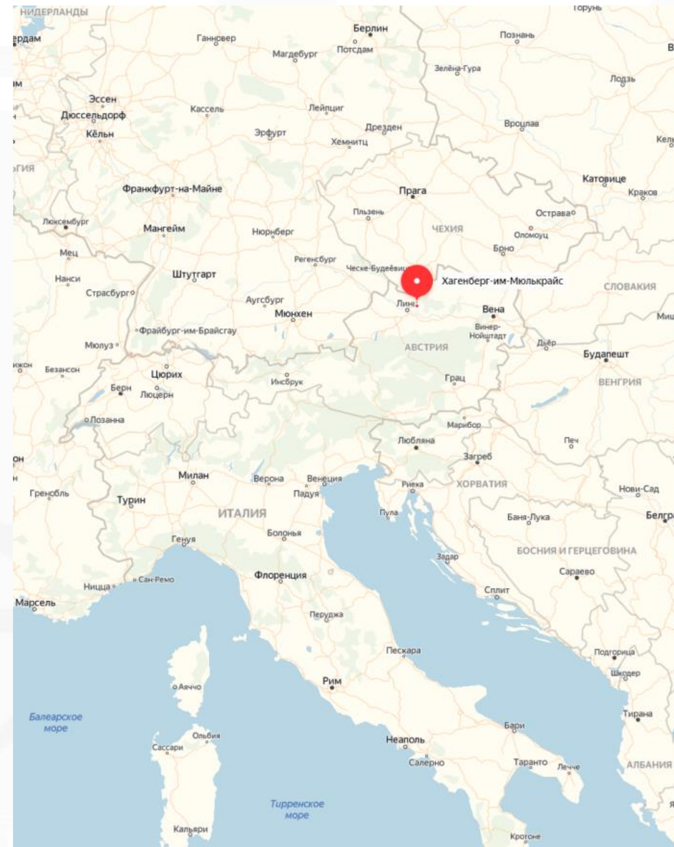
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Исследование технологий обработки и анализа энергоданных на базе платформы Flex+

Результаты научной стажировки в
Центре компетенций по программному обеспечению
г. Хагенберг, Австрия

к.т.н., доцент кафедры ИТБ
А.В. Кычкин

Software Competence Center Hagenberg (SCCH)



Softwarepark Hagenberg - 20 лет

Более 300 научных и индустриальных партнеров

- Университет Иоганна Кеплера
- Университет прикладных наук верхней Австрии

Программа стажировки

Период стажировки: 16.09.2019 - 06.12.2019

№	Содержание	Длительность, дни
1	Знакомство с научной организацией, ее структурой и проектом Flex+, командой проекта	1
2	Установка и настройка среды Python, базы данных временных рядов InfluxDB и веб-сервера Grafana	3
3	Анализ модели хранения данных в csv файлах. Импорт реальных энергетических данных объектов	2
4	Реализация в Python модели энергобаланса, дискретного автомата для управления инвертором	25
5	Реализация краткосрочных моделей прогнозирования электрической нагрузки зданий для DA (на сутки вперед) рынка управления спросом	23
6	Верификация и оптимизация моделей	17
7	Подготовка публикации для 21-го Всемирного конгресса по автоматическому управлению IFAC 2020 в Берлине, Германия	11
	Всего:	82

О проекте Flex+

Рынок управления спросом на электрическую энергию на сутки вперед
Агрегаторы от 50 МВт
Потребители - жилые дома
Установки:
- солнечная панель
- инвертор
- аккумуляторная батарея
- тепловой насос
- электрический бойлер
- бытовые устройства

Задачи:
- повышение гибкости потребителей
- оптимизация управления инвертором

SCCH | Innovation & Research | Networking | Publications | Career | Maps & Directions

DAS DATA ANALYSIS SYSTEMS

OVERVIEW | RESEARCH TOPICS | PROJECTS | PUBLICATIONS | CONTACT

DAS

FLEX+

Large-scale use of prosumer flexibility in short-term electricity markets, taking into account prosumer interests

INITIAL SITUATION, CHALLENGES AND MOTIVATION

The integration of prosumers into the energy markets is promoted both at the European level, for example in the so-called winter package, but also at the national level in order to actively involve the prosumers in the market and to integrate fluctuating renewable energies by using their flexibility. From a technical point of view, automatically-controllable prosumer components such as **heat pumps, electric boilers, battery storage and e-mobility** are particularly suitable for this purpose. Surveys in the MBS+ and EcoGrid EU projects have shown that there is a great interest on the prosumers side to make their flexibility available externally, in order to contribute to a rapid energy supply that is affordable for society. Unlike in Austria, there are already existing business models in the area of private flexibility marketing in Germany and Switzerland. However, due to the different legal, regulatory and economic circumstances, these can not be transferred one-by-one to Austria. Studies in the above research projects also show that the willingness of the prosumer to participate is very strongly influenced by certain factors, such as their own interests - an aspect which is not or only insufficiently taken into account in existing business models in the other countries. By comparison, financial interests play a less important role. In order to exploit the existing potential from an economic point of view, the needs of the prosumer must therefore be taken into account accordingly.

OBJECTIVES AND INNOVATION

The project develops scalable optimization algorithms at the aggregator and prosumer level, which take into account not only the interests of the aggregator, but also the needs / interests of the prosumer. Under this premise, an optimal market-wide use and marketing of the existing flexibility in private households is to be made possible for all stakeholders. This premise is subsequently tested in large real-time operation and evaluated for selected use cases, such as the commercialization of spot and balanced energy markets as well as for the minimization of balancing energy. While existing business models are only available for a single component / technology, the planned project is designed to increase the flexibility of several different components in a household, such as heat pumps, electric boilers, battery storage and e-mobility, and to make use of these markets across a range of selected (system) services.

DAS

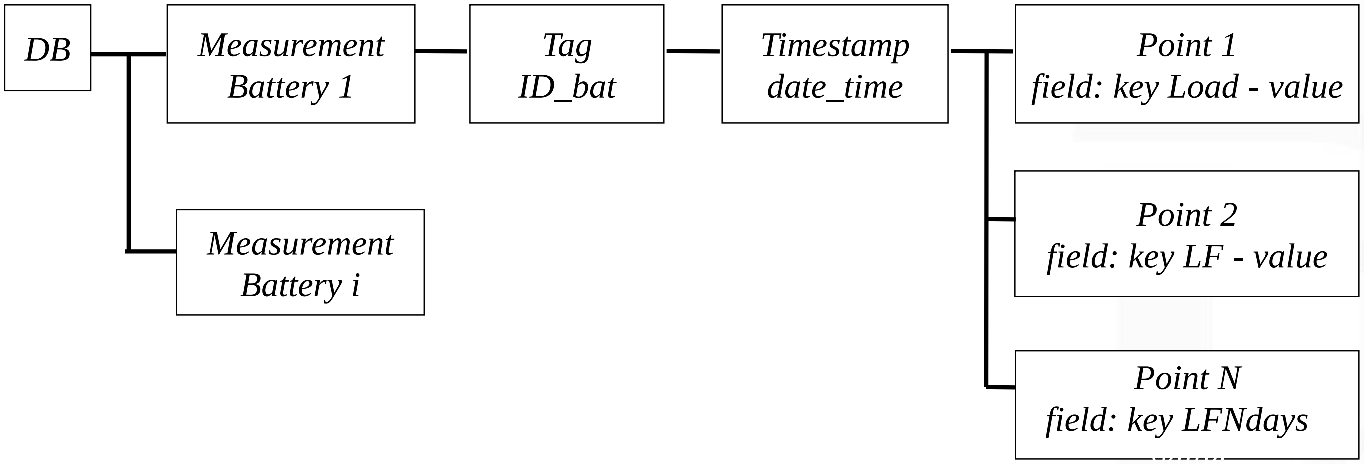
KEYWORD SEARCH

data centric technologies
Data Mining
Model Predictive Control
Modelling
Process data analysis
Fault detection
Predictive analytics
Predictive maintenance
Stream data analysis
data management
prognosis
control and optimization
knowledge representation and semantics
big data
knowledge representation
deep learning

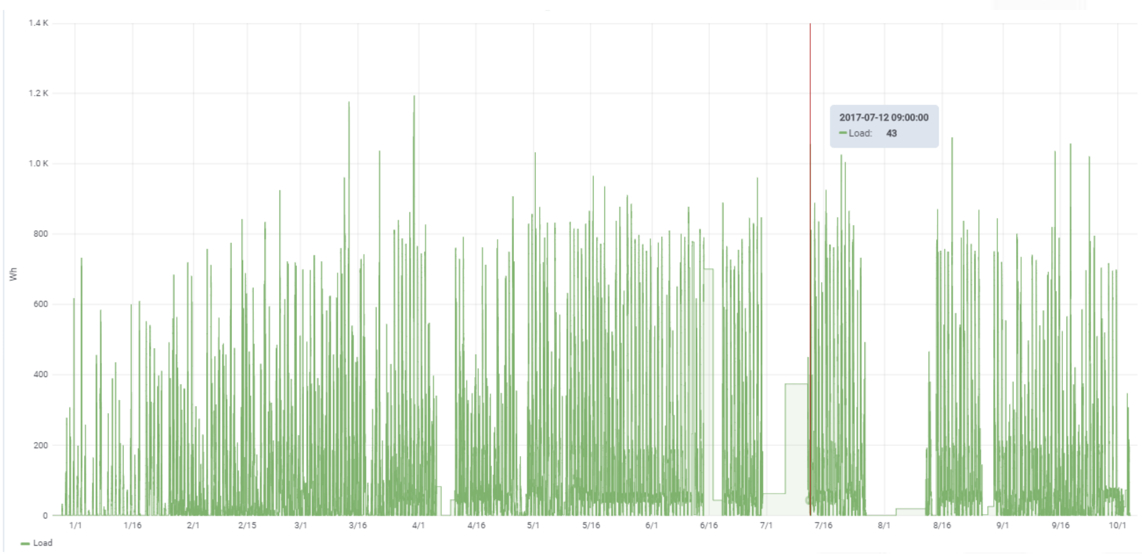
Исходные данные

Zeitpunkte innerhalb des Optimierungszeitraums (+24h)	Время в период оптимизации
Batterieladezustand in %	Заряд аккумулятора
Last Prognose als Energie in 15 Minuten Intervallen. Multipliziert mit 4 ergibt sich die Leistung	Прогнозное значение нагрузки на 15 минут
PV Prognose als Energie in 15 Minuten Intervallen. Multipliziert mit 4 ergibt sich die Leistung	Прогнозное значение PV на 15 минут
Kaufpreis des Stroms (inkl. Netzgebühren). Formel: Netzgebühren + (awattarPreis+Ökostromabgate)*Ust; Ökostromabgabe = 0,001€, Ust=1,2	Покупная цена на ЭЭ, включая сетевые сборы
Verkaufspreis des Stroms	Цена продажи ЭЭ
Batteriekapazität in Wh	Емкость батареи
Maximale Ladeleistung der Batterie	Максимальная зарядная мощность батареи
Maximale Entladeleistung der Batterie	Максимальная разрядка аккумулятора
Maximale Wechselrichter Leistung von DC zu AC	Максимальная мощность инвертора от постоянного тока до переменного
Maximale Wechselrichter Leistung von AC zu DC	Максимальная мощность инвертора от переменного тока до постоянного
Tatsächlicher Energiefluss ins Netz	Фактический поток энергии в сеть
Tatsächlicher Energiefluss vom Netz	Фактический поток энергии из сети
Tatsächlicher Energiefluss an Verbraucher/Last	Фактический поток энергии потребителю / нагрузке
Tatsächlicher Energiefluss vom Wechselrichter auf AC-Seite gemessen	Фактический поток энергии от инвертора, измеренный на стороне переменного тока
Tatsächlciher Energiefluss zum Wechselrichter auf AC-Seite gemessen	Фактический поток энергии к инвертору измеряется на стороне переменного тока
Tatsächlicher Energiefluss vom PV-Modul	Фактический поток энергии от фотоэлектрического модуля
Tatsächlicher Energiefluss von Batterie	Фактический поток энергии из батареи
Tatsächlicher Energiefluss zur Batterie	Фактический поток энергии в батарею
SoC laut SolarWeb am Ende des 15 Minuten Intervalls	SoC в соответствии с SolarWeb в конце 15-минутного интервала

