019



ANN daily model 2017-2018 checked in 2019

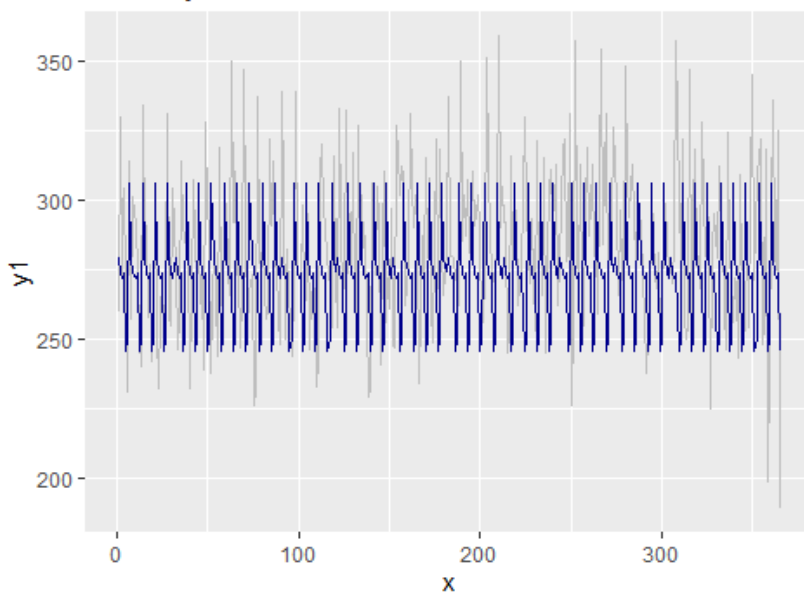
Model dzienny ANN 2017-2018 sprawdzony w 2019 r.

## MSE: 1876.212 MAPE 0.1276797

Wyglądanie wykładowe

## Dynamiczny

### ES daily model as union of 11 models



ES daily model as union of 11 models

Model dzienny ES jako połączenie 11 modeli

```
## #####
## Model 1
## Start date: 2017-01-01 End date: 2019-01-01
## Checkup dates
## Start date: 2019-01-01 End date: 2019-03-01
## Time elapsed: 3.44 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0894 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0061
## Error standard deviation: 0.0787
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6558.710 6559.016 6604.641 6605.649
## #####
## Model 2
## Start date: 2017-02-01 End date: 2019-02-01
## Checkup dates
## Start date: 2019-02-01 End date: 2019-04-01
## Time elapsed: 3.23 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0699 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0061
## Error standard deviation: 0.0784
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6557.404 6557.710 6603.334 6604.343
## #####
## Model 3
## Start date: 2017-03-01 End date: 2019-03-01
## Checkup dates
## Start date: 2019-03-01 End date: 2019-05-01
## Time elapsed: 3.03 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0581 0.0000
## Initial values were optimised.
```

```
##
## Loss function type: MSE; Loss function value: 0.006
## Error standard deviation: 0.0782
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6558.069 6558.376 6604.000 6605.009
## #####
## Model 4
## Start date: 2017-04-01 End date: 2019-04-01
## Checkup dates
## Start date: 2019-04-01 End date: 2019-06-01
## Time elapsed: 3.03 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
##  alpha  gamma
## 0.0613 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.006
## Error standard deviation: 0.078
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6552.280 6552.586 6598.210 6599.219
## #####
## Model 5
## Start date: 2017-05-01 End date: 2019-05-01
## Checkup dates
## Start date: 2019-05-01 End date: 2019-07-01
## Time elapsed: 3.04 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
##  alpha  gamma
## 0.0537 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0059
## Error standard deviation: 0.0776
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6548.683 6548.989 6594.614 6595.622
## #####
```

```
## Model 6
## Start date: 2017-06-01 End date: 2019-06-01
## Checkup dates
## Start date: 2019-06-01 End date: 2019-08-01
## Time elapsed: 3.39 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0465 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0059
## Error standard deviation: 0.0772
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6543.838 6544.144 6589.768 6590.777
## #####
## Model 7
## Start date: 2017-07-01 End date: 2019-07-01
## Checkup dates
## Start date: 2019-07-01 End date: 2019-09-01
## Time elapsed: 3.35 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0537 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0059
## Error standard deviation: 0.0771
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6542.037 6542.343 6587.968 6588.976
## #####
## Model 8
## Start date: 2017-08-01 End date: 2019-08-01
## Checkup dates
## Start date: 2019-08-01 End date: 2019-10-01
## Time elapsed: 3.25 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0547 0.0000
## Initial values were optimised.
##
```

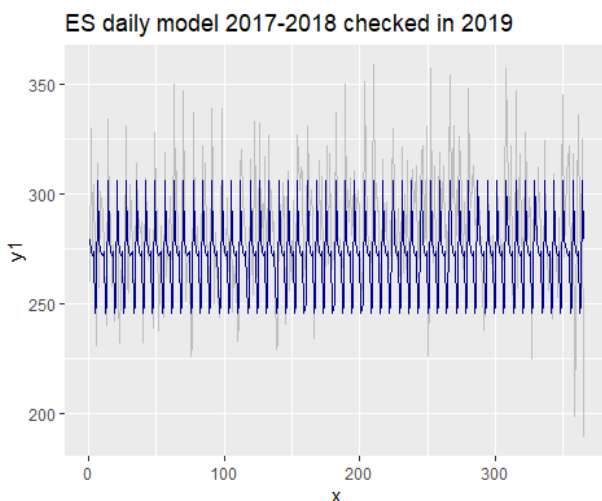
```
## Loss function type: MSE; Loss function value: 0.0058
## Error standard deviation: 0.0766
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6534.466 6534.772 6580.396 6581.405
## #####
## Model 9
## Start date: 2017-09-01 End date: 2019-09-01
## Checkup dates
## Start date: 2019-09-01 End date: 2019-11-01
## Time elapsed: 3.43 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
##  alpha  gamma
## 0.0421 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0056
## Error standard deviation: 0.0754
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6513.563 6513.869 6559.494 6560.502
## #####
## Model 10
## Start date: 2017-10-01 End date: 2019-10-01
## Checkup dates
## Start date: 2019-10-01 End date: 2019-12-01
## Time elapsed: 3.43 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
##  alpha  gamma
## 0.0441 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0056
## Error standard deviation: 0.0753
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6516.047 6516.353 6561.978 6562.986
## #####
## Model 11
```

```
## Start date: 2017-11-01 End date: 2019-11-01
## Checkup dates
## Start date: 2019-11-01 End date: 2020-01-01
## Time elapsed: 3.32 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0579 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0054
## Error standard deviation: 0.0743
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6499.918 6500.224 6545.849 6546.858
```

##	Model	MSE	MAPE
##	:-----:	:-----:	:-----:
##	Model 1	334	0,054
##	Model 2	1.243	0,109
##	Model 3	1.279	0,107
##	Model 4	1.116	0,099
##	Model 5	1.161	0,104
##	Model 6	1.483	0,119
##	Model 7	1.177	0,104
##	Model 8	1.672	0,119
##	Model 9	1.560	0,118
##	Model 10	569	0,071
##	Model 11	1.245	0,107
##	Average	1.167	0,101

Stacjonarny





ES daily model 2017-2018 checked in 2019

Model dzienny ES 2017-2018 sprawdzony w 2019 r.

```
## Time elapsed: 3.44 seconds
## Model estimated: ETS(MNM)
## Persistence vector g:
## alpha gamma
## 0.0894 0.0000
## Initial values were optimised.
##
## Loss function type: MSE; Loss function value: 0.0061
## Error standard deviation: 0.0787
## Sample size: 730
## Number of estimated parameters: 10
## Number of provided parameters: 1
## Number of degrees of freedom: 720
## Information criteria:
##      AIC      AICc      BIC      BICc
## 6558.710 6559.016 6604.641 6605.649
```

```
## MSE: 567.548    MAPE 0.06866617
```

## WNIOSKI

W poniższej tabeli zestawiono wskaźniki MSE i MAPE dla każdego modelu:

MODEL	MSE	MAPE
Dynamiczny model addytywny	395	0,056
Stacjonarny model addytywny	501	0,064
Dynamiczny model sieci neuronowej	597	0,070
Stacjonarny model sieci neuronowej	1,876	0,128
Dynamiczny model wygładzania wykładniczego	1,167	0,101
Stacjonarny model wygładzania wykładniczego	567	0.068



W oparciu o uzyskane wyniki, do prognozowania przyjęć na oddziały ratunkowe sugerujemy wybór dynamicznego modelu addytywnego. Model ten ma najniższe wartości MSE i MAPE oraz jest najłatwiejszy do obliczenia i interpretacji przez osoby nie zajmujące się matematyką. Niemniej jednak, sugerujemy częste szacowanie przedstawionych przez nas oraz innych modeli, ponieważ nowe dane mogą zachowywać się inaczej, sprawiając, że inne modele będą lepiej pasować.

## Kod R

### Dynamiczny model addytywny

```
rm(list = ls())

pckgs<-c("forecast", "readxl", "digest", "stringr", "summarytools", "dplyr",
         "Hmisc", "chron", "xts", "tidyverse", "tseries", "seastests", "smooth",
         "sarima", "writexl", "MLmetrics", "lubridate", "fable", "Metrics",
         "nnfor", "rlist", "gridExtra" )
pckgs2Install<-pckgs[!(pckgs %in% library())$results[,1]]
pckgs2Load<-pckgs[!(pckgs %in% (.packages()))]
for(pckg in pckgs2Install) {install.packages(pckg)}
for(pckg in pckgs2Load) {library(pckg,character.only = TRUE)}

#Home Laptop
setwd("C:/Users/mguer/Dropbox/Docs Varsovia per Manel/1. Demand forecasting")

#laptop toshiba
#setwd("C:/Users/User1/Dropbox/Docs Varsovia per Manel/1. Demand forecasting")

#Home Laptop
pathdata<-"C:/Users/mguer/Dropbox/Docs Varsovia per Manel/1. Demand forecasting/HClinic Data"

#laptop toshiba
#pathdata<-"C:/Users/User1/Dropbox/Docs Varsovia per Manel/1. Demand forecasting/HClinic Data"
#Data of 2019
filename<-"Clean_Selected_data_No_Maternitat_2019.RData"
df_clean19<-get(load(file=paste(pathdata, filename, sep="/")))
#Data of 2017 and 2018
filename<-"Clean_Selected_data_No_Maternitat_20172018.RData"
df_clean1718<-get(load(file=paste(pathdata, filename, sep="/")))

#Data of 2019
filename<-"Clean_Selected_data_No_Maternitat_2019.RData"
df_clean19<-get(load(file=paste(pathdata, filename, sep="/")))
#Data of 2017 and 2018
filename<-"Clean_Selected_data_No_Maternitat_20172018.RData"
df_clean1718<-get(load(file=paste(pathdata, filename, sep="/")))

df_clean<-rbind.data.frame(df_clean1718,df_clean19)
ts_all<-df_clean %>% group_by(Data) %>% summarise(n=n())

dates_mini<-c("2017-01-01", "2017-02-01", "2017-03-01", "2017-04-01", "2017-05-01", "2017-06-01", "2017-07-01", "2017-08-01", "2017-09-01", "2017-10-01", "2017-11-01")
dates_mfin<-c("2019-01-01", "2019-02-01", "2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01")
dates_cini<-c("2019-01-01", "2019-02-01", "2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01")
dates_cfin<-c("2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01", "2019-12-01", "2020-01-01")

model_list<-list()

for(i in 1:11){
  ts_model<-ts_all %>% filter(Data>=dates_mini[i] & Data<dates_mfin[i])
  ts_check<-ts_all %>% filter(Data>=dates_cini[i] & Data<dates_cfin[i])

  ts<-ts_model
  ts$yday<-lubridate::yday(ts$Data, week_start = 1)
  ts$month<-month(ts$Data)
  ts$day<-seq(1:730)

  ts$m1 <- ifelse(ts$month == 1, 1, 0)
  ts$m2 <- ifelse(ts$month == 2, 1, 0)
  ts$m3 <- ifelse(ts$month == 3, 1, 0)
  ts$m4 <- ifelse(ts$month == 4, 1, 0)
  ts$m5 <- ifelse(ts$month == 5, 1, 0)
  ts$m6 <- ifelse(ts$month == 6, 1, 0)
  ts$m7 <- ifelse(ts$month == 7, 1, 0)
```

```
ts$m8 <- ifelse(ts$month == 8, 1, 0)
ts$m9 <- ifelse(ts$month == 9, 1, 0)
ts$m10 <- ifelse(ts$month == 10, 1, 0)
ts$m11<- ifelse(ts$month == 11, 1, 0)
ts$m12 <- ifelse(ts$month == 12, 1, 0)

ts$w1 <- ifelse(ts$wday == 1, 1, 0)
ts$w2 <- ifelse(ts$wday == 2, 1, 0)
ts$w3 <- ifelse(ts$wday == 3, 1, 0)
ts$w4 <- ifelse(ts$wday == 4, 1, 0)
ts$w5 <- ifelse(ts$wday == 5, 1, 0)
ts$w6 <- ifelse(ts$wday == 6, 1, 0)
ts$w7 <- ifelse(ts$wday == 7, 1, 0)

if(i==1){
  model <- lm(n ~ m1 + m2 + m7 + m9 + w1 + w2 + w3 + w4 + w5, data=ts)
}

if(i==2 | i==3){
  model <- lm(n ~ m7+ w1 + w2 + w3 + w4 + w5 , data=ts)
}

if(i==5 | i==6){
  model <- lm(n ~ day + m6 + m7 + m9 + w1 + w2 + w3 + w4 + w5 , data=ts)
}

if(i==4 | i>6){
  model <- lm(n ~ day + m7 + m9 + w1 + w2 + w3 + w4 + w5 , data=ts)
}

model_list<-list.append(model_list,model)
}

#Predictions for two months
ts_all$day<-seq(1: length(ts_all$Data))

predicchio_list<-list()
mse_list<-list()
mape_list<-list()

##### i=1 #####3
i<-1
#Calculate predictions for two months
#Make predictors
model<-model_list[[1]]
ts_check<-ts_all %>% filter(Data==dates_cini[i] & Data<dates_cfin[i])
ts<-ts_check
ts$wday<-lubridate::wday(ts$Data, week_start = 1)
ts$month<-month(ts$Data)

pos_ini<-which(dates_cini[i]==ts_all$Data)
pos_fin<-which(dates_cfin[i]==ts_all$Data)-1
ts$day<-unname(unlist(ts_all[pos_ini:pos_fin,"day"]))

ts$m1 <- ifelse(ts$month == 1, 1, 0)
ts$m2 <- ifelse(ts$month == 2, 1, 0)
ts$m7 <- ifelse(ts$month == 7, 1, 0)
ts$m9 <- ifelse(ts$month == 9, 1, 0)

ts$w1 <- ifelse(ts$wday == 1, 1, 0)
ts$w2 <- ifelse(ts$wday == 2, 1, 0)
ts$w3 <- ifelse(ts$wday == 3, 1, 0)
ts$w4 <- ifelse(ts$wday == 4, 1, 0)
ts$w5 <- ifelse(ts$wday == 5, 1, 0)

#Calculate predictions for two months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, 6:14]))
predicchio<-data.frame(Data,month,n)
#Error prediction in 2019
mse_lm<-mse(predicchio$n,ts$n)
```

```

mape_lm<-mape(prediccion,ts$n)

prediccion_list<-list.append(prediccion_list,prediccion)
mse_list<-list.append(mse_list,mse_lm)
mape_list<-list.append(mape_list,mape_lm)

##### i = 2 #####

i<-2
#Calculate predictions for two months
#Make predictors
model<-model_list[[2]]
ts_check<-ts_all %>% filter(Data>=dates_cini[i] & Data<dates_cfin[i])
ts<-ts_check
ts$yday<-lubridate::yday(ts$Data, week_start = 1)
ts$month<-month(ts$Data)

pos_ini<-which(dates_cini[i]==ts_all$Data)
pos_fin<-which(dates_cfin[i]==ts_all$Data)-1
ts$day<-unname(unlist(ts_all[pos_ini:pos_fin,"day"]))

ts$m7 <- ifelse(ts$month == 7, 1, 0)

ts$w1 <- ifelse(ts$yday == 1, 1, 0)
ts$w2 <- ifelse(ts$yday == 2, 1, 0)
ts$w3 <- ifelse(ts$yday == 3, 1, 0)
ts$w4 <- ifelse(ts$yday == 4, 1, 0)
ts$w5 <- ifelse(ts$yday == 5, 1, 0)

#Calculate predictions for two months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, 6:11]))
prediccion<-data.frame(Data,month,n)
#Error prediction in 2019
mse_lm<-mse(prediccion,ts$n)
mape_lm<-mape(prediccion,ts$n)

prediccion_list<-list.append(prediccion_list,prediccion)
mse_list<-list.append(mse_list,mse_lm)
mape_list<-list.append(mape_list,mape_lm)

##### i = 3 #####

i<-3
#Calculate predictions for two months
#Make predictors
model<-model_list[[3]]
ts_check<-ts_all %>% filter(Data>=dates_cini[i] & Data<dates_cfin[i])
ts<-ts_check
ts$yday<-lubridate::yday(ts$Data, week_start = 1)
ts$month<-month(ts$Data)

pos_ini<-which(dates_cini[i]==ts_all$Data)
pos_fin<-which(dates_cfin[i]==ts_all$Data)-1
ts$day<-unname(unlist(ts_all[pos_ini:pos_fin,"day"]))

ts$m7 <- ifelse(ts$month == 7, 1, 0)

ts$w1 <- ifelse(ts$yday == 1, 1, 0)
ts$w2 <- ifelse(ts$yday == 2, 1, 0)
ts$w3 <- ifelse(ts$yday == 3, 1, 0)
ts$w4 <- ifelse(ts$yday == 4, 1, 0)
ts$w5 <- ifelse(ts$yday == 5, 1, 0)

#Calculate predictions for two months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, 6:11]))
prediccion<-data.frame(Data,month,n)
#Error prediction in 2019
mse_lm<-mse(prediccion,ts$n)
mape_lm<-mape(prediccion,ts$n)

