Time Series Forecasting Problem

About me

Aleksei Riabykh

Lead Data Scientist at VTB

PhD at MIPT

DCAM MIPT & ITTP RAS & Skoltech Alumnus

aryabyh@vtb.ru

Researcher
IITP RAS,
Feb 2016 – Dec 2017

Data Scientist
CDS Office at Sberbank,
Feb 2017 – Apr 2019

Senior Data Scientist
DataLab at Gazprombank,
Apr 2019 – Sep 2019

Lecturer
CS Department at MIPT,
Sep 2019 - Present

Lead Data Scientist
DADM at VTB,
Sep 2019 - Present

Terminology

- Model fitting, parameter estimation: model training
- In-sample: training set
- Out-of-sample: test set
- Forecast horizon: target variable
- Forecast origin: from where you do the forecasting from, the last known observation
- Rolling origin: the forecast origin changes (for every point in the test set)
- Fixed origin: the forecast origin is fixed

Terminology

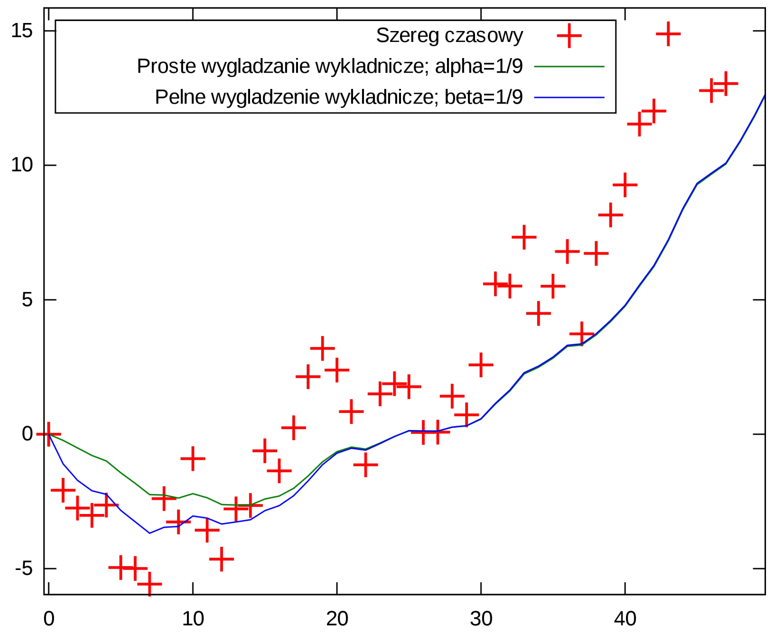
- Forecast combination: ensembling
- Lags, independent variables, regressors, covariates,
- Predictors: features, inputs
- Dummy variable, indicator variable: one-hot encoding
- Seasonality, seasonal period: cyclic change in mean of the series. Its length is know a priori and will not change in the future
- Trend: (smooth) change in the mean of the series

Trading off the native ant the mean forecast

- Naive and mean forecast are two very extreme cases
- Naive forecast: We only consider the last observation, none of the others
- Mean forecast: All observations are weighted equally
- A central assumption in forecasting is often that the more recent past is more important than the more distant past.

Idea: What about weighting the observations with an exponential decay?

–> Exponential smoothing



https://cbergmeir.com/talks/acml-tutorial/
https://ru.wikipedia.org/wiki/Экспоненциальное_сглаживание

Exponential smoothing

- Exponential smoothing applied to components: level, seasonality, and trend
- Holt (1957), and his student Winters (1960)
- Was used for a long time as a relatively ad-hoc method without a theoretical underpinning
- Hyndman et al. (2002) gave it a solid statistical foundation in state-space models
- ETS stands for both ExponenTial Smoothing and Error, Trend, and Seasonality

Linear modelling: ARMA

Autoregressive moving average model (ARMA):

$$\hat{x}_{t+1} = c + \phi_1 x_t + \cdots + \phi_p x_{t-p+1} + \theta_1 \epsilon_t + \cdots + \theta_q \epsilon_{t-q+1}$$

- Model is linear in the lags and linear in the errors
- When fitting the model, where do the errors come from?
- Need to estimate some initial conditions and step through
- the whole series (in ETS as well)
- No closed-form solution
- Need a non-linear fitting procedure
- Fitting can be slow in long time series

Linear modelling: (S)ARIMA

- Addresses non-stationarity
- Preprocessing step: Do differencing of the series before we do the ARMA
- Pro: Hopefully makes the series stationary
- Contra: We loose some information about the scale in the preprocessing
- Widely used and very competitive

- Any (stationary) AR(1) model has an equivalent MA(∞) model, and any (invertible) MA(1) model has an equivalent AR(∞) model (Hyndman and Athanasopoulos, 2018)
- Thus, we can approximate the MA part of the ARMA model with a higher order AR part

–> Can be seen as a re-parametrisation, to get fewer parameters and a smaller input window, at the cost of more complex model fitting

Problem: Non-stationarity

- Data Distribution changes over time
- Many (most?) real-world problems have a time component and changing distributions
- Think about detecting cars on the street with a dataset from the 1970s
- In time series it is more explicit though and has more impact

Typical non-stationarities in time series:

- change in mean: **seasonality, trend**
- change in variance: **heteroskedasticity**

How to achieve stationarity?

- Differencing only solves some forms of non-stationarity, not others
- Loosing information about the scale.
  Can have an additional input as the (log of) the original scale.
- Differencing can help to make ML models more robust

Trend

Detrending

- Problem: it is not well specified what a trend is
- Essentially just a smoothed version of the series
- We still need to forecast the trend then

Logarithm or Box-Cox transform

- Makes exponential trends linear
- Also stabilizes the variance
- Box-Cox Transformation

$$w_t = \begin{cases} log(y_t) & \text{if } \lambda = 0, \\ (y_t^\lambda - 1)/\lambda & \text{if } \lambda \neq 0 \end{cases}$$

- Choice of optimal value for λ is difficult

Seasonality

Seasonal indicator variables:

- Categorical variable: Monday, Tuesday, Wednesday, ...
- One-hot encoded version of this variable: “Seasonal dummy”

Fourier terms:

$$\boxed{sin\left(\frac{2\pi kt}{s}\right), cos\left(\frac{2\pi kt}{s}\right)}$$

t is the time point
s is the seasonal periodicity of the time series and
k is the number of sine cosine pairs used with the transformation

The number of Fourier terms controls the smoothness of the seasonal pattern.

ML Approach

- Seasonality and trend handling as discussed earlier
- Build one model per horizon
- Especially suitable with (diverse) external variables
- Engineer features such as rolling means, rolling sds
- Use differences as additional inputs (of different lags, e.g., lag1 difference, lag 12 difference)

Holiday effects

- one-hot encoded
- as distance maps (days before/after holiday)
- Monthly series: number of trading days in the month

Ensembling works in forecasting just as well as in any other area

- Ensembles of GBTs, NNs, pooled regression

Global models

Name “global models” was introduced by Januschowski et al. (2020)

- Traditionally, one time series is seen as a dataset
- One model is built per time series
- Low sampling frequencies and non-stationarities like structural breaks make usually that we don’t have enough data to fit complex (ML) models

Paradigm shift:

- a set of time series is a dataset (e.g., a set of series from retail, smart meters, etc.)
- build a model across the series

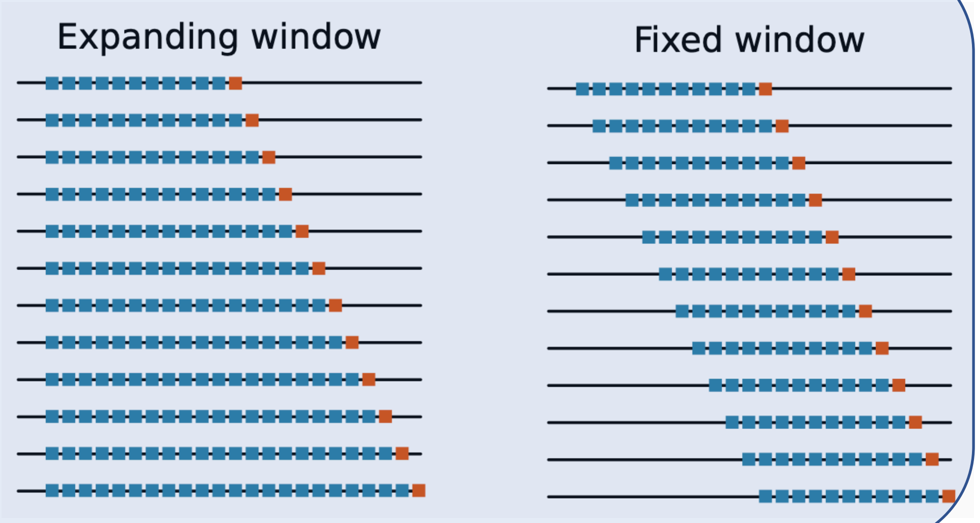
–> Now, enough data, due to more series
–> ML methods are competitive now

Evaluation

evaluation in general regression

target value	quadratic loss	RMSE	cross-validation
fit models / compute results	calculate error (with loss function)	compute error measure (average)	use model selection procedure
Which values to predict? Using which values? (one-step-ahead, one-to-24-step-ahead, ...)	absolute and quadratic losses of errors, percentage errors, relative errors, scaled errors, ...	RMSE, MAE, MAPE, MDAPE, sMAPE, MDRAE, GMRAE, MASE, RELMAE, ...	last block validation, (cross-validation without dependencies, ...)

traditional time series forecast evaluation



$$\mathrm{SE}_t = (y_t - \hat{y}_t)^2$$

$$\mathrm{AE}_t = |y_t - \hat{y}_t|$$

$$\mathrm{RMSE} = \sqrt{\frac{1}{n}\sum_{t=1}^{n}(y_t - \hat{y}_t)^2}$$

$$\mathrm{MAE} = \frac{1}{n}\sum_{t=1}^{n}|y_t - \hat{y}_t|$$

$$\mathrm{sMAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|100\,\frac{y_t - \hat{y}_t}{\frac{|y_t|+|\hat{y}_t|}{2}}\right|$$

$$\mathrm{MASE} = \frac{\sum_{t=1}^{n}|\hat{y}_t - y_t|}{\frac{n}{m-s}\sum_{k=s+1}^{m}|y_k - y_{k-s}|}$$

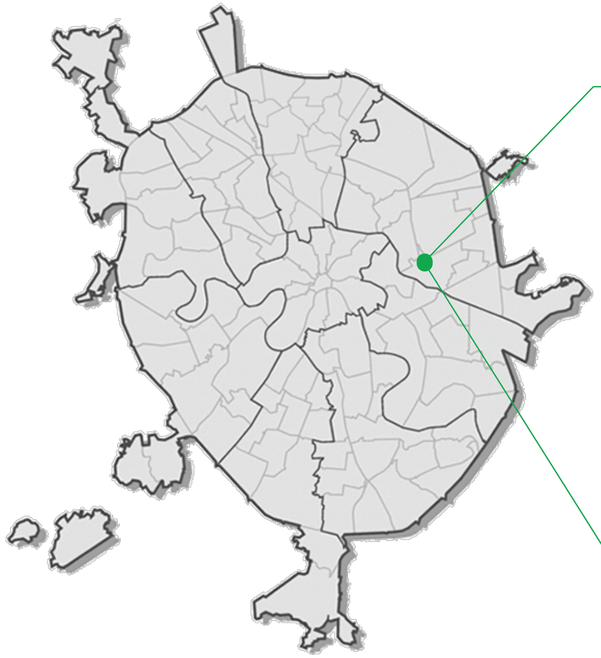
$$\mathrm{PE}_t = 100\,\frac{y_t - \hat{y}_t}{y_t},$$

$$\mathrm{MAPE} = \frac{1}{n}\sum_{t=1}^{n}\left|100\,\frac{y_t - \hat{y}_t}{y_t}\right|$$

Time Series Forecasting in Cash Management

Stating the Problem

Administrative division of Moscow

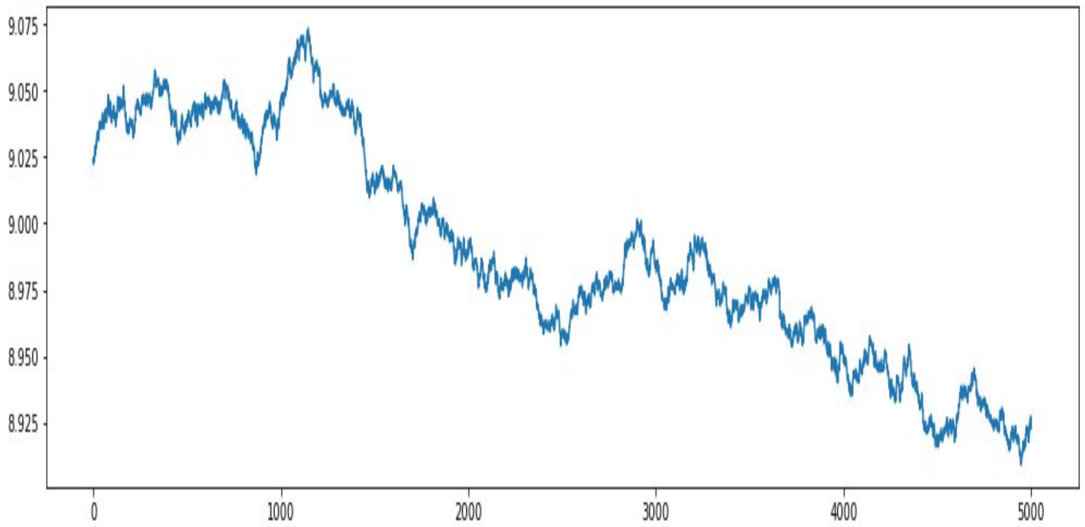


Cash Optimization Meets Consumer Demand

Optimal schedule

id	27 авг	28 авг	29 авг	30 авг	31 авг	1 сен	2 сен	3 сен	4 сен	5 сен	6 сен	7 сен	8 сен	9 сен	10 сен	11 сен	12 сен
10355	1	0	0	0	0	1	0	0	0	1	0	0	0	1	0	1	0
12456	0	1	0	0	0	1	0	0	0	0	0	0	1	0	0	0	1
10432	0	0	1	0	0	0	1	0	0	1	0	0	0	1	0	0	0
667456	0	1	0	0	0	1	0	0	1	0	0	1	0	0	1	0	0
34529	0	0	1	0	0	0	0	1	0	0	1	0	0	1	0	0	1
128437	1	0	0	0	1	0	0	1	0	0	1	0	1	0	0	1	0
34098	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53921	1	0	0	0	1	0	1	0	1	0	0	0	1	0	1	1	0