Хранение и визуализация данных

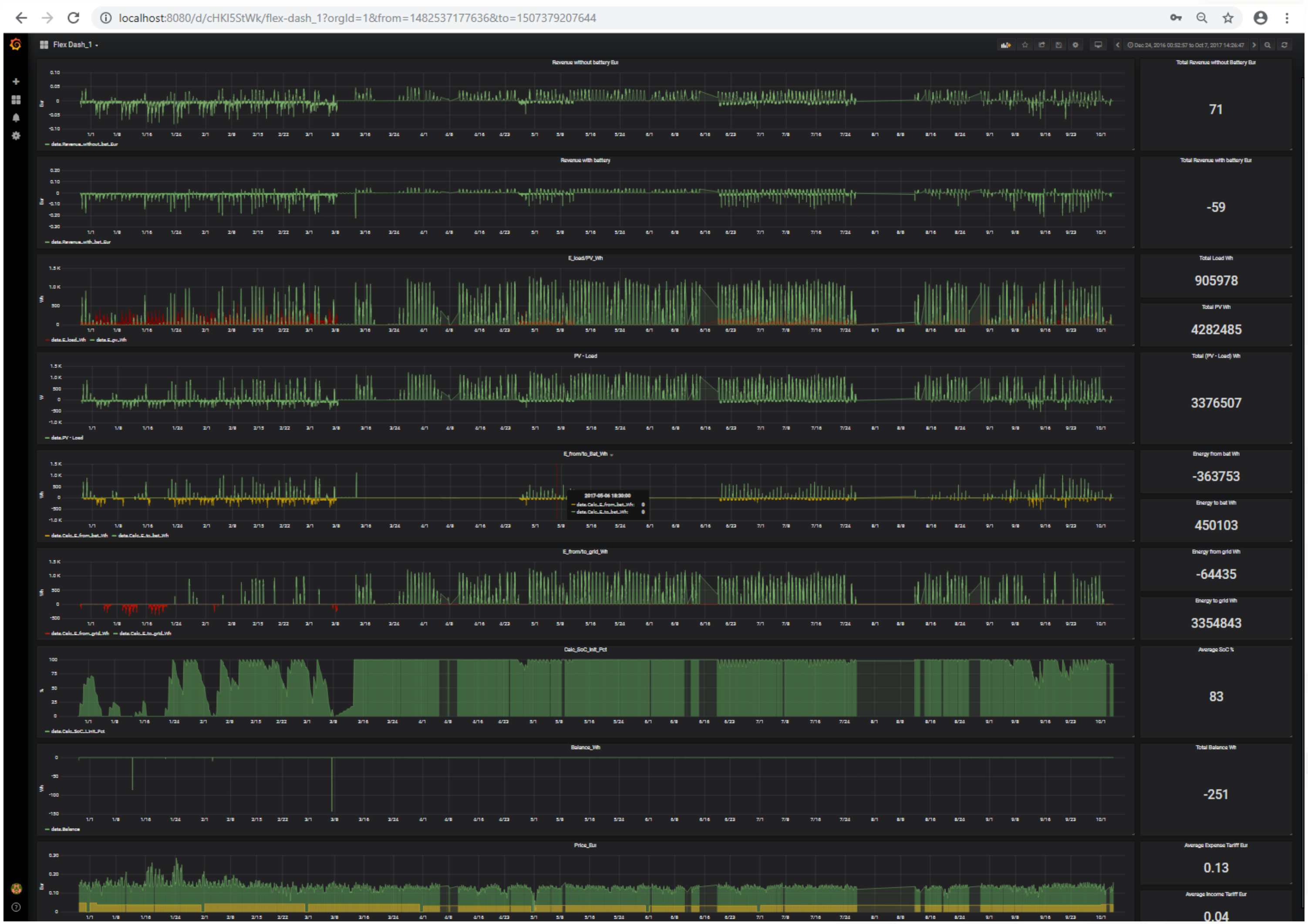


СУБД – InfluxDB
Порт: 8086



WEB – Grafana
Порт: 8080

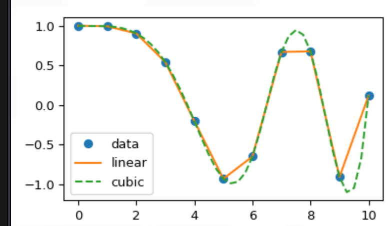
Подготовка данных

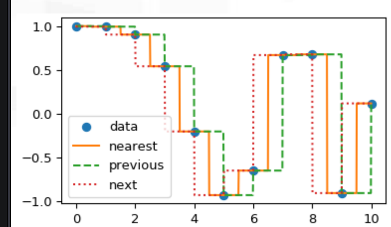


Фильтрация отрицательных значений и величин, выходящих за пределы

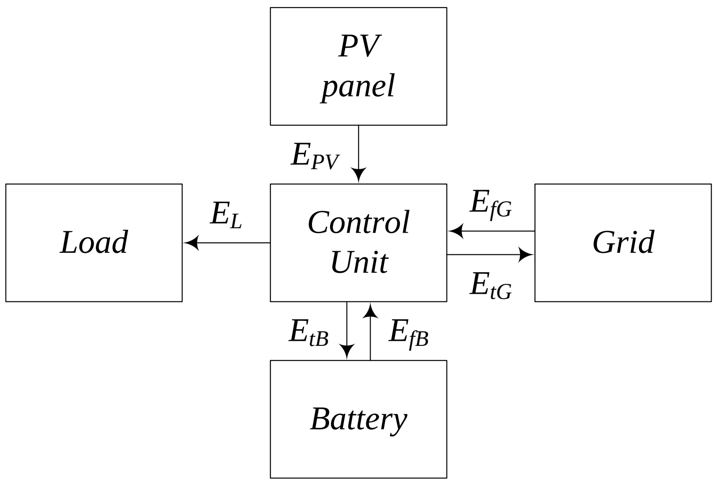
Индексация по времен 15 мин

Заполнение пустых значений





Расчет энергобаланса



E_{PV} – energy which goes from PV panel
E_L – energy consumption by Load
E_{tB} – charged energy to battery, $E_{tB} < 0$
E_{fB} – discharged energy from battery, $E_{fB} >= 0$
E_{tG} – energy to grid (energy to sell), $E_{tG} < 0$
E_{fG} – energy from grid (energy to buy), $E_{fG} >= 0$

Balance can be calculated by:
$B = \Delta E - (E_{tB} + E_{fB} + E_{tG} + E_{fG})$
where $\Delta E = E_{PV} - E_L$

Расчет базовой мощности, потенциальной гибкости и состояния заряда батареи

Algorithm 1 Baseline power to battery i at time interval t

procedure $\mathcal{B}_i(\Delta P_i(t),\, \mathrm{SOC}_i(t))$
  Step 1a. maximum possible (average) charging power to battery
  $\Delta T^*_{c,i}(t) = (1 - \mathrm{SOC}_i(t)) \cdot \frac{\chi_i}{c_{\max,i}\eta_{c,i}}$ ▷ We make the assumption here that the *maximum charging power*, $c_{\max,i}$, is evaluated at the output of the battery. Thus, the actual charging power (that directly affects the $\mathrm{SOC}_i(t)$) will be a bit smaller than $c_{\max,i}$ due to the energy losses during charging (captured by the efficiency rate $\eta_{c,i}$). $\Delta T^*_{c,i}(t)$ is the minimum time needed for the battery to reach its full energy capacity, denoted by χ_i, at the maximum charging power $c_{\max,i}$.
  $P_{b,c,\max,i}(t) = c_{\max,i} \cdot \min\left\{\Delta T^*_{c,i}, \Delta T\right\}/\Delta T$ ▷ This is the maximum (average over a ΔT time-interval) charging power that could be achieved within ΔT.
  Step 1b. maximum possible (average) discharging power to battery
  $\Delta T^*_{d,i}(t) = \mathrm{SOC}_i(t) \cdot \frac{\chi_i \eta_{d,i}}{d_{\max,i}}$ ▷ Analogously to the charging case, we make the assumption here that the maximum discharging power, denoted by $d_{\max,i}$, is evaluated at the output of the battery. Thus, the actual discharging power (that directly affects the $\mathrm{SOC}_i(t)$) will be a bit larger than $d_{\max,i}$ due to the energy losses during discharging (captured by the efficiency rate $\eta_{d,i}$). $\Delta T^*_{d,i}$ is the minimum time needed for the battery to empty at the maximum discharging speed $d_{\max,i}$.
  $P_{b,d,\max,i}(t) = -d_{\max,i} \cdot \min\left\{\Delta T^*_{d,i}, \Delta T\right\}/\Delta T$ ▷ This is the maximum (average over a ΔT time-interval) discharging power that could be achieved within ΔT.
  Step 2. baseline power to the battery
  if $\Delta P_i(t) \geq 0$ **then**
    $P_{b,\mathrm{base},i}(t) = \min\left\{P^*_{b,c,i}(t), \Delta P_i(t)\right\} \geq 0$ ▷ if there is an excess of power, a battery is always charged at maximum possible (average) power
  else
    $P_{b,\mathrm{base},i}(t) = \max\left\{P^*_{b,d,i}(t), \Delta P_i(t)\right\} \leq 0$ ▷ if there is a shortage of power, the battery is always discharged at maximum possible (average) power
  return $P_{b,\mathrm{base},i}(t)$ ▷ baseline power to battery at interval t

[*Chasparis et al., 2019*]

Algorithm 2 Flexibility potential of household i at time interval t

procedure $\mathcal{V}_i(P_{b,\mathrm{base},i}(t),\, \mathrm{SOC}_i(t))$
  Maximum charging (flexibility) potential
  $v_{c,i}(t) = [P_{b,c,\max,i}(t) - P_{b,\mathrm{base},i}]_+ \cdot \Delta T$
  Maximum discharging (flexibility) potential
  $v_{d,i}(t) = [P_{b,d,\max,i}(t) - P_{b,\mathrm{base},i}]_- \cdot \Delta T$
  return $v_{c,i}(t)\,, v_{d,i}(t)$ ▷ flexibility potential at interval t

Algorithm 3 State-of-Charge of battery i at time $t+1$

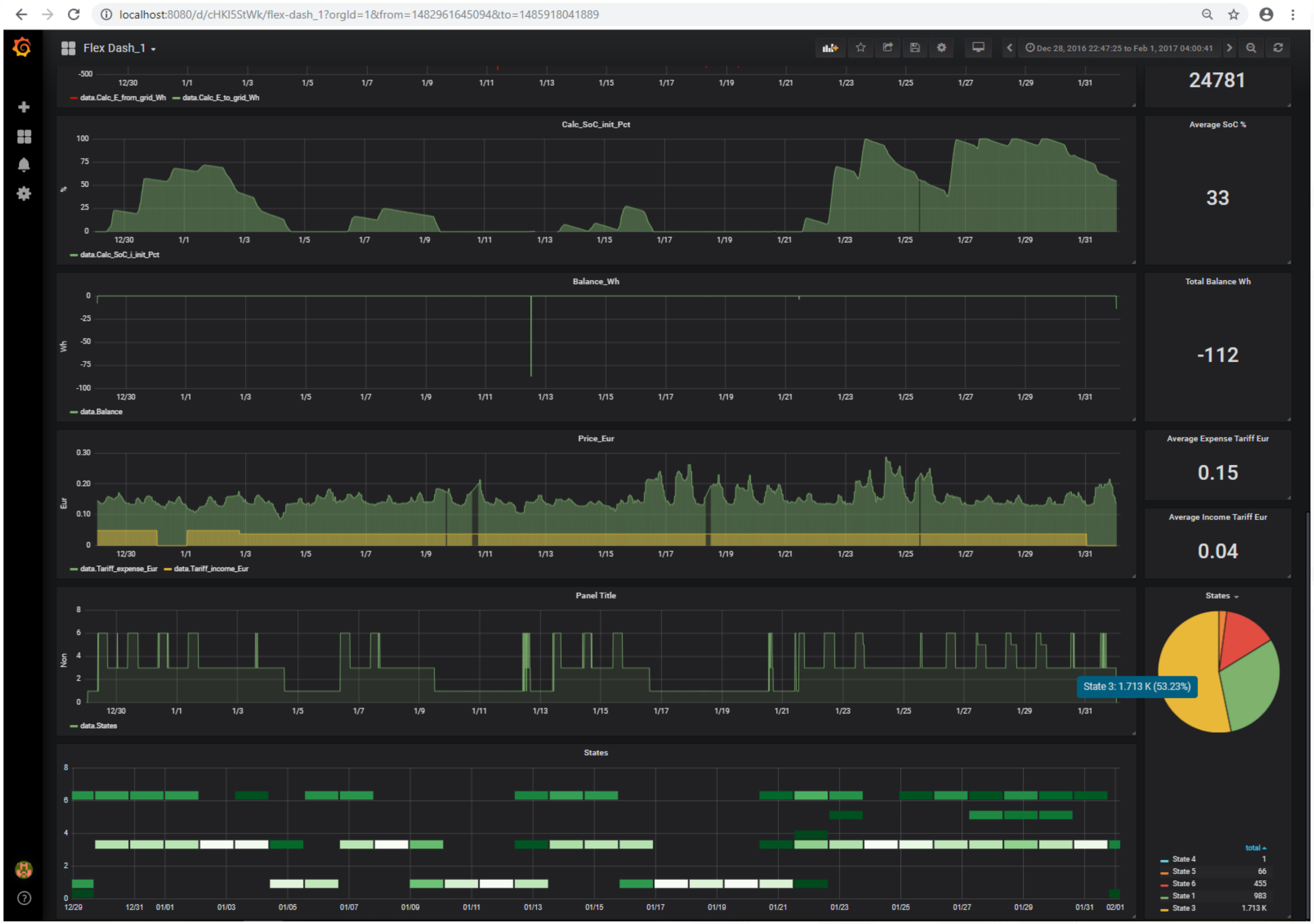
procedure $\Sigma_i(\mathrm{SOC}_i(t), v_{d,i}(t), v_{c,i}(t), u_i(t))$
  Step 1. *average power to battery under* $u_i(t)$,
  if $u_i(t) > 0$ **then**
    $P_{b,i}(t) = P_{b,\mathrm{base},i}(t) + u_i(t) \cdot v_{c,i}(t)/\Delta T$ ▷ Charging flexibility potential $v_{c,i}(t)$ during time interval t has been computed by first applying Algorithm 1 to compute $P_{b,\mathrm{base},i}(t)$ and then applying Algorithm 2 for computing the available charging potential $v_{c,i}(t)$
  else
    $P_{b,i}(t) = P_{b,\mathrm{base},i}(t) - u_i(t) \cdot v_{d,i}(t)/\Delta T$ ▷ Discharging flexibility potential $v_{d,i}(t)$ during time interval t has been computed by first applying Algorithm 1 to compute $P_{b,\mathrm{base},i}$ and then applying Algorithm 2 for computing the available discharging potential $v_{d,i}(t)$
  Step 2. *State-of-Charge at time* $t+1$
  if $P_{b,i}(t) > 0$ **then**
    $\mathrm{SOC}_i(t+1) = \mathrm{SOC}_i(t) + \eta_{c,i}\frac{P_{b,i}(t)\Delta T}{\chi_i}$
  else
    $\mathrm{SOC}_i(t+1) = \mathrm{SOC}_i(t) + \frac{P_{b,i}(t)\Delta T}{\eta_{d,i}\chi_i}$
  return $\mathrm{SOC}_i(t+1)$ ▷ State-of-Charge at $t+1$

Дискретный автомат для управления инвертором

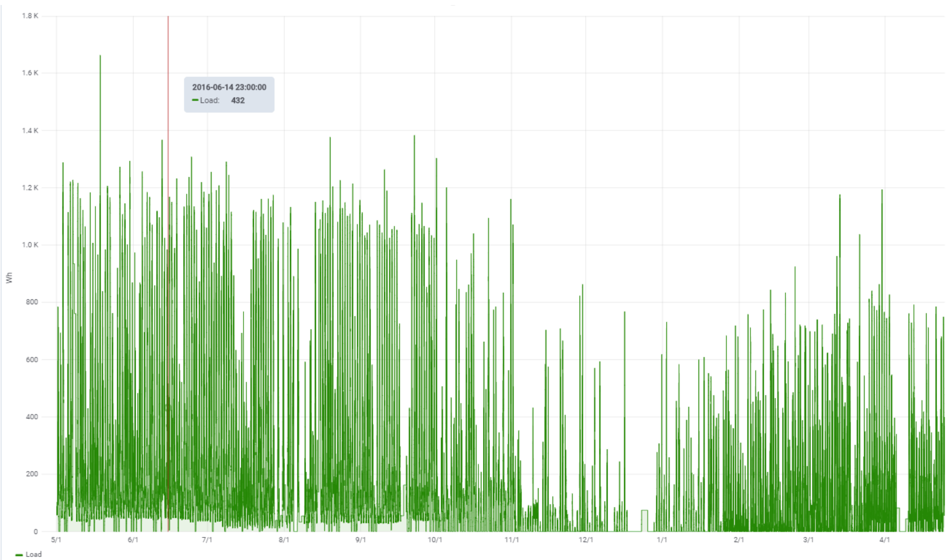
ΔE >=0	$SoC(t+1)$ ">0%"	$SoC(t+1)$ "<100%"		EtB	EfB	EfG	EtG	st
0	0	0		x	x	x	x	x
0	0	1	bat empty	0	0	1	0	1
0	1	0	bat full	0	1	0	0	2
0	1	1		0	1	0	0	3
1	0	0		x	x	x	x	x
1	0	1	bat empty	1	0	0	0	4
1	1	0	bat full	0	0	0	1	5
1	1	1		1	0	0	0	6