```

```

predicchio_list<-list.append(predicchio_list,predicchio)
mse_list<-list.append(mse_list,mse_lm)
mape_list<-list.append(mape_list,mape_lm)

##### i = 4 #####

i<-4
#Calculate predictions for two months
#Make predictors
model<-model_list[[4]]
ts_check<-ts_all %>% filter(Data>=dates_cini[i] & Data<dates_cfin[i])
ts<-ts_check
ts$yday<-lubridate::yday(ts$Data, week_start = 1)
ts$month<-month(ts$Data)

pos_ini<-which(dates_cini[i]==ts_all$Data)
pos_fin<-which(dates_cfin[i]==ts_all$Data)-1
ts$day<-unname(unlist(ts_all[pos_ini:pos_fin,"day"]))

ts$m7 <- ifelse(ts$month == 7, 1, 0)
ts$m9 <- ifelse(ts$month == 9, 1, 0)

ts$w1 <- ifelse(ts$yday == 1, 1, 0)
ts$w2 <- ifelse(ts$yday == 2, 1, 0)
ts$w3 <- ifelse(ts$yday == 3, 1, 0)
ts$w4 <- ifelse(ts$yday == 4, 1, 0)
ts$w5 <- ifelse(ts$yday == 5, 1, 0)

#Calculate predictions for two months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, c(3,6:12)]))
predicchio<-data.frame(Data,month,n)
#Error prediction in 2019
mse_lm<-mse(predicchio$n,ts$n)
mape_lm<-mape(predicchio$n,ts$n)

predicchio_list<-list.append(predicchio_list,predicchio)
mse_list<-list.append(mse_list,mse_lm)
mape_list<-list.append(mape_list,mape_lm)

##### i = 5 #####

i<-5
#Calculate predictions for two months
#Make predictors
model<-model_list[[5]]
ts_check<-ts_all %>% filter(Data>=dates_cini[i] & Data<dates_cfin[i])
ts<-ts_check
ts$yday<-lubridate::yday(ts$Data, week_start = 1)
ts$month<-month(ts$Data)

pos_ini<-which(dates_cini[i]==ts_all$Data)
pos_fin<-which(dates_cfin[i]==ts_all$Data)-1
ts$day<-unname(unlist(ts_all[pos_ini:pos_fin,"day"]))

ts$m6 <- ifelse(ts$month == 6, 1, 0)
ts$m7 <- ifelse(ts$month == 7, 1, 0)
ts$m9 <- ifelse(ts$month == 9, 1, 0)

ts$w1 <- ifelse(ts$yday == 1, 1, 0)
ts$w2 <- ifelse(ts$yday == 2, 1, 0)
ts$w3 <- ifelse(ts$yday == 3, 1, 0)
ts$w4 <- ifelse(ts$yday == 4, 1, 0)
ts$w5 <- ifelse(ts$yday == 5, 1, 0)

#Calculate predictions for two months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, c(3,6:13)]))
predicchio<-data.frame(Data,month,n)

```

```
#Error prediction in 2019
mse_lm<-mse(prediccion,ts$n)
mape_lm<-mape(prediccion,ts$n)

prediccion_list<-list.append(prediccion_list,prediccion)
mse_list<-list.append(mse_list,mse_lm)
mape_list<-list.append(mape_list,mape_lm)

##### i = 6 #####

i<-6
#Calculate predictions for two months
#Make predictors
model<-model_list[[6]]
ts_check<-ts_all %>% filter(Data==dates_cini[i] & Data<dates_cfin[i])
ts<-ts_check
ts$yday<-lubridate::yday(ts$Data, week_start = 1)
ts$month<-month(ts$Data)

pos_ini<-which(dates_cini[i]==ts_all$Data)
pos_fin<-which(dates_cfin[i]==ts_all$Data)-1
ts$day<-unname(unlist(ts_all[pos_ini:pos_fin,"day"]))

ts$m6 <- ifelse(ts$month == 6, 1, 0)
ts$m7 <- ifelse(ts$month == 7, 1, 0)
ts$m9 <- ifelse(ts$month == 9, 1, 0)

ts$w1 <- ifelse(ts$yday == 1, 1, 0)
ts$w2 <- ifelse(ts$yday == 2, 1, 0)
ts$w3 <- ifelse(ts$yday == 3, 1, 0)
ts$w4 <- ifelse(ts$yday == 4, 1, 0)
ts$w5 <- ifelse(ts$yday == 5, 1, 0)

#Calculate predictions for two months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, c(3,6:13)]))
prediccion<-data.frame(Data,month,n)
#Error prediction in 2019
mse_lm<-mse(prediccion,ts$n)
mape_lm<-mape(prediccion,ts$n)

prediccion_list<-list.append(prediccion_list,prediccion)
mse_list<-list.append(mse_list,mse_lm)
mape_list<-list.append(mape_list,mape_lm)

##### i = 7 a 11 #####

for(i in 7:11){

  #Calculate predictions for two months
  #Make predictors
  model<-model_list[[i]]
  ts_check<-ts_all %>% filter(Data==dates_cini[i] & Data<dates_cfin[i])
  ts<-ts_check
  ts$yday<-lubridate::yday(ts$Data, week_start = 1)
  ts$month<-month(ts$Data)

  pos_ini<-which(dates_cini[i]==ts_all$Data)
  if(i==11){
    pos_fin<-which("2019-12-31"==ts_all$Data)}
  else{
    pos_fin<-which(dates_cfin[i]==ts_all$Data)-1}

  ts$day<-unname(unlist(ts_all[pos_ini:pos_fin,"day"]))

  ts$m7 <- ifelse(ts$month == 7, 1, 0)
  ts$m9 <- ifelse(ts$month == 9, 1, 0)

  ts$w1 <- ifelse(ts$yday == 1, 1, 0)
  ts$w2 <- ifelse(ts$yday == 2, 1, 0)
  ts$w3 <- ifelse(ts$yday == 3, 1, 0)
  ts$w4 <- ifelse(ts$yday == 4, 1, 0)
  ts$w5 <- ifelse(ts$yday == 5, 1, 0)
```



```
#Calculate predictions for two months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, c(3,6:12)]))
prediccio<-data.frame(Data,month,n)
#Error prediction in 2019
mse_lm<-mse(prediccio$n,ts$n)
mape_lm<-mape(prediccio$n,ts$n)

prediccio_list<-list.append(prediccio_list,prediccio)
mse_list<-list.append(mse_list,mse_lm)
mape_list<-list.append(mape_list,mape_lm)
}

prediccio_all<-NULL
for(i in 1:11){
  prediccio<-prediccio_list[[i]]
  pred_aux<-prediccio %>% filter(month(Data)==i)
  prediccio_all<-rbind.data.frame(prediccio_all,pred_aux)
}

prediccio<-prediccio_list[[11]]
pred_aux<-prediccio %>% filter(month(Data)==12)
prediccio_all<-rbind.data.frame(prediccio_all,pred_aux)
library(pander)
panderOptions('knitr.auto.asis', FALSE)
cat("\n")
for(i in 1:11)
{
  cat("#####", "\n")
  cat("Model", i, "\n")
  cat("Start date:", dates_mini[i], "End date:", dates_mfin[i], "\n")
  cat("Checkup dates", "\n")
  cat("Start date:", dates_cini[i], "End date:", dates_cfin[i], "\n")
  pander(summary(model_list[[i]]), style='rmarkdown')
}

cat("\n")

mapes<-unlist(mape_list[])
mses<-unlist(mse_list[])
method<-c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6", "Model 7", "Model 8", "Model 9", "Model 10", "Model 11")
met_metrics<-cbind(method,format(round(mses,digits=0), big.mark=".", decimal.mark=","),format(round(mapes,digits=3),decimal.mark=","))
met_metrics<-rbind.data.frame(met_metrics, c("Average",format(round(mean(mses),digits=0), big.mark=".", decimal.mark=","),format(round(mean(mapes),digits=3),decimal.mark=",")))
colnames(met_metrics)<-c("Model","MSE","MAPE")

pander(met_metrics, style='rmarkdown')

x <-seq(1:365)
y1 <- ts_all[ts_all$Data>="2019-01-01","n"]
y2 <- prediccio_all$n
df <- data.frame(x,y1,y2)
colnames(df)<-c("x","y1","y2")

ggplot(df, aes(x)) +
  geom_line(aes(y=y1), colour="grey") + # first layer
  geom_line(aes(y=y2), colour="darkblue")+
  ggtitle("ADDITIVE daily model as union of 11 models") # second layer
```

## Stacjonarny model addytywny

```
##### i=1 #####
i<-1
#Calculate predictions for two months
#Make predictors
model<-model_list[[1]]
```

```
ts_check<-ts_all %>% filter(Data>="2019-01-01")
ts<-ts_check
ts$yday<-lubridate::yday(ts$Data, week_start = 1)
ts$month<-month(ts$Data)
ts$day<-ts_check$day

ts$m1 <- ifelse(ts$month == 1, 1, 0)
ts$m2 <- ifelse(ts$month == 2, 1, 0)
ts$m7 <- ifelse(ts$month == 7, 1, 0)
ts$m9 <- ifelse(ts$month == 9, 1, 0)

ts$w1 <- ifelse(ts$yday == 1, 1, 0)
ts$w2 <- ifelse(ts$yday == 2, 1, 0)
ts$w3 <- ifelse(ts$yday == 3, 1, 0)
ts$w4 <- ifelse(ts$yday == 4, 1, 0)
ts$w5 <- ifelse(ts$yday == 5, 1, 0)

#Calculate predictions for 12 months
Data<-ts$Data
month<-ts$month
n<-unname(predict(model, ts[, 6:14]))
prediccio<-data.frame(Data,month,n)
#Error prediction in 2019
mse_lm<-mse(prediccio$ts$ts$yday)
mape_lm<-mape(prediccio$ts$ts$yday)

cat("#####", "\n")
## #####
cat("Model 1", "\n")
## Model 1
cat("Start date:", dates_mini[1], "End date:", dates_mfin[1], "\n")
## Start date: 2017-01-01 End date: 2019-01-01
cat("Checkup dates", "\n")
## Checkup dates
cat("Start date:", "01-01-2019", "End date:", "31-12-2019", "\n")
## Start date: 01-01-2019 End date: 31-12-2019
pander(summary(model), style='rmarkdown')

cat("\n")
cat("MSE:", mse_lm, " MAPE", mape_lm)
## MSE: 500.9011 MAPE 0.06412862
cat("\n")
#
x <-seq(1:365)
y1 <- ts_all[ts_all$Data>="2019-01-01", "n"]
y2 <- prediccio$ts$ts$yday
df <- data.frame(x,y1,y2)
colnames(df)<-c("x", "y1", "y2")

ggplot(df, aes(x)) + # basic graphical object
  geom_line(aes(y=y1), colour="grey") + # first Layer
  geom_line(aes(y=y2), colour="darkblue")+
  ggtitle("ADDITIVE daily model 2017-2018 checked in 2019") # second Layer
```

## Dynamiczny model sieci neuronowej

```
rm(list = ls())

pckgs<-c("forecast", "readxl", "digest", "stringr", "summarytools", "dplyr",
        "Hmisc", "chron", "xts", "tidyverse", "tseries", "seastests", "smooth",
        "sarima", "writexl", "MLmetrics", "lubridate", "fable", "Metrics",
        "nnfor", "rlist", "gridExtra")
pckgs2Install<-pckgs[!(pckgs %in% library()$results[,1])]
pckgs2Load<-pckgs[!(pckgs %in% (.packages()))]
for(pckg in pckgs2Install) {install.packages(pckg)}
for(pckg in pckgs2Load) {library(pckg,character.only = TRUE)}

#home Laptop
setwd("C:/Users/mguer/Dropbox/Docs Varsovia per Manel/1. Demand forecasting")

#Laptop toshiba
#setwd("C:/Users/User1/Dropbox/Docs Varsovia per Manel/1. Demand forecasting")

#home Laptop
pathdata<- "C:/Users/mguer/Dropbox/Docs Varsovia per Manel/1. Demand forecasting/HCLinic Data"

#Laptop toshiba
#pathdata<- "C:/Users/User1/Dropbox/Docs Varsovia per Manel/1. Demand forecasting/HCLinic Data"

#Data of 2019
filename<- "Clean_Selected_data_No_Maternitat_2019.RData"
df_clean19<-get(load(file=paste(pathdata, filename, sep="/")))
#Data of 2017 and 2018
filename<- "Clean_Selected_data_No_Maternitat_20172018.RData"
df_clean1718<-get(load(file=paste(pathdata, filename, sep="/")))

df_clean<-rbind.data.frame(df_clean1718,df_clean19)
ts_all<-df_clean %>% group_by(Data) %>% summarise(n=n())

dates_mini<-c("2017-01-01", "2017-02-01", "2017-03-01", "2017-04-01", "2017-05-01", "2017-06-01", "2017-07-01", "2017-08-01", "2017-09-01", "2017-10-01", "2017-11-01")
dates_mfin<-c("2019-01-01", "2019-02-01", "2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01")
dates_cini<-c("2019-01-01", "2019-02-01", "2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01")
dates_cfin<-c("2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01", "2019-12-01", "2020-01-01")

model_list<-list()

for(i in 1:11){
  ts_model<-ts_all %>% filter(Data>=dates_mini[i] & Data<dates_mfin[i])

  myts<-ts_model$n
  model<-nnetar(myts)

  model_list<-list.append(model_list,model)
}

prediccio_list<-list()
mse_list<-list()
mape_list<-list()

for(i in 1:11){
  ts_check<-ts_all %>% filter(Data>=dates_cini[i] & Data<dates_cfin[i])

  n_days<-nrow(ts_check)

  prediccio<-forecast(model_list[[i]], h=n_days)

  mse_ann<-mse(prediccio$mean,ts_check$n)
  mape_ann<-mape(prediccio$mean,ts_check$n)

  pred<-cbind.data.frame(prediccio$mean,ts_check$Data)
  colnames(pred)<-c("n", "Data")
}
```

```

predicchio_list<-list.append(predicchio_list,pred)
mse_list<-list.append(mse_list,mse_ann)
mape_list<-list.append(mape_list,mape_ann)
}

predicchio_all<-NULL
for(i in 1:11){
  predicchio<-predicchio_list[[i]]
  pred_aux<-predicchio %>% filter(month(Data)==i)
  predicchio_all<-rbind.data.frame(predicchio_all,pred_aux)
}

predicchio<-predicchio_list[[11]]
pred_aux<-predicchio %>% filter(month(Data)==12)
predicchio_all<-rbind.data.frame(predicchio_all,pred_aux)

library(pander)
panderOptions('knitr.auto.asis', FALSE)
cat("\n")
for(i in 1:11)
{
  cat("#####", "\n")
  cat("Model", i, "\n")
  cat("Start date:", dates_mini[i], "End date:", dates_mfin[i], "\n")
  cat("Checkup dates", "\n")
  cat("Start date:", dates_cini[i], "End date:", dates_cfin[i], "\n")
  print(model_list[[i]]$model, "\n")
  pander(summary(model_list[[i]]), style='rmarkdown')
}

mapes<-unlist(mape_list[])
mses<-unlist(mse_list[])
method<-c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6", "Model 7", "Model 8", "Model 9", "Model 10", "Model 11")
met_metrics<-cbind(method,format(round(mses,digits=0), big.mark=".", decimal.mark="),format(round(mapes,digits=3),decimal.mark=","))
met_metrics<-rbind.data.frame(met_metrics, c("Average",format(round(mean(mses),digits=0), big.mark=".", decimal.mark="),format(round(mean(mapes),digits=3),decimal.mark=",")))
colnames(met_metrics)<-c("Model", "MSE", "MAPE")

pander(met_metrics, style='rmarkdown')

x <-seq(1:365)
y1 <- ts_all[ts_all$Data>="2019-01-01", "n"]
y2 <- predicchio_all$n
df <- data.frame(x,y1,y2)
colnames(df)<-c("x", "y1", "y2")

ggplot(df, aes(x)) +
  geom_line(aes(y=y1), colour="grey") + # first layer
  geom_line(aes(y=y2), colour="darkblue")+
  ggtitle("ANN daily model as union of 11 models") # second layer