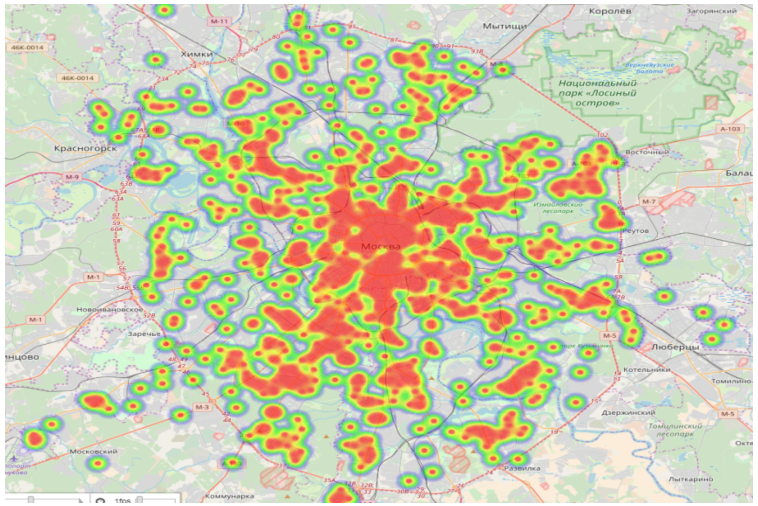
Cost of schedule



Stating the Problem

Different types of ATMs

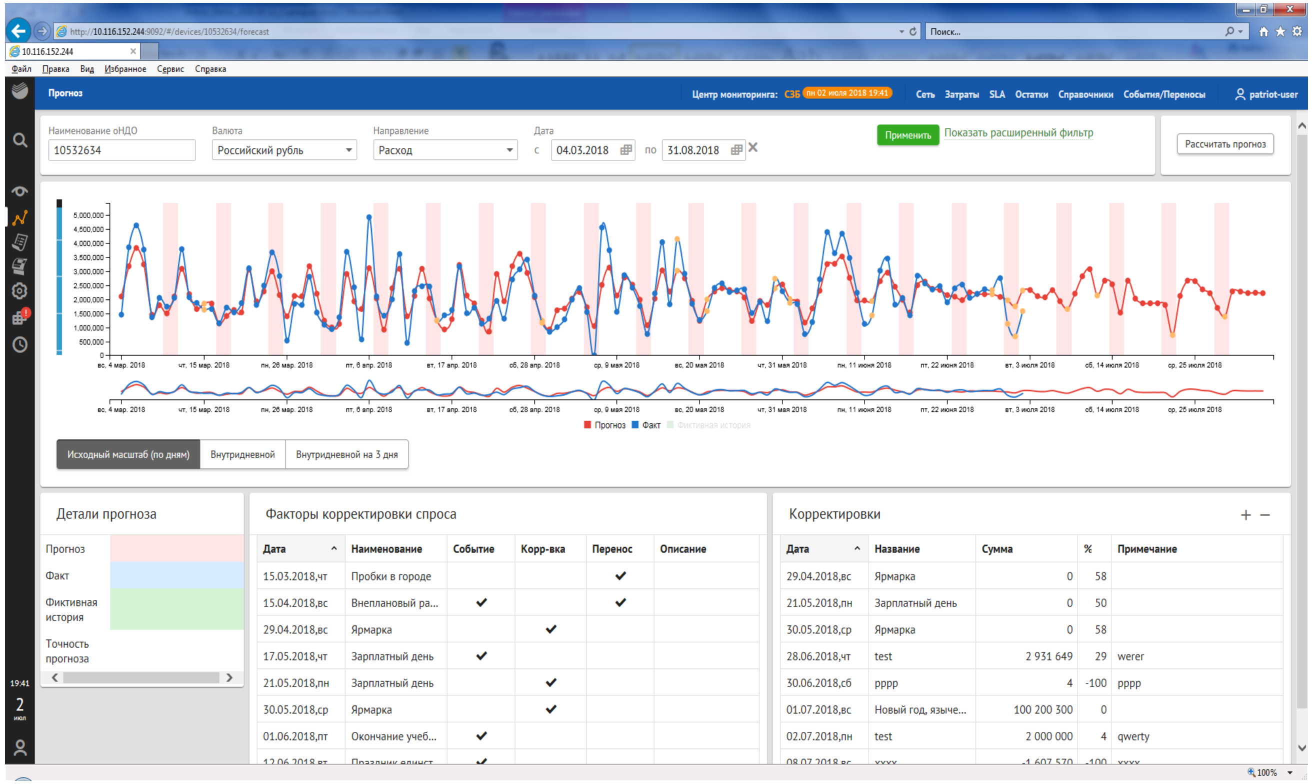
Heat map of client's demand in Moscow



- 8,700 (3900 ATMs + 3300 IPTs + 1500 DMs) time series of customer withdrawals / deposits in ATMs only in Moscow, 80 000 in Russia

- A large number of business process constraints

Stating the Problem



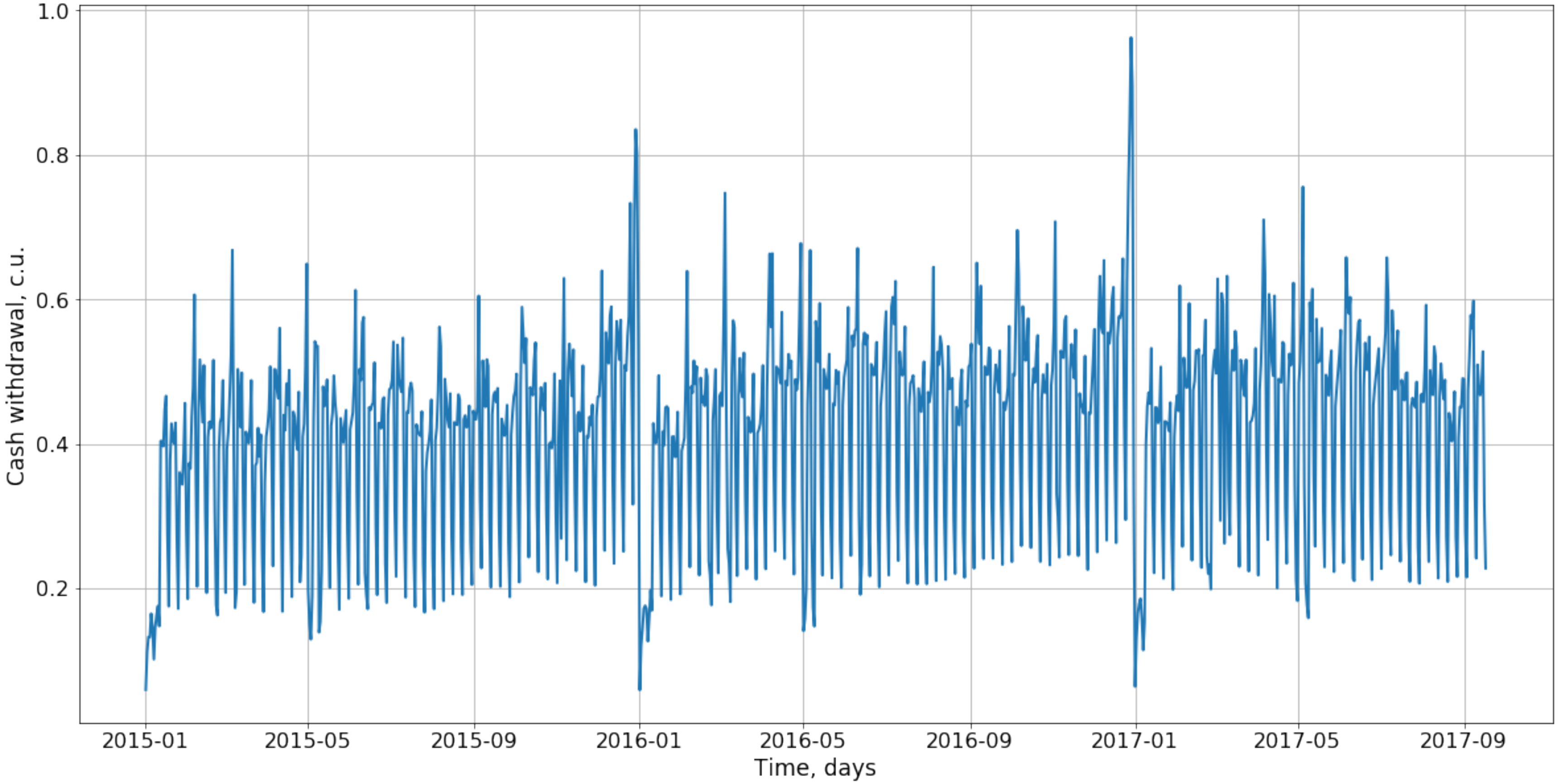
- Problems of ATM's downtimes due to internal and external factors, outliers of a random and regular nature, a sharp change in the distribution of demand, lack of history in the case of ATM relocation and others are solved by the analyst

- In market solutions we have high cost, low quality of demand forecasting and schedule optimization, closed source code and <u>very high analyst influence</u>