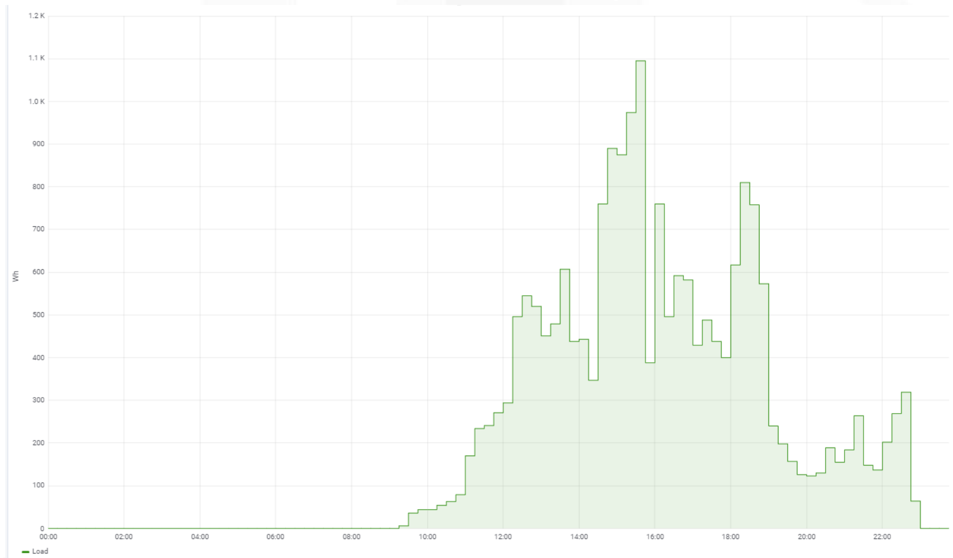
For “True” is used “1”
For “False” is used “0”
“False” in case the “$SoC(t+1) > 0\%$” means that the battery is empty
“False” in case the “$SoC(t+1) < 100\%$” means that the battery is full
“1” in case E_{fB} means that energy goes from battery
“1” in case E_{tB} means that energy goes to battery
“1” in case E_{fG} means that energy goes from grid
“1” in case E_{tG} means that energy goes to grid

Анализ переходов состояний



Модели прогнозирования электрической нагрузки

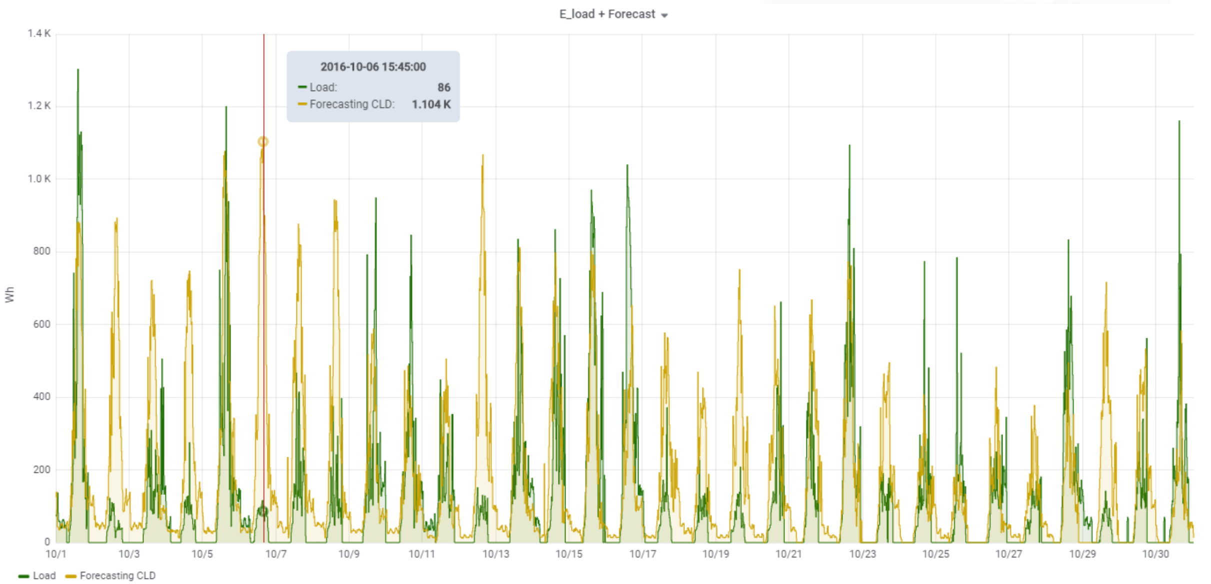


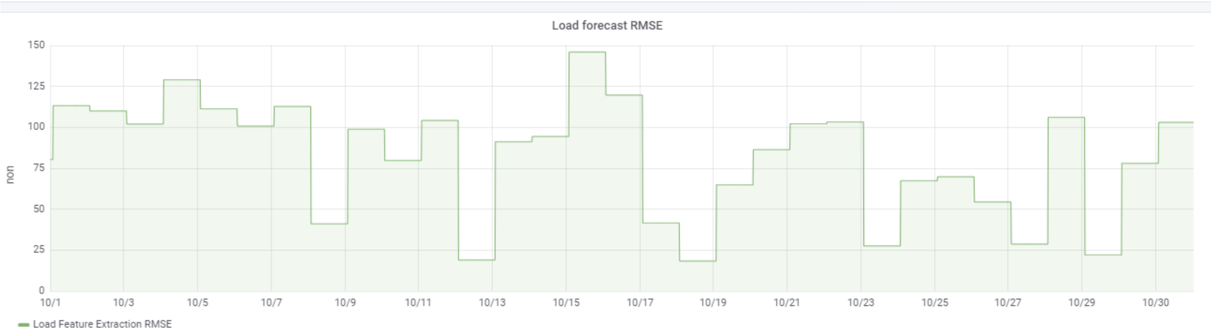


16-11-2016 .. 15-05-2017

Модель 1. Устойчивая модель

$$\hat{y}_t = \frac{y_{t-7N} + y_{t-14N} + y_{t-21N}}{3} \qquad N = 96$$





Модель 2. Базовая линия

1) Formation of the window for constructing the Load data prediction (during construction, the last N (N = 10) working days from no more than 45 calendar days preceding the day for which the base load schedule (target day) is built taking into account the following restrictions

   The load window does not include:
   - weekend days;
   - days with daily consumption less than 50% of the average daily consumption over the last N working days;
   - days for which there is no data during few hours;
   - any other exeptions.

Note: in the absence of the possibility of constructing a window for the Load data predicion with due regard for the restrictions imposed, the value of the Load is assumed to be zero.

2) The selection of N working days for the construction of the Load data prediction window, each of them will be used in the calculation.
3) The calculation of Load prediction is based on finding the arithmetic average consumption for each 15 minute interval of the day included in the construction of Load data prediction; carried out according to the following formula:

$$b_t = \frac{\sum_{d=1}^{N} c_{td}}{N},$$ where

b_t – the value of the Load data prediction per 15 minute interval t,
c_{td} – the electricity consumption per 15 minute interval t on day d,
t – the serial number of the 15 minute interval on day d, takes values from 1 to 96,
d – the day from the set of days, takes values from 1 to N.

4) Adjustment of the calculated Load data prediction use a symmetric additive adjustment.
   $b_{adj\_t} = b_t + a$, where
   b_{adj_t} – the value of the load per 15 minute interval t, taking into account the adjustment,
   a – the value of adjustment.

Note: The adjustment value is determined as the arithmetic average of the difference between the total energy consumption in working day, preceding settlement target day, determined according to the Load data prediction for the same day:

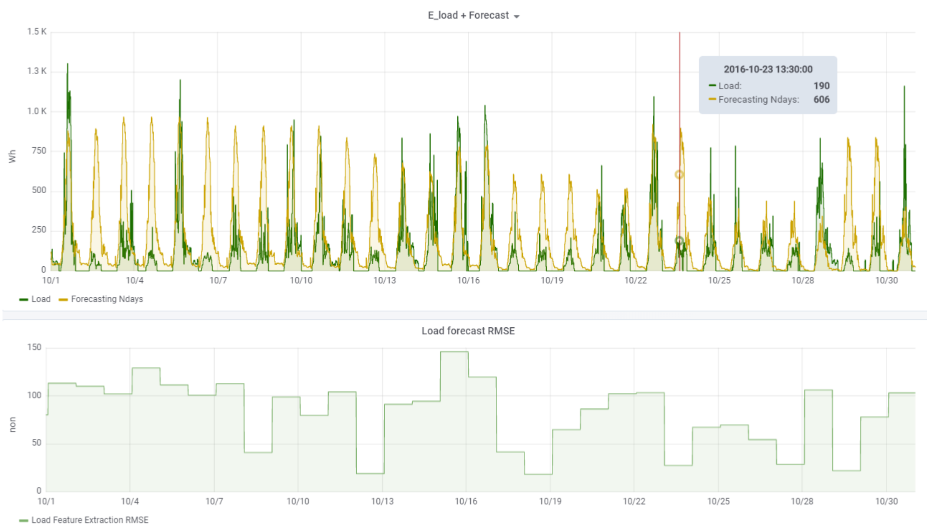
$$a = \frac{\sum_{t=1}^{96}(c_{t(d-1)} - b_{t(d)})}{2}$$

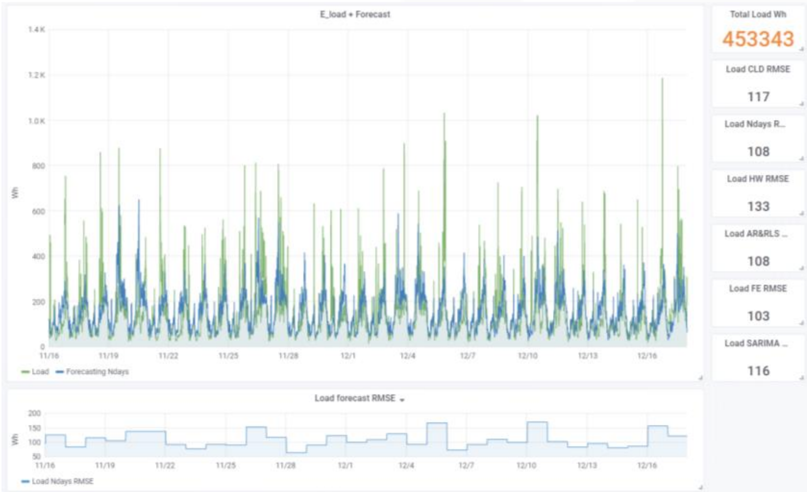
5) Check the need for tuning restrictions, apply the restriction (if necessary). The value of the Load data prediction taking into account the adjustment should not be less than 0.8 of the value of the Load data prediction without adjustment and should not exceed 1.2 values of the Load data prediction without adjustment.
   IF $b_{adj\_t} < 0.8 \times b_t$ THEN $b_{adj\_t} = 0.8 \times b_t$
   IF $b_{adj\_t} > 1.2 \times b_t$ THEN $b_{adj\_t} = 1.2 \times b_t$

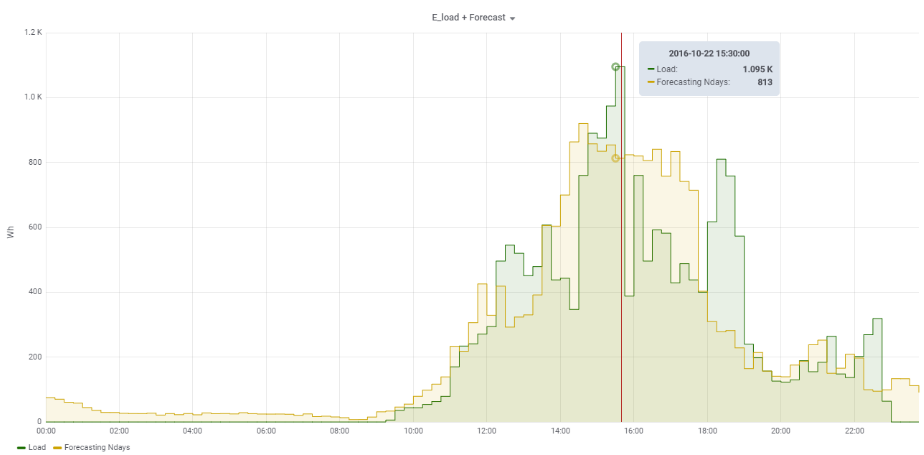
[*Определение объема снижения потребления энергопринимающего устройства с использованием графика базовой нагрузки и порядок построения такого графика*]

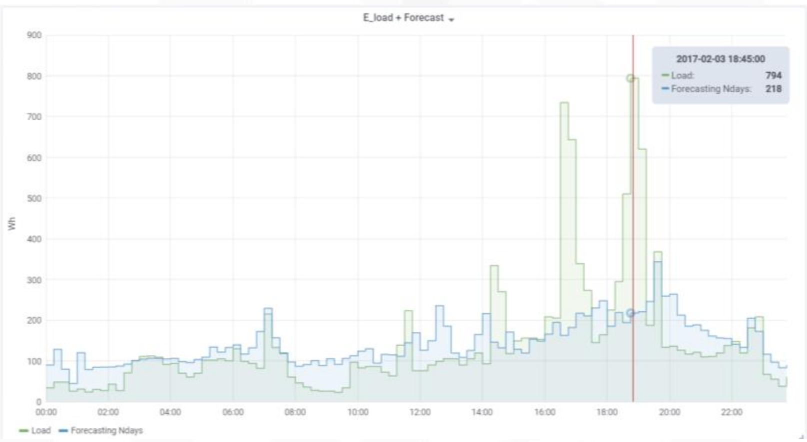
[*Постановление Правительства РФ от 20 марта 2019 г. № 287 “О внесении изменений в некоторые акты Правительства Российской Федерации по вопросам функционирования агрегаторов управления спросом на электрическую энергию в Единой энергетической системе России, а также совершенствования механизма ценозависимого снижения потребления электрической энергии и оказания услуг по обеспечению системной надежности”*]

Модель 2. Базовая линия









Модель 3. Тройное экспоненциальное сглаживание

The basic equation for the method is given by:

$\hat{y}_t = L_{t-1} + P_{t-1} + S_{t-T}$, where

L – level component given by: $L_t = \alpha \cdot (y_t - S_{t-T}) + (1-\alpha) \cdot (L_{t-1} + P_{t-1})$

P – trend component given by: $P_t = \beta \cdot (L_t - L_{t-1}) + (1-\beta) \cdot P_{t-1}$