```

## Stacjonarny model sieci neuronowej

```
##### i=1 #####
i<-1

ts_check<-ts_all %>% filter(Data>="2019-01-01")
n_days<-nrow(ts_check)
prediccio<-forecast(model_list[[i]], h=n_days)

mse_ann<-mse(prediccio$mean,ts_check$n)
mape_ann<-mape(prediccio$mean,ts_check$n)

cat("#####", "\n")
## #####
cat("Model 1", "\n")
## Model 1
cat("Start date:", dates_mini[1], "End date:", dates_mfin[1], "\n")
## Start date: 2017-01-01 End date: 2019-01-01
cat("Checkup dates", "\n")
## Checkup dates
cat("Start date:", "01-01-2019", "End date:", "31-12-2019", "\n")
## Start date: 01-01-2019 End date: 31-12-2019
print(model_list[[i]]$model, "\n")
pander(summary(model_list[[i]]), style='rmarkdown')
| 2 | -none- | call |
cat("\n")
cat("MSE:", mse_ann, " MAPE", mape_ann)
cat("\n")
#
x <-seq(1:365)
y1 <- ts_all[ts_all$Data>="2019-01-01", "n"]
y2 <- prediccio$mean
df <- data.frame(x,y1,y2)
colnames(df)<-c("x", "y1", "y2")

ggplot(df, aes(x)) + # basic graphical object
  geom_line(aes(y=y1, colour="grey")) + # first layer
  geom_line(aes(y=y2, colour="darkblue"))+
  ggtitle("ANN daily model 2017-2018 checked in 2019") # second layer
```

## Dynamiczny model wygładzania wykładniczego

```
rm(list = ls())

pckgs<-c("forecast", "readxl", "digest", "stringr", "summarytools", "dplyr",
        "Hmisc", "chron", "xts", "tidyverse", "tseries", "seastests", "smooth",
        "sarima", "writexl", "MLmetrics", "lubridate", "fable", "Metrics",
        "nnfor", "rlist", "gridExtra")
pckgs2Install<-pckgs[!(pckgs %in% library()$results[,1])]
pckgs2Load<-pckgs[!(pckgs %in% (.packages()))]
for(pckg in pckgs2Install) {install.packages(pckg)}
for(pckg in pckgs2Load) {library(pckg,character.only = TRUE)}

#Home Laptop
setwd("C:/Users/mguer/Dropbox/Docs Varsovia per Manel/1. Demand forecasting")

#Laptop toshiba
#setwd("C:/Users/User1/Dropbox/Docs Varsovia per Manel/1. Demand forecasting")

#Home Laptop
pathdata<-"C:/Users/mguer/Dropbox/Docs Varsovia per Manel/1. Demand forecasting/HCLinic Data"

#Laptop toshiba
#pathdata<-"C:/Users/User1/Dropbox/Docs Varsovia per Manel/1. Demand forecasting/HCLinic Data"

#install.packages("devtools")
#library(devtools)
#devtools::install_github("kassambara/easyGgplot2")
#library(easyGgplot2)
#Data of 2019
filename<-"Clean_Selected_data_No_Maternitat_2019.RData"
df_clean19<-get(load(file=paste(pathdata, filename, sep="/")))
#Data of 2017 and 2018
filename<-"Clean_Selected_data_No_Maternitat_20172018.RData"
df_clean1718<-get(load(file=paste(pathdata, filename, sep="/")))

df_clean<-rbind.data.frame(df_clean1718,df_clean19)
ts_all<-df_clean %>% group_by(Data) %>% summarise(n=n())

dates_mini<-c("2017-01-01", "2017-02-01", "2017-03-01", "2017-04-01", "2017-05-01", "2017-06-01", "2017-07-01", "2017-08-01", "2017-09-01", "2017-10-01", "2017-11-01")
dates_mfin<-c("2019-01-01", "2019-02-01", "2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01")
dates_cini<-c("2019-01-01", "2019-02-01", "2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01")
dates_cfin<-c("2019-03-01", "2019-04-01", "2019-05-01", "2019-06-01", "2019-07-01", "2019-08-01", "2019-09-01", "2019-10-01", "2019-11-01", "2019-12-01", "2020-01-01")

model_list<-list()

for(i in 1:11){
  ts_model<-ts_all %>% filter(Data>=dates_mini[i] & Data<dates_mfin[i])

  myts<-ts_model$n
  myts_season<-ts(myts, frequency=7)
  model<-es(myts_season, loss= "MSE", silent="none", h=365)

  model_list<-list.append(model_list,model)
}

prediccio_list<-list()
mse_list<-list()
mape_list<-list()

for(i in 1:11){
  ts_check<-ts_all %>% filter(Data>=dates_cini[i] & Data<dates_cfin[i])

  n_days<-nrow(ts_check)

  prediccio<-model_list[[1]]$forecast[1:n_days]
```

```
mse_ann<-mse(predicchio,ts_check$n)
mape_ann<-mape(predicchio,ts_check$n)

pred<-cbind.data.frame(predicchio,ts_check$Data)
colnames(pred)<-c("n", "Data")

predicchio_list<-list.append(predicchio_list,pred)
mse_list<-list.append(mse_list,mse_ann)
mape_list<-list.append(mape_list,mape_ann)
}

predicchio_all<-NULL
for(i in 1:11){
  predicchio<-predicchio_list[[i]]
  pred_aux<-predicchio %>% filter(month(Data)==i)
  predicchio_all<-rbind.data.frame(predicchio_all,pred_aux)
}

predicchio<-predicchio_list[[11]]
pred_aux<-predicchio %>% filter(month(Data)==12)
predicchio_all<-rbind.data.frame(predicchio_all,pred_aux)
library(pander)
panderOptions('knitr.auto.asis', FALSE)
cat("\n")
for(i in 1:11)
{
  cat("#####", "\n")
  cat("Model", i, "\n")
  cat("Start date:", dates_mini[i], "End date:", dates_mfin[i], "\n")
  cat("Checkup dates", "\n")
  cat("Start date:", dates_cini[i], "End date:", dates_cfin[i], "\n")
  #print(summary(model_list[[i]]))
  #print(model_list[[i]]$model, "\n")
  #pander(summary(model_list[[i]]$model), style='rmarkdown')
  pander(summary(model_list[[i]]), style='rmarkdown')
  #pander(anova(model_list[[i]]), style='rmarkdown')
}

cat("\n")
mapes<-unlist(mape_list[])
mses<-unlist(mse_list[])
method<-c("Model 1", "Model 2", "Model 3", "Model 4", "Model 5", "Model 6", "Model 7", "Model 8", "Model 9", "Model 10", "Model 11")
met_metrics<-cbind(method,format(round(mses,digits=0), big.mark=".", decimal.mark=","),format(round(mapes,digits=3),decimal.mark=","))
met_metrics<-rbind.data.frame(met_metrics, c("Average",format(round(mean(mses),digits=0), big.mark=".", decimal.mark=","),format(round(mean(mapes),digits=3),decimal.mark=",")))
colnames(met_metrics)<-c("Model", "MSE", "MAPE")

pander(met_metrics, style='rmarkdown')
x <-seq(1:365)
y1 <- ts_all[ts_all$Data>="2019-01-01", "n"]
y2 <- predicchio_all$n
df <- data.frame(x,y1,y2)
colnames(df)<-c("x", "y1", "y2")

ggplot(df, aes(x)) +
  geom_line(aes(y=y1), colour="grey") + # first layer
  geom_line(aes(y=y2), colour="darkblue")+
  ggtitle("ES daily model as union of 11 models") # second layer
```



## Stacjonarny model wygładzania wykładniczego

```
##### i=1 #####
i<-1

ts_check<-ts_all %>% filter(Data>="2019-01-01")
n_days<-nrow(ts_check)
prediccio<-model_list[[i]]$forecast

mse_ann<-mse(prediccio,ts_check$n)
mape_ann<-mape(prediccio,ts_check$n)

cat("#####", "\n")
## #####
cat("Model 1", "\n")
## Model 1
cat("Start date:", dates_mini[1], "End date:", dates_mfin[1], "\n")
## Start date: 2017-01-01 End date: 2019-01-01
cat("Checkup dates", "\n")
## Checkup dates
cat("Start date:", "01-01-2019", "End date:", "31-12-2019", "\n")
## Start date: 01-01-2019 End date: 31-12-2019
#print(model_list[[i]]$model, "\n")
#pander(summary(model_list[[i]]$model), style='rmarkdown')
pander(summary(model_list[[i]]), style='rmarkdown')
cat("\n")
cat("MSE:", mse_ann, " MAPE", mape_ann)
cat("\n")
#
x <-seq(1:365)
y1 <- ts_all[ts_all$Data>="2019-01-01", "n"]
y2 <- prediccio
df <- data.frame(x,y1,y2)
colnames(df)<-c("x", "y1", "y2")

ggplot(df, aes(x)) + # basic graphical object
  geom_line(aes(y=y1), colour="grey") + # first layer
  geom_line(aes(y=y2), colour="darkblue")+
  ggtitle("ES daily model 2017-2018 checked in 2019") # second layer
```

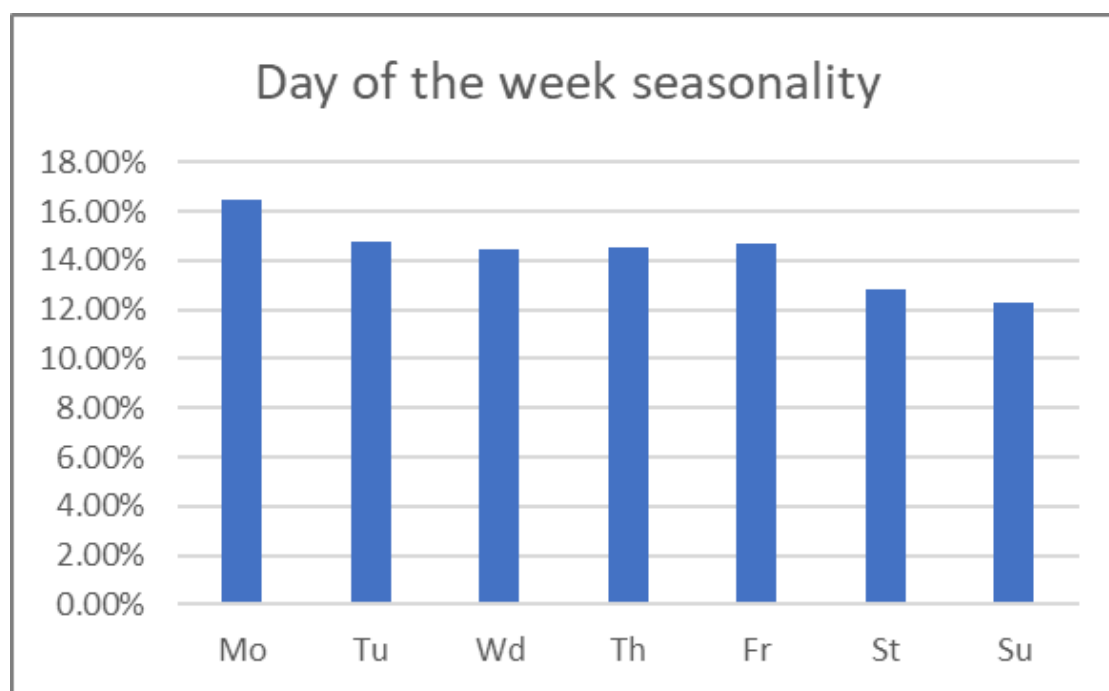
## 2. Wymiarowanie dla zasobów stałych

Algorytm przedstawiony w tej części ma na celu ułatwienie wymiarowania wydajności zasobów stałych w różnych trybach opieki (triage, resuscytacja, opieka medyczna w trybie pilnym, opieka doraźna, szybka ścieżka i obszary obserwacyjne, a także diagnostyka obrazowa, laboratorium i poczekalnie). Ponieważ zasoby te nie jest łatwo zmodyfikować, algorytm musi przeprowadzić analizę maksymalnej wydajności w długim okresie.

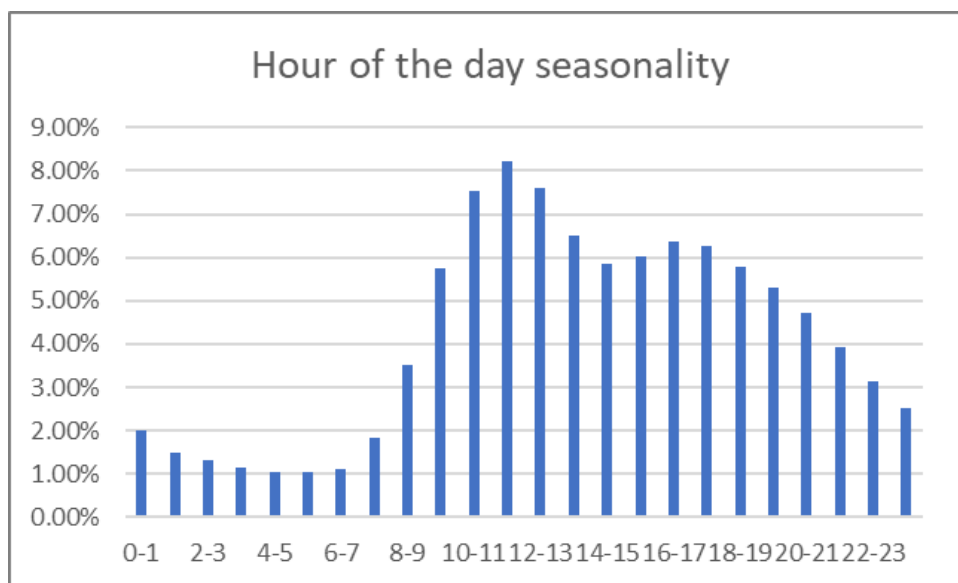
Algorytmy wykorzystują dane dot. przyjęć w szpitalnych SOR-ach z poprzednich lat. Jak opisano w poprzednich etapach tego projektu, przygotowaliśmy algorytmy w taki sposób, aby można je było wykorzystać do przetwarzania danych pochodzących z różnych sytuacji, zarówno z tego samego szpitala, jak i z różnych szpitali. W ten sposób algorytmy można wykorzystać w celu dostosowania wymagań dotyczących wydajności w przypadku zwiększenia zapotrzebowania SOR-u w przyszłym roku, a także można je zastosować do danych z innych szpitali w celu powtórzenia analizy przeprowadzonej dla CSK MSWiA.