NCR OptiCash / OptiNet System

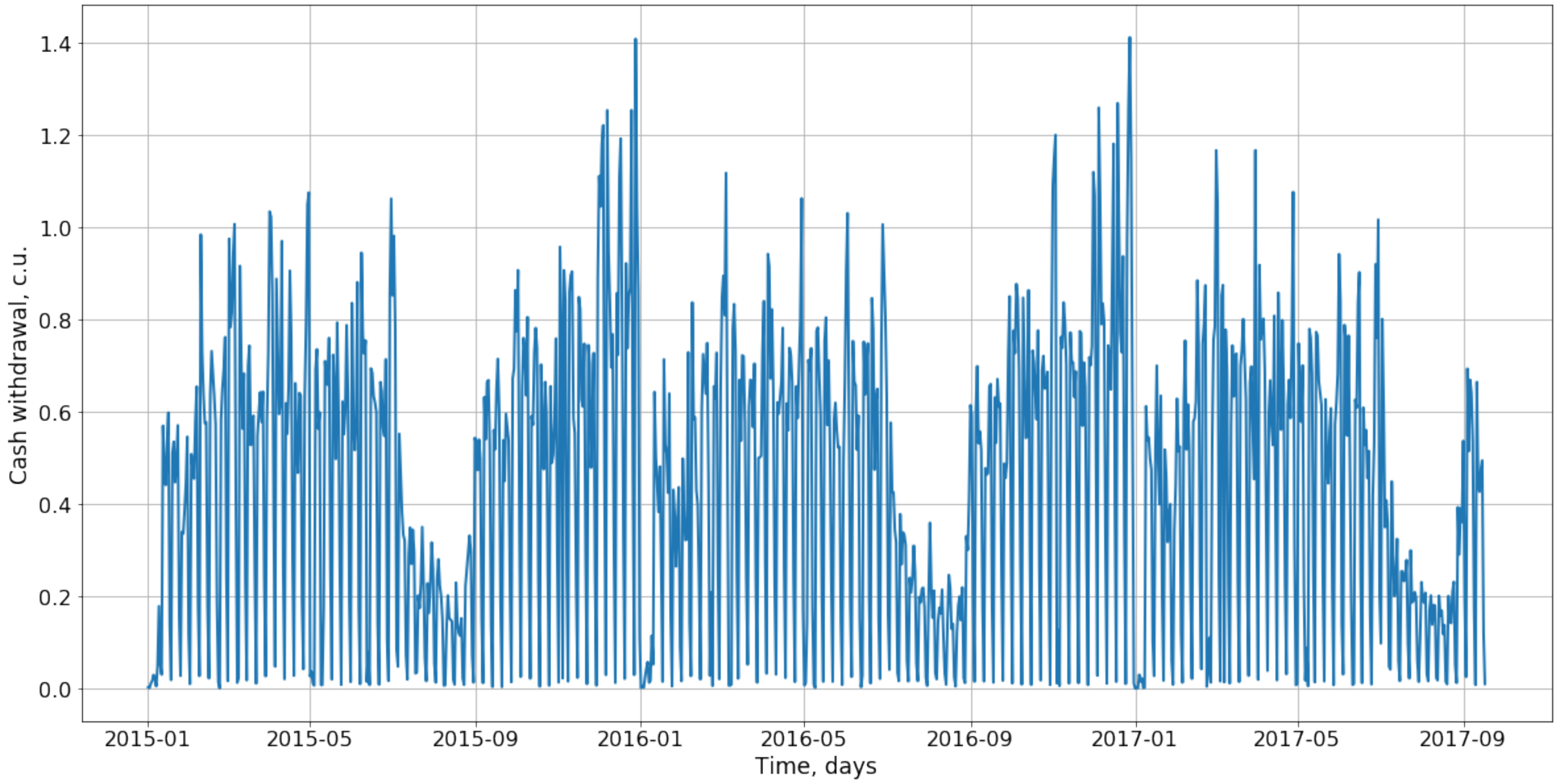
Stating the Problem

An example of almost ideal ATM case



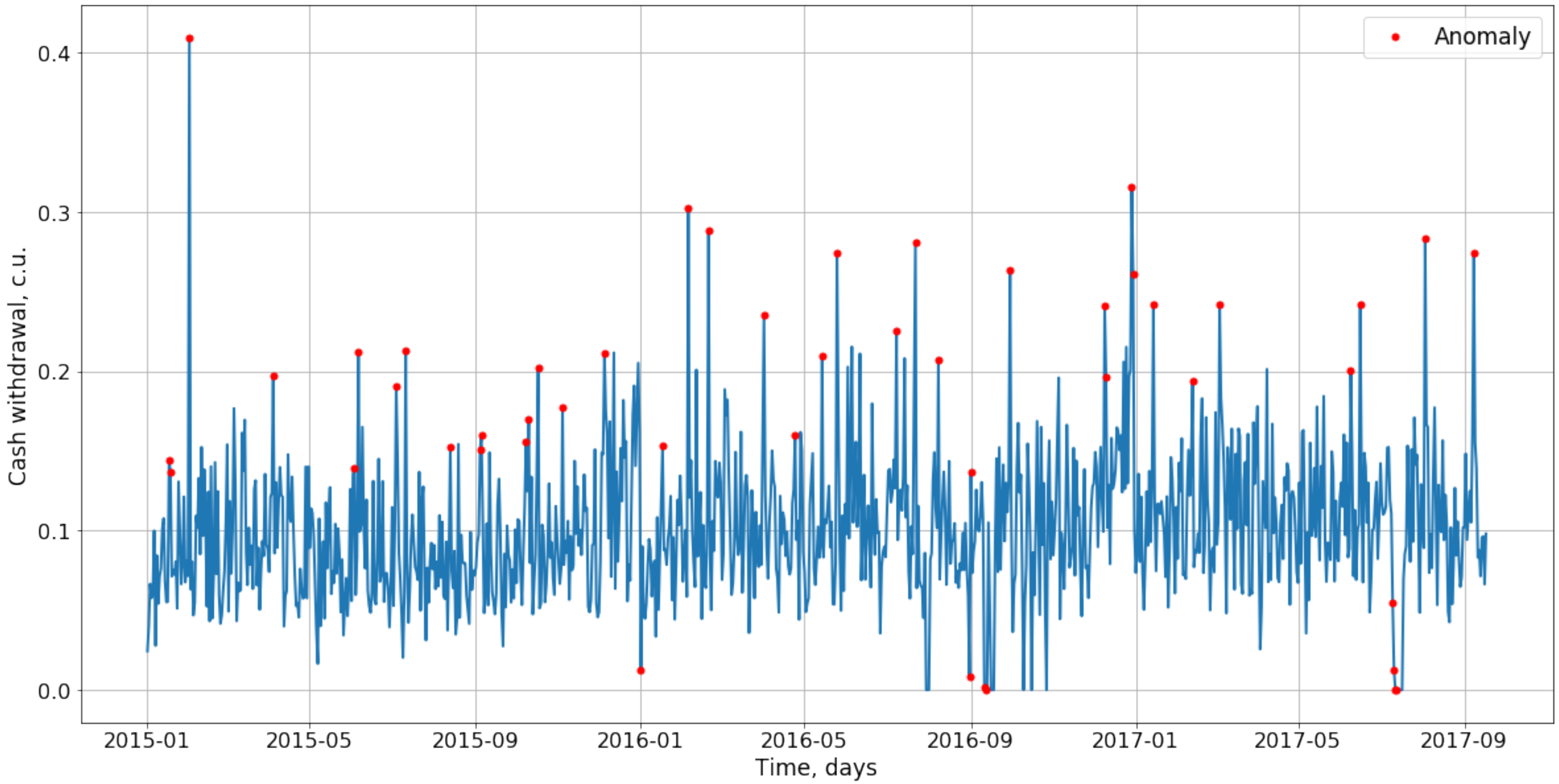
Stating the Problem

An example of ATM located in education facilities



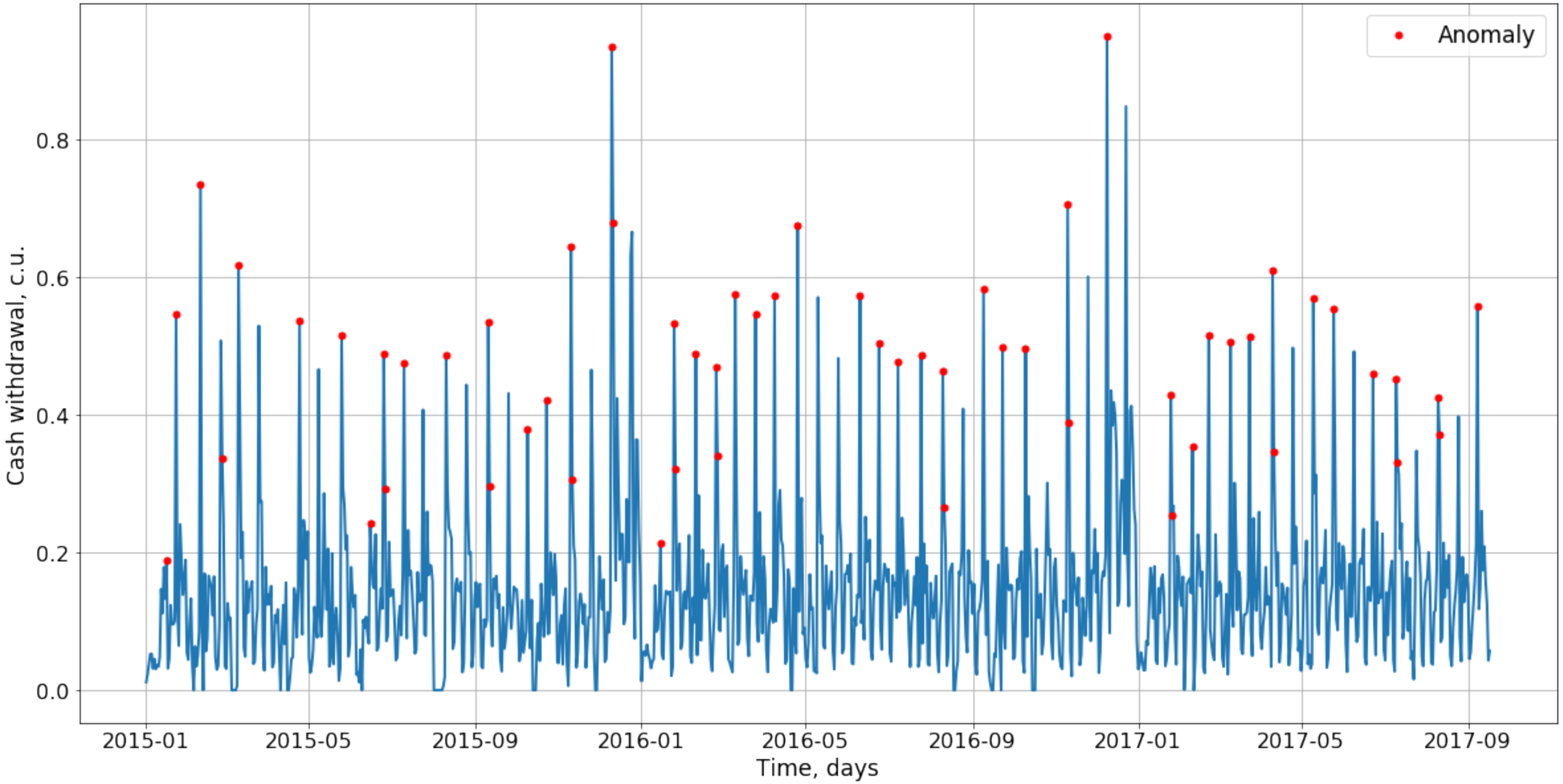
Stating the Problem

An example of ATM operating with a huge amount of stochastic outliers



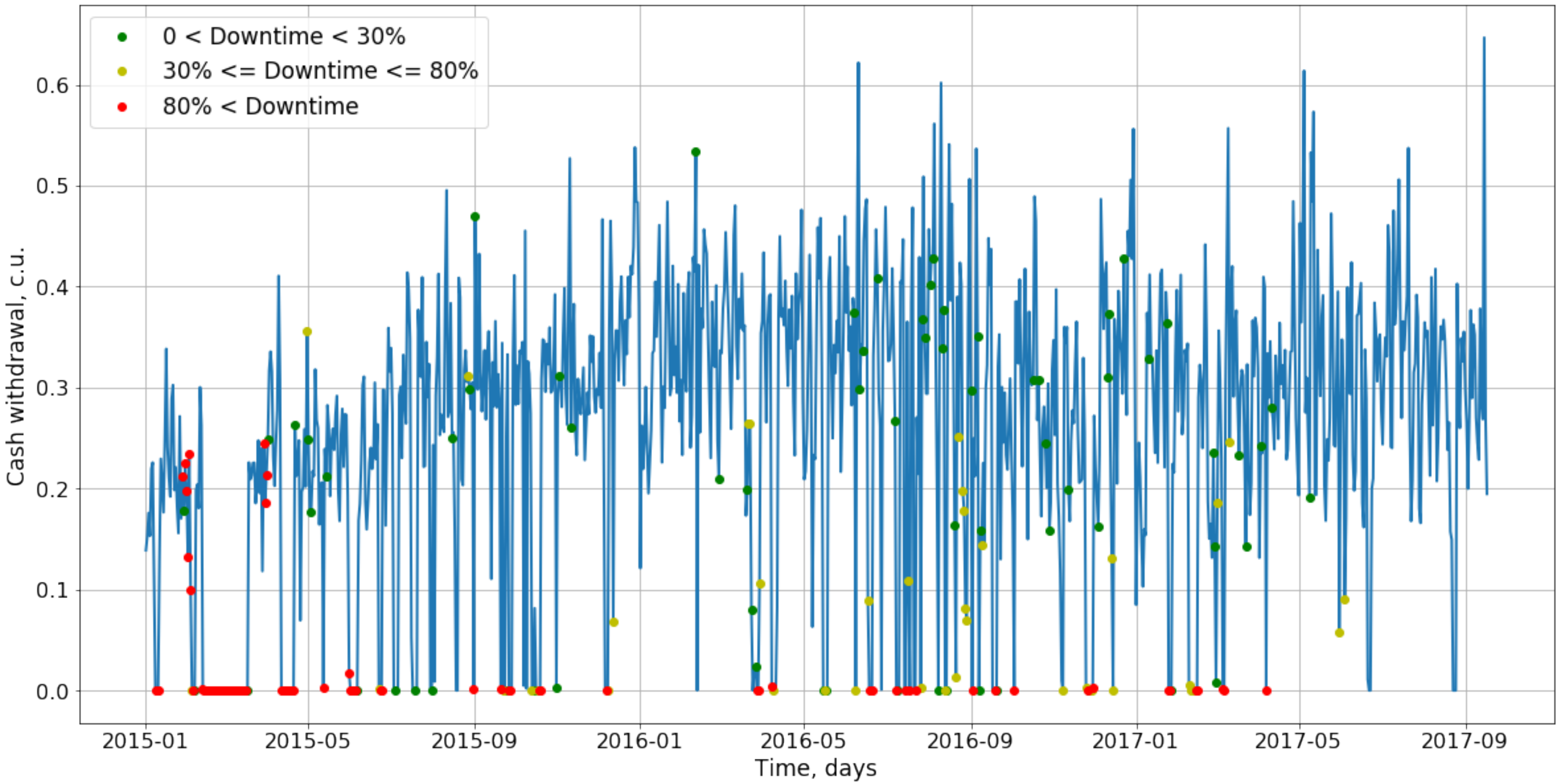
Stating the Problem

An example of ATM with a hue number of anomalies, associated with salary days



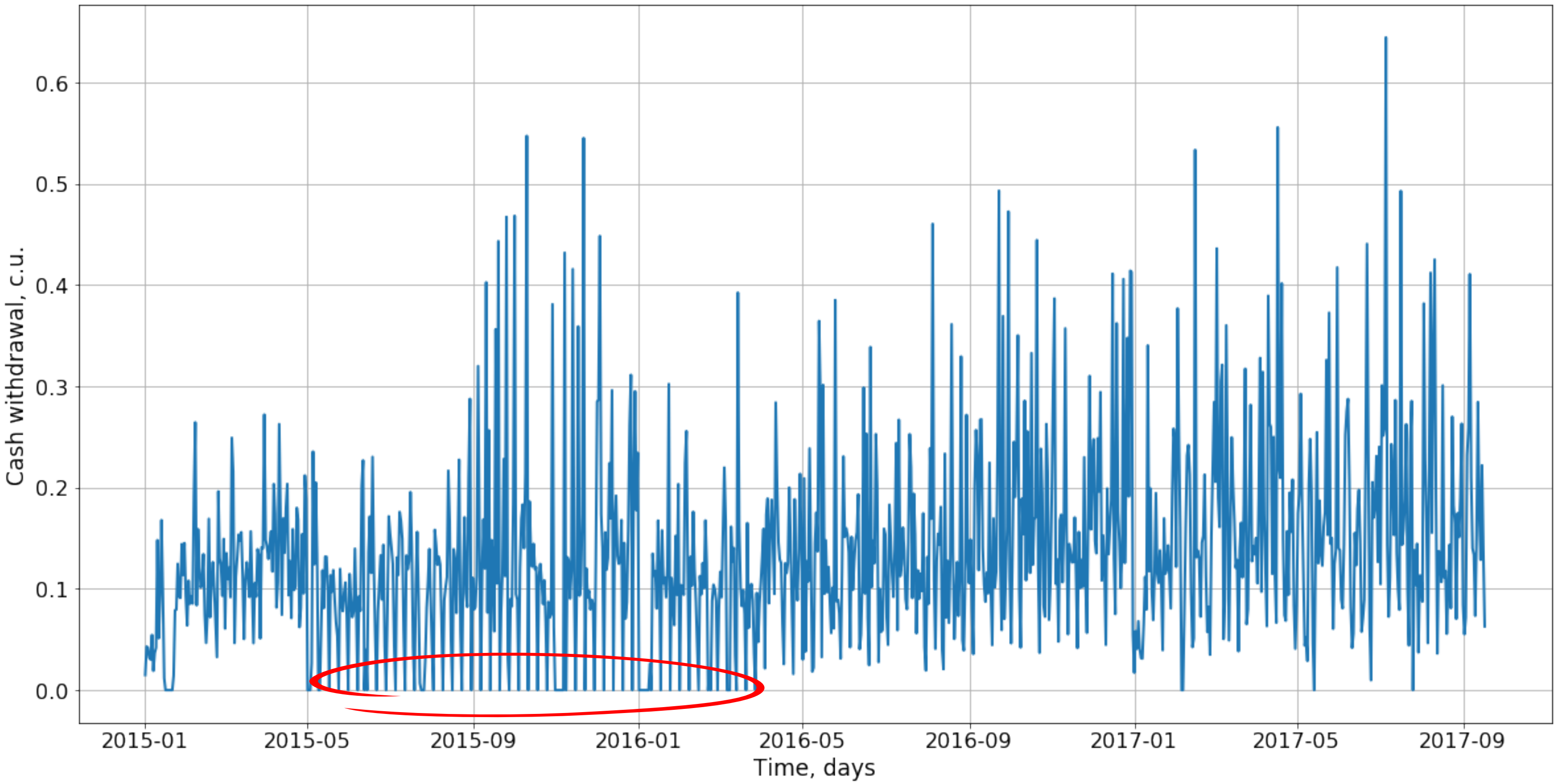
Stating the Problem

An example of ATM working with a huge amount of forced downtimes



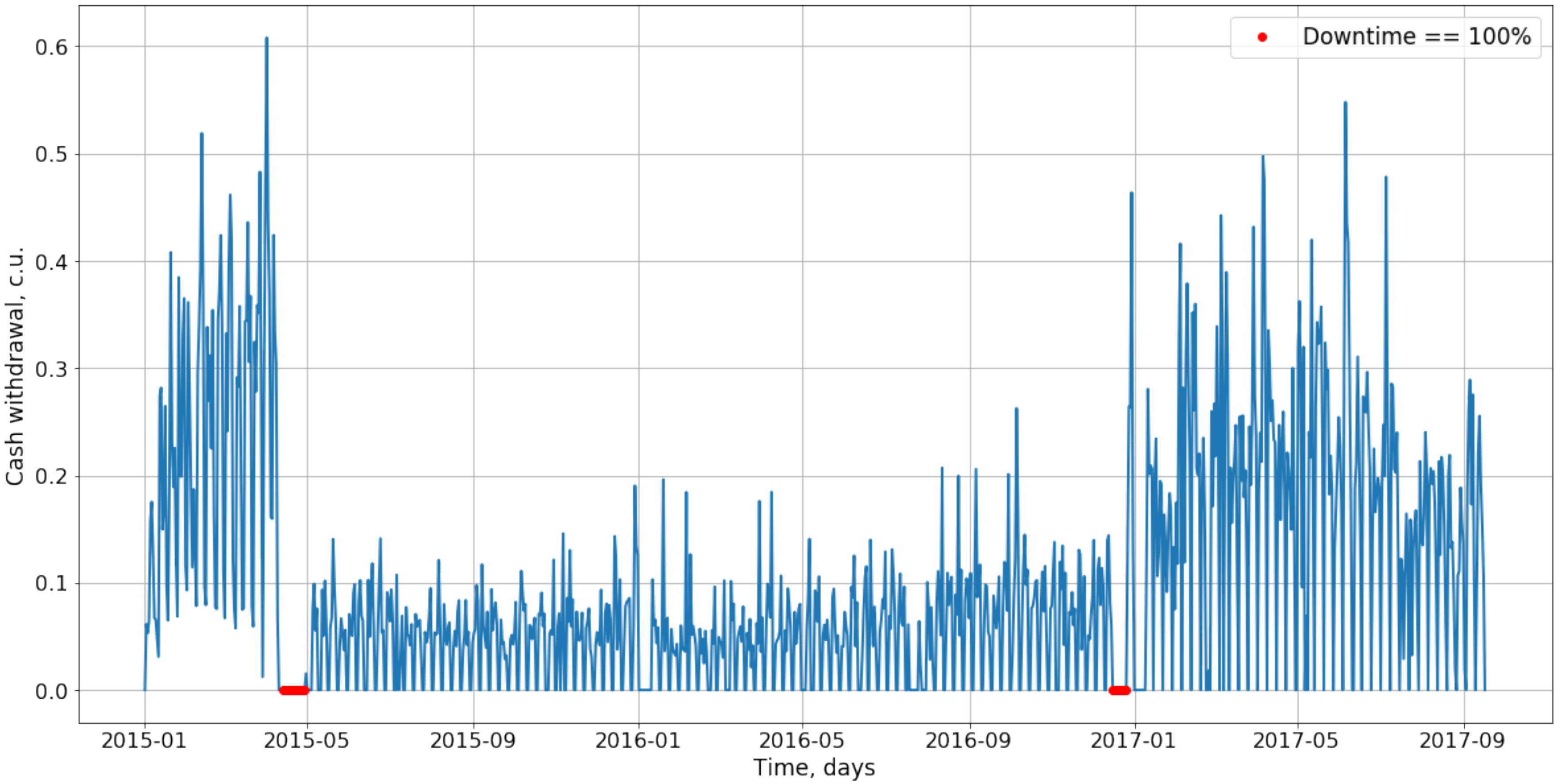
Stating the Problem

An example of ATM operating with a change in the nature of the distribution of client's demand associated with a change in operating hours in the place it is located