S – season component given by: $S_t = \gamma \cdot (y_t - L_t) + (1-\gamma) \cdot S_{t-T}$

t – moment in time

$\hat{y}_t$ – predicted value of Load in moment t

y_t – real (measured) value of Load in moment t

T – time series period

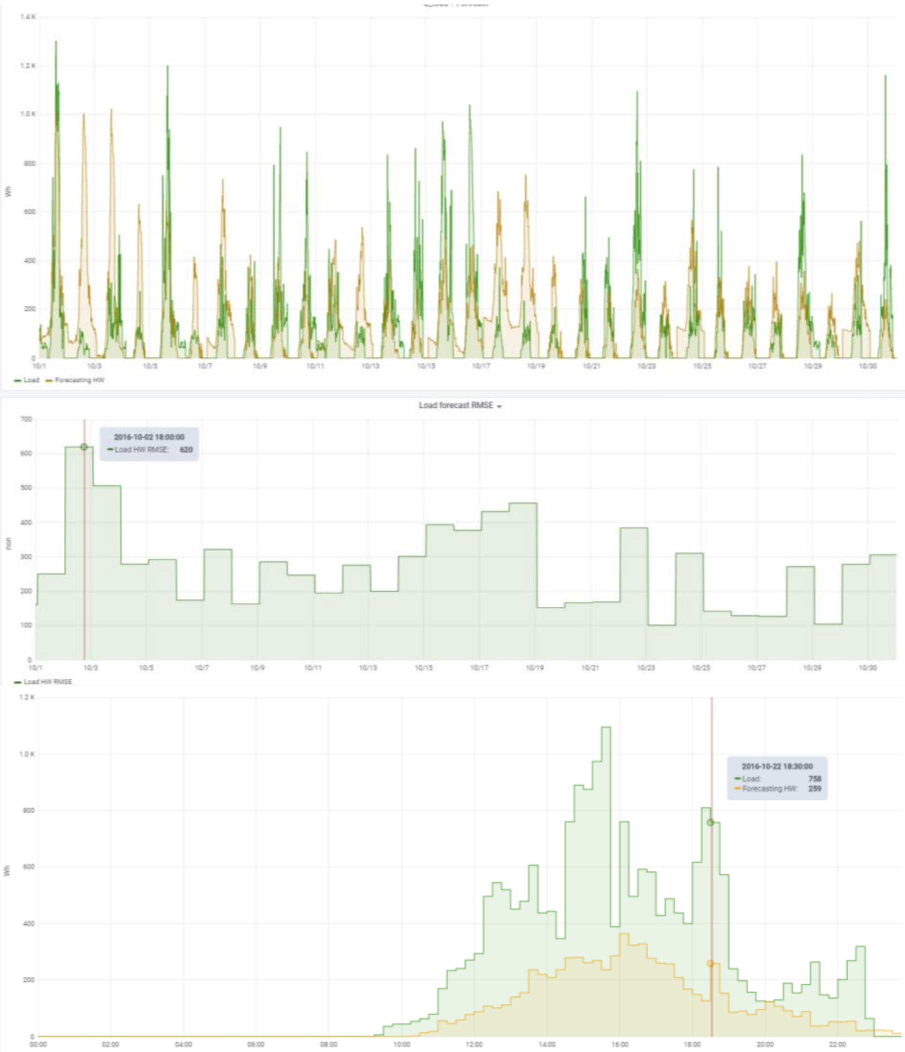
α – data smoothing factor

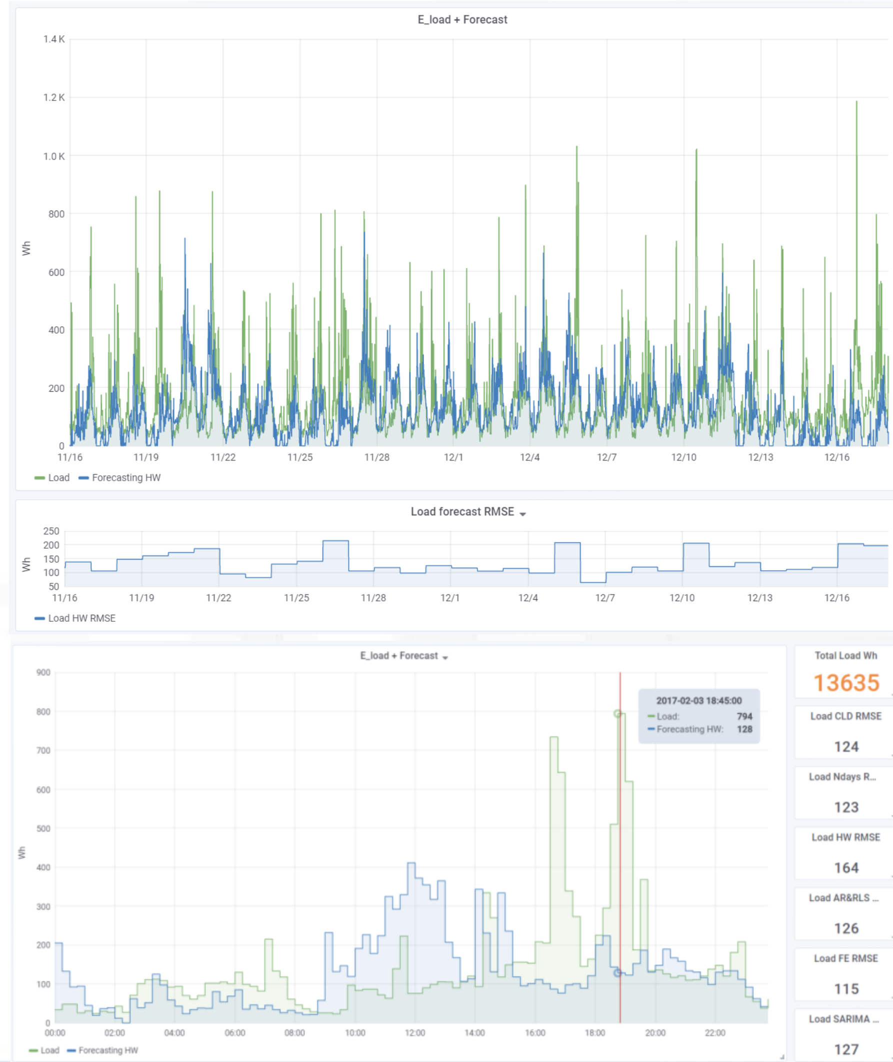
β – trend smoothing factor

γ – seasonal change smoothing factor

[*Maciej Szmit, Anna Szmit, Sławomir Adamus, Sebastian Bugała. Implementation of Brutlag's algorithm in Anomaly Detection 3.0. Proceedings of the Federated Conference on Computer Science and Information Systems 2012. pp. 685–691*]

Модель 3. Тройное экспоненциальное сглаживание





Модель 4. Сезонная интегрированная модель авторегрессии – скользящего среднего (SARIMA)

SARIMA *(p, d, q)(P, D, Q)$_S$* :

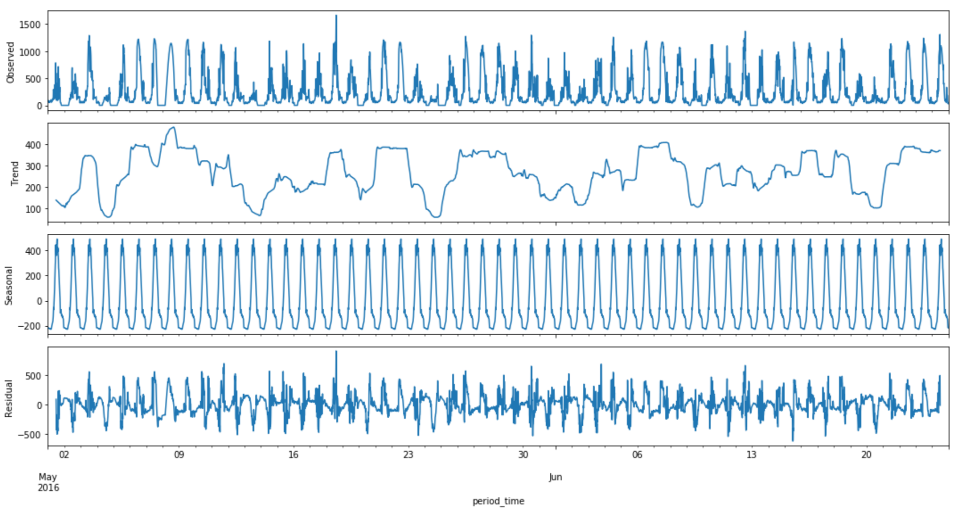
$$\phi(L)\Phi(L^S)(1-L)^d(1-L^S)^D y_i = \theta(L)\Theta(L^S)\varepsilon_i$$

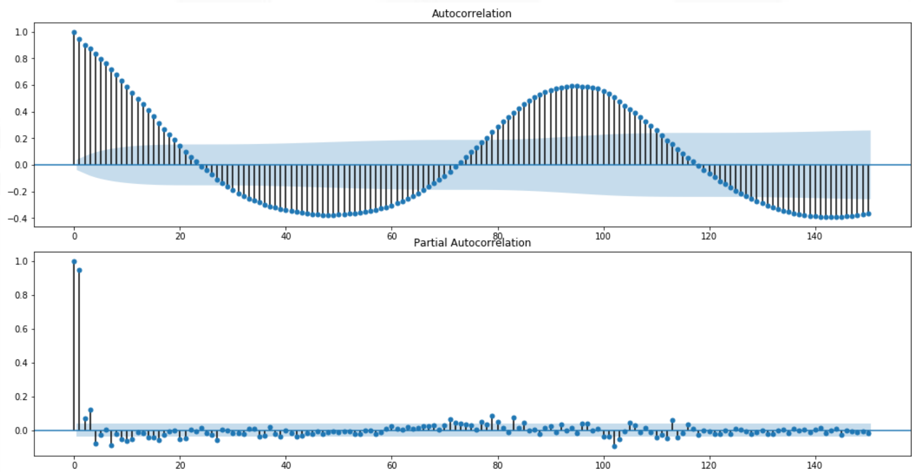
$$\Phi(L)y_i = (1-\Phi_1 L^1 - \Phi_2 L^2 - \ldots - \Phi_P L^P)y_i$$

$$\Theta(L)\varepsilon_i = (1+\Theta_1 L^1 + \Theta_2 L^2 + \ldots + \Theta_Q L^Q)\varepsilon_i$$

p: Trend autoregression order
d: Trend difference order
q: Trend moving average order

P: Seasonal autoregressive order (with coefficients $\Phi_1, \ldots, \Phi P$)
D: Seasonal difference order based on S seasonal periods
Q: Seasonal moving average order (with coefficients $\Theta_1, \ldots, \Theta Q$)
S: The number of time steps for a single seasonal period





Модель 4. Сезонная интегрированная модель авторегрессии – скользящего среднего (SARIMA)

Algorithm for SARIMA (1,1,1)(1,1,1)96 forecasting can be shown by:

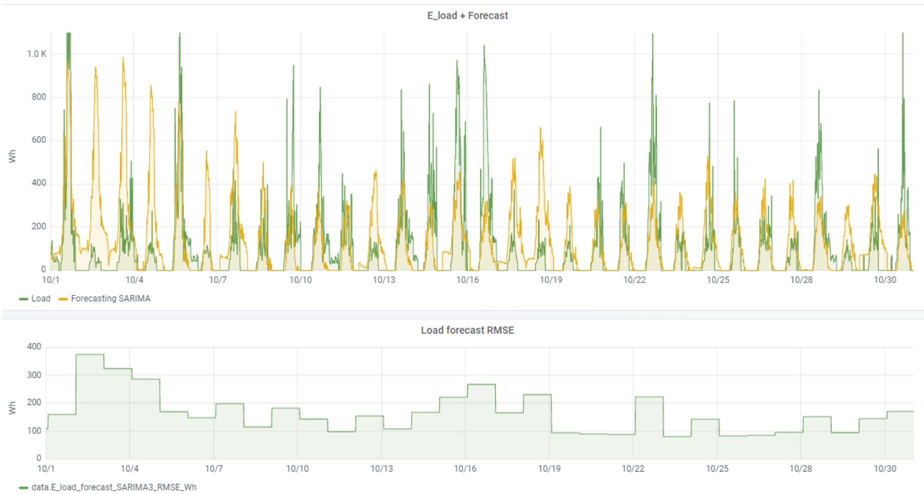
1. $y = [y_1, y_2, \dots y_i]$ – load energy data
2. First differencing corresponding to time series shifting d can be written:
   $y'_i = y_i - y_{i-1}$
3. Second differencing corresponding to seasonal shifting D can be written:
   $y''_i = y'_i - y'_{i-96} = (y_i - y_{i-1}) - (y_{i-96} - y_{i-96-1})$
4. The residuals can be calculated in the end of day by:
   $\varepsilon_i = y''_i - \phi_0 - \phi_1 y''_{i-1} - \Phi_1 y''_{i-96} + \phi_1\Phi_1 y''_{i-96-1} - \theta_1\varepsilon_{i-1} - \Theta_1\varepsilon_{i-96} - \theta_1\Theta_1\varepsilon_{i-96-1}$

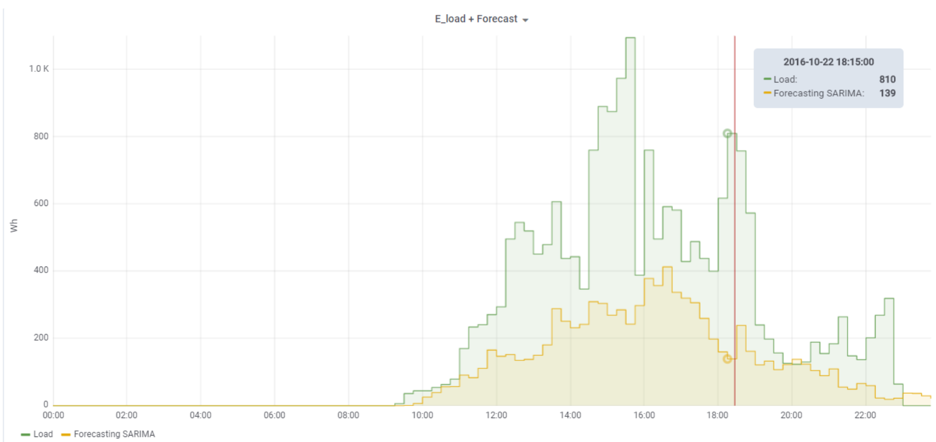
   This equation takes the following form:
   $\phi(L)\Phi(L^{96})y''_i = \phi_0 + \theta(L)\Theta(L^{96})\varepsilon_i$
   $(1-\phi_1 L^1)(1-\Phi_1 L^{96})y''_i = \phi_0 + (1+\theta_1 L^1)(1+\Theta_1 L^{96})\varepsilon_i$
   $(1-\phi_1 L^1 - \Phi_1 L^{96} + \phi_1\Phi_1 L^{96+1})y''_i = \phi_0 + (1+\theta_1 L^1 + \Theta_1 L^{96} + \theta_1\Theta_1 L^{96+1})\varepsilon_i$
   $y''_i - \phi_1 y''_{i-1} - \Phi_1 y''_{i-96} + \phi_1\Phi_1 y''_{i-96-1} = \phi_0 + \varepsilon_i + \theta_1\varepsilon_{i-1} + \Theta_1\varepsilon_{i-96} + \theta_1\Theta_1\varepsilon_{i-96-1}$

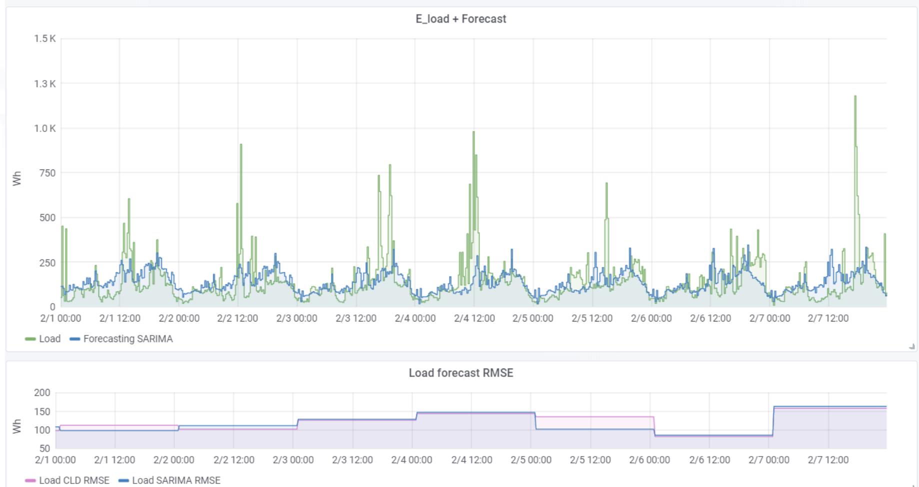
5. Coefficients $\phi_0, \phi_1, \Phi_1, \theta_1, \Theta_1$ should be calculated in this way that the sum of the squares of the residuals (SSE) will be minimized
6. The estimated coefficients can be used for forecasting:
   $y'' = \phi_0 + \phi_1 y''_{i-1} + \Phi_1 y''_{i-96} - \phi_1\Phi_1 y''_{i-96-1} + \theta_1\varepsilon_{i-1} + \Theta_1\varepsilon_{i-96} + \theta_1\Theta_1\varepsilon_{i-96-1}$
7. The predicted values can be calculated:
   $\hat{y}_i = y''_i + y_{i-1} + y_{i-96} - y_{i-96-1}$