Przeanalizowaliśmy prognozowane zapotrzebowanie SOR-u, biorąc pod uwagę zmiany sezonowe w różnych dniach tygodnia i godzinach doby.

Zmienność tę ilustrują dwa poniższe wykresy.



Day of the week seasonality	Sezonowość z pod względem dnia tygodnia
Mo	Pn
Tu	Wt
Wd	Śr
Th	Czw
Fr	Pt
St	Sb
Su	Nd



Hour of the day seasonality

Sezonowość pod względem pory dnia

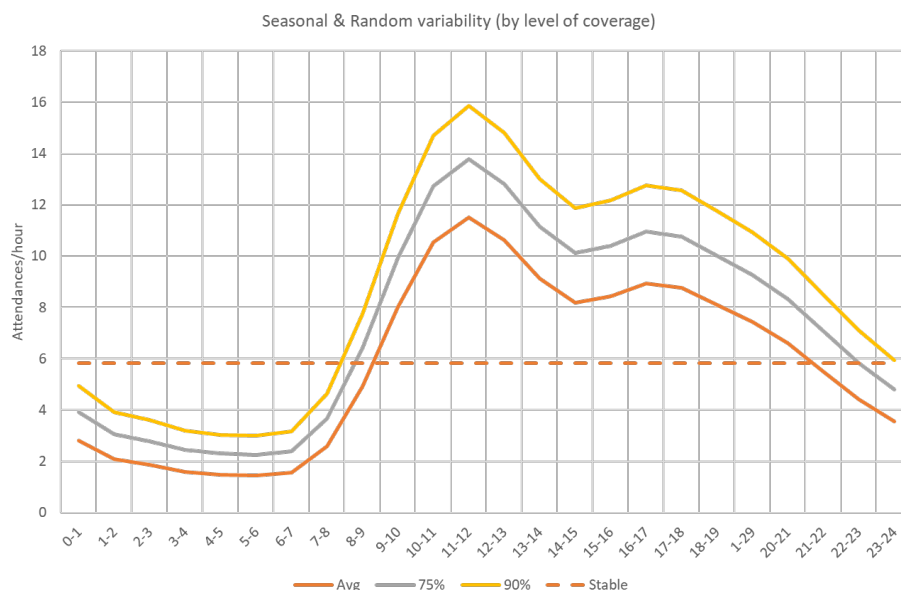
Średnie tygodniowe zapotrzebowanie zostało podzielone na 168 okresów (7 dni po 24 godziny), a średni odsetek przyjęć pacjentów na oddział w tygodniu w każdym z tych okresów jest następujący:

Total		Mo	Tu	We	Th	Fr	Sa	Su
		16.46%	14.73%	14.47%	14.54%	14.65%	12.85%	12.29%
0-1	2.00%	0.33%	0.29%	0.29%	0.29%	0.29%	0.26%	0.25%
1-2	1.48%	0.24%	0.22%	0.21%	0.22%	0.22%	0.19%	0.18%
2-3	1.33%	0.22%	0.20%	0.19%	0.19%	0.19%	0.17%	0.16%
3-4	1.13%	0.19%	0.17%	0.16%	0.16%	0.17%	0.15%	0.14%
4-5	1.05%	0.17%	0.16%	0.15%	0.15%	0.15%	0.14%	0.13%
5-6	1.04%	0.17%	0.15%	0.15%	0.15%	0.15%	0.13%	0.13%
6-7	1.11%	0.18%	0.16%	0.16%	0.16%	0.16%	0.14%	0.14%
7-8	1.84%	0.30%	0.27%	0.27%	0.27%	0.27%	0.24%	0.23%
8-9	3.51%	0.58%	0.52%	0.51%	0.51%	0.51%	0.45%	0.43%
0-19	5.73%	0.94%	0.84%	0.83%	0.83%	0.84%	0.74%	0.70%
10-11	7.53%	1.24%	1.11%	1.09%	1.09%	1.10%	0.97%	0.93%
11-12	8.21%	1.35%	1.21%	1.19%	1.19%	1.20%	1.06%	1.01%
12-13	7.59%	1.25%	1.12%	1.10%	1.10%	1.11%	0.97%	0.93%
13-14	6.52%	1.07%	0.96%	0.94%	0.95%	0.95%	0.84%	0.80%
14-15	5.85%	0.96%	0.86%	0.85%	0.85%	0.86%	0.75%	0.72%
15-16	6.03%	0.99%	0.89%	0.87%	0.88%	0.88%	0.77%	0.74%
16-17	6.38%	1.05%	0.94%	0.92%	0.93%	0.93%	0.82%	0.78%
17-18	6.26%	1.03%	0.92%	0.91%	0.91%	0.92%	0.80%	0.77%
18-19	5.79%	0.95%	0.85%	0.84%	0.84%	0.85%	0.74%	0.71%
19-20	5.31%	0.87%	0.78%	0.77%	0.77%	0.78%	0.68%	0.65%
20-21	4.71%	0.78%	0.69%	0.68%	0.69%	0.69%	0.61%	0.58%
21-22	3.92%	0.65%	0.58%	0.57%	0.57%	0.57%	0.50%	0.48%
22-23	3.15%	0.52%	0.46%	0.46%	0.46%	0.46%	0.40%	0.39%
23-24	2.53%	0.42%	0.37%	0.37%	0.37%	0.37%	0.32%	0.31%
		100.00%						

Total	Ogółem
Mo	Pn
Tu	Wt
We	Śr

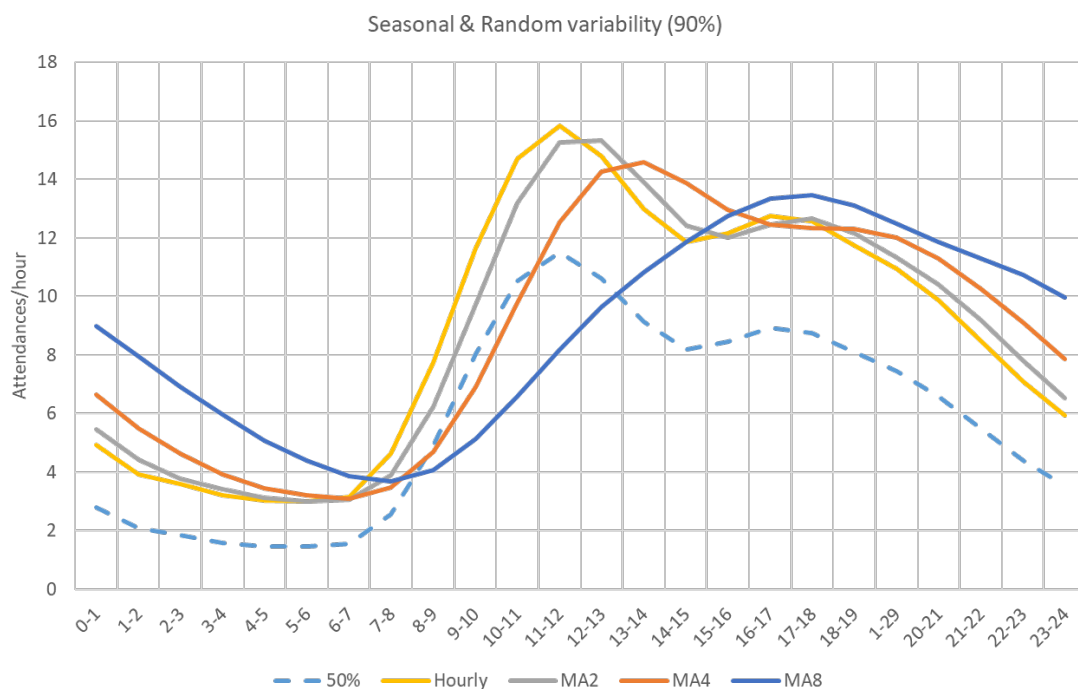
Th	Czw
Fr	Pt
Sa	Sb
Su	Nd

Przy przewidywanej liczbie przyjęć pacjentów na SOR wynoszącej 51 000 pacjentów (około 140 pacjentów dziennie), liczbę przyjęć w ciągu przeciętnego dnia można oszacować według godzin, jak pokazano na poniższym rysunku.



Attendances/hour	Przyjęcia/godzina
Seasonal & Random variability (by level of coverage)	Zmienność sezonowa i losowa (według poziomu pokrycia)
Avg	Średnia
Stable	Stabilna

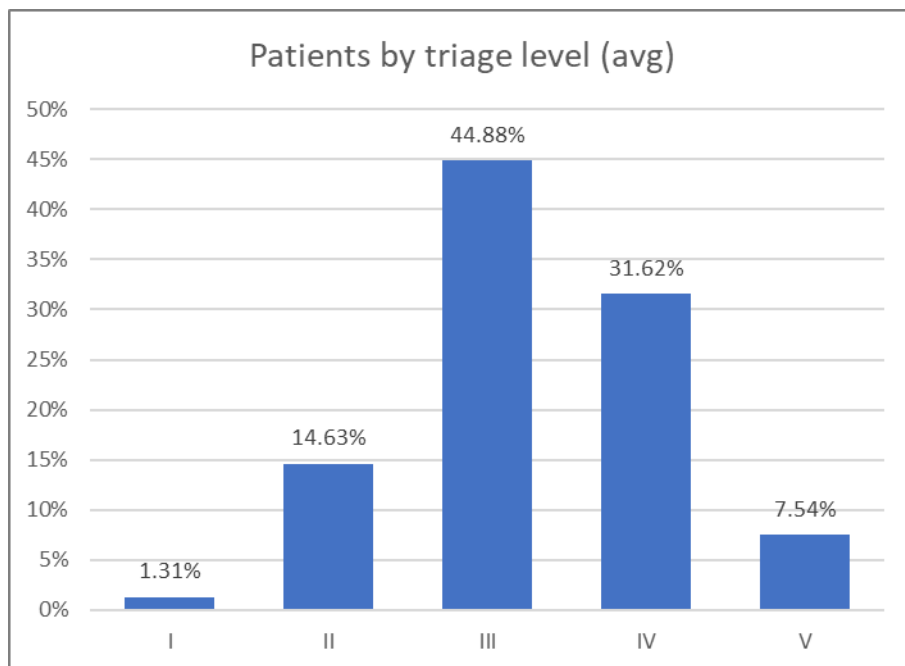
Na tym rysunku przedstawiono przewidywaną liczbę pacjentów (linia Avg). Zapotrzebowanie to będzie jednak podlegać zmienności losowej, którą poddaliśmy procesowi modelowania w procesie Poissona. Stosując średnie dane szacunkowe, pokrylibyśmy tylko 50% zmienności, a więc w 50% przypadków nie mielibyśmy wystarczających zasobów do pokrycia zapotrzebowania. Na wykresie widać wzrost liczby przyjęć, który należy uwzględnić, aby osiągnąć poziom obsługi wynoszący 75% i 90% w ciągu każdej godziny. Można by rozważyć poziom obsługi nie w ciągu godziny, ale w ciągu kilku godzin, np. 2, 4 godzin. Na poniższym wykresie przedstawiono wpływ takiego uśredniania danych z kilku godzin na liczbę pacjentów do zwymiarowania.



Seasonal & Random variability (90%)	Zmienność sezonowa i losowa (90%)
Hourly	Godzinowo
Attendances/hour	Przyjęcia/godzina

Należy zauważyć, że wpływ ten nie jest tak duży, gdy średnia wynosi 2 lub 4 godziny. Biorąc pod uwagę, że wymiarujemy SOR w CSK MSWiA z myślą o przyszłości, a zapotrzebowanie prawdopodobnie wzrośnie, sugerujemy zastosowanie wysokiego poziomu obsługi.

Kolejnym elementem, który należy rozważyć, jest poziom podziału pacjentów na grupy (ang. patient streaming). Terminu „podział pacjentów na grupy” używamy w odniesieniu do proponowanej polityki, która dzieli pacjentów (i zasoby) na różne grupy, zgodnie z przewidywaną dyspozycją (przyjęcie do szpitala lub wypis) lub grupą zgodnie z systemem triage (I-V). Konwencjonalna polityka polegająca na jednoczesnym leczeniu wszystkich rodzajów pacjentów (z połączonymi zasobami) jest znana jako „pooling”. Z literatury dotyczącej zarządzania operacyjnego wiadomo, że leczenie wszystkich pacjentów bez podziału na grupy przynosi korzyści w zakresie efektywności wynikające z lepszego ich wykorzystania. Okazało się jednak, że podział pacjentów na grupy ma zalety, które równoważą jego nieodłączną wadę, która przemawia przeciwko stosowaniu tego rozwiązania. Aby rozważyć możliwość włączenia dzielenia pacjentów na grupy do naszej polityki, a także aby móc dokładniej oszacować wykorzystanie zasobów przez pacjenta, zaklasyfikowaliśmy przybywających pacjentów do przewidywanej grupy w systemie triage. Na podstawie badań przeprowadzonych w podobnych szpitalach w Hiszpanii i USA szacujemy, że prawdopodobieństwo zaklasyfikowania pacjenta do określonej grupy w systemie triage może być następujące:



Patients by triage level (avg)

Klasyfikacja wg. grupy w systemie triage (średnia)

Oszacowaliśmy również godziny, w których pacjent korzysta z łóżka na SOR-ze, wykorzystując następujące prawdopodobieństwa przepływu pacjentów:

#### Wykorzystanie stanowisk

Ods. Godz./sta  
nowisko

Triage	100,0%	0,08
1. Resuscytacja	1,31%	6
2. Zagrożenie życia	14,63%	6
3. Nagły przypadek, duże zagrożenie	44,88%	5
4. Nagły przypadek, małe zagrożenie	31,62%	2.5
5. Brak pilnej potrzeby opieki	7,54%	1.5
Obserwacja	11,0%	12
Przyjęcie do szpitala (czas przebywania na oddziale)	40,0%	6

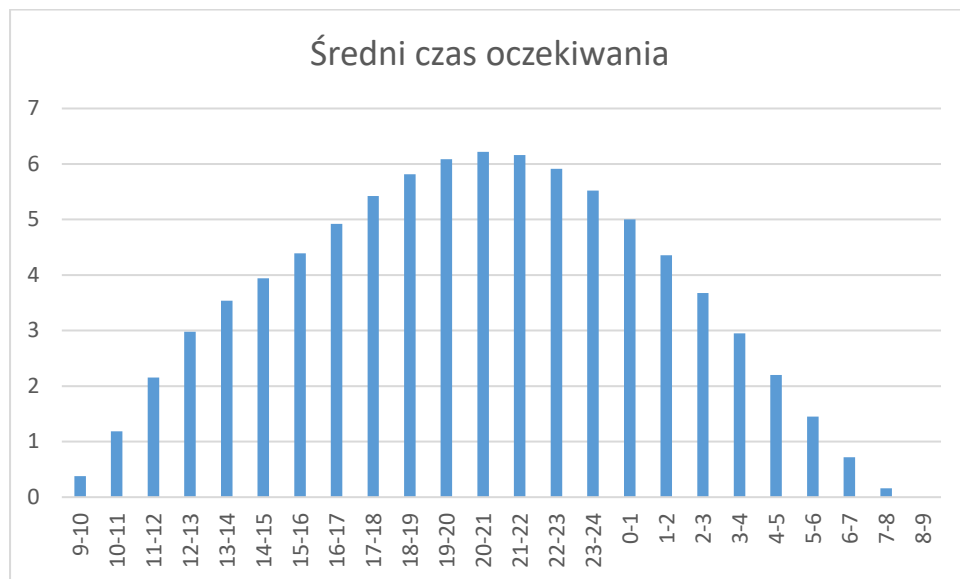
Zakładamy, że wszyscy pacjenci przechodzą przez system triage, a następnie są klasyfikowani według poziomu nasilenia zagrożenia zdrowia (przy czym każdy poziom nasilenia zagrożenia zdrowia wymaga innego czasu, w którym pacjent zajmuje łóżko). Część pacjentów (11%) wymaga obserwacji po leczeniu na SOR-ze i przebywa na obserwacji średnio 12 godzin. Wreszcie zakładamy, że 40% pacjentów z SOR-u jest ostatecznie przyjmowanych na oddział szpitalny i ponownie szacujemy, że opóźnienie w przyjęciu tych pacjentów na oddział wynosi 6 godzin. Oczywiście wszystkie te parametry są w naszych modelach zmiennymi, które z łatwością można dostosować do bardziej precyzyjnych szacunków, gdy tylko szpital będzie w stanie je przedstawić.