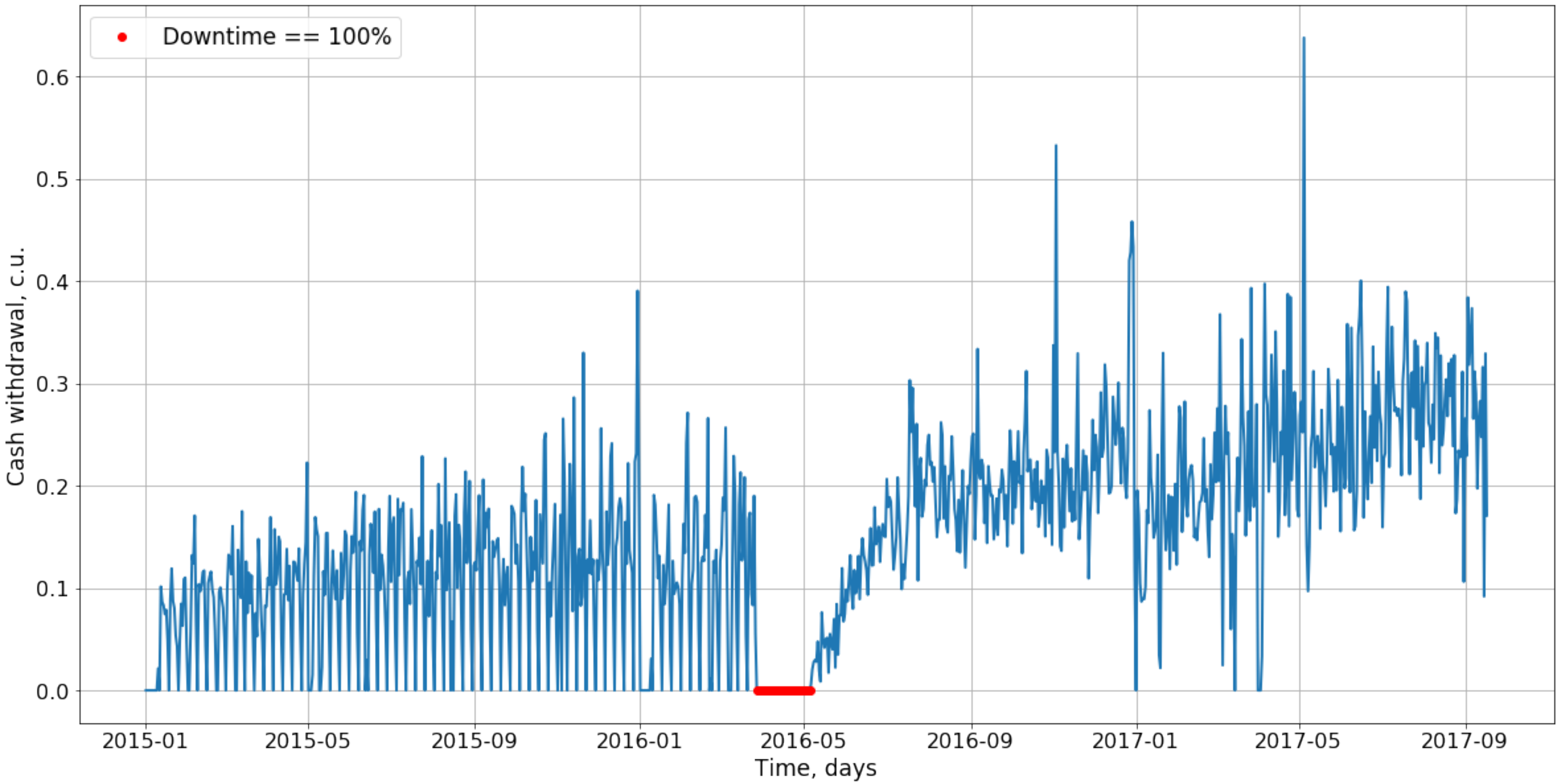
Stating the Problem

An example of ATM moved two times to new set up places



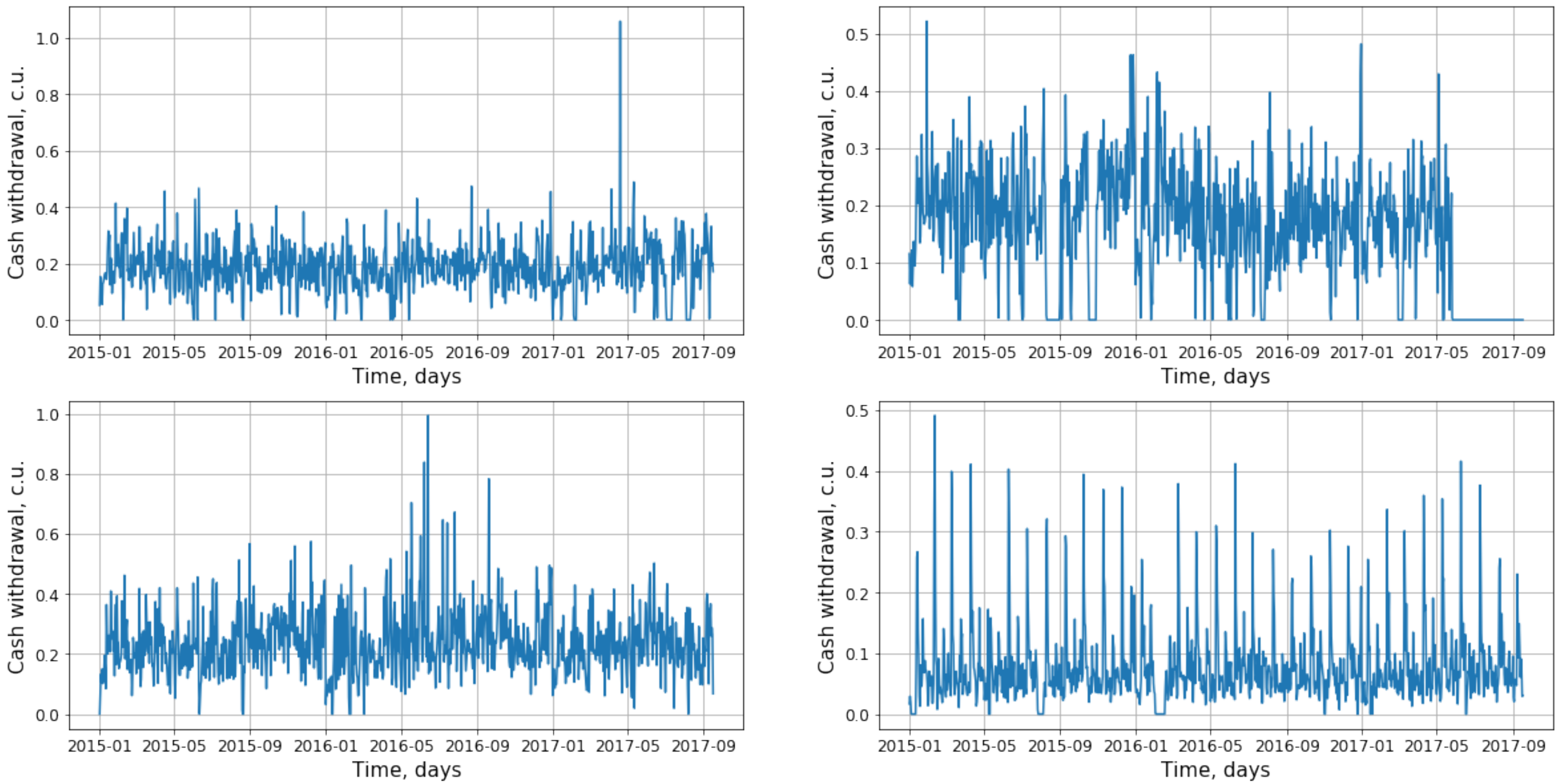
Stating the Problem

An example of ATM with a growing trend after its replacement



Stating the Problem

An example of 4 ATMs operating in one local branch bank



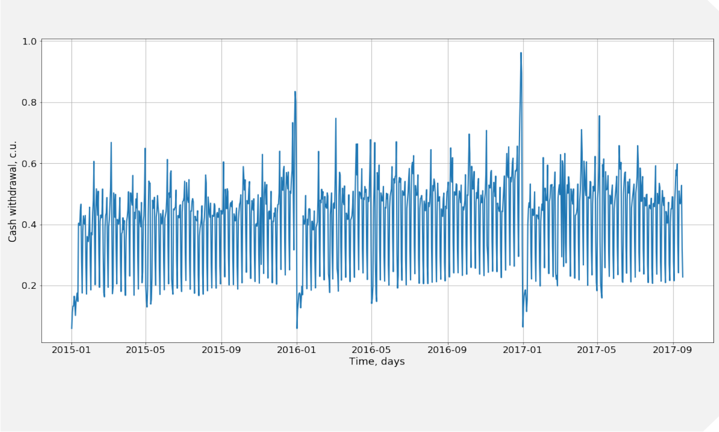
Aim

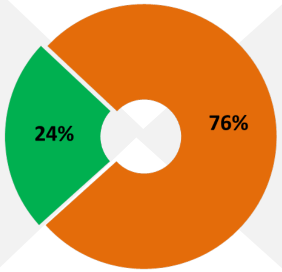
- To develop a <u>fully automatic </u>pipeline forecasting of customer's demand for cash in each ATM

- And as consequence to minimize the total costs of cash management (cost of funding + cost of collection) with a fixed probability of default.

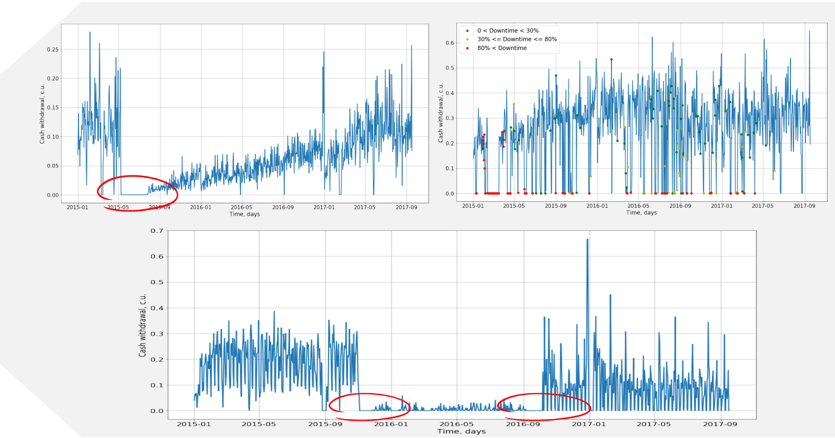
Pipeline

Example of “good” ATM history





Example of “bad” ATM history



Dividing histories into GH and BH

Good history (GH) → Downtime replacement → Detection salary/prepaid expense ATM → Outlier replacement → Repeat downtime replacement → Good fictitious history (GFH-1)

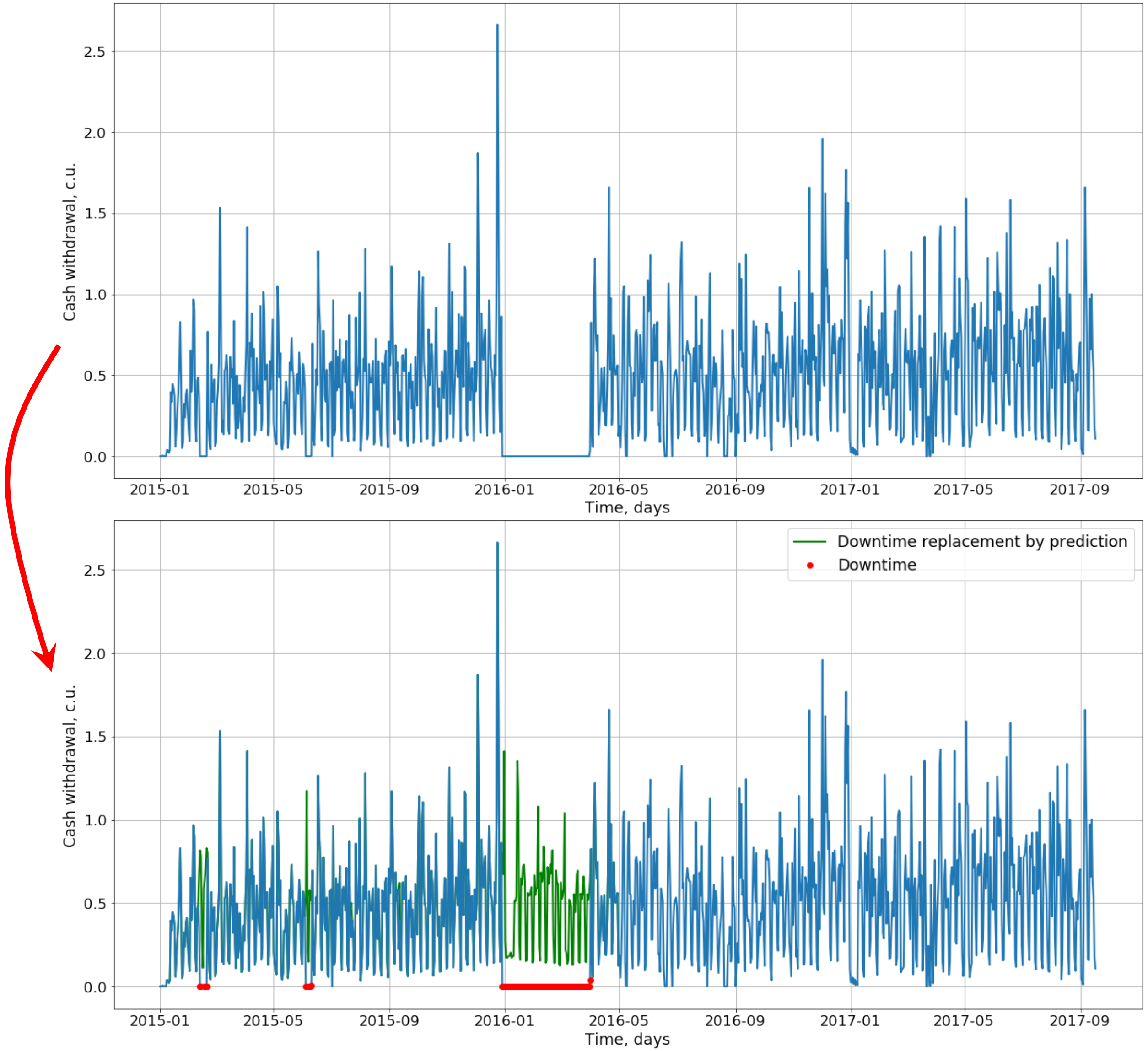
Bad history (BH) → Detection of the last correct history → Downtime replacement → Detection salary/prepaid expense ATM → Outlier replacement → Selection and replacement of history

Good fictitious history (GFH-1) → Selection and replacement of history → Good fictitious history (GFH-2)

Replacement of downtimes

- Replacement of downtimes is as follows: using data from the monitoring system, we determine the day of the first forced downtime, train the model in the time interval before downtime, predict one day ahead, replace the value of client's demand on the downtime with prediction. Then we repeat the procedure for the second, third, …, n-th day of forced downtime.

Visualization of the ATM downtime replacement by predicted values of the model being trained in an expanding window

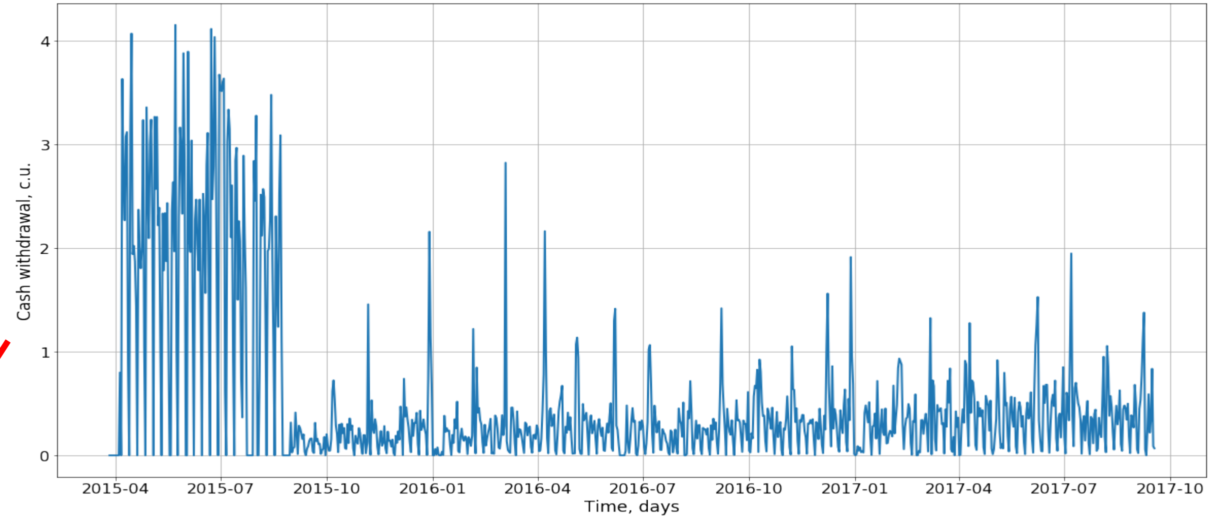


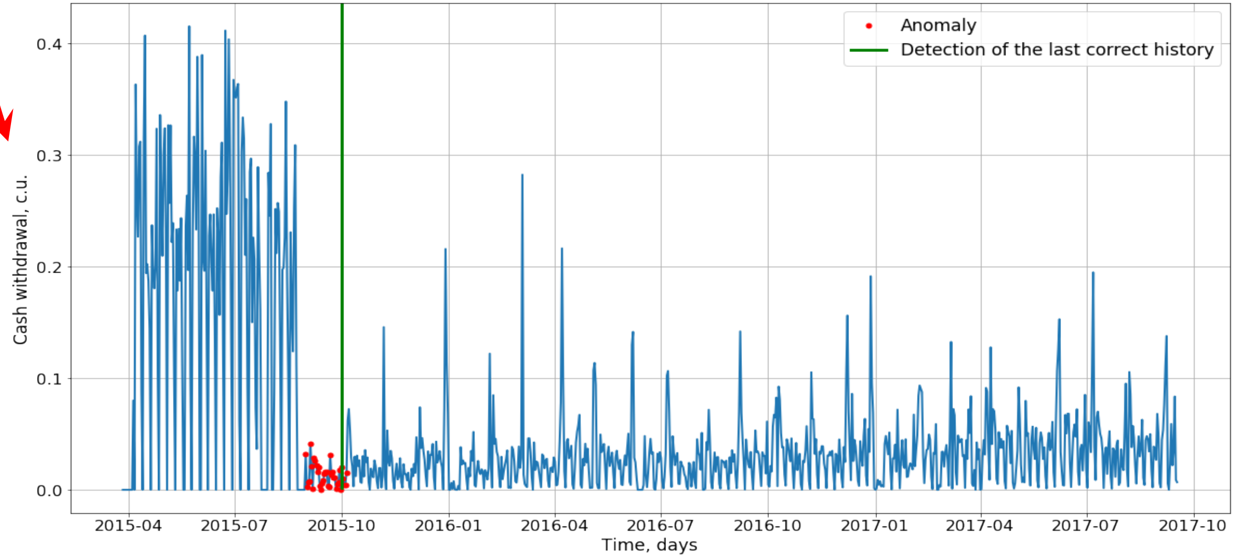
Detection of the last correct history

Detection of the last correct history is implemented when one of the conditions below is fulfilled:

- If there is a presence of the unbroken downtime for more than 14 days
- If the hypothesis of homogeneity of samples formed from client's demand values before and after downtimes is rejected in the Kolmogorov-Smirnov criterion
- With a significant increase of anomalies numbers found by the preset algorithm of cumulative sums in the sliding window or with a significant disorder in the prediction error

Visualization of the detection results of the last correct history based on a significant increase in the number of anomalies





Anomaly detection. CUSUM

CUSUM-algorithm detects changes in time series structure

Algorithm uses 4 parameters :

- the expected mean of the process, **μ**
- the expected standard deviation of the process, **σ**
- the size of the shift that is to be detected, **k**
- the control limit, **H**

Parameters **μ** and **σ** are defined in sliding window on historical values, **k** and **H** are usually defined by the problem

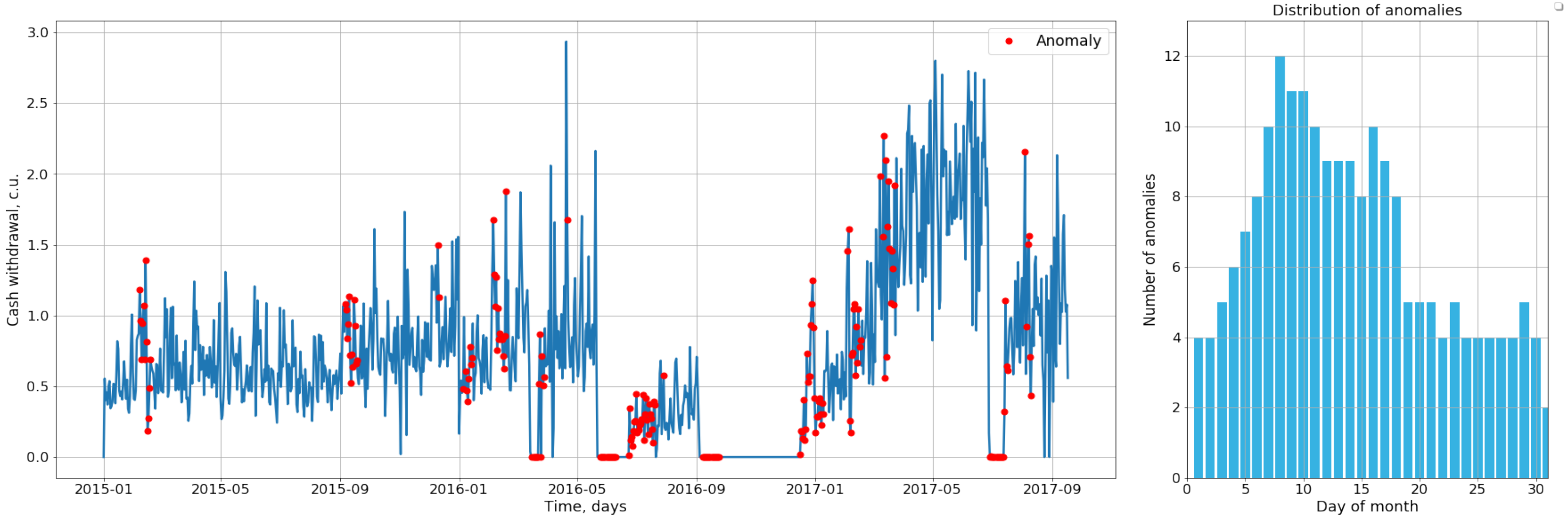
The algorithm is based on the process of accumulation of atypical deviations of the time series from the average value.

As soon as the cumulative sum exceeds the control threshold H, an anomaly is detected, the subsequent segment of the time series continues to be considered abnormal until the cumulative sum relaxes to the value of the smaller control threshold.

$$S_i^- = Max(0, S_{i-1}^- + x_i - \mu + k),\ S_0^- = 0\ for\ i = 1, 2, \dots, N$$

$$S_i^+ = Max(0, S_{i-1}^+ + x_i - \mu - k),\ S_0^+ = 0\ for\ i = 1, 2, \dots, N$$

An example of the results of applying the CUSUM algorithm to the time series of client withdrawals

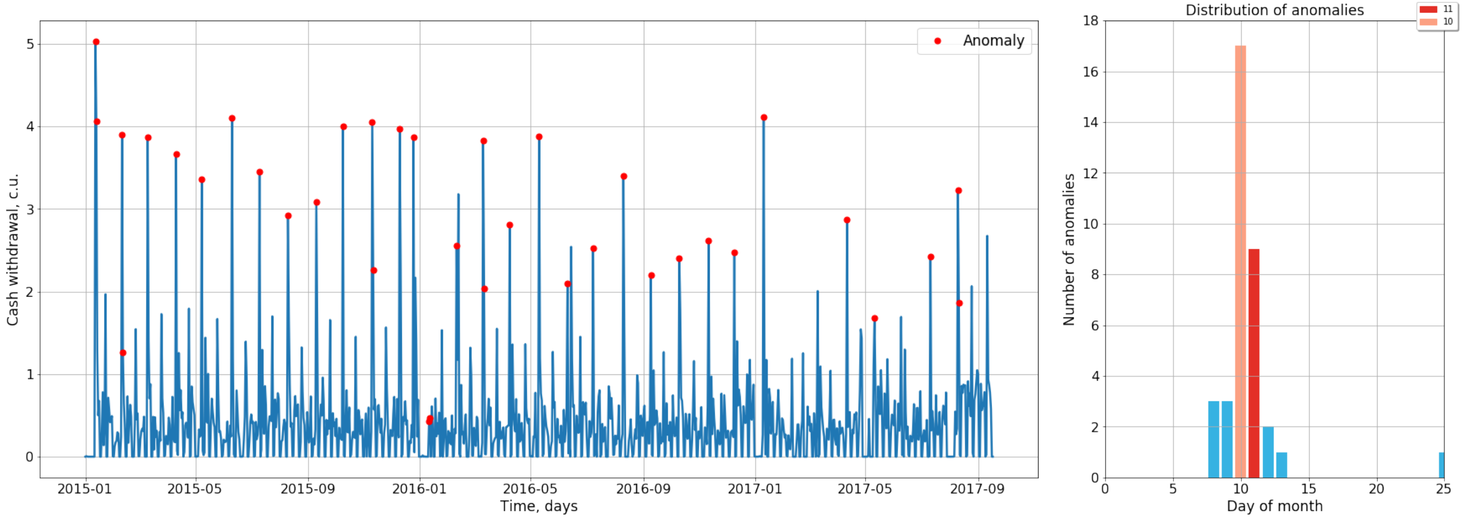


Detection of mass payment days

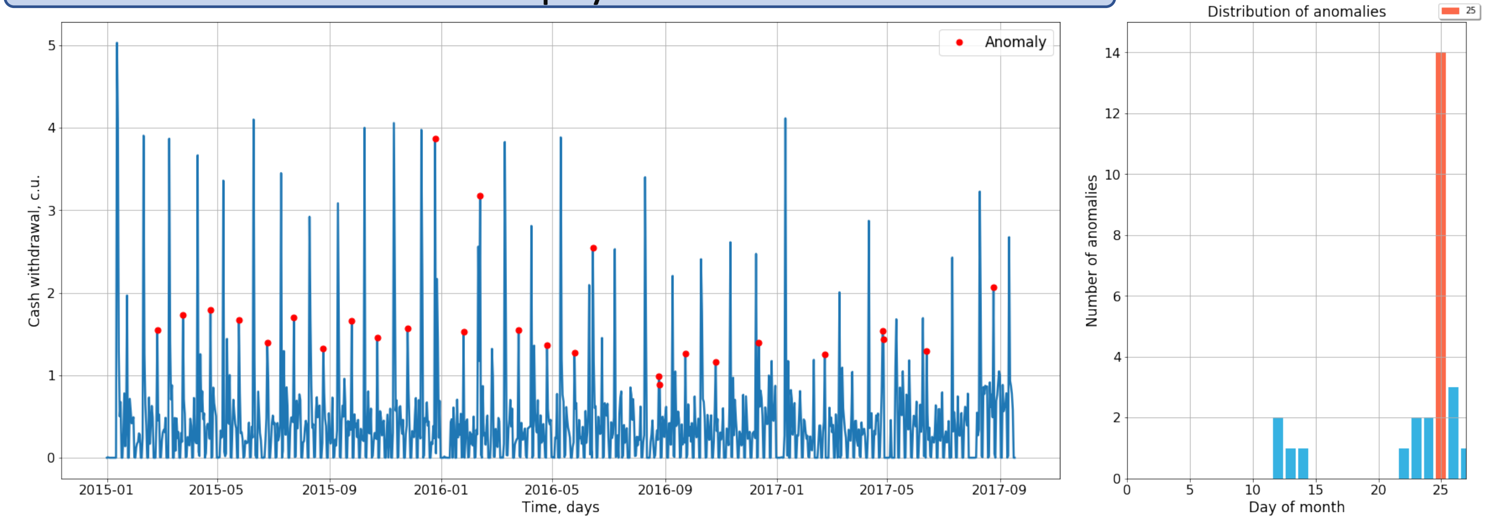
CUSUM-algorithm also helps detect <u>periodical outliers caused by mass payment days</u>: salary, prepayment, pension etc.

<u>Recursive application</u> of CUSUM in sliding window helps to extract relevant and actual features

Salary payments



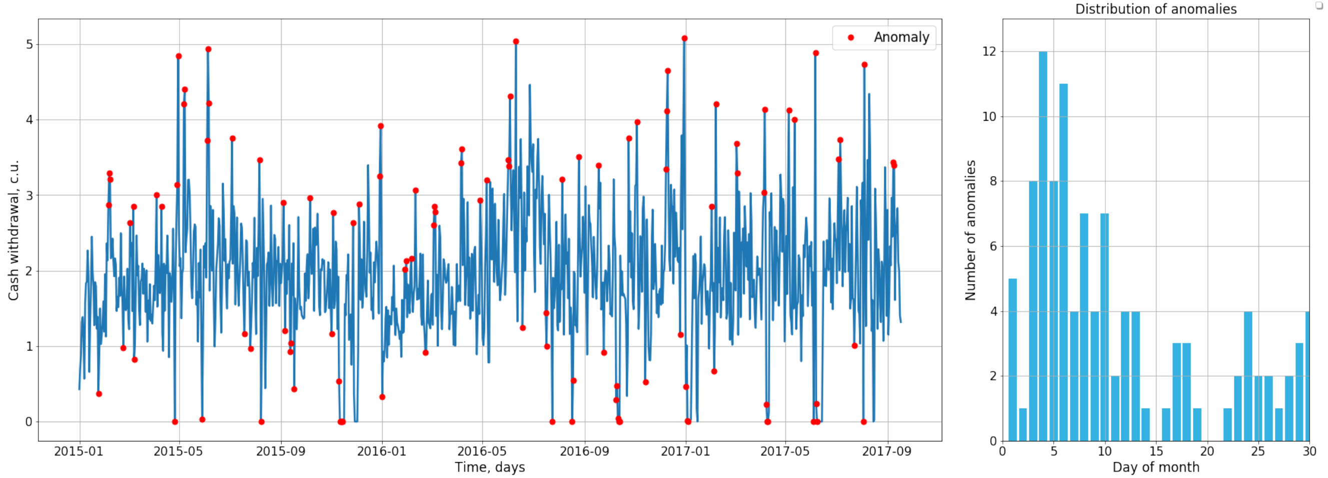
Avance payments



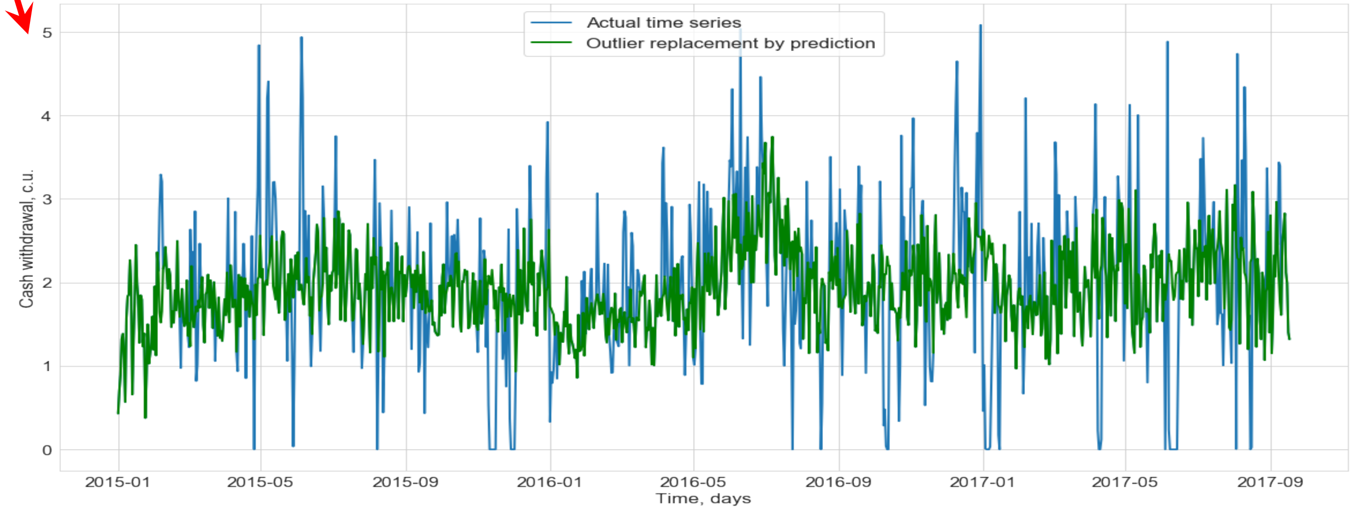
Replacements of stochastic outliers

If the found by the algorithm anomalies are <u>not regular</u>, then they are replaced by prediction in an expanding window like forced downtime.

An example of the results of applying the CUSUM algorithm to the time series of client withdrawals with a large number of stochastic emissions



Visualization of the replacement results of stochastic outliers by a prediction from trained model in an expanding window



Selection and replacement of client's demand history

Our algorithm uses a number of "proximity" criteria of ATM:

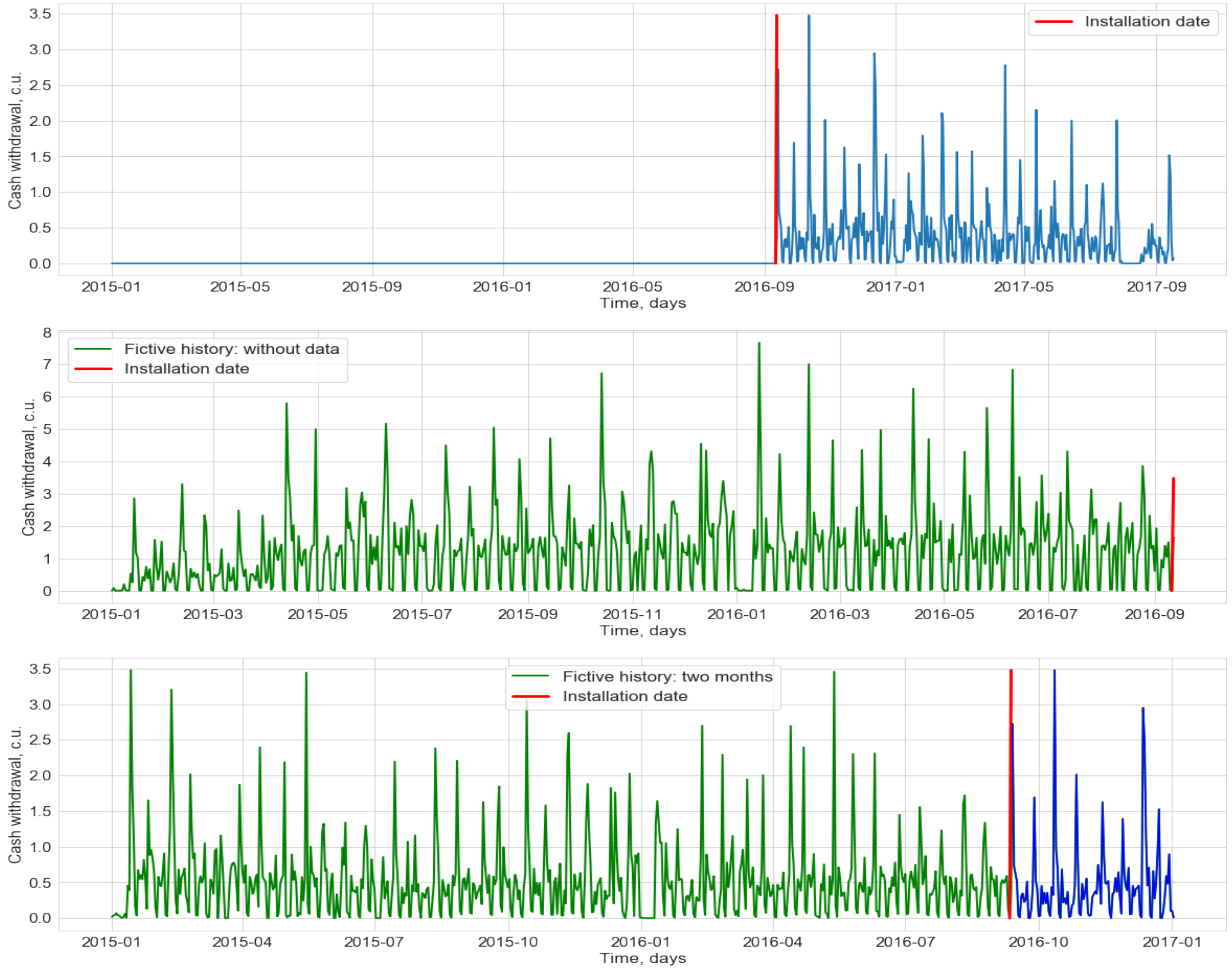
- Geolocation (geodesic distance between ATMs does not exceed r kilometers)
- Matching type of location (shopping center, urban recreation areas, business center, government buildings, educational facilities, etc.)
- Coincidence of accessibility of an ATM for a mass client (5/2, 24/7, etc.)
- Coincidence of mass payment day
- Correlation and mutual information between client's demand time series

Visualization of the fictive history selection from the various criteria

Cold start algorithm

- Based on the geo-location criteria, the type of the location and the accessibility mode, time series of ATM with similar characteristics are determined.
- As a fictive history, the time series closest to the average in the resulting group is selected.
- A week after the statistics on real client's withdrawals have accumulated, the selection of a fictive history is restarted with a new data.