[*https://towardsdatascience.com/time-series-in-python-part-2-dealing-with-seasonal-data-397a65b74051*]

Модель 4. Сезонная интегрированная модель авторегрессии – скользящего среднего (SARIMA)







Модель 5. Комбинация авторегрессии и устойчивой модели

This model describes dependentses of the variable with one or more lagged values of itself. AR(p) model can be define by formula:

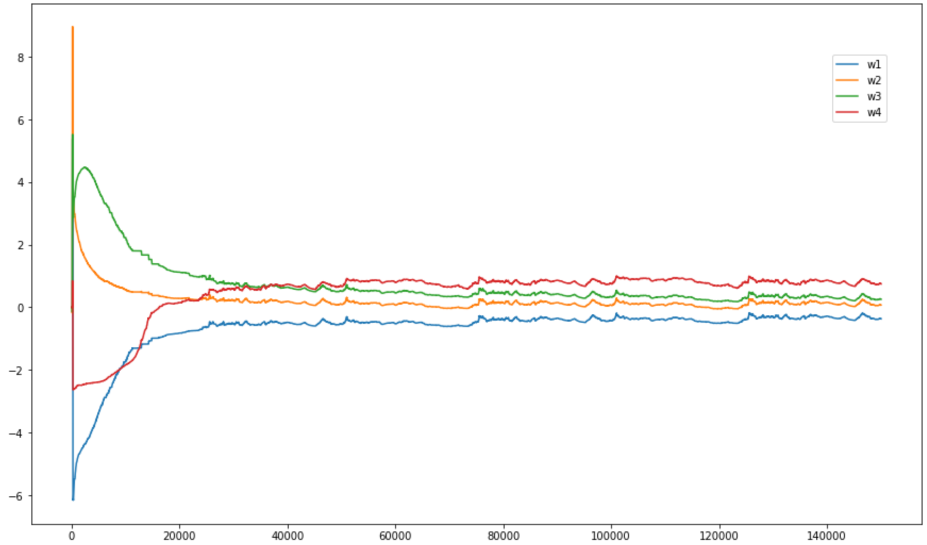
$$\hat{y}_t = w_1 y_{t-1} + \ldots + w_p y_{t-p} + w_{p+1} \frac{y_{t-7N} + \ldots + y_{t-m\cdot 7N}}{m} = \sum_{i=1}^{P} w_i y_{t-i} + w_{P+1} \frac{1}{m} \sum_{i=1}^{m} y_{t-i\cdot 7N}$$

A combination of 4 variables is used: 3 of them are the previous data values of the Load time series, and 4-th is the average of the three previous the same week position day values:

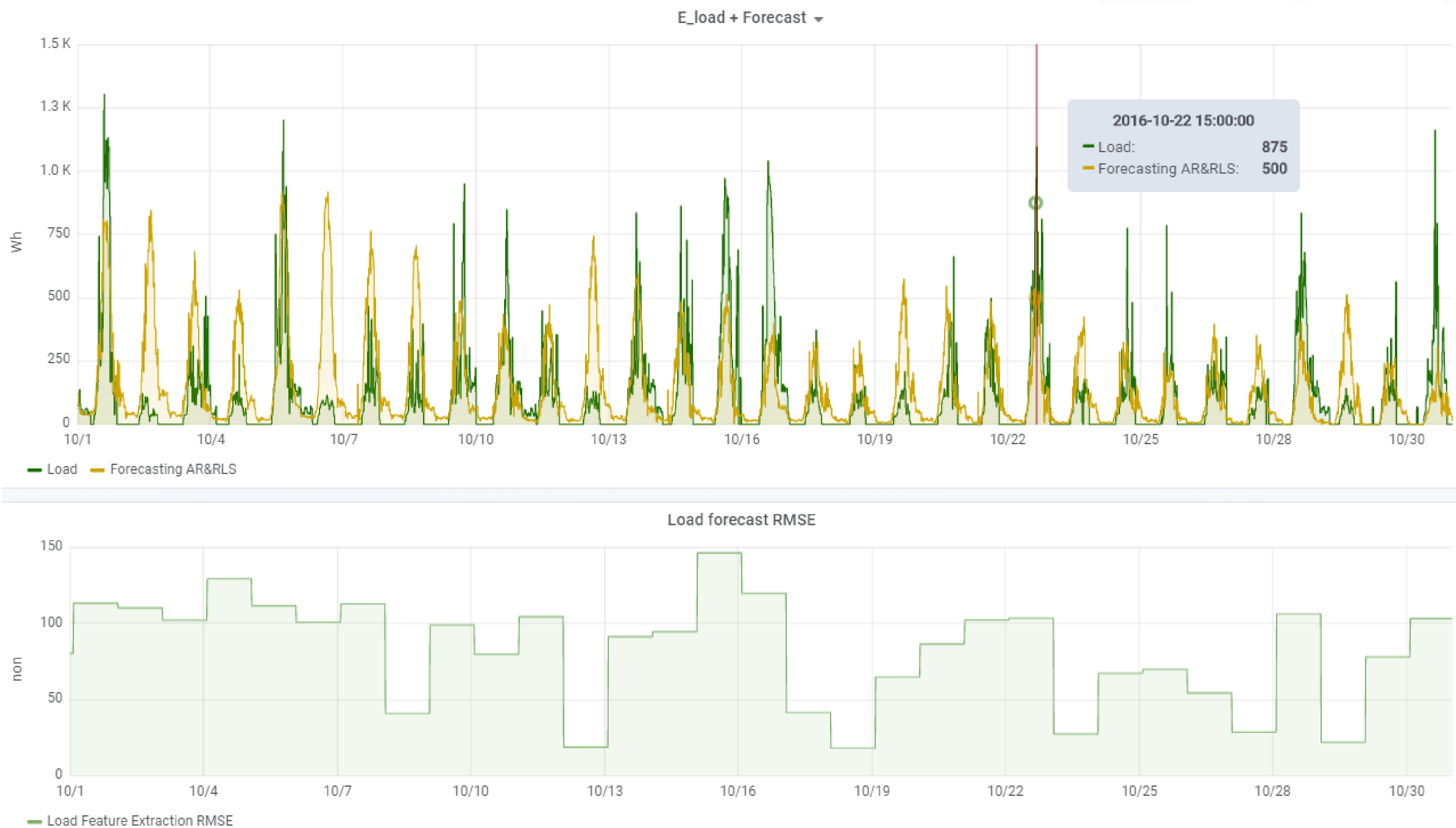
$$\hat{y}_t = w_1 y_{t-1} + w_2 y_{t-2} + w_3 y_{t-3} + w_4 \frac{y_{t-7N} + y_{t-14N} + y_{t-21N}}{3} = \sum_{i=1}^{3} w_i y_{t-i} + w_4 \frac{1}{3} \sum_{i=1}^{3} y_{t-i\cdot 7N}$$

This model uses the Recursive Least Squares (RLS) method for the optimization the weights:

Ali H Sayed and Thomas Kailath. Recursive least-squares adaptive filters. The Digital Signal Processing Handbook, pages 21–1, 1998.



Модель 5. Комбинация авторегрессии и устойчивой модели



Модель 6. Линейная регрессия на извлеченных признаках

1: *rs*(4) – rolling sum in a hour window (four 15-min inretval) for previouse day devided by 4

```
LOAD_data[1,j,i] = copy_from_data.rolling(4).sum()[EL][j]/4
```

2: *d* – bussines day or weekend, $d \in [0, 1]$

```
LOAD_data[2,j,i] = copy_from_data.index.weekday.isin([5,6])[j]*1
```

3: *Lh* – total Load in one hour

```
if (j%4==0):
      temp_sum = (LOAD_data[0,j,i] + LOAD_data[0,j+1,i] + LOAD_data[0,j+2,i]
      + LOAD_data[0,j+3,i])/4
      LOAD_data[3,j,i] = temp_sum
```

4: *Ld* – part of the Load in Mean LOAD for this day

```
for j in range(N):
            sum_LOAD = sum_LOAD + LOAD_data[0,j,i]
sum_LOAD=sum_LOAD/N
LOAD_data[4,j,i] = LOAD_data[0,j,i]/sum_LOAD
```

5: *DLh* – difference in hour LOAD that detects changes in LOAD

```
LOAD_data[5,j,i] = LOAD_data[3,j,i] - LOAD_data[3,j-4,i]
```

6: *LC* – low consumption detection, if Load less than 20% of Mean LOAD

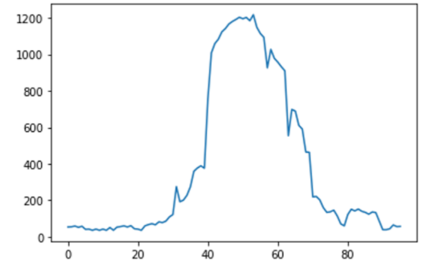
```
if (LOAD_data[0,j,i]>(sum_LOAD*0.2)):
      LOAD_data[6,j,i] = 1
else:
      LOAD_data[6,j,i] = 0
```

7: *PC* – peak consumption detection, if Load more than 150% of Mean LOAD

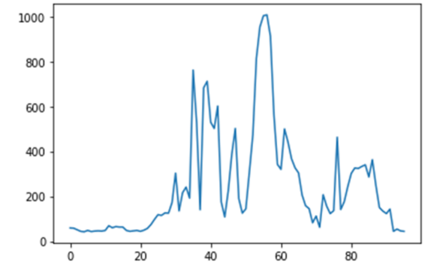
```
if (LOAD_data[0,j,i]>(sum_LOAD*1.5)):
      LOAD_data[7,j,i] = 1
else:
      LOAD_data[7,j,i] = 0
```

This features and LOAD are used for prediction with the differerent weights:

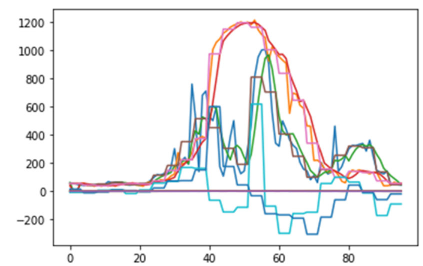
Day	*LOAD*	*rs*(4)	*d*	*Lh*	*Ld*	*DLh*	*LC*	*PC*
Target X	-	-	w_4	-	-	-	-	-
X-1	w_0	w_2	-	w_5	w_7	w_9	w_{11}	w_{13}
X-7	w_1	w_3	-	w_6	w_8	w_{10}	w_{12}	w_{14}



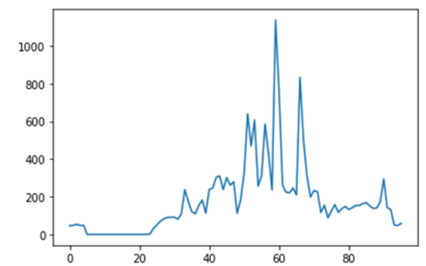
One week before (*d-7*)



One day before (*d-1*)



Features for the load forecasting for a day - *d*

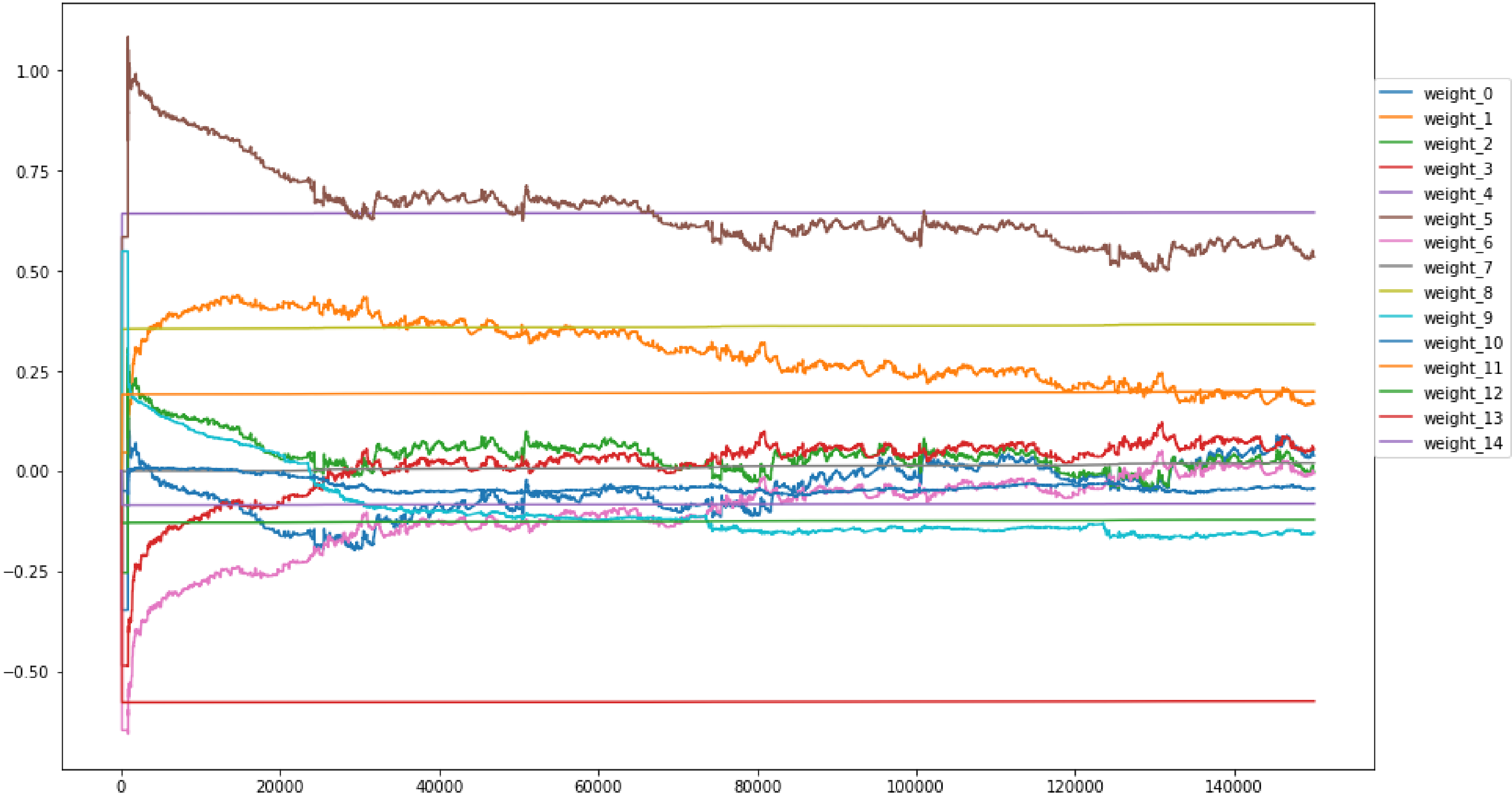


Measured load in a day - *d*

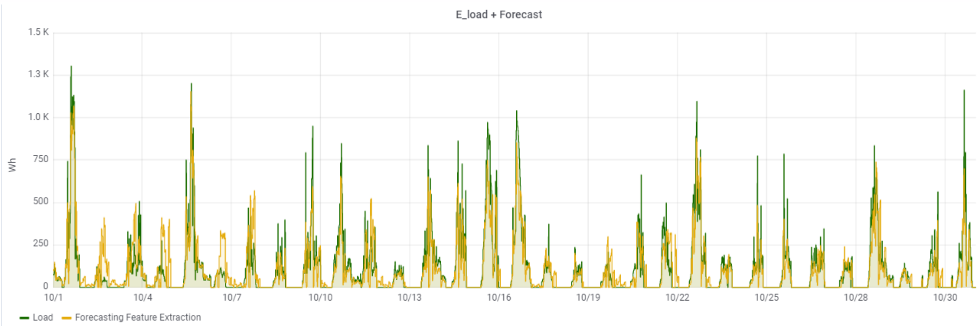
Weights were adopted by RLS method.