Planując zapotrzebowanie stałe, można by posłużyć się średnim zapotrzebowaniem godzinowym (5,8 pacjenta na godzinę = 140 pacjentów dziennie podzielonych przez 24 godziny), ale w ten sposób

**SORTECH**

Projekt współfinansowany przez Narodowe Centrum Badań i Rozwoju w ramach projektu realizowanego na rzecz obronności i bezpieczeństwa państwa w ramach konkursu nr 10/2019

można by pokryć tylko około 50% tygodniowego czasu pracy, a pozostałe 50% oddział pracowałby poniżej swoich możliwości, co powodowałoby długie kolejki. Byłoby to niedopuszczalne, ponieważ mogłoby oznaczać, że średnio pacjenci przybywający w godzinach wieczornych (gdzie kolejka liczyłaby aż 37 osób oczekujących na leczenie) oczekiwaliby ponad 6 godzin na leczenie.



Average waiting time

Średni czas oczekiwania

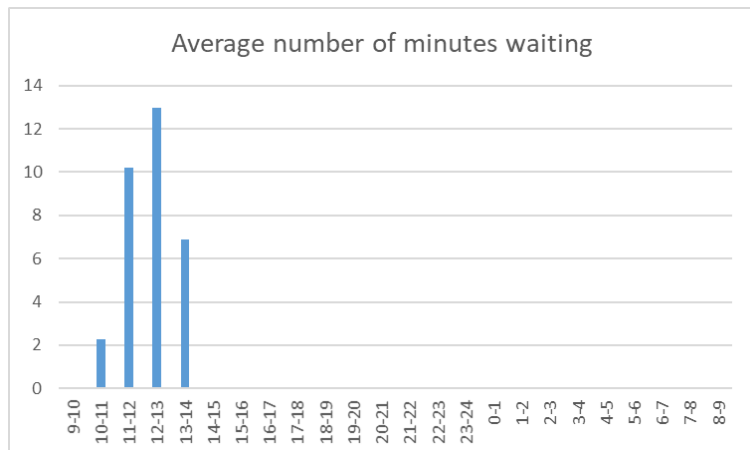
W naszym algorytmie uwzględniliśmy parametr, dzięki któremu kierownik SOR-u może decydować o poziomie wystarczającego pokrycia godzin w tygodniu. Na przykład, na poniższym wykresie zdecydowaliśmy się pokryć 97% tygodniowego czasu pracy, co daje cel godzinowy w postaci pokrycia 1,19% tygodniowego zapotrzebowania. W niniejszym raporcie będziemy nazywać ten poziom pokrycia sezonowością poziomą usług (ang. seasonality service level, SSL).



Total		Mo	Tu	We	Th	Fr	Sa	Su
		<b>16.46%</b>	<b>14.73%</b>	<b>14.47%</b>	<b>14.54%</b>	<b>14.65%</b>	<b>12.85%</b>	<b>12.29%</b>
0-1	<b>2.00%</b>	0.33%	0.29%	0.29%	0.29%	0.29%	0.26%	0.25%
1-2	<b>1.48%</b>	0.24%	0.22%	0.21%	0.22%	0.22%	0.19%	0.18%
2-3	<b>1.33%</b>	0.22%	0.20%	0.19%	0.19%	0.19%	0.17%	0.16%
3-4	<b>1.13%</b>	0.19%	0.17%	0.16%	0.16%	0.17%	0.15%	0.14%
4-5	<b>1.05%</b>	0.17%	0.16%	0.15%	0.15%	0.15%	0.14%	0.13%
5-6	<b>1.04%</b>	0.17%	0.15%	0.15%	0.15%	0.15%	0.13%	0.13%
6-7	<b>1.11%</b>	0.18%	0.16%	0.16%	0.16%	0.16%	0.14%	0.14%
7-8	<b>1.84%</b>	0.30%	0.27%	0.27%	0.27%	0.27%	0.24%	0.23%
8-9	<b>3.51%</b>	0.58%	0.52%	0.51%	0.51%	0.51%	0.45%	0.43%
0-19	<b>5.73%</b>	0.94%	0.84%	0.83%	0.83%	0.84%	0.74%	0.70%
10-11	<b>7.53%</b>	1.24%	1.11%	1.09%	1.09%	1.10%	0.97%	0.93%
11-12	<b>8.21%</b>	1.35%	1.21%	1.19%	1.19%	1.20%	1.06%	1.01%
12-13	<b>7.59%</b>	1.25%	1.12%	1.10%	1.10%	1.11%	0.97%	0.93%
13-14	<b>6.52%</b>	1.07%	0.96%	0.94%	0.95%	0.95%	0.84%	0.80%
14-15	<b>5.85%</b>	0.96%	0.86%	0.85%	0.85%	0.86%	0.75%	0.72%
15-16	<b>6.03%</b>	0.99%	0.89%	0.87%	0.88%	0.88%	0.77%	0.74%
16-17	<b>6.38%</b>	1.05%	0.94%	0.92%	0.93%	0.93%	0.82%	0.78%
17-18	<b>6.26%</b>	1.03%	0.92%	0.91%	0.91%	0.92%	0.80%	0.77%
18-19	<b>5.79%</b>	0.95%	0.85%	0.84%	0.84%	0.85%	0.74%	0.71%
19-20	<b>5.31%</b>	0.87%	0.78%	0.77%	0.77%	0.78%	0.68%	0.65%
20-21	<b>4.71%</b>	0.78%	0.69%	0.68%	0.69%	0.69%	0.61%	0.58%
21-22	<b>3.92%</b>	0.65%	0.58%	0.57%	0.57%	0.57%	0.50%	0.48%
22-23	<b>3.15%</b>	0.52%	0.46%	0.46%	0.46%	0.46%	0.40%	0.39%
23-24	<b>2.53%</b>	0.42%	0.37%	0.37%	0.37%	0.37%	0.32%	0.31%
100.00%								

Total	Ogółem
Mo	Pn
Tu	Wt
We	Śr
Th	Czw
Fr	Pt
Sa	Sb
Su	Nd

Przy takim percentylu będziemy dążyć do pokrycia zapotrzebowania godzinowego wynoszącego  $1,19 \times 140 \times 7 = 11,7$  pacjenta. Przy takiej liczbie miejsc, w najgorszym dniu tygodnia, czyli w poniedziałek, przewidywany czas oczekiwania pacjentów będzie następujący



Average number of minutes waiting	Średnia liczba minut oczekiwania
-----------------------------------	----------------------------------

Do tej pory rozpatrywaliśmy jedynie zmienność sezonową w obrębie dni tygodnia i godzin dnia. Zapotrzebowanie na SOR-ze podlega jednak także zmienności losowej. Na przykład z powyższej tabeli wynika, że w środę w godzinach 11-12 możemy spodziewać się 11,7 pacjentów, ale liczba ta będzie się wahać w zależności od tygodnia. W niektórych tygodniach będzie wyższa, a w innych niższa, czyli w połowie tygodni czas oczekiwania będzie krótszy niż wynika to z powyższego wykresu, a w drugiej połowie dłuższy.

Dlatego w naszym algorytmie wprowadziliśmy kolejny parametr poziomu bezpieczeństwa, aby uwzględnić naturalną zmienność ilości przybywających pacjentów. W tym przypadku, jeśli chcemy objąć opieką nie tylko 50% przypadków w danej godzinie, ale - powiedzmy - 75% przypadków, powinniśmy zwymiarować obsługę do 14 pacjentów, a nie tylko do 11,7. Taki zakres opieki będziemy nazywać naturalnym poziomem bezpieczeństwa zmienności (ang. natural variability safety level, NSL). Określenie wymaganego poziomu obsługi w oparciu o przepisy i dostępny personel jest obowiązkiem kierownika SOR-u.

Po ustaleniu dwóch poziomów bezpieczeństwa (pokrycie w ramach sezonowej zmienności godzin w tygodniu, SSL, i pokrycie w ramach naturalnej zmienności w ciągu jednej godziny, NSL) (w naszym przykładzie odpowiednio 97% i 75%) algorytm określa liczbę łóżek potrzebnych w różnych sekcjach, w zależności od wyboru grupy.

		Patients/hr					
		Avg	Std	75%	Bed time	No. Beds	
<b>Triage</b>	100%	11.70	3.42	14.01	0.08	1.12	
<b>No streaming</b>	100%	11.70	3.42	14.01	4.10	57.52	
<b>Streaming (2)</b>							
I+II	15.95%	1.87	1.37	2.79	6.00	16.73	61.50
III+IV+V	84.05%	9.84	3.14	11.95	3.75	44.77	
<b>Streaming (3)</b>							
I+II	15.95%	1.87	1.37	2.79	6.00	16.73	62.53
III+IV	76.51%	8.96	2.99	10.97	3.97	43.53	
V	7.54%	0.88	0.94	1.52	1.50	2.28	
<b>Streaming (5)</b>							
I	1.31%	0.15	0.39	0.42	6.00	2.51	66.85
II	14.63%	1.71	1.31	2.60	6.00	15.57	
III	44.88%	5.25	2.29	6.80	5.00	34.00	
IV	31.62%	3.70	1.92	5.00	2.50	12.50	
V	7.54%	0.88	0.94	1.52	1.50	2.28	
<b>Observation</b>	11.0%	1.29	1.13	2.05	12	24.63	
<b>Boarding</b>	40.0%	4.68	2.16	6.14	6	36.85	
<b>Summary number of beds</b>		No str.	Str. 2	Str. 3	Str. 5		
Triage		2	2	2	2		
Treatment		58	62	63	67		
Observation		25	25	25	25		
Boarding		37	37	37	37		
<b>Total</b>		122	126	127	131		

Triage	Triage
No streaming	Bez podziału pacjentów na grupy

Streaming (2)	Podział pacjentów na grupy (2)
Streaming (3)	Podział pacjentów na grupy (3)
Streaming (5)	Podział pacjentów na grupy (5)
Observation	Obserwacja
Boarding	Przyjmowanie pacjentów
Summary number of beds	Podsumowanie liczby łóżek
No str.	Bez podziału pacjentów na grupy
Str. 2	Podział pacjentów na grupy 2
Str. 3	Podział pacjentów na grupy 3
Str. 5	Podział pacjentów na grupy 5
Patients/hr	Pacjenci/godz.
Avg	Średnia
Std	Std
Bed time	Czas zajmowania łóżka przez pacjenta
No. Beds	Liczba łóżek
Triage Treatment Observation Boarding Total	Triage Leczenie Obserwacja Zajmowanie łóżka przez pacjenta Ogółem

Należy zauważyć, że „efekt rozproszenia” wynikający z podziału pacjentów na grupy wymaga zwiększenia liczby łóżek z 57 do 67 w obszarze leczenia w przypadku przydzielenia pacjentów do pięciu różnych grup. Ponadto w przykładzie tym konieczna jest duża liczba łóżek obserwacyjnych i zajmowanych przez pacjentów.

W poniższej tabeli przedstawiono wpływ dwóch poziomów bezpieczeństwa uwzględnionych w algorytmie na liczbę łóżek wymaganych w warunkach braku podziału pacjentów na grupy.

Hourly % coverage	Safety level				
	60%	70%	80%	90%	95%
50%	61	67	73	83	88
60%	69	76	83	92	98
70%	78	85	92	104	109
80%	89	97	104	116	122
90%	103	111	121	133	140
95%	115	124	133	146	152
98%	130	139	148	162	169
100%	183	196	208	227	235

Hourly % coverage	% Pokrycia godzinowo
Safety level	Poziom bezpieczeństwa

Dzięki opracowanemu algorytmowi jesteśmy w stanie zwymiarować liczbę potrzebnych łóżek. Zastosowaliśmy ten algorytm do ustalenia wymaganej liczby łóżek, ale można go również zastosować do innych zasobów stałych, takich jak łóżka na oddziałach intensywnej terapii, sal operacyjnych, sprzętu radiologicznego, rezonansu magnetycznego itp. Jesteśmy gotowi do współpracy z ordynatorami szpitalnych SOR-ów w celu dopracowania tych decyzji.

Dane i decyzje wykorzystywane przez algorytm są następujące:

- Prognoza liczby przybywających pacjentów na godzinę (dane wyjściowe z części 1 niniejszego dokumentu)
- Typologia pacjentów (według systemu triage, według wykorzystania zasobów, ...)

- Klastry o małej zmienności międzygrupowej (do tej pory stosowaliśmy podział wg. systemu triage)
- % pacjentów, którzy wymagają okresu obserwacji
- % pacjentów, którzy zostali przyjęci na oddziały
- Wykorzystanie zasobów stałych według typu pacjenta (łóżka, sprzęt do diagnostyki obrazowej, laboratorium, sala operacyjna, OIOM, sala obserwacyjna, sala przyjęć,...)
- Pożądane poziomy podziału pacjentów na grupy
- Poziomy bezpieczeństwa (percentyl w tygodniu i zmienność w ciągu godziny)

### 3. Wymiarowanie pod kątem zasobów zmiennych (personel)

Algorytm przedstawiony w tej części ma na celu ułatwienie wymiarowania wydajności dla zmiennych zasobów (pielęgniarki z obszaru triage, lekarze, pielęgniarki, ratownicy medyczni,...). Ponieważ zasoby te łatwiej jest zmodyfikować, algorytm musi przeprowadzić analizę wymaganej wydajności w wymiarze godzinowym.

Pierwsza część tego algorytmu jest podobna do opisanego już algorytmu dla zasobów stałych, ale teraz określimy wymaganą liczbę lekarzy, pielęgniarek i ratowników medycznych dla każdej godziny w danym dniu, ponieważ możemy zmieniać przydział personelu na różnych zmianach w ciągu dnia i w różne dni tygodnia. Tak więc, stosując podejście podobne do opisanego w części 2, po ustaleniu pokrycia sezonowej zmienności godzin w tygodniu, SSL, oraz pokrycia naturalnej zmienności w ciągu godziny, NSL (w naszym przykładzie odpowiednio 97% i 75%), możemy określić przewidywaną liczbę pacjentów, dla których należy zaplanować opiekę w każdej godzinie danego dnia. Na poniższym wykresie przedstawiono obliczenia dla poniedziałku.