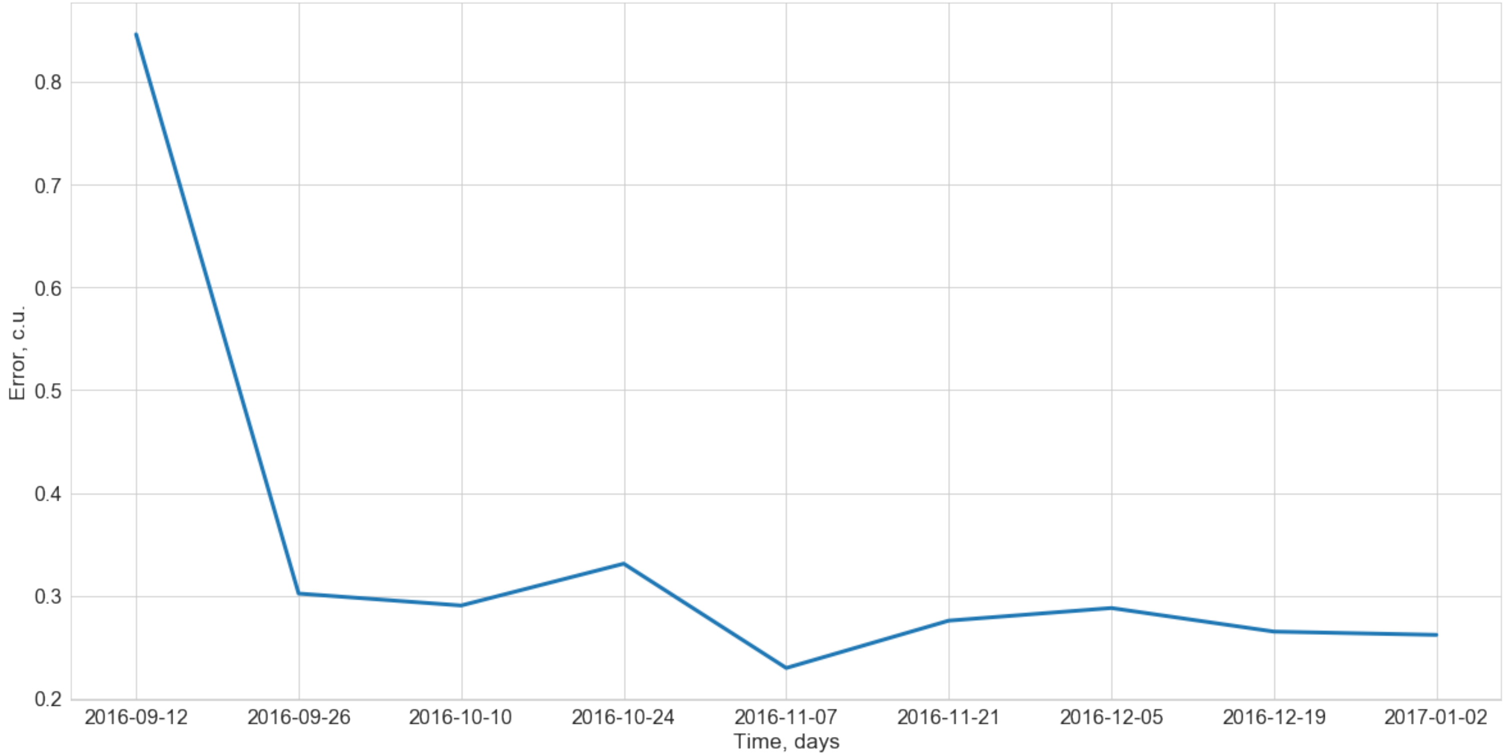
Visualization of the selected fictive history results for a new ATM: on the day of installation and after two months of operation



Cold start algorithm

- This continues until the prediction error stabilizes or until a sufficient amount of real data accumulates.

Graph of forecasting error from the date of selecting fictive history for a new ATM



Prediction model: approaches

In-house library for time series forecasting

svanga
0.0.1
Search docs

- Детектирование аномалий
- Периодичные аномалии
- Построение предсказаний
- API

Docs » svanga: библиотека для анализа и предсказаний временных рядов View page source

svanga: библиотека для анализа и предсказаний временных рядов

Author: Sberbank

Date: Feb 12, 2018

Version: 0.0.1

SVanga - это библиотека для прогнозирования временных рядов, которая способна выявлять периодичные и непериодичные аномалии, что позволяет точнее строить прогноз для сезонных рядов. SVanga устойчива к пропускам в данных, аномальным выбросам. Она работает хорошо как на чистых, так и на зашумленных данных.

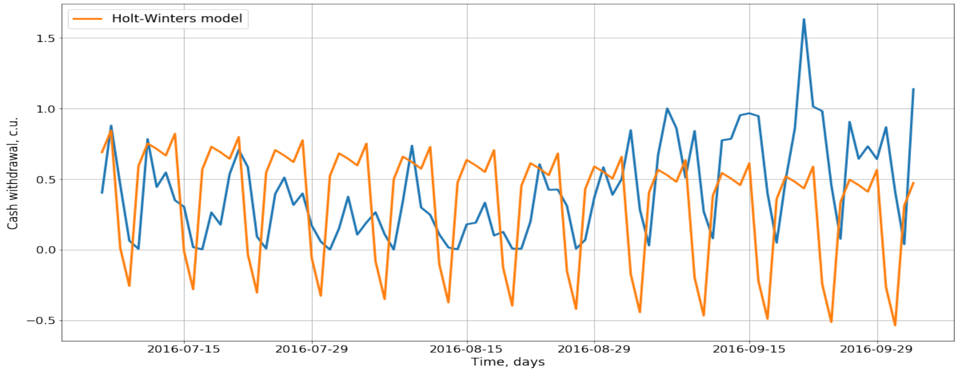
Для того, чтобы установить последний релиз, напишите:

```
pip install svanga
```

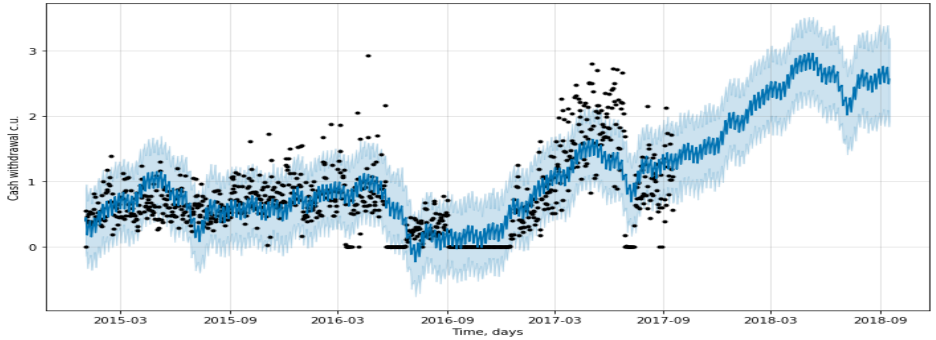
Контент

- Детектирование аномалий
  - Инициализация класса детекции аномалий
    - Краткое описание метода CUSUM
  - Простая детекция аномальных значений
  - Визуализация аномалий
- Периодичные аномалии
  - Примеры:
    - Курс цены акций Сбербанка
    - Выдача наличных в банкоматах
  - Разметка
- Построение предсказаний
  - Генерация признаков
    - Возможные признаки
  - Прогноз на исторических данных (backtest)
  - Прогноз в реальном времени (realtime)

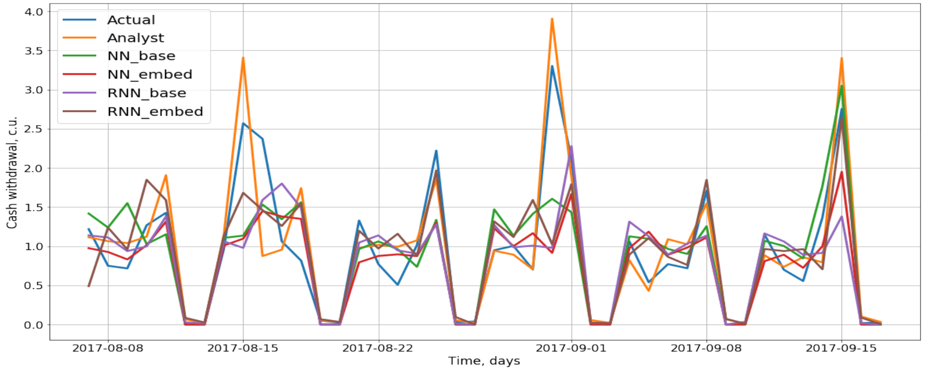
Classic TS approach (HW)



FB Prophet



NN approach



Prediction model: feature space and forecasting process

For each time series of pools GFH1 and GFH2, an extensive feature space is created (more than 50 features):

- Features based on the industrial calendar, encoded with one-hot transformation (day of the week, month, holiday, pre-holiday, last day of the month, pre-New Year, etc.)

- Lag features (time series values for previous days)

- Rolling statistics (minimum, maximum, average, variance, median of the time series for the previous period of size w)

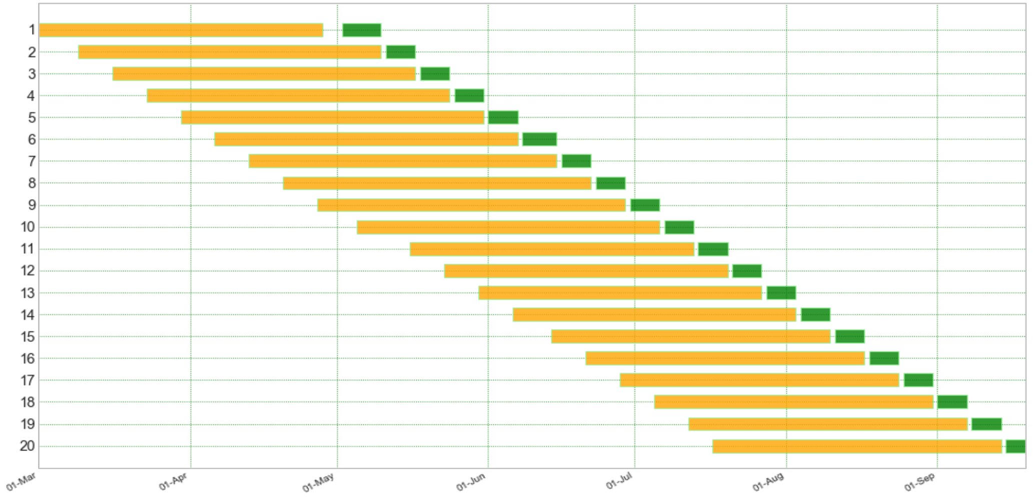
- Rolling statistics grouped by calendar (for example, statistics of a time series from a paragraph above on Tuesdays, public holidays, etc.)

- Marking the mass payment days

An example of feature space

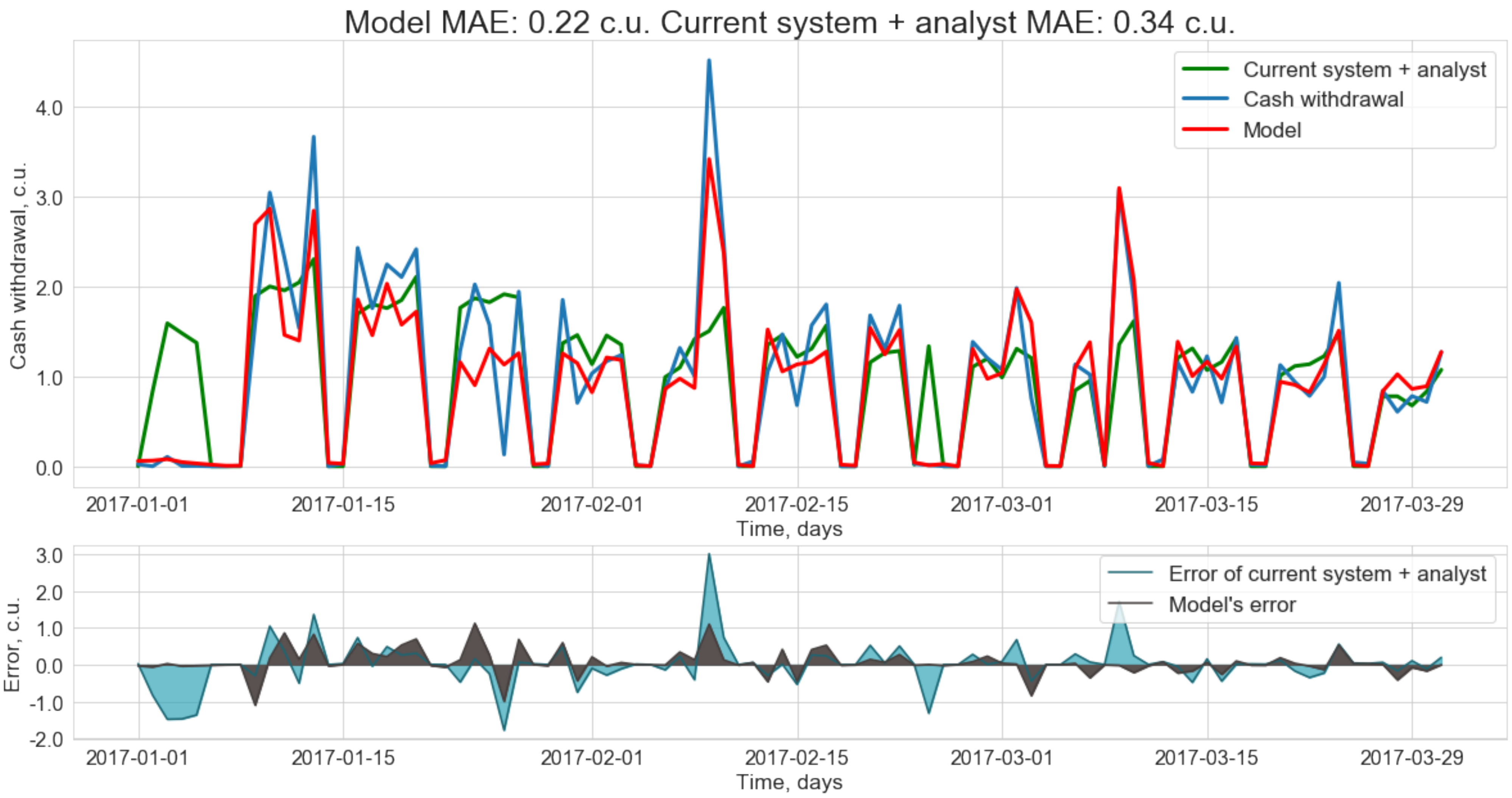
TimeStamp	Target	One-Hot-Encoding					Lags		Rolling Statistics				One-Hot-Encoding	
	y	пн	вт	ср	чт	пт	lag_1	lag_2	rolling_mean	rolling_std	rolling_mean_weekday	rolling_min_weekday	зарплата 15	праздник
2017-09-08	1273900.0	0	0	0	0	1	1118000.0	896500.0	653100.000000	391381.906412	513666.666667	306200.0	0	0
2017-09-09	344700.0	0	0	0	0	0	1273900.0	1118000.0	726185.714286	457375.871790	143900.000000	72500.0	0	0
2017-09-10	45200.0	0	0	0	0	0	344700.0	1273900.0	746514.285714	432713.001224	36466.666667	6200.0	0	0
2017-09-11	930000.0	1	0	0	0	0	45200.0	344700.0	740328.571429	443853.266723	843400.000000	443700.0	0	0
2017-09-12	43900.0	0	1	0	0	0	930000.0	45200.0	735400.000000	441132.391314	494900.000000	468100.0	0	0
2017-09-13	544300.0	0	0	1	0	0	43900.0	930000.0	664600.000000	511907.166063	546400.000000	319400.0	0	0