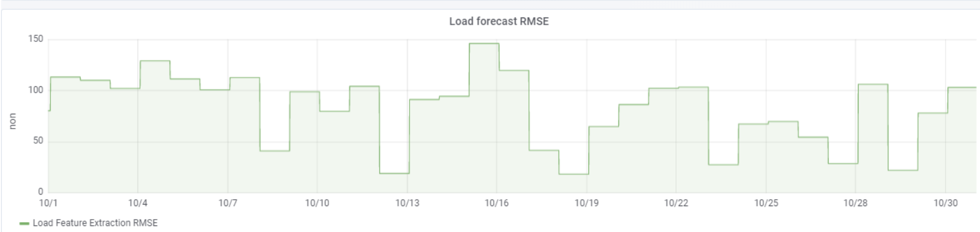
w_0	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	w_{10}	w_{11}	w_{12}	w_{13}	w_{14}
0.037780	0.167459	-0.001879	0.053854	0.644751	0.53366	-0.006949	0.019924	0.366183	-0.154432	-0.043700	0.197787	-0.121735	-0.574366	-0.081549

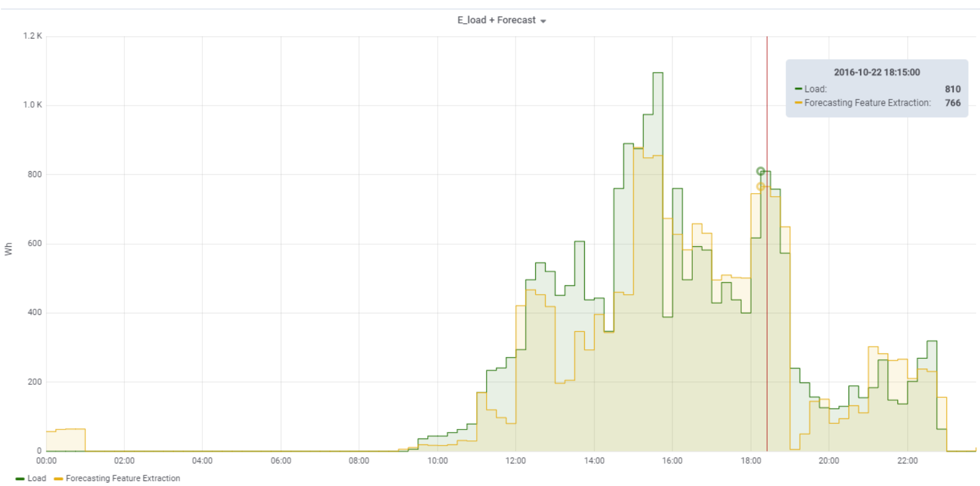
Модель 6. Линейная регрессия на извлеченных признаках

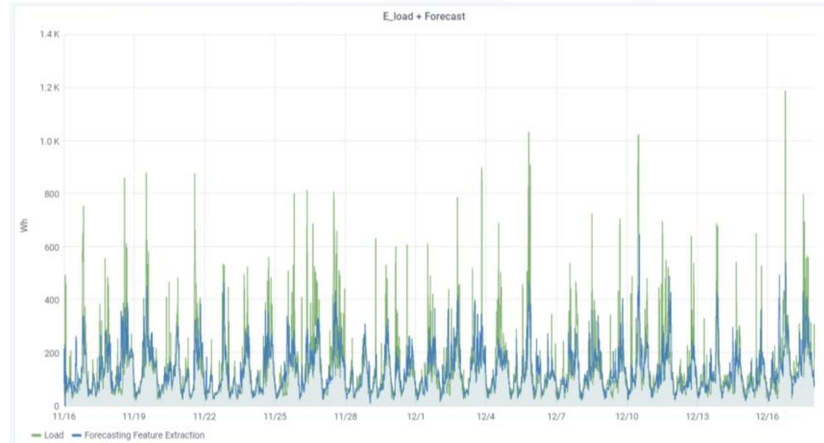


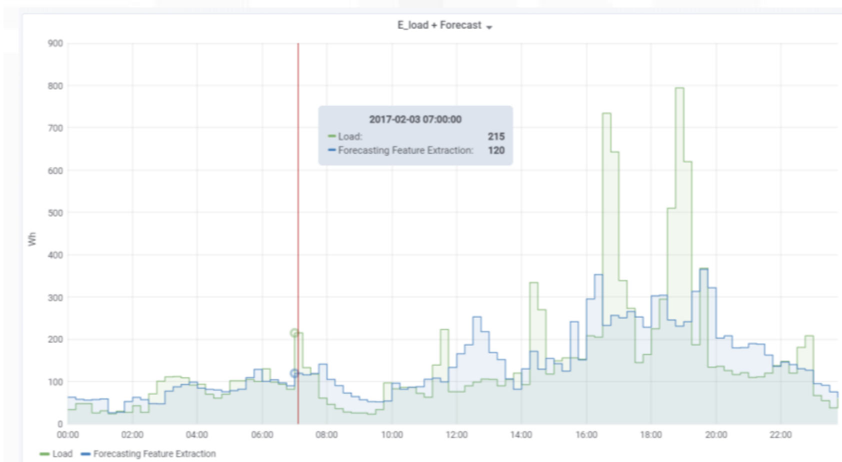
Модель 6. Линейная регрессия на извлеченных признаках