DoW		
1		
		No. Pat.
	z→	0.674
Hour	Exp.Visits	75%
0-1	3.22	4.43
1-2	2.39	3.44
2-3	2.14	3.13
3-4	1.83	2.74
4-5	1.70	2.58
5-6	1.67	2.54
6-7	1.80	2.70
7-8	2.96	4.12
8-9	5.66	7.26
0-19	9.25	11.30
10-11	12.15	14.50
11-12	13.25	15.71
12-13	12.25	14.61
13-14	10.52	12.70
14-15	9.44	11.51
15-16	9.73	11.83
16-17	10.29	12.45
17-18	10.10	12.25
18-19	9.34	11.40
19-20	8.57	10.54
20-21	7.61	9.47
21-22	6.33	8.02
22-23	5.08	6.60
23-24	4.08	5.44

DoW	DoW
Hour	Godzina
Exp.Visits	Przewidywane wizyty
No. Pat.	Liczba Pacjentów

W kolejnym kroku algorytmu wykorzystuje się przewidywane wykorzystanie każdego ze zmiennych zasobów (lekarze, pielęgniarki, ratownicy medyczni,...) do obsługi jednego pacjenta. Istnieją dwa możliwe podejścia do uzyskania tego współczynnika: (1) dla każdego zasobu można oszacować liczbę minut, której będzie wymagał każdy pacjent (np. jeden pacjent na 4. poziomie w systemie triage wymaga 20 minut pracy lekarza) lub można użyć wskaźnika Pacjent/Profesjonalista (ang. Patient/Professional ratio, PPR - liczba pacjentów, którymi może zająć się specjalista). Do testowania algorytmów zastosowaliśmy następujące dane szacunkowe, zaczynając od podejścia 1 (czas szacowany przez każdego specjalistę) i obliczając odpowiedni współczynnik. Mogliśmy też zacząć bezpośrednio od współczynników.

Resources utilization	Perc.	Time/patient		Pat/profes. ratio	
		Doctor min	Nurse min	Doctors	Nurses
Triage	100.0%	0	4.5		
1. Resuscitation	1.31%	90	120	0.67	0.50
2. Life threatening	14.63%	60	90	1.00	0.67
3. Urgent high acuity	44.88%	40	40	1.50	1.50
4. Urgent low acuity	31.62%	20	10	3.00	6.00
5. Non urgent	7.54%	20	10	3.00	6.00
Observation	11.0%	72	72	0.83	0.83
Boarding	40.0%	10	30	6.00	2.00
		<b>47.67</b>	<b>61.04</b>	<b>1.26</b>	<b>0.98</b>

Resources utilization	Wykorzystanie zasobów
Triage	Triage
1. Resuscitation	1. Resuscytacja
2. Life threatening	2. Zagrożenie życia
3. Urgent high acuity	3. Nagły przypadek, duże zagrożenie
4. Urgent low acuity	4. Nagły przypadek, małe zagrożenie
5. Non urgent Observation Boarding	5. Zajmowanie łóżka przez pacjenta, który nie wymaga opieki w trybie pilnym
Perc.	Ods.
Time/patient	Czas/pacjent
Pat/profes. ratio	Współczynnik pacjent/specjalista
Doctor min	Lekarz min
Nurse min	Pielęgniarka min

Należy pamiętać, że uwzględniliśmy wymagania dotyczące nadzoru w czasie, gdy pacjent przebywa na obserwacji i gdy został przyjęty na SOR, oczekując na przyjęcie do oddziału szpitalnego. Jeśli obszary te nie zostaną uwzględnione, współczynniki pacjent/specjalista znacznie wzrastają - do 1,68 w przypadku lekarzy i 1,46 w przypadku pielęgniarek.

Na podstawie tych danych szacunkowych algorytm oblicza wymaganą liczbę członków personelu<sup>1</sup> w poszczególnych godzinach dnia. Na przykład w poniższej tabeli przedstawiono wyniki działania algorytmu dla poniedziałku o SSL równym 97% i NSL równym 75%, bez podziału pacjentów na różne grupy.

<sup>1</sup> Do tej pory obliczyliśmy jedynie zapotrzebowanie na pielęgniarki i lekarzy, ponieważ nie mamy danych szacunkowych dotyczących współczynników dla ratowników medycznych i tego, w jaki sposób zmniejszają one zapotrzebowanie na pielęgniarki. Jednak gdy tylko te dane szacunkowe będą dostępne, będziemy mogli dodać ratowników medycznych do naszych algorytmów.

DoW											
1											
Hour	Triage	Care		Observation		Boarding		TOTAL		Required doctors	Required nurses
	4.50 Nurses	35.75 Doctors	36.62 Nurses	7.92 Doctors	7.92 Nurses	4.00 Doctors	12.00 Nurses	(except triage) Doctors Nurses			
0-1	0.33	2.64	2.71	0.59	0.59	0.30	0.89	3.52	4.18	4	4
1-2	0.26	2.05	2.10	0.45	0.45	0.23	0.69	2.73	3.24	3	3
2-3	0.23	1.86	1.91	0.41	0.41	0.21	0.63	2.49	2.95	2	3
3-4	0.21	1.63	1.67	0.36	0.36	0.18	0.55	2.18	2.58	2	3
4-5	0.19	1.54	1.57	0.34	0.34	0.17	0.52	2.05	2.43	2	2
5-6	0.19	1.52	1.55	0.34	0.34	0.17	0.51	2.02	2.40	2	2
6-7	0.20	1.61	1.65	0.36	0.36	0.18	0.54	2.14	2.54	2	3
7-8	0.31	2.46	2.52	0.54	0.54	0.27	0.82	3.28	3.89	3	4
8-9	0.54	4.33	4.43	0.96	0.96	0.48	1.45	5.77	6.84	6	7
0-19	0.85	6.74	6.90	1.49	1.49	0.75	2.26	8.98	10.65	9	11
10-11	1.09	8.64	8.85	1.91	1.91	0.97	2.90	11.52	13.66	12	14
11-12	1.18	9.36	9.59	2.07	2.07	1.05	3.14	12.48	14.80	12	15
12-13	1.10	8.70	8.91	1.93	1.93	0.97	2.92	11.60	13.76	12	14
13-14	0.95	7.57	7.75	1.68	1.68	0.85	2.54	10.09	11.97	10	12
14-15	0.86	6.86	7.02	1.52	1.52	0.77	2.30	9.14	10.85	9	11
15-16	0.89	7.05	7.22	1.56	1.56	0.79	2.37	9.40	11.15	9	11
16-17	0.93	7.42	7.60	1.64	1.64	0.83	2.49	9.89	11.74	10	12
17-18	0.92	7.30	7.47	1.62	1.62	0.82	2.45	9.73	11.54	10	12
18-19	0.85	6.79	6.96	1.50	1.50	0.76	2.28	9.05	10.74	9	11
19-20	0.79	6.28	6.43	1.39	1.39	0.70	2.11	8.38	9.94	8	10
20-21	0.71	5.64	5.78	1.25	1.25	0.63	1.89	7.52	8.92	8	9
21-22	0.60	4.78	4.90	1.06	1.06	0.53	1.60	6.37	7.56	6	8
22-23	0.49	3.93	4.03	0.87	0.87	0.44	1.32	5.24	6.22	5	6
23-24	0.41	3.24	3.32	0.72	0.72	0.36	1.09	4.32	5.13	4	5

DoW	DoW
Hour	Godzina
Triage	Triage
Nurses	Pielęgniarki
Care	Opieka
Observation	Obserwacja
Boarding	Przyjmowanie pacjentów
TOTAL	OGÓŁEM
(except triage)	(za wyjątkiem systemu triage)
Required doctors	Wymagana liczba lekarzy
Required nurses	Wymagana liczba pielęgniarek

W podobny sposób można obliczyć liczbę pielęgniarek potrzebnych w każdej godzinie w ciągu tygodnia



### Nurses in treatment required by hour and day of the week

Hour	Mo	Tu	We	Th	Fr	Sa	Su	Average
0-1	4	4	4	4	4	3	3	3.7
1-2	3	3	3	3	3	3	3	3.0
2-3	3	3	3	3	3	2	2	2.7
3-4	3	2	2	2	2	2	2	2.1
4-5	2	2	2	2	2	2	2	2.0
5-6	2	2	2	2	2	2	2	2.0
6-7	3	2	2	2	2	2	2	2.1
7-8	4	4	3	3	4	3	3	3.4
8-9	7	6	6	6	6	5	5	5.9
0-19	11	10	9	10	10	9	8	9.6
10-11	14	12	12	12	12	11	10	11.9
11-12	15	13	13	13	13	12	11	12.9
12-13	14	12	12	12	12	11	11	12.0
13-14	12	11	11	11	11	10	9	10.7
14-15	11	10	10	10	10	9	8	9.7
15-16	11	10	10	10	10	9	9	9.9
16-17	12	11	10	10	11	9	9	10.3
17-18	12	10	10	10	10	9	9	10.0
18-19	11	10	10	10	10	9	8	9.7
19-20	10	9	9	9	9	8	8	8.9
20-21	9	8	8	8	8	7	7	7.9
21-22	8	7	7	7	7	6	6	6.9
22-23	6	6	6	6	6	5	5	5.7
23-24	5	5	5	5	5	4	4	4.7

Nurses in treatment required by hour and day of the week	Liczba pielęgniarek zapewniających opiekę medyczną w podziale na godziny i dni tygodnia
Hour	Godzina
Mo	Pn
Tu	Wt
We	Śr
Th	Czw
Fr	Pt
Sa	Sb
Su	Nd
Average	Średnia

Wysokie wymagania naszego algorytmu mogą zaskakiwać, ale należy pamiętać, że odpowiada to rozwiązaniu, które pokrywa 75% zmienności losowej w każdej godzinie, pokrywa 97% godzin w tygodniu i umożliwia obsługę pacjenta w godzinie jego przybycia. Oczywiście, jeśli zmniejszymy zakres, uzyskamy niższe wymagania dotyczące personelu.

### Symulacja

## 4. Optymalizacja zmian

Problemy z układaniem harmonogramu pracy personelu dotyczą przydzielania pewnej liczby zadań pewnej liczbie specjalistów z zachowaniem szeregu ograniczeń, które często czynią to zadanie

**SORTECH**

Projekt współfinansowany przez Narodowe Centrum Badań i Rozwoju w ramach projektu realizowanego na rzecz obronności i bezpieczeństwa państwa w ramach konkursu nr 10/2019

trudniejszym niż w przypadku normalnego przydzielania zadań. Ogólnie rzecz biorąc, takie przypisanie musi być dokonane w celu ścisłego spełnienia wymagań dotyczących usług i umów oraz w celu maksymalizacji preferencji specjalistów i/lub minimalizacji kosztów, jeśli takie cele mają zastosowanie.

W dziedzinie opieki zdrowotnej, a w szczególności na SOR-ach, problemy związane z układaniem harmonogramu pracy personelu są szczególnie rozpowszechnione, ponieważ wiele usług trzeba świadczyć w sposób ciągły, przez 24 godziny na dobę, 7 dni w tygodniu, a zatem każda organizacja pracy musi opierać się na harmonogramie dyżurów. Algorytmy przedstawione w tym i następnym rozdziale dotyczą problemu planowania pracy specjalistów (lekarzy, pielęgniarek, ratowników medycznych) w różnych sekcjach SOR-u, gdzie plan polega na przydziale personelu do wykonywania różnych usług medycznych w tych sekcjach. Planowanie pracy personelu medycznego jest na ogół szczególnie trudne. Jakość tych planów jest jednak bardzo ważna, ponieważ trudno jest osiągnąć wysokie wyniki w służbie zdrowia, gdy stosuje się niskiej jakości grafiki, np. grafiki powodujące niezrównoważone obciążenie pracą lub nadmierny stres u pracowników.

Odbywanie dyżurów na SOR-ze należy do obowiązków lekarza. W przedstawionych algorytmach skupiliśmy się na zaspokojeniu potrzeb na SOR-ze, oddzielając je od ewentualnych obciążeń, na jakie specjaliści są narażeni w innych częściach szpitala, tzn. założyliśmy, że pracują wyłącznie na Szpitalnym Oddziale Ratunkowym. Pracujemy nad tymi problemami w dwóch etapach. W **pierwszym etapie**, przedstawionym w tej części, proponujemy kilka możliwych zmian (spełniających wymagania ewentualnego układu zbiorowego pracy) i optymalizujemy przydział pracowników na te zmiany, dążąc do jak najlepszego pokrycia zapotrzebowania przez dostępny personel. Zmiany trwają zwykle 10/12 godzin i rozpoczynają się w różnych okresach w ciągu dnia. Mimo że możliwe byłoby wprowadzenie różnych zmian w różnych dniach tygodnia, zakładamy, że zmiany zostaną zachowane, tyle że w niektóre dni do niektórych zmian przypisana będzie mniejsza liczba pracowników.

Założmy, że w części 3 powyżej ustaliliśmy, że godzinowe zapotrzebowanie na dany rodzaj personelu (np. pielęgniarki) w ciągu danego dnia jest określone przez  $p_t$ , gdzie  $t$  jest godziną w ciągu dnia. Przygotowaliśmy także listę możliwych zmian opisanych macierzą  $S_{it}$ ,  $i=1, \dots, I$  (liczba rozważanych zmian) oraz  $t=1, \dots, 24$  (godziny jednej doby).  $S_{it}$  będzie równe 1, jeśli zmiana  $i$  obejmuje okres  $t$ , lub w innym przypadku będzie równe 0. Na przykład, obecnie na szpitalnym SOR-ze stosowane są tylko dwie zmiany, każda po 12 godzin: zmiana dzienna D1 rozpoczynająca się o 7:00 i kończąca o 19:00 oraz zmiana nocna N1 rozpoczynająca się o 19:00 i kończąca o 7:00. W tym przypadku macierz  $S$  będzie miała następującą postać:

Shifts	Period starting at t																							
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
D1 (7-19)								1	1	1	1	1	1	1	1	1	1	1	1					
N1 (19-7)	1	1	1	1	1	1	1													1	1	1	1	1

Period starting at t	Okres rozpoczynający się o t
Shifts	Zmiany

Jedną z możliwych konstrukcji algorytmu optymalizacyjnego jest konstrukcja prosta, w której staramy się zminimalizować całkowite niepokryte zapotrzebowanie godzinowe, przydzielając dostępny personel (maksymalna wartość  $D$ ) na różne zmiany.

Model programowania matematycznego jest następujący:

$$\text{Min } \sum_t u_t \quad (1)$$

z następującymi ograniczeniami

$$u_t = \max(p_t - c_t, 0) \quad t=1, \dots, 24 \text{ (liczba okresów)} \quad (2)$$

$$c_t = \sum_i S_{it} \times d_i \quad t=1, \dots, 24 \quad (3)$$

$$\sum_i d_i \leq D \quad i=1, \dots, I \text{ (liczba of zmian)} \quad (4)$$

$$d_i \geq 0, \text{ liczba całkowita} \quad i=1, \dots, I \quad (5)$$

- (1) To funkcja optymalizacji, w której chcemy zminimalizować niepokryte zapotrzebowanie godzinowe
- (2) To wzór, według którego oblicza się zapotrzebowanie niepokryte w okresie  $t$
- (3) Definicja pokrycia każdego okresu, przy założeniu, że personel jest przydzielony do zmian ( $d_i$  to liczba pracowników przydzielonych do zmiany  $i$ )
- (4) Jest to ograniczenie liczby członków personelu przydzielonych do pracy na wszystkich zmianach, tak aby nie przekroczyć liczby dostępnych członków personelu ogółem.
- (5) Liczba członków personelu przypisanych do danej zmiany musi być dodatnią liczbą całkowitą.

### Przykład

Założmy, że wymagania godzinowe dla danego rodzaju personelu (założmy, że są to pielęgniarki) obliczone w części 3 niniejszego dokumentu są następujące:

	Period starting at																							
$t \rightarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Needs $p_t \rightarrow$	4	3	3	2	2	2	2	4	7	10	13	14	13	12	10	11	11	11	10	10	9	7	6	5

Period starting at	Okres rozpoczynający się o
Needs $p_t$	Potrzuje $p_t$
$t$	$t$

Mamy do dyspozycji 18 pielęgniarek i zgodnie z polityką obowiązującą w CSK MSWiA połowę z nich (9) przydzielamy na zmianę dzienną D1 (7:00 - 19:00), a drugą połowę (9) na zmianę nocną N1 (19:00 - 7:00).