Visualization of the cross-validation procedure in a sliding window



Prediction model (based on LR + RF): an example of results

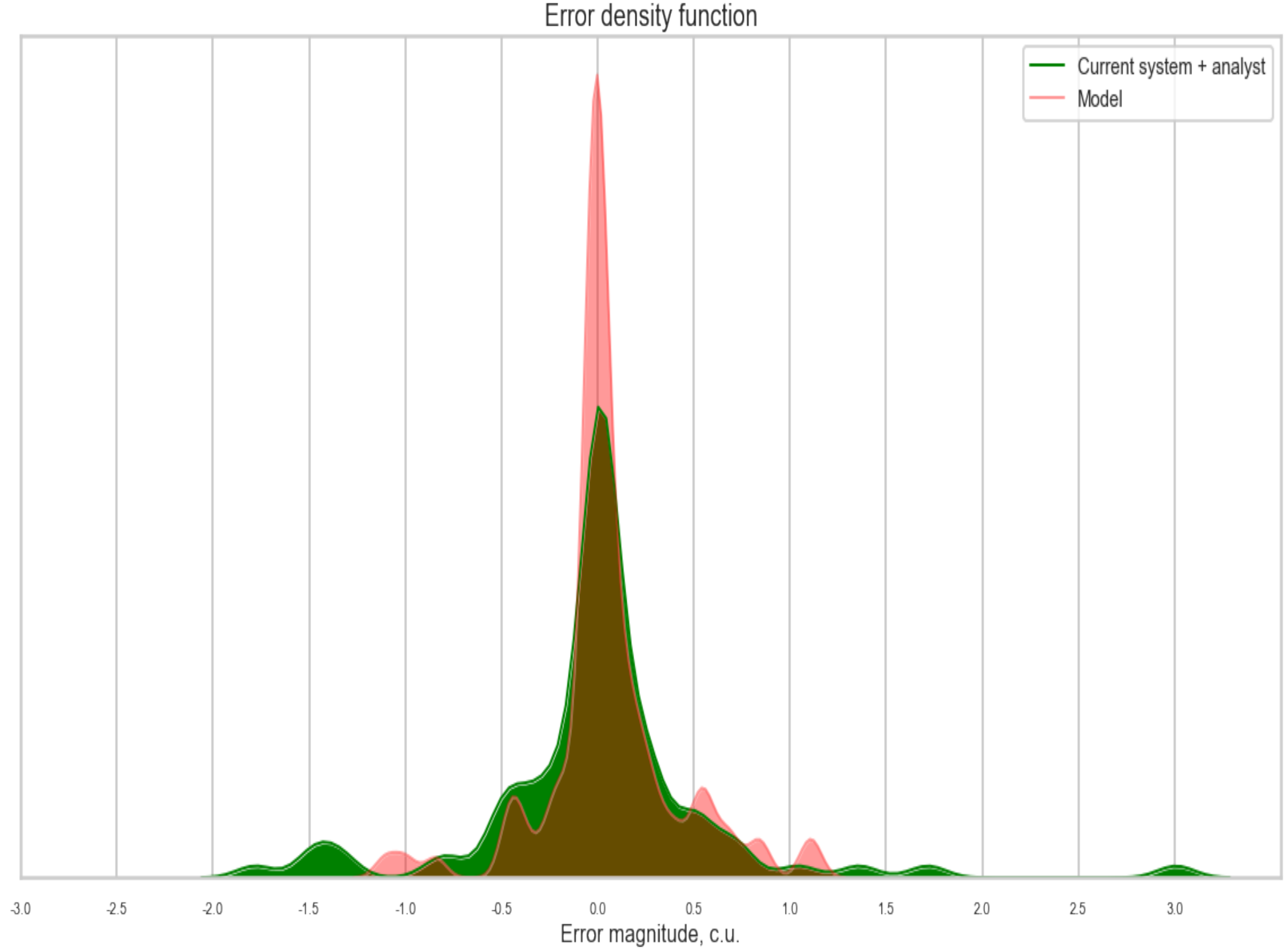
Visualization of forecasts and errors of the proposed model and the current system with the analyst's correction



Prediction model (based on LR + RF): an example of results

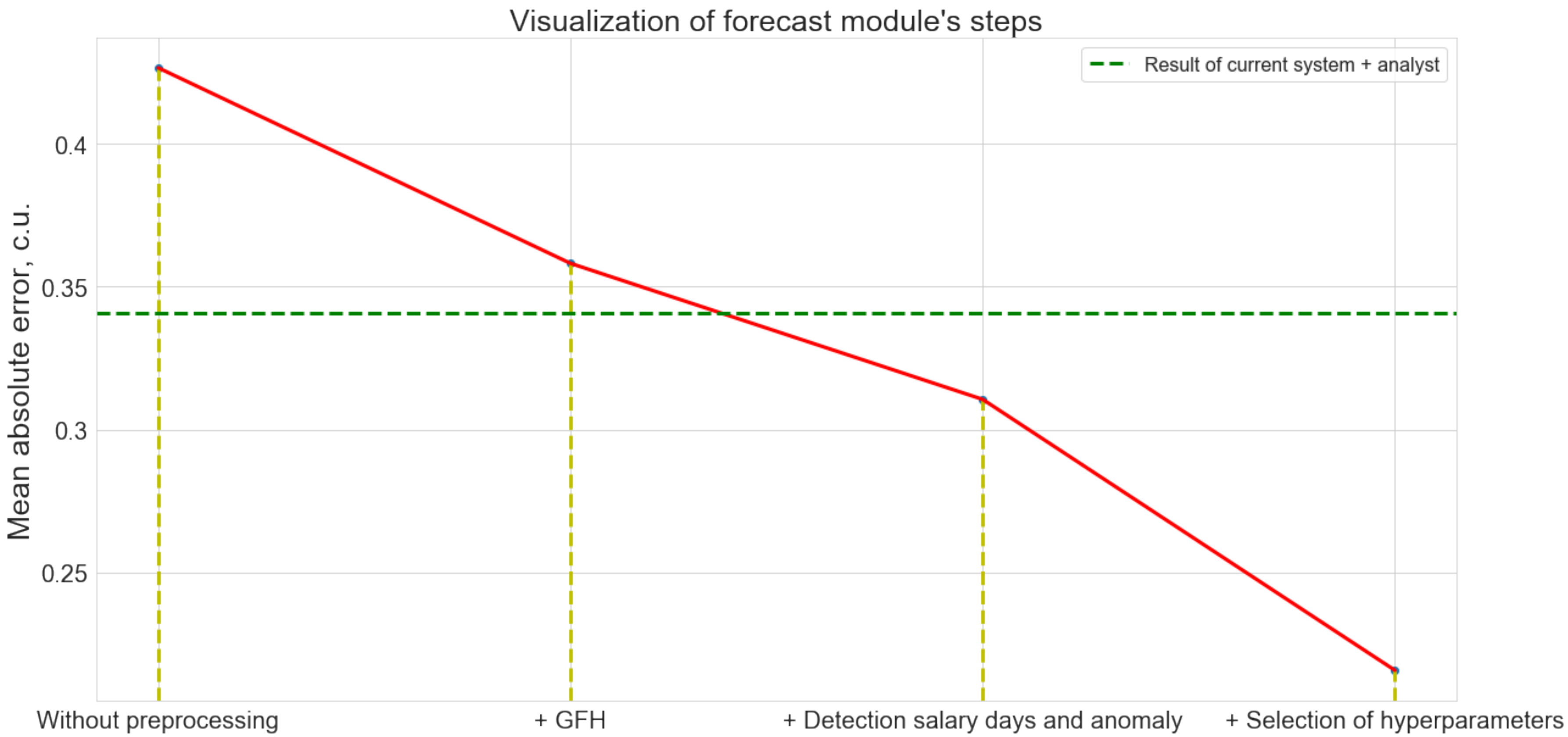
- The symmetry of the error density function of the proposed model indicates the absence of shifts towards a systematic over forecast or under forecast.
- On the contrary, the bias of the tails of the error density function of the current approach towards positive values reflects the analyst’s desire to minimize risks of downtime occurrences.
- In general, error density of our model is more narrower, and therefore a greater number of deviations fall into a less wide confidence interval.

The density function of forecasting errors of the proposed model and the current system with the analyst's correction



Prediction model: mean absolute error metric

Visualization of the automated stages contribution in data preprocessing to minimization MAE

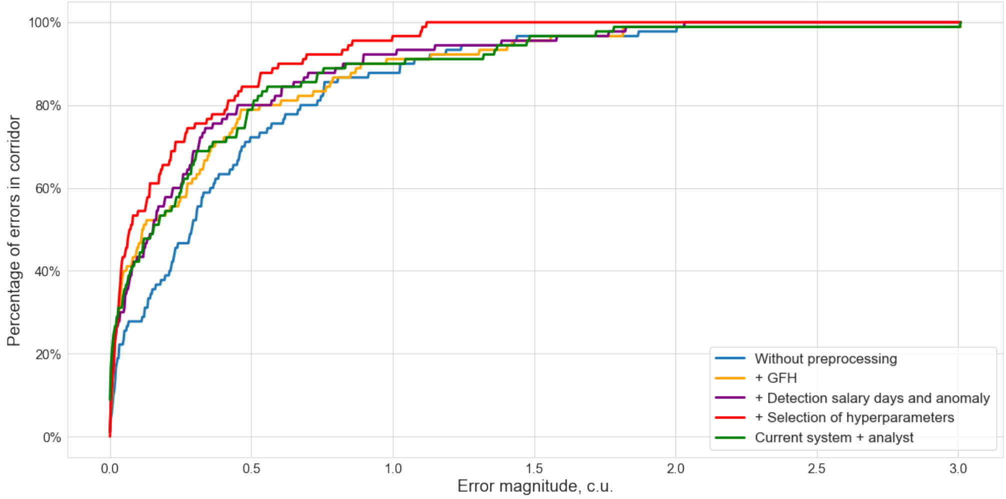


Prediction model: AUC-AE metric

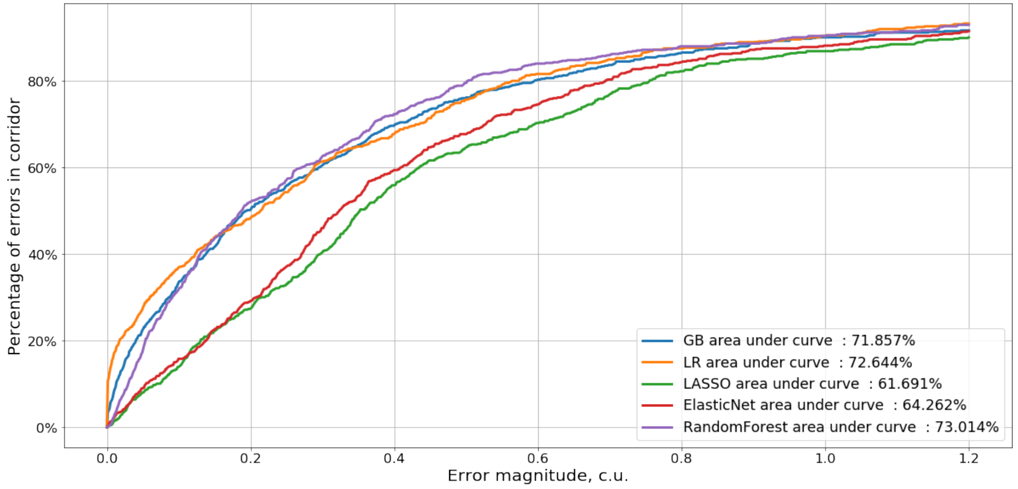
- Let us introduce a metric, allowing to take into account different confidence intervals of the error depending on the risk appetite: *AUC - AE* - the area under the curve, reflecting the percentage of hitting a given corridor of deviations.

- The value of the AUC - AE metric is a relative value ranging from 0 to 100% and is calculated as follows:
  - The cutoff bound is determined.
  - In a rectangle [0, bound] × [0, 100] the area under the curve is calculated AUC -- AE
  - The resulting area is normalized to the area of the rectangle: $area = \frac{under\_curve\_area}{bound}$

Visualization of the contribution stages of automated data preprocessing using *AUC - AE* curves



Visualization of the prediction quality of different machine learning methods by metric *AUC - AE*



Prediction model: results

We indirectly associate the average absolute error (*MAE* metric) with cost of funding, and the metric, which takes into account various risk appetites (*AUC - AE* metric), with the probability of downtime due to the lack of cash occurrence.

Business restrictions on the length of downtime are given independently for each ATM, besides, ATMs can have radically different cash flows, and therefore it is not entirely correct to compare them with *MAE*.

We compare the prediction results of the proposed approach with the current metrics for the mentioned above two metrics for each time series separately.
The comparison results are presented below:

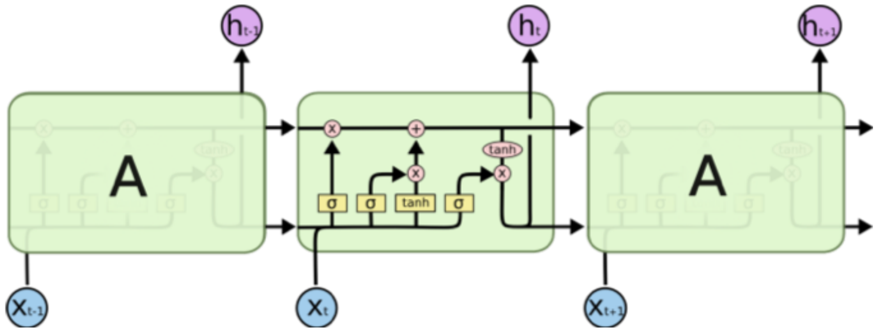
- In 67% cases: *MAE* in our model, on average 36% lower than the current approach, and in the remaining 33% of cases: on average, 12% higher, while 95% percentile is 24%.

- In 61% cases: *AUC - AE* exceeds the current approach by 3.4 points on average, in the remaining 39% of cases: on average, it is lower by 1.7 points, and the 95-percentile is 2.1 points.

Compare with NN approach results

We also considered an approach in which all time series are considered not individually, but together and are fed to the input of one single model. Such a model, we have chosen the architecture of recurrent neural networks based on *LSTM.*

Architecture scheme LSTM



Except *LSTM* layer in model, there are three fully connected layers, dropout with p = 0.2 and activation function is *leaky relu*. The results of parallel comparison:

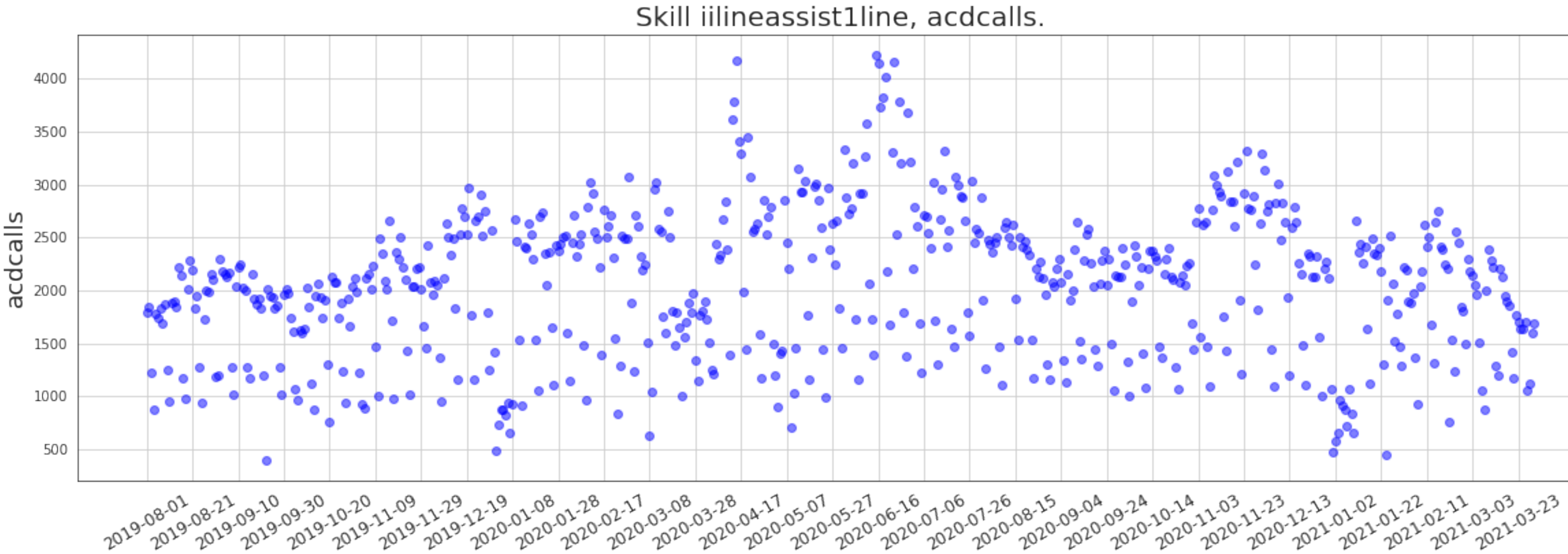
- In 46% cases: *MAE RNN - LSTM* on average 17% lower than the current approach, and in the remaining 54% cases: on average 22% higher, while 95% percentile is 31 %.