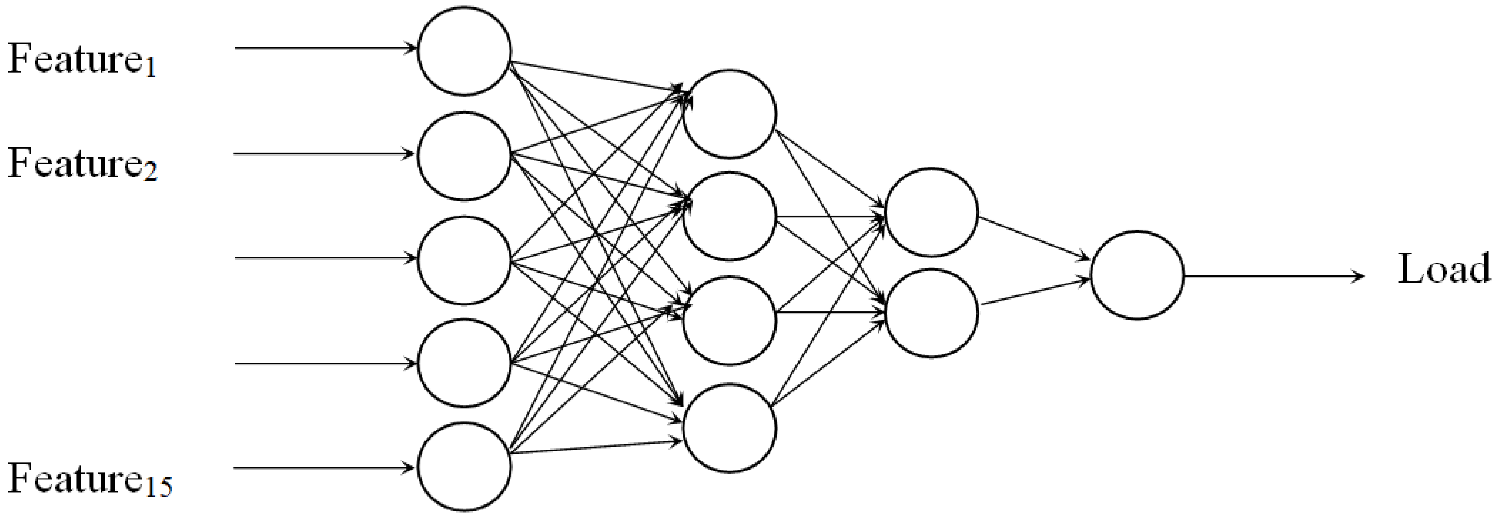




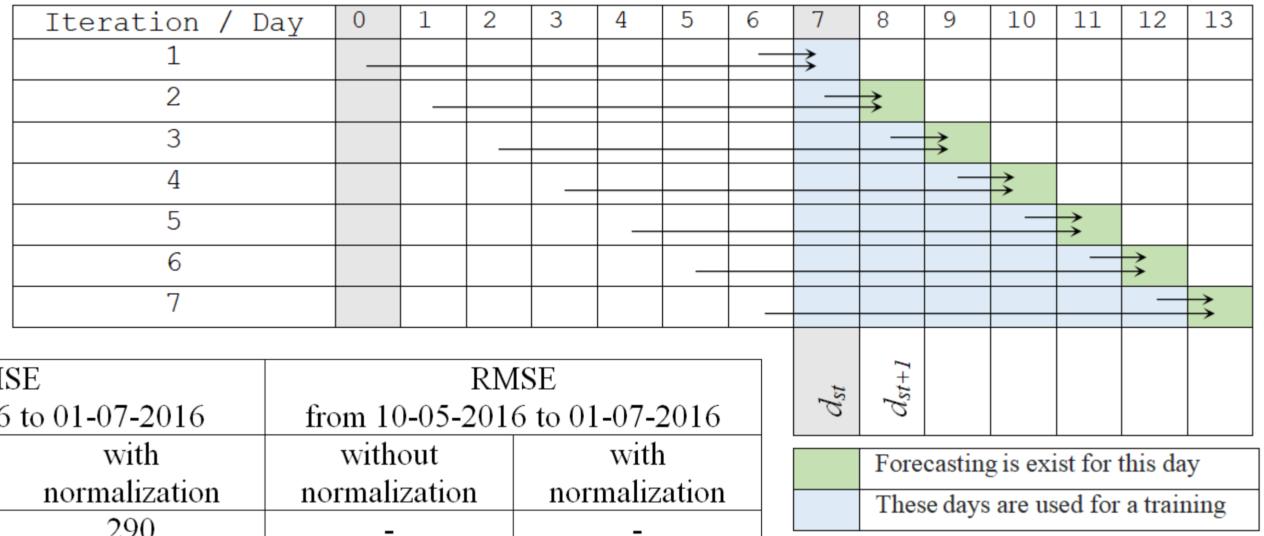




Модель 7. Нелинейная регрессия на извлеченных признаках

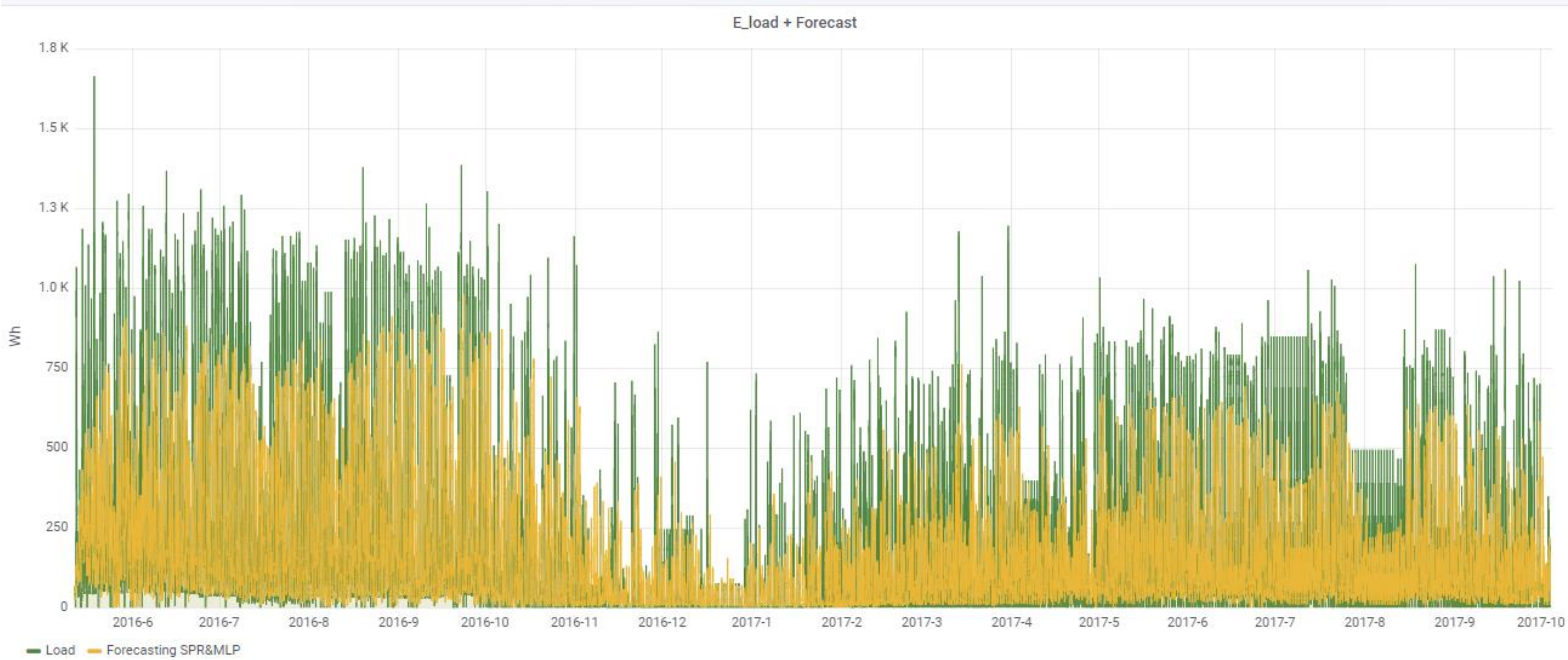


Модель 7. Нелинейная регрессия на извлеченных признаках



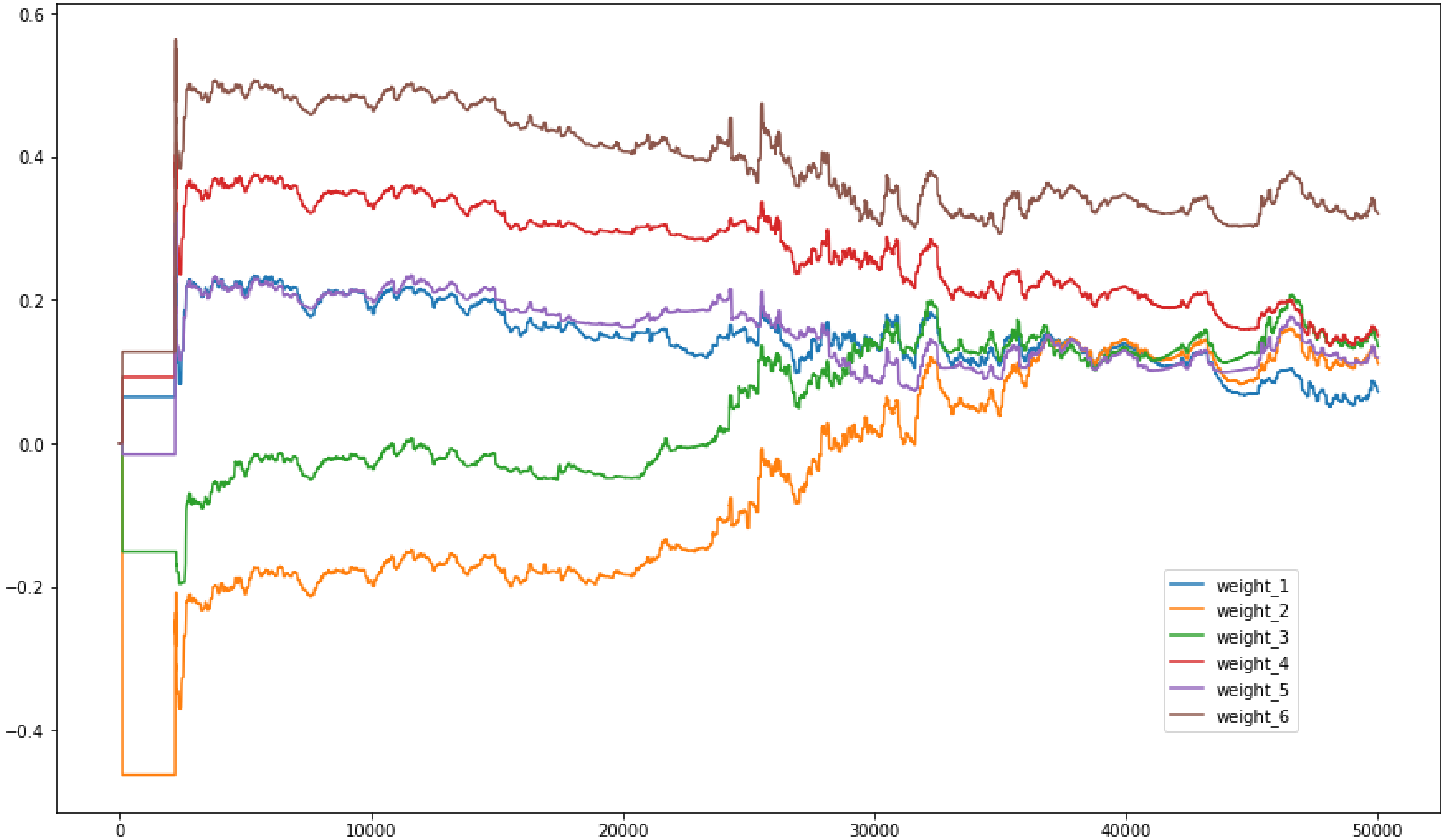
Name	RMSE from 10-05-2016 to 01-07-2016		RMSE from 10-05-2016 to 01-07-2016	
	without normalization	with normalization	without normalization	with normalization
10	222-230	290	-	-
20	219-220	289	-	-
40	220-222	259	-	-
60	218-220	232	150	-
80	221-225	220	-	-
100	220-225	225	-	-
120	225-230	219	-	146
140	225-230	221	-	-
160	230-240	220	-	-
200	230-240	221	-	-
300	>300	221	-	-
400	>300	227	-	-

Модель 7. Нелинейная регрессия на извлеченных признаках

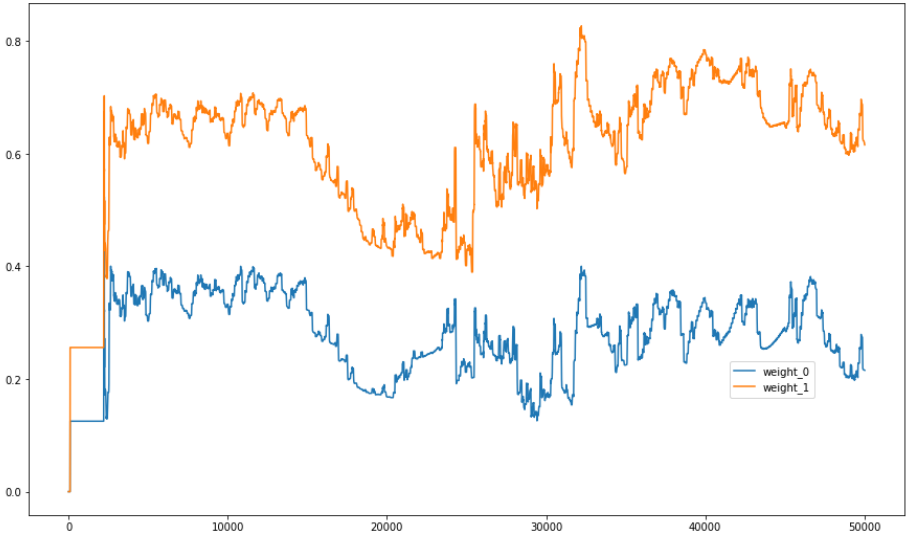


Модель 8. Комбинированная модель

$$\hat{y}_t = w_1\hat{y}_t^1 + w_2\hat{y}_t^2 + w_3\hat{y}_t^3 + w_4\hat{y}_t^4 + w_5\hat{y}_t^5 + w_6\hat{y}_t^6$$



Модель 8. Комбинированная модель



$\hat{y}_t^0$ SPR model based on feature extraction with RLS

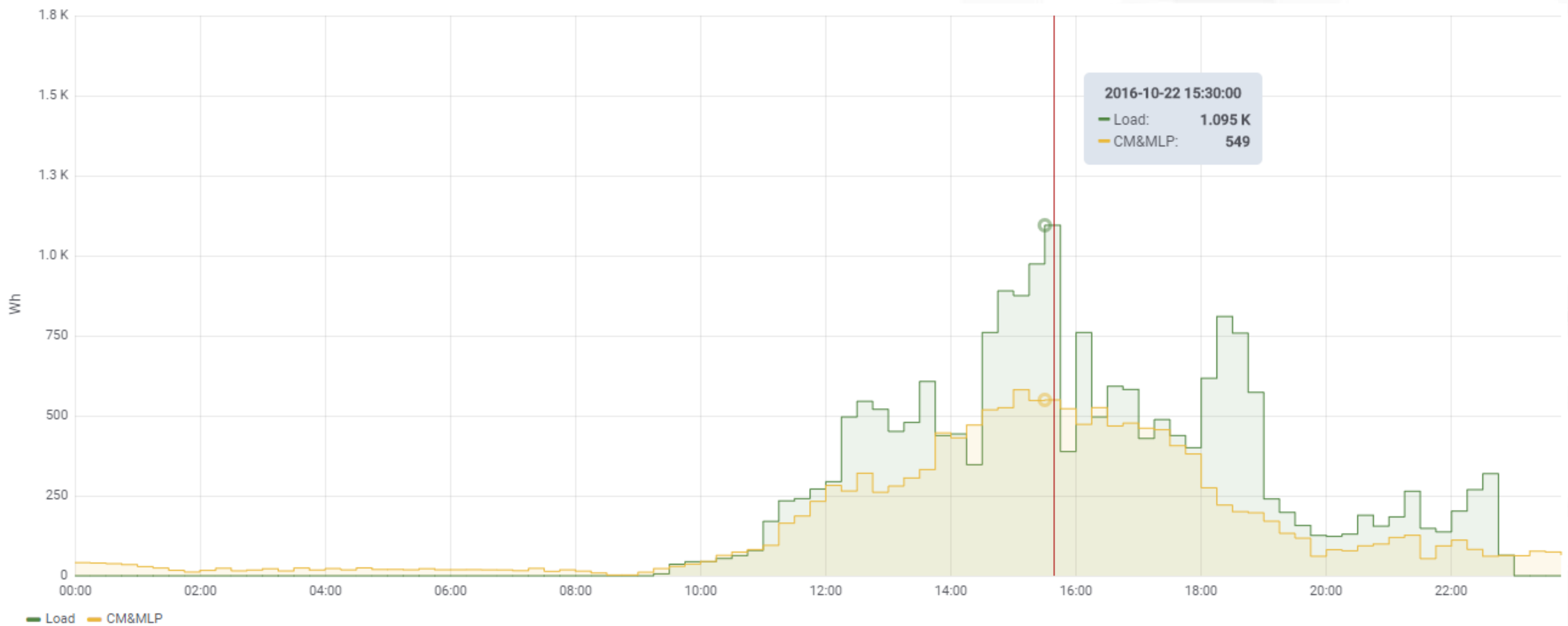
$\hat{y}_t^1$ SPR model based on feature extraction with MLP$_{NN = [15,4,1]}$ as a regressor

$\hat{y}_t^0$ Copy last N days (N = 10)

$\hat{y}_t^1$ SPR model based on feature extraction with RLS

Модель 9. Комбинированная модель с нелинейной регрессией

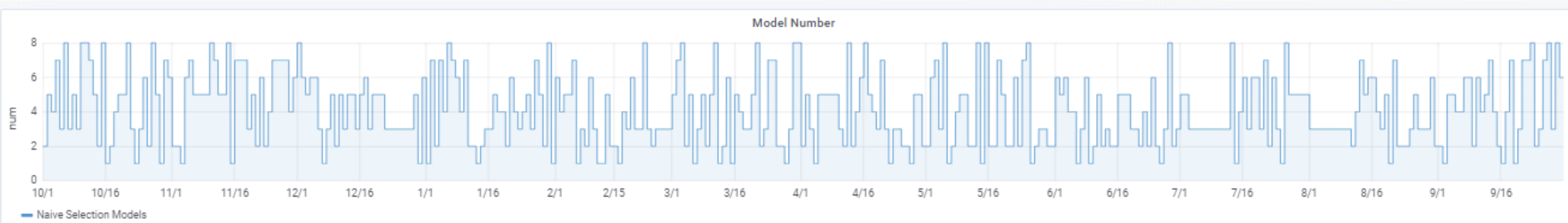
The next data are used as the training set:
- load as source vectors [*N*]
- input matrix [$d_{st}*N-1:(d+1)*N-1$, *7*] – predictions from other models (CLD, Nday, HW, SARIMA, PAR, SPR, SPR&MLP) for the load forecasting in a day – *d*
- targets for input matrix [$d_{st}*N-1:(d+1)*N-1$] - Measured load in a day - *d*



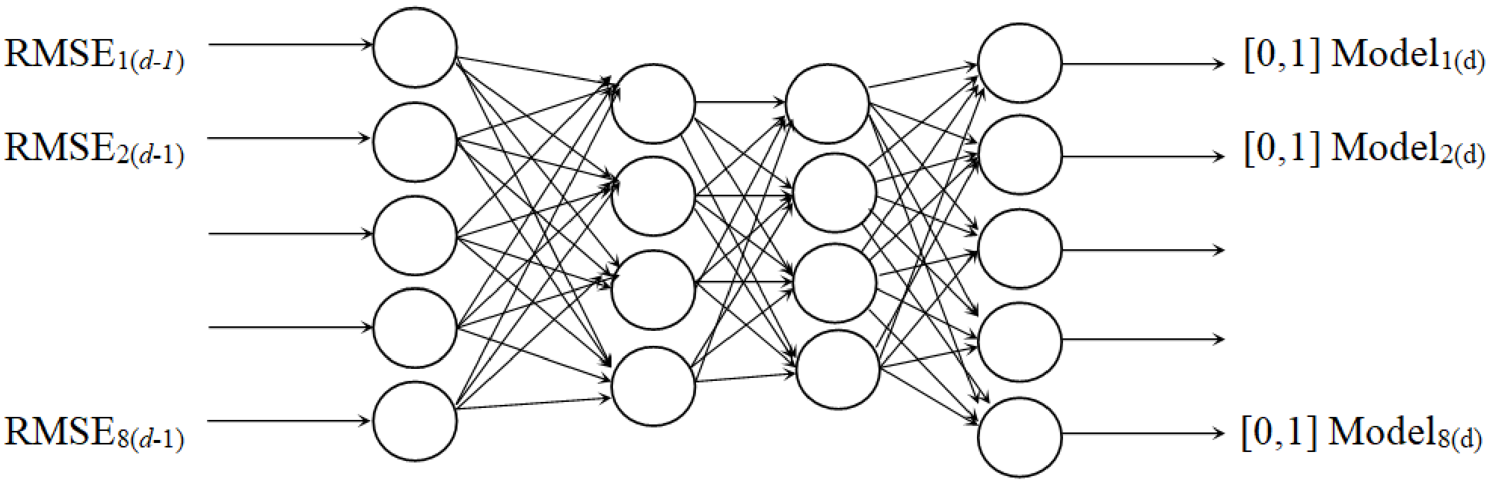
Модель 10. Наивный выбор модели

Для прогноза на следующий день выбирается всегда модель с наименьшим RSME в предыдущий день

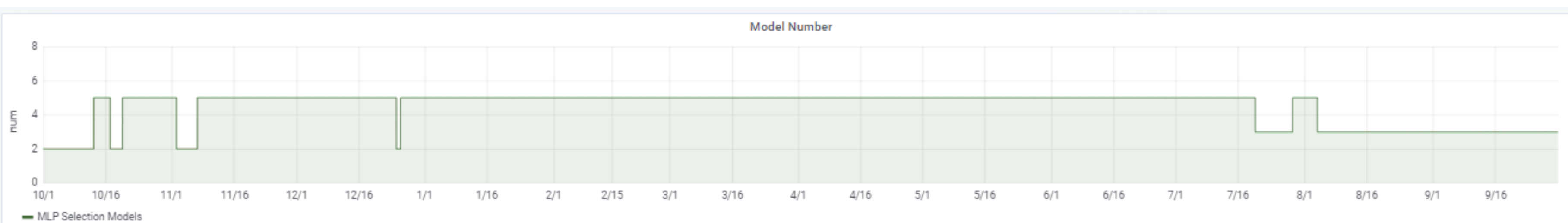
index	Model name
1	Coping previous 3 days
2	Copy last N days (N = 10)
3	Holt Winters (slen = 96)
	(calculations are not finished) SARIMA $(1,1,1)(1,1,1)_{96}$
4	PAR - persistent and AR model with RLS
5	SPR model based on feature extraction with RLS
6	SPR model based on feature extraction with $\mathrm{MLP}_{NN = [15,4,1]}$ as a regressor
7	Combine model (CM) with linear regression
8	Combine model (CM) with nonlinear regression by MLP



Модель 11. Выбор модели с использованием нейронной сети



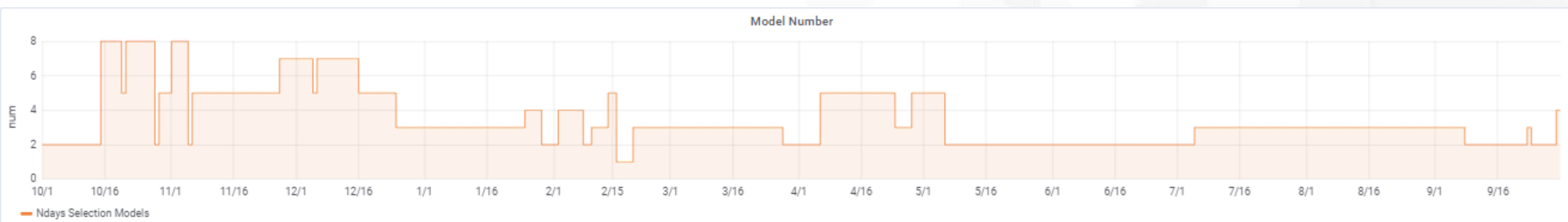
The day d has only one model, that gives the less RMSE. For example, model 4 can be written by vector [0 0 0 1 0 0 0 0].