Pokrycie (i jego brak) w każdym okresie przedstawia się następująco

Max No. Prof	Period starting at																							
18	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
Needs $p_t \rightarrow$	4	3	3	2	2	2	2	4	7	10	13	14	13	12	10	11	11	11	10	10	9	7	6	5
Prof/shift	Shifts $\downarrow$																							
9	D1 (7-19)																							
9	N1 (19-7)																							
18	Coverage $c_t$	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9	9
Uncovered $u_t$	0	0	0	0	0	0	0	0	0	1	4	5	4	3	1	2	2	2	1	1	0	0	0	0
																								26

14.4%

Max No. Prof	Maks. liczba spec.
Period starting at	Okres rozpoczynający się o
Prof/shift	Spec./zmiana
Shifts	Zmiany
Needs $p_t$	Potrzuje $p_t$
Coverage $c_t$	Pokrycie $c_t$
Uncovered $u_t$	Brak pokrycia $u_t$
$t$	$t$

Daje to 26 przypadków niepokrytego zapotrzebowania (jeden przypadek to jedna godzina, podczas której, w razie konieczności, zapotrzebowanie nie jest pokryte przez jedną pielęgniarkę).

Jeśli pozwolimy, aby liczba pielęgniarek na każdej zmianie była różna, przy zachowaniu tej samej liczby pielęgniarek ogółem, możemy uzyskać znacznie lepsze pokrycie zapotrzebowania:

Max No. Prof	Period starting at	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
18	$t \rightarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	Needs $p_t \rightarrow$	4	3	3	2	2	2	2	4	7	10	13	14	13	12	10	11	11	11	10	10	9	7	6	5
Prof/shift	Shifts $\downarrow$																								
12	D1 (7-19)									1	1	1	1	1	1	1	1	1	1	1					
6	N1 (19-7)	1	1	1	1	1	1	1													1	1	1	1	1
18	Coverage $c_t$	6	6	6	6	6	6	6	12	12	12	12	12	12	12	12	12	12	12	12	6	6	6	6	6
	Uncovered $u_t$	0	0	0	0	0	0	0	0	0	0	1	2	1	0	0	0	0	0	0	4	3	1	0	0

6.6%

Max No. Prof	Maks. liczba spec.
Period starting at	Okres rozpoczynający się o
Prof/shift	Spec./zmiana
Shifts	Zmiany
Needs $p_t$	Potrzuje $p_t$
Coverage $c_t$	Pokrycie $c_t$
Uncovered $u_t$	Brak pokrycia $u_t$
$t$	$t$

Należy zauważyć, że w przypadku optymalizacji zmian, przy tej samej liczbie członków personelu, tylko poprzez przesunięcie ich na inne zmiany, uzyskujemy rozwiązanie, które zmniejsza poziom braku pokrycia przez specjalistów z 26 okresów\* (14,4%) do zaledwie 12 (6,6%).

Jeśli stworzymy jeszcze jedną zmianę dzienną (od 9:00 do 21:00) i zoptymalizujemy przydział 18 pielęgniarek na trzy różne zmiany, otrzymamy

Max No. Prof	Period starting at	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
18	$t \rightarrow$	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
	Needs $p_t \rightarrow$	4	3	3	2	2	2	2	4	7	10	13	14	13	12	10	11	11	11	10	10	9	7	6	5
Prof/shift	Shifts $\downarrow$																								
5	D1 (6-18)							1	1	1	1	1	1	1	1	1	1	1	1						
8	D2 (9-21)									1	1	1	1	1	1	1	1	1	1	1	1	1			
5	N (19-7)	1	1	1	1	1	1	1													1	1	1	1	1
18	Coverage $c_t$	5	5	5	5	5	5	10	5	5	13	13	13	13	13	13	13	13	13	8	13	13	5	5	5
	Uncovered $u_t$	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	2	0	0	2	1	0

4.4%

Max No. Prof	Maks. liczba spec.
Period starting at	Okres rozpoczynający się o
Prof/shift	Spec./zmiana
Shifts	Zmiany
Needs $p_t$	Potrzuje $p_t$
Coverage $c_t$	Pokrycie $c_t$
Uncovered $u_t$	Brak pokrycia $u_t$
$t$	$t$

Należy zauważyć, że taki przydział pielęgniarek do zmian pozwala uzyskać znacznie lepsze pokrycie zapotrzebowania (4,4% przypadków braku pokrycia w porównaniu z 14,4% w planie pierwotnym).

Zaprojektowany algorytm został wdrożony do narzędzia Excel Solver dla małego przykładu wykorzystanego w tej prezentacji, ale przygotowano go do rozwiązywania bardziej złożonych sytuacji, które można rozwiązać przy pomocy narzędzia OpenSolver.

Założyliśmy, że lekarze i pielęgniarki są tak samo wydajni podczas wszystkich godzin na zmianie. W literaturze istnieją dowody na to, że lekarze na SOR-ze nie badają pacjentów w stałym tempie, tylko oceniają znacznie więcej przypadków na początku, a mniej na końcu zmiany. Udokumentowano zmniejszenie wydajności o 32-38%. Gdybyśmy chcieli uwzględnić to zmniejszenie wydajności, można je łatwo włączyć do opisanego algorytmu. W macierzy Sit, gdzie przypisaliśmy 1, jeśli zmiana i obejmuje okres  $t$ , możemy przypisać wartości większe od 1 na początku zmiany i mniejsze od 1 pod koniec zmiany.

## 5. Optymalizacja przydziału poszczególnych pracowników do dostępnych zmian

W tym ostatnim algorytmie zajmujemy się przydzielaniem określonych członków personelu do różnych zmian. Biorąc pod uwagę czynniki związane z tym modułem naszego badania, zaproponowano model planowania harmonogramu pracy specjalistów z uwzględnieniem czynników związanych z: (1) utrzymaniem wskaźnika pacjent/specjalista (PPR, liczba pacjentów, którymi powinien zająć się każdy specjalista); (2) sprawiedliwym przydziałem zmian; oraz (3) satysfakcją zawodową. Sekwencyjność ram decyzyjnych wynika z różnych wymiarów opieki, które mają różne poziomy ważności, przy czym najwyższą wagę przypisuje się wskaźnikowi PPR.

Aby uwzględnić priorytety charakterystyczne dla tego typu problemów w procesie podejmowania decyzji, zaproponowano ramy sekwencyjnego programowania celowego (ang. sequential goal programming, SGP) (Masud & Hwang, 1981). SGP w sposób iteracyjny rozwiązuje każdy cel w modelu w kolejności zgodnej z hierarchią ważności. Wykazano, że wskaźnik PPR ma bezpośredni związek z wynikami leczenia pacjentów (Aiken i in., 2002; Stimpfel i Aiken, 2013; Stone i in., 2006). Sprawiedliwy przydział zmian odnosi się do ważniejszego celu zarządzania w porównaniu z poziomem satysfakcji pracowników. Takie sekwencyjne podejście do podejmowania decyzji odzwierciedla zatem rzeczywiste ramy decyzyjne dotyczące zarządzania obserwowane w dużych szpitalach, w których wyniki pacjentów są zawsze najważniejsze w porównaniu z innymi perspektywami (Berwick, Nolan, & Whittington, 2008; Bodenheimer & Sinsky, 2014).

W ramach SGP zastosowano miary wyników, które z powodzeniem wykorzystano w układaniu harmonogramu pracy specjalistów, a w projektowaniu harmonogramu uwzględniono zarówno perspektywę pacjenta, jak i świadczeniodawcy. Model ułatwia także podejście do rozwiązań, w którym cele rozpatruje się zgodnie z hierarchią ważności (Haimes i Li, 1988). Może on modelować wyraźne priorytety dotyczące zarządzania pod względem lepszych wyników opieki, uwzględniając jednak mniej ważne wyniki, związane z preferencjami personelu, a także sprzyja postrzeganiu sprawiedliwego przydziału zmian poprzez opracowanie harmonogramu pracy specjalistów z udziałem użytkowników.

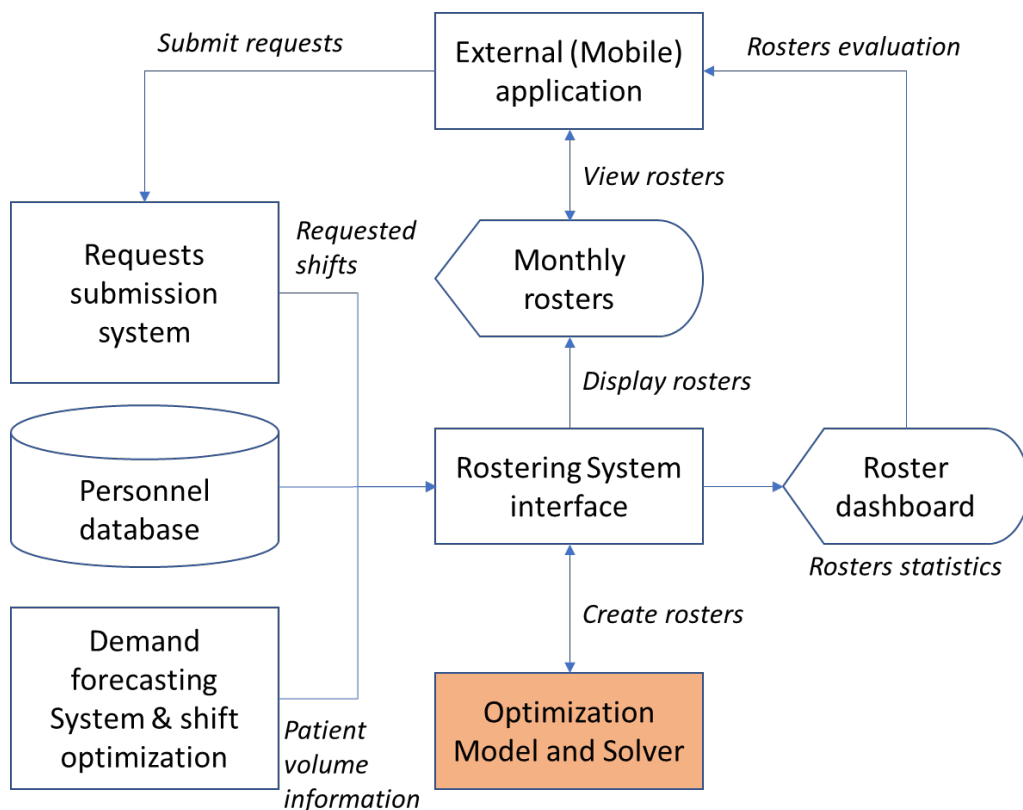
Ponieważ ważne jest także opracowanie systemu operacyjnego, który będzie skutecznie przekazywał informacje o harmonogramie pracownikom, a także pozwalał na otrzymywanie informacji zwrotnych, sugerujemy opracowanie elektronicznego grafiku pracy specjalistów w postaci pulpitu zintegrowanego z interfejsem aplikacji mobilnej. Zintegrowany pulpit i system aplikacji mobilnych ułatwią zarówno sprawozdawczość zarządczą, jak i rozpowszechnianie informacji o harmonogramie wśród specjalistów. Do zilustrowania tła wykorzystano studium przypadku zastosowania modelu SGP i ram projektowania harmonogramów z udziałem użytkowników na oddziale ratunkowym (SOR) dużego szpitala klinicznego w Singapurze.

W celu opracowania harmonogramu pracy specjalistów zaproponowano różne modele programowania matematycznego (ang. mathematical programming, MP). Aby uwzględnić wyniki i poziom satysfakcji zarówno pacjentów, jak i świadczeniodawców, zaproponowano również modele programowania wielocelowego (ang. goal programming, GP). W modelu GP przyjmuje się, że zbiór pożądaných wartości to cele podstawowe, a celem ogólnym jest maksymalne zmniejszenie odchyień dla każdego z celów podstawowych.

Wyspecjalizowanie podejścia modelu GP poprzez zastosowanie ram SGP polega na realizacji każdego ogólnego celu w modelu w sposób iteracyjny, zgodnie z hierarchią ważności. Priorytetyzacja celów opiera się na względnym znaczeniu poszczególnych celów w danym systemie. W publicznej służbie zdrowia głównym celem jest zazwyczaj osiągnięcie jak najlepszych wyników klinicznych i jakości opieki dla wszystkich pacjentów. Preferencje specjalistów dotyczące zmian i postrzeganie harmonogramu jako sprawiedliwego to również ważne cele ogólne, które zostały słabiej zbadane przy użyciu metod MP do tworzenia harmonogramów.

Trudności z operacjonalizacją systemów tworzenia harmonogramów opartych na modelach MP zostały obszernie opisane w literaturze (Clark, Moule, Topping, i Serpell, 2015; Kellogg i Walczak, 2007). Kwestie wynikające z faktycznego wdrożenia modeli grafików obejmują nieufność osób zarządzających pracą pielęgniarek wobec modeli matematycznych (Hung, 1991), zdolność tych modeli do uchwycenia złożoności ograniczeń leżących u podstaw planowania pracy specjalistów (zwłaszcza przy uwzględnieniu preferencji personelu i postrzegania sprawiedliwego przydziału zmian) (Kellogg i Walczak, 2007), zmiany harmonogramów i potrzebę podejścia do planowania w zamkniętej pętli z udziałem użytkowników (Clark i in., 2015). Dostosowanie parametrów jakościowych grafiku do celów oddziału lub organizacji oraz zapewnienie odpowiednich środków umożliwiających tworzenie danych statystycznych objętych obowiązkiem sprawozdawczym ma zasadnicze znaczenie dla oceny jakości grafiku, a także ułatwi zastosowanie wyników. Kwestie te zostały uwzględnione w niniejszym opracowaniu w celu stworzenia systemu, który może zostać wdrożony do wspomagania zarządzania personelem pielęgniarskim na różnych poziomach planowania.

Proponowany algorytm powinien stanowić element profesjonalnego systemu tworzenia harmonogramów, tak jak to opisano na poniższym schemacie:



Submit requests	Wysyłanie zapytań
External (Mobile) application	Zewnętrzna aplikacja (mobilna)
Rosters evaluation	Ocena grafików



View rosters	Podgląd grafików
Monthly rosters	Grafiki miesięczne
Display rosters	Wyświetlanie grafików
Roster dashboard	Pulpit z grafikami
Rosters statistics	Dane statystyczne dotyczące grafików
Create rosters	Tworzenie grafików
Optimization	Optymalizacja
Model and Solver	Model i Solver
Rostering System interface	Interfejs systemu do tworzenia grafików
Requested shifts	Wnioski o zmiany
Requests submission system	System dodawania wniosków
Personnel database	Baza danych dotycząca personelu
Demand forecasting System & shift optimization	System prognozowania zapotrzebowania i optymalizacja zmian
Patient volume information	Informacje o ilości pacjentów

Istnieją różne możliwe podejścia do modelowania preferencji specjalistów w zakresie określonych zmian. Proponujemy dostosowanie metody zaproponowanej przez Boon Yew Ang i in. (2017). W tej części opisujemy proponowaną metodę. Badany algorytm został zakodowany i jesteśmy gotowi do rozpoczęcia rozmów z menedżerami i lekarzami CSK MSWiA w celu dostosowania algorytmu do ich kultury i wymagań.

Wdrożono model MISGP w celu określenia optymalnego harmonogramu uwzględniającego wiele priorytetowych celów poprzez przeformułowanie modelu za pomocą ograniczeń równoważnych, co pozwala na odchylenia miar wyników od ich pożądaných wartości docelowych. Elastyczność celów ogólnych wynika z tolerancji dla pewnego poziomu niewłaściwej alokacji, kosztów lub niezadowolenia z punktu widzenia pracowników i kierownictwa.