Модель 12. Выбор модели с наименьшей ошибкой на периоде в N дней

	0	1	2	3	4	5	6	7
56	152.873	210.063	174.485	170.39	199.428	243.386	202.275	212.613
57	191.878	271.721	209.786	198.568	173.302	179.529	184.711	226.105
58	124.537	170.062	134.991	137.473	277.026	221.587	179.507	200.853
59	161.364	143.574	154.179	196.346	124.643	197.862	150.997	157.928
60	155.23	213.42	163.363	156.873	225.632	181.686	199.641	208.204
61	194.463	198.578	171.068	212.478	241.447	228.584	186.266	226.184
62	188.487	186.09	163.407	160.878	181.467	184.598	194.26	180.119
63	160.004	160.511	167.007	133.809	163.32	156.316	148.847	144.117
64	298.857	119.341	142.733	199.039	256.856	259.506	225.982	129.051
65	136.351	115.12	135.369	135.837	134.38	162.112	133.548	143.418
66	240.41	252.948	216.05	274.311	257.465	229.67	306.296	228.962
67	207.757	213.545	191.815	209.359	284.712	267.742	228.508	222.614
68	142.144	124.183	112.688	137.064	150.247	157.241	139.55	121.662

	0	1	2	3	4	5	6	7
155	0	0	0	0	0	0	1	0
156	0	0	1	0	0	0	0	0
157	0	0	0	0	0	0	0	1
158	0	0	1	0	0	0	0	0
159	0	0	0	0	1	0	0	0
160	0	0	1	0	0	0	0	0
161	0	0	0	0	0	0	0	1
162	0	0	0	0	0	0	0	1
163	0	0	0	0	0	0	1	0
164	0	0	0	0	1	0	0	0
165	0	1	0	0	0	0	0	0
166	0	0	0	0	0	0	0	1
167	1	0	0	0	0	0	0	0

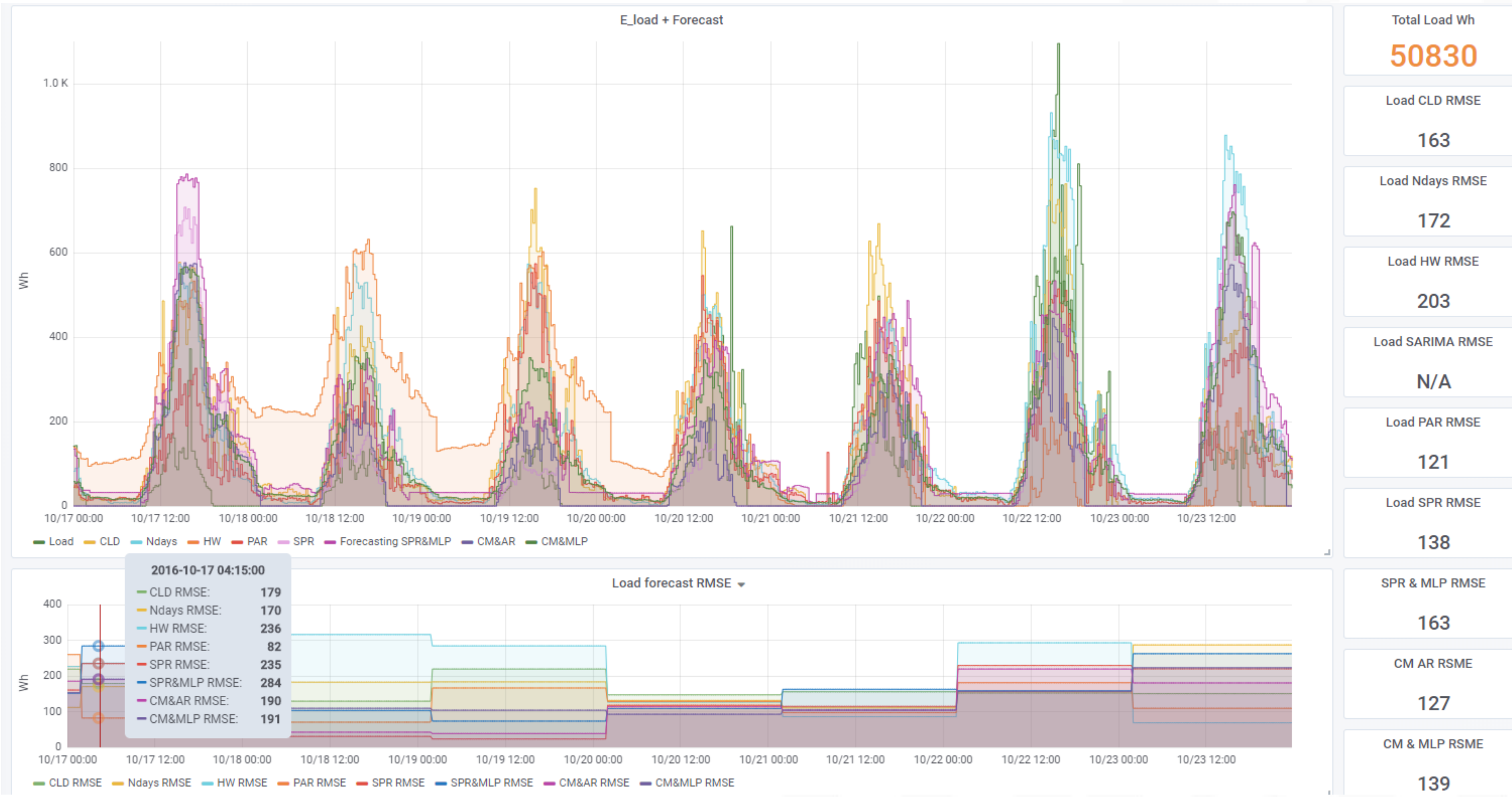


Анализ выбора моделей

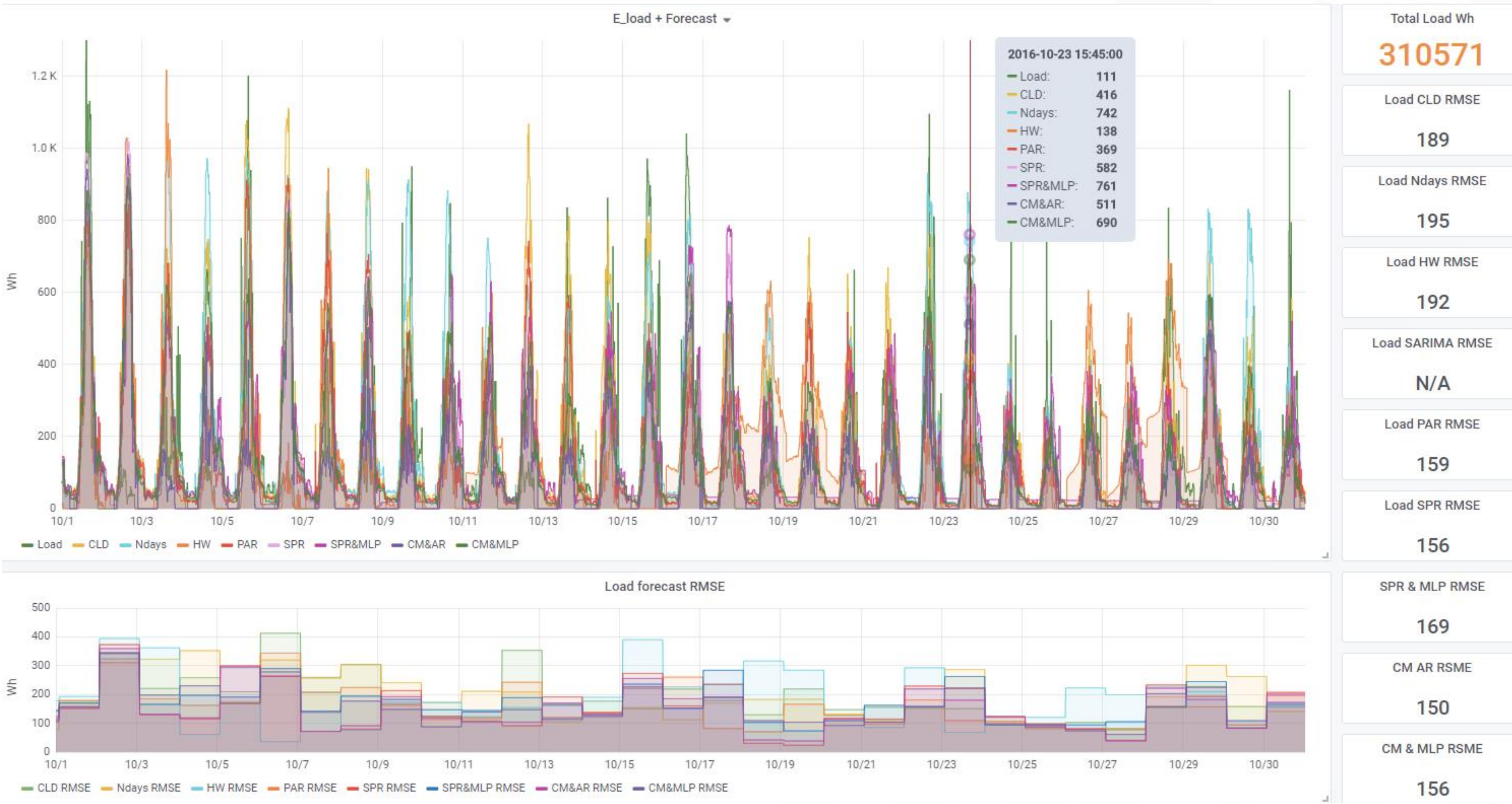
Оценка моделей на реальных данных

Duration	1 day	1 week	1 month	½ year	1 year
Dates	22-10-2016	17-10-2016 .. 23-10-2016	01-10-2016 .. 31-10-2016	16-10-2016 .. 15-04-2017	01-10-2016 .. 30-09-2017
CLD	153	163	189	121	149
N-day	153	188	205	118	139
HW	276	203	284	121	142
SARIMA	211	141	163	126	135
PAR	174	121	159	111	137
SPR	220	138	154	105	127
SPR&MLP	165	160	165	117	131
CM&LR	**210**	**127**	**150**	**104**	**124**
CM&MLP	152	138	155	106	127
NC	277	134	161	108	126

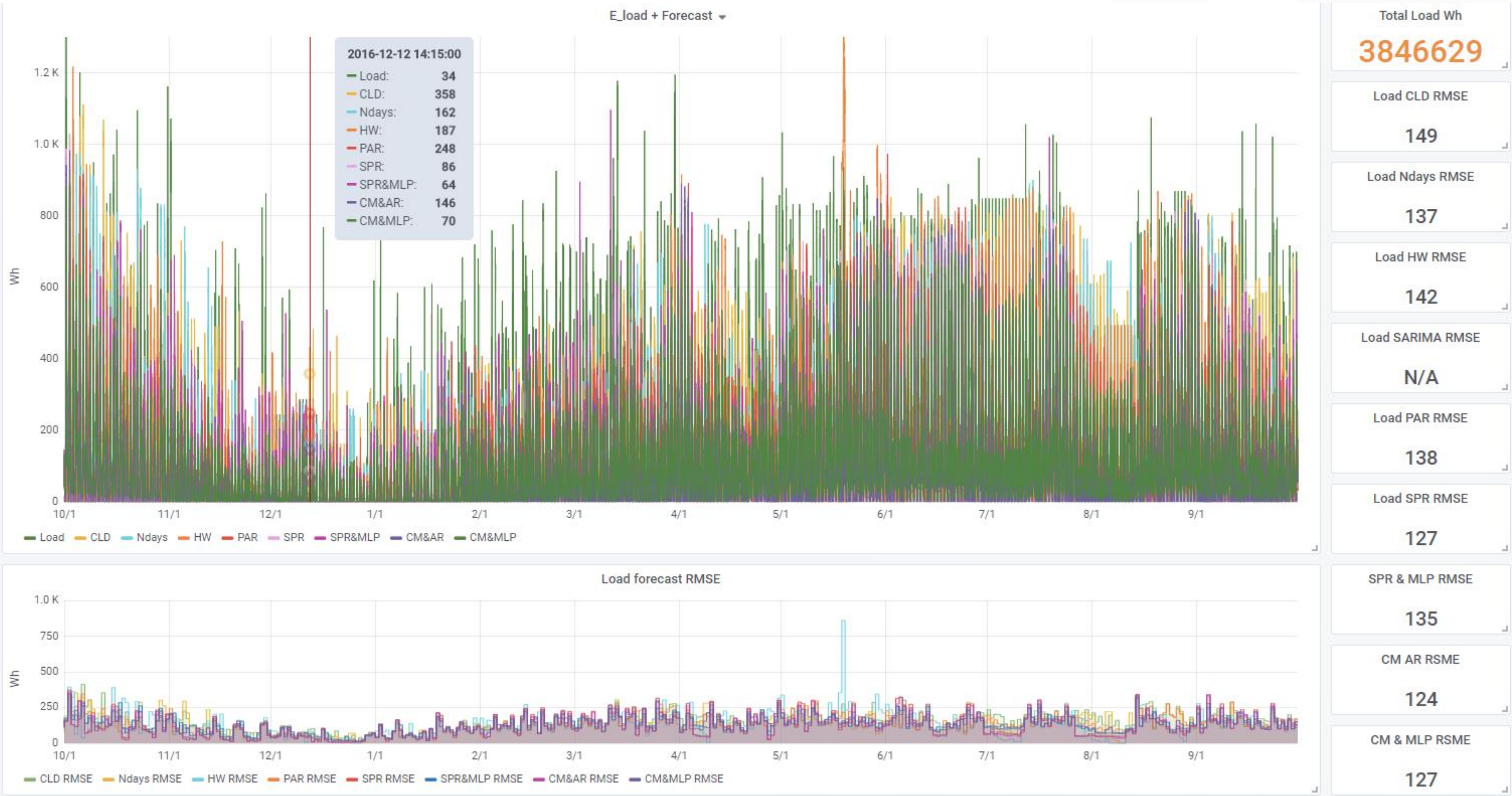
Оценка моделей на реальных данных



Оценка моделей на реальных данных



Оценка моделей на реальных данных



Результаты

Проведена установка и настройка среды Python, базы данных временных рядов InfluxDB и веб-сервера Grafana

Реализован анализ модели хранения данных в csv файлах более 100 различных объектов. Произведен импорт реальных энергетических данных объектов

Реализована модель энергобаланса и дискретный автомат для управления инвертором

Реализованы краткосрочные моделей прогнозирования электрической нагрузки зданий для DA (на сутки вперед) рынка управления спросом. Проведена оценка моделей

Подготовлена публикация для 21-го Всемирного конгресса по автоматическому управлению IFAC 2020 в Берлине, Германия

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ

Спасибо за внимание!