W modelu MISGP uwzględniono następujące cele w kolejności od najważniejszego do najmniej ważnego:

- (1) minimalizacja średnich odchyżeń od docelowej wartości wskaźnika PPR
- (2) minimalizacja maksymalnych odchyżeń od docelowej wartości wskaźnika PPR
- (3) minimalizacja średniego odchylenia od docelowej liczby niepożądanych zmian;
- (4) maksymalizacja całkowitej przydzielonej liczby pożądaných zmian
- (5) maksymalne rozproszenie dni odpoczynku

Zmienne decyzyjne są binarnymi zmiennymi przypisania, które wskazują, czy specjalista jest przypisany do określonej kombinacji zmian i dni.

W następnym paragrafie opisano cele i ograniczenia modelu:

**Cele 1 i 2:** minimalizacja średnich i maksymalnych odchyżeń od docelowych wartości wskaźników pielęgniarstwa/pacjent

Jako docelową wartość wskaźnika PPR w badaniach pilotażowych przyjęto średnią wartość wskaźnika PPR w całym horyzoncie planowania. Określenie docelowego wskaźnika PPR na podstawie średniej horyzontalnej pozwala na zrównoważenie obciążenia pracą w całym okresie uwzględnionym w grafiku, zgodnie z liczbą przyjęć pacjentów w każdym oknie zmianowym. Docelowa wartość PPR może być różna dla różnych oddziałów i specjalizacji. Przewidywane zapotrzebowanie pacjentów pochodzi z danych historycznych (część 1 niniejszego dokumentu) i jest niejednorodne na różnych zmianach w ciągu każdego dnia. Na podstawie tych celów model może oszacować idealny wskaźnik PPR dla każdej

kombinacji dzień/zmiana, aby zrównoważyć obciążenie pracą poprzez zminimalizowanie odchyleń od docelowego wskaźnika PPR.

Minimalizując średnie odchylenia od docelowego wskaźnika PPR, cel 1 nie gwarantuje ograniczenia wskaźnika PPR dla każdej zmiany. Aby obciążenie pracą nie odbiegało zbyt wiele od średniego wskaźnika PPR, uwzględniono drugorzędny cel PPR (cel 2) oparty na minimalizacji maksymalnego „żalu”. „Żal” definiuje się następnie jako bezwzględne odchylenia od docelowego wskaźnika PPR.

#### **Cel 3 i 4:** minimalizacja średnich odchyleń od docelowej liczby niekorzystnych/ korzystnych zmian

Niekorzystne zmiany - Na SOR-ze niektóre zmiany (np. święta, zmiany wieczorne lub nocne) mogą być uważane za niekorzystne. Aby sprawiedliwie podzielić niekorzystne zmiany wśród specjalistów, w ramach Celu 3 minimalizuje się średnie odchylenia liczby takich zmian przydzielonych każdej pielęgniarsce od średniej liczby zmian przydzielonych całej puli pielęgniarek. Cel ten można dalej rozszerzyć, uwzględniając średnią ważoną niekorzystnych zmian z poprzedniego i bieżącego okresu uwzględnionego w grafiku.

Korzystne zmiany - W studium przypadku każdy specjalista mógł złożyć maksymalnie tylko osiem wniosków o zmianę w każdym horyzoncie planowania, z których każdy uwzględniał 1 miesiąc. Wnioski o zmianę to wyrażenie określonych preferencji dotyczące określonych zmian. Takie preferencje mogą wynikać z przyczyn osobistych i są zazwyczaj różne dla poszczególnych specjalistów. Założono, że ten cel ma niższy priorytet w porównaniu z potrzebą zminimalizowania alokacji niekorzystnych zmian.

#### **Cel 5:** maksymalne rozproszenie dni odpoczynku w harmonogramie

Oprócz preferencji specjalistów dotyczących zmian w pracy, przepisy BHP mogą ograniczać maksymalną liczbę następujących po sobie dni roboczych. Przepisy te można łatwo wprowadzić do modelu jako ograniczenia. Zauważono jednak, że choć ze względu na przepisy istnieją sztywne ograniczenia co do liczby kolejnych dni roboczych, to dla specjalistów korzystne jest także skrócenie czasu pracy i umożliwienie wystarczającej regeneracji sił między okresami pracy.

Długie okresy pracy z wydłużoną liczbą dni odpoczynku między nimi nie są atrakcyjne dla specjalistów, ponieważ mogą powodować zmęczenie i pogorszenie jakości opieki.

Aby zmniejszyć częstotliwość występowania długich okresów pracy, w modelu można uwzględnić dodatkowy cel polegający na ograniczeniu liczby długich sekwencji dni roboczych za pomocą zestawu ograniczeń logicznych, co pozwoli zmaksymalizować częstotliwość występowania schematów pracy po godzinach w optymalnym harmonogramie.

#### **Ograniczenia**

Model jest dość elastyczny i można w nim zastosować wiele różnych ograniczeń, które odzwierciedlają kulturę i zwyczaje panujące w poszczególnych krajach i instytucjach. Na przykład inne kluczowe ograniczenia wprowadzone w modelu MISGP w oparciu o politykę kadrową i przepisy dotyczące zatrudnienia mogą obejmować<sup>2</sup>:

- (1) 2 dni wolne w tygodniu dla każdego specjalisty;
- (2) minimalny współczynnik liczby mężczyzn do liczby kobiet wśród specjalistów;

<sup>2</sup> Ograniczenia te są oparte na przepisach obowiązujących w Singapurze w czasie przeprowadzania optymalizacji.



- (3) minimalny współczynnik liczby starszych specjalistów na zmianie;
- (4) minimalna liczba specjalistów przeszkolonych w zakresie systemu triage i resuscytacji;
- (5) maksymalnie 5 kolejnych dni roboczych;
- (6) maksymalna liczba specjalistów dostępnych do tworzenia harmonogramu;
- (7) maksymalnie 3 kolejne zmianyienne;
- (8) zestaw zasad przydzielania nocnych zmian.

Ograniczenia (2)-(4) opierają na istniejących zasadach dotyczących pracy szpitala i SOR-u. Można je zmieniać zgodnie z wymaganiami opieki klinicznej dla innych oddziałów. Ograniczenia (1) i (5) wprowadzają rządowe przepisy dotyczące zatrudnienia, które określają maksymalną liczbę godzin pracy w tygodniu.

Na podstawie ogólnych zasad dotyczących siły roboczej założyliśmy, że specjaliści zatrudnieni w pełnym wymiarze czasu pracy są zobowiązani do pracy przez określoną liczbę godzin na zmianę. W modelu podstawowym zajmujemy się ustalaniem harmonogramów pracy specjalistów na podstawie założenia o stałym budżecie kadrowym (liczba specjalistów do uwzględnienia w harmonogramie), opisanym w ograniczeniu (6). Ograniczenie (7) opisuje istniejący wymóg dotyczący polityki kadrowej, zgodnie z którym należy wprowadzić maksymalnie 3 kolejne zmiany nocne. Ograniczenie (8) zawiera zestaw wymagań dotyczących polityki kadrowej dla nocnych zmian. W zespole specjalistów niektórzy wyznaczeni specjaliści pracujący na nocne zmiany mogą być przydzieleni do maksymalnie 4 kolejnych nocnych zmian, podczas gdy w przypadku innych specjalistów dozwolone są maksymalnie 3 kolejne nocne zmiany, przy czym w każdym miesiącu mogą być najwyżej 2 takie sekwencje. Po wszystkich zmianach nocnych muszą nastąpić obowiązkowe dni przeznaczone na sen i odpoczynek.

Ponadto zmiany nocne muszą być przydzielane na zasadzie rotacji. W okresie dwóch kolejnych horyzontów planowania żaden specjalista nie może być przydzielony do więcej niż trzech bloków nocnych zmian. Ze względu na charakter nocnych zmian, które są uważane za niekorzystne, wszelkie wnioski o przydział na nocną zmianę mogą być rozpatrywane automatycznie z dodatkowymi ograniczeniami. Inne zasady dotyczące nocnych zmian można z łatwością modelować w podejściu MP.

Jak można sobie wyobrazić, powyższy zestaw ograniczeń i celów odzwierciedla sytuację konkretnego szpitala w danym kraju. Będziemy mogli uwzględnić specyficzne cele i ograniczenia, które dotyczą CSK MSWiA w Polsce, i dostosować model do ich potrzeb.

Zdajemy sobie sprawę, że sytuacja, którą modelowaliśmy, może bardzo różnić się od sytuacji, w jakiej znajdzie się SOR w Warszawie po wdrożeniu projektu.

### **Konkretne sformułowanie optymalizacji (przykład)**

W tym miejscu opisujemy algorytm, który po pewnych modyfikacjach można by wykorzystać do optymalizacji przydziału pielęgniarek/ratowników medycznych. Używamy ogólnego terminu „pielęgniarka”. Można go łatwo dostosować do innych specjalistów.

#### **Warunki**

- i. Okres objęty harmonogramem wynosi jeden miesiąc (cztery tygodnie). W ciągu dnia pracy zakładamy kilka zmian, obejmujących zmianyienne (D), popołudniowe/wieczorne (A) i nocne (N).

- ii. Liczba pielęgniarek na każdej zmianie musi spełniać określone wymagania, wynikające z optymalizacji liczby zmian (oddzielny dokument).
- iii. Pielęgniarki są klasyfikowane jako poziom 1 i poziom 2 (w oparciu o doświadczenie) - *można to dostosować, aby uwzględnić ratowników medycznych.*
- iv. Istnieją ograniczenia dotyczące stosunku liczby pielęgniarek na poziomie 1 do liczby pielęgniarek na poziomie 2 na danej zmianie. (stosunek  $\geq \alpha$ )
- v. Minimalna i maksymalna liczba zmian, które mają być przepracowane przez każdą pielęgniarkę (np. od 22 do 24).
- vi. Musi mieć co najmniej jeden dzień wolny w tygodniu
- vii. Może pracować tylko na jedną zmianę na dobę
- viii. Nie więcej niż  $\beta$  zmian nocnych w miesiącu
- ix. Niedozwolone są następujące po sobie zmiany nocne
- x. Pielęgniarka przełożona zawsze pracuje na porannej zmianie i ma dzień wolny w niedzielę
- xi. Wskaźniki
  - i: Liczba pielęgniarek ( $i=1, \dots, I$ )
  - j: Liczba zmian (M, A, N)
  - k: Liczba dni w horyzoncie planowania ( $k=1, \dots, K$ )
- xii. Parametry wejściowe
  - $R_j$ : Wymagana liczba pielęgniarek na każdej zmianie j
  - $E_i$ : Równa się 1, jeśli pielęgniarka i jest na poziomie 1, w innym przypadku równa się 0.
  - $PS_{i,j,k}$ : Preferencje pielęgniarki i do pracy na zmianie j w dniu k (1=tak, 0=nie)
  - $PD_{i,k}$ : Wynik preferencji pielęgniarek w zakresie brania dni wolnych (=3 dla najbardziej preferowanego dnia wolnego, 1 dla drugiego najbardziej preferowanego)
  - $WS_{min}$ : Minimalna dopuszczalna liczba zmian roboczych
  - $WS_{max}$ : Maksymalna dopuszczalna liczba zmian roboczych
  - $S_{target}$ : Cel dotyczący liczby zmian roboczych
  - $PS_{target}$ : docelowa liczba preferowanych zmian
  - $PD_{target}$ : docelowe preferowane dni wolne od pracy
- xiii. Zmienne decyzyjne
  - $X_{i,j,k}$  =1, jeśli pielęgniarka i przydzielono pracę na zmianie j w dniu k, w innym przypadku 0. Binarna.

- $Y_{i,k} = 1$ , jeśli pielęgniarka i przydzielono dzień wolny od pracy w dniu  $k$ , w innym przypadku 0. Binarna.
- $S_i^+$ ,  $S_i^-$ : dodatnie i ujemne odchylenia liczby zmian od wartości docelowej dla pielęgniarki  $i$
- $S_i^+$ ,  $S_i^-$ : Dodatnie i ujemne odchylenia liczby preferowanych zmian od wartości docelowej dla pielęgniarki  $i$
- $PD_i^+$ ,  $PD_i^-$ : Dodatnie i ujemne odchylenia wyników preferowanych dni wolnych od pracy od wartości docelowej dla pielęgniarki  $i$

CEL 1 : Liczba zmian, na których każda pielęgniarka przepracowała zbyt dużą lub zbyt małą liczbę godzin

$$(1) \quad \sum_{j=1}^J \sum_{k=1}^K X_{i,j,k} - S_i^+ + S_i^- = S_{target} \quad \forall i$$

CEL 2: Liczba preferowanych zmian przypisanych do każdej pielęgniarki

$$(2) \quad \sum_{j=1}^J \sum_{k=1}^K PS_{i,j,k} X_{i,j,k} - PS_i^+ + PS_i^- = PS_{target} \quad \forall i$$

CEL 3: Liczba preferowanych dni wolnych przypisanych do każdej pielęgniarki

$$(3) \quad \sum_{k=1}^K PD_{i,k} Y_{i,k} - PD_i^+ + PD_i^- = PD_{target} \quad \forall i$$

Celem tworzenia harmonogramu jest jednocześnie zminimalizowanie niepożądanych wyników tworzenia harmonogramu związanych z tymi trzema celami. Niepożądany wynik każdego celu jest normalizowany do swojej wartości docelowej. Funkcja celu zdefiniowana w równaniu (4) polega na minimalizacji sumy znormalizowanych niepożądanych wyników, jak pokazano poniżej. Każdemu z celów można przypisać inną wagę.

$$(4) \quad \text{minimize } Z = \left( \frac{\sum (S_i^+ + S_i^-)}{S_{target} \cdot I} \right) + \left( \frac{\sum (PS_i^-)}{PS_{target} \cdot I} \right) + \left( \frac{\sum (PD_i^-)}{PD_{target} \cdot I} \right)$$

Z następującymi ograniczeniami:

$$(5) \quad \sum_i X_{i,j,k} = R_j \quad \forall j, k \quad \text{Zapewnia, że pielęgniarki przydzielone do każdej zmiany spełniają wymagania}$$

$$(6) \quad WS_{min} \leq \sum_j \sum_k X_{i,j,k} \leq WS_{maks} \quad \forall i \quad \text{Ogranicza minimalną i maksymalną liczbę zmian roboczych w miesiącu dla pielęgniarek}$$

$$(7) \quad \sum_j (X_{i,j,k}) \leq 1 \quad \forall i, k \quad \text{umożliwia pielęgniarkom pracę tylko na jednej zmianie w ciągu dnia}$$

$$(8) \quad \sum_i E_i X_{i,j,k} \geq \alpha R_j \quad \forall j, k \quad \text{zapewnia odpowiednią liczbę doświadczonych pielęgniarek na każdej zmianie (a oznacza wymagany stosunek liczby pielęgniarek z doświadczeniem)}$$

$$(9) \quad \sum_j X_{i,j,k} + Y_{i,k} = 1 \quad \forall i, k \quad \text{W danym dniu pielęgniarka jest albo na zmianie, albo ma dzień wolny od pracy.}$$

(10)  $\sum_{j \in N} \sum_k X_{i,j,k} \leq \beta \quad \forall i$  Egzekwowanie maksymalnej (b) dopuszczalnej liczby zmian nocnych w miesiącu.

(11)  $\sum_{j \in N} \sum_{f=k}^{k+1} X_{i,j,f} \leq 1 \quad \forall i, k$  Zabrania przydzielaniu kolejnych zmian nocnych.

(12)  $\sum_{f=k}^{k+6} Y_{jeśli} \geq 1 \quad \forall i, k = 1, 8, 15, 21$  Zapewnia, że każda pielęgniarka ma prawo do co najmniej jednego dnia wolnego w tygodniu.

(13)  $X_{i=HeadNurse, j=M, k} = 1 \quad \forall k$  Pielęgniarka przełożona (ang. head nurse) pracuje na ranną zmianę

(14)  $Y_{i=HeadNurse, k} = 1 \quad \forall k = Sunday$  Pielęgniarka przełożona nie pracuje w niedziele

(15)  $S_i^+, S_i^-, PS_i^+, PS_i^-, PD_i^+, PD_i^- \in Z^+ \quad \forall i$  Zmienne odchylen muszą być dodatnimi liczbami całkowitymi