- In 35% cases: *AUC - AE RNN - LSTM* exceeds the current approach by 1.67 points on average, in the remaining 65% cases: on average, it is lower by 2.9 points, and the 95-percentile is 3.8 points.

Time Series Forecasting
in Call Center Management

Stating the Problem

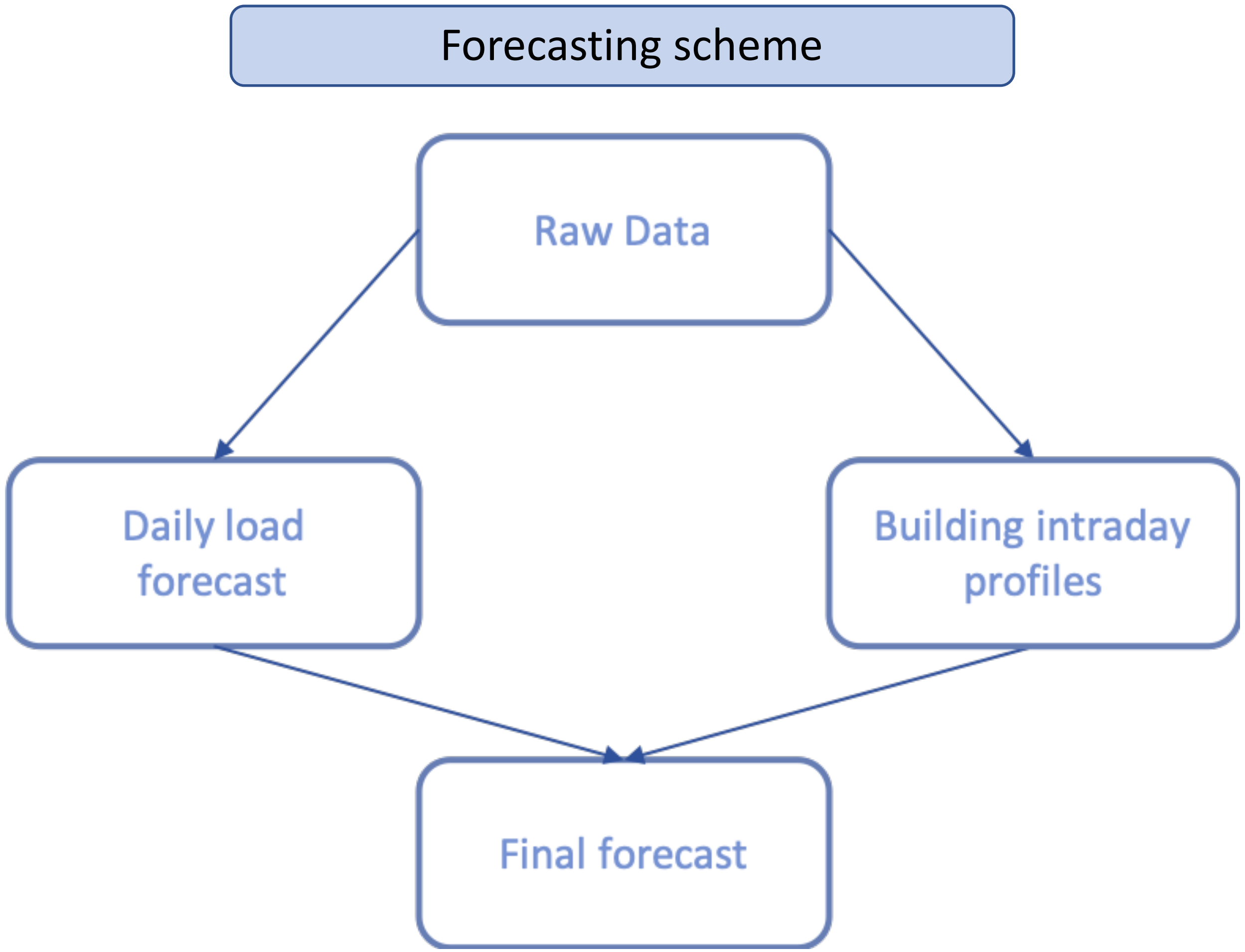
Number of calls per 15-minutes on a certain direction



Stating the Problem

The problem is divided into two subtasks:

- Daily load forecast
- Forecast of the intraday load profile by classification patterns
- The final prediction is formed by multiplying forecasts of daily and intraday loads.



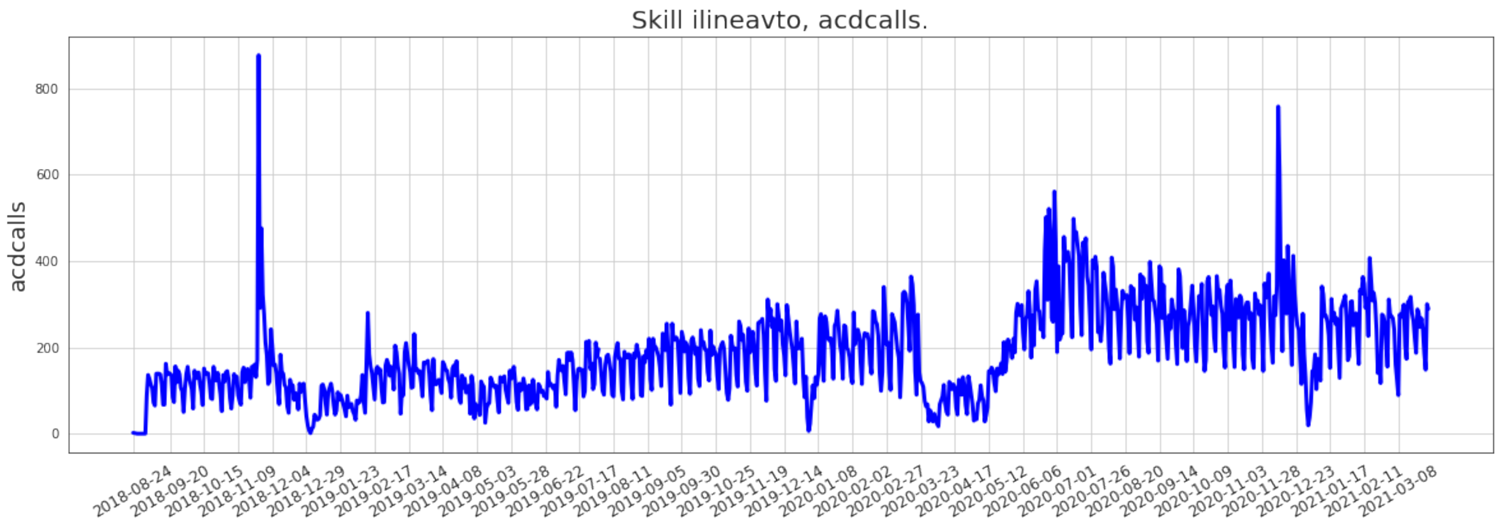
Daily data

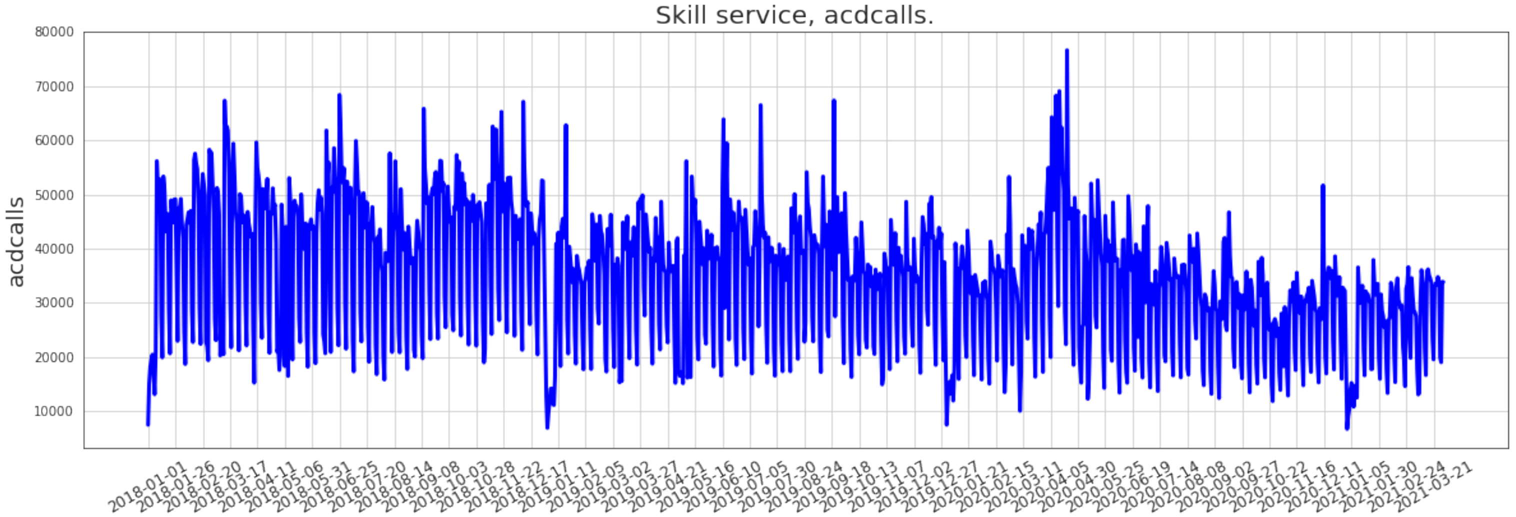
All directions have a strong weekly seasonality and a low load at the beginning of January.

Some directions have a down-trend. Some have an up-trend.

For some skills, you can note an atypical seasonality.

In all time series, one way or another, strong peaks of increased load, sagging with abnormally low values are noticeable.





Daily Feature Space

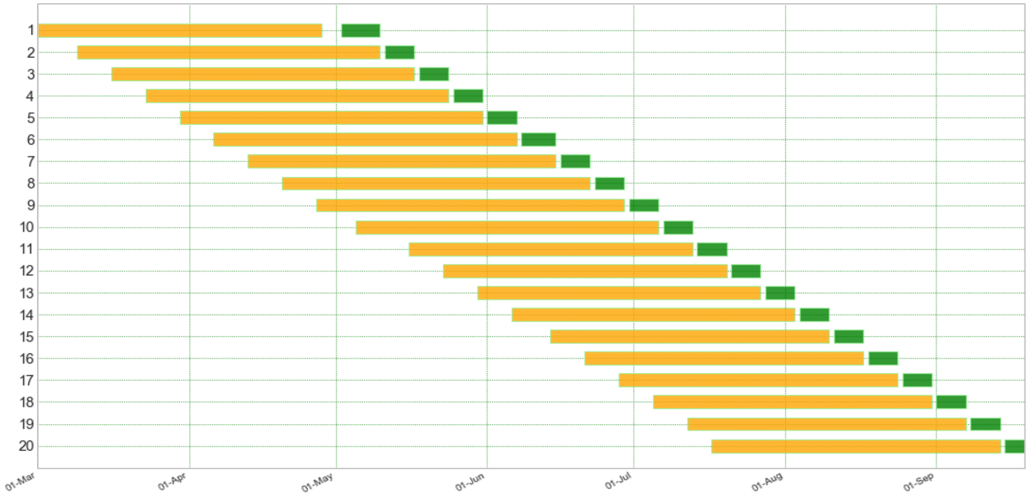
The following common features we use in the forecasting feature space:

- days of the week, months
- months indicators
- weekends / working days
- pre-holiday, holidays, post-holiday days, holidays other than Saturday or Sunday
- last days of the month, last working days of the month
- days of transfers (weekend, which became work and vice versa) and with reduction of working hours
- days of Bank's releases and incidents

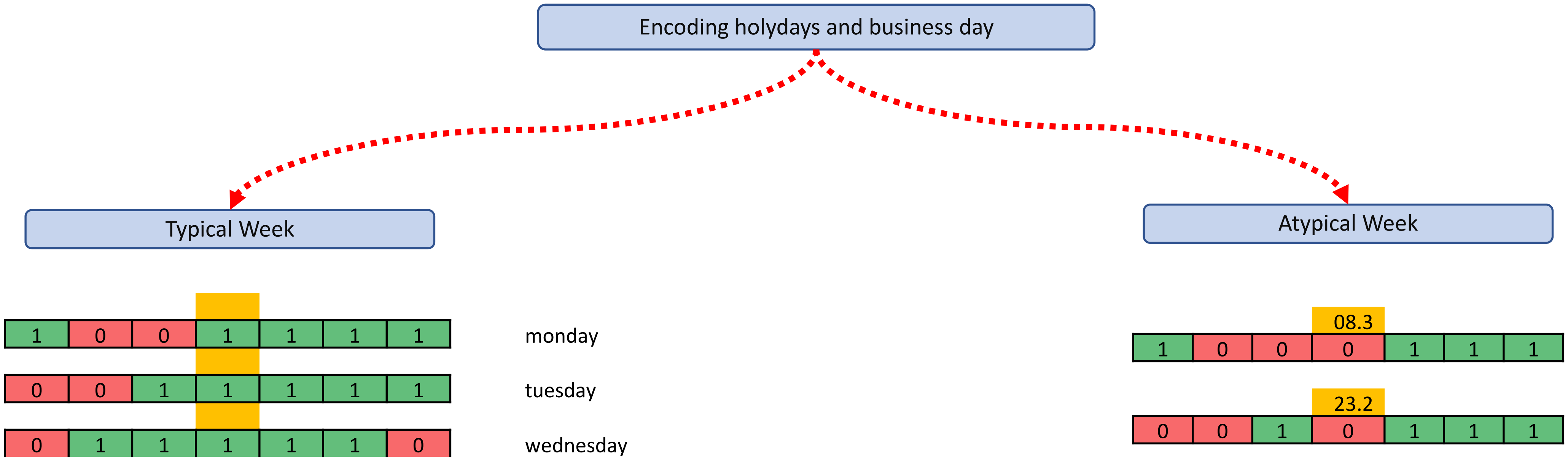
An example of feature space

TimeStamp	Target	One-Hot-Encoding					Lags		Rolling Statistics				One-Hot-Encoding	
	y	пн	вт	ср	чт	пт	lag_1	lag_2	rolling_mean	rolling_std	rolling_mean_weekday	rolling_min_weekday	зарплата 15	праздник
2017-09-08	1273900.0	0	0	0	0	1	1118000.0	896500.0	653100.000000	391381.906412	513666.666667	306200.0	0	0
2017-09-09	344700.0	0	0	0	0	0	1273900.0	1118000.0	726185.714286	457375.871790	143900.000000	72500.0	0	0
2017-09-10	45200.0	0	0	0	0	0	344700.0	1273900.0	746514.285714	432713.001224	36466.666667	6200.0	0	0
2017-09-11	930000.0	1	0	0	0	0	45200.0	344700.0	740328.571429	443853.266723	843400.000000	443700.0	0	0
2017-09-12	43900.0	0	1	0	0	0	930000.0	45200.0	735400.000000	441132.391314	494900.000000	468100.0	0	0
2017-09-13	544300.0	0	0	1	0	0	43900.0	930000.0	664600.000000	511907.166063	546400.000000	319400.0	0	0

Visualization of the cross-validation procedure in a sliding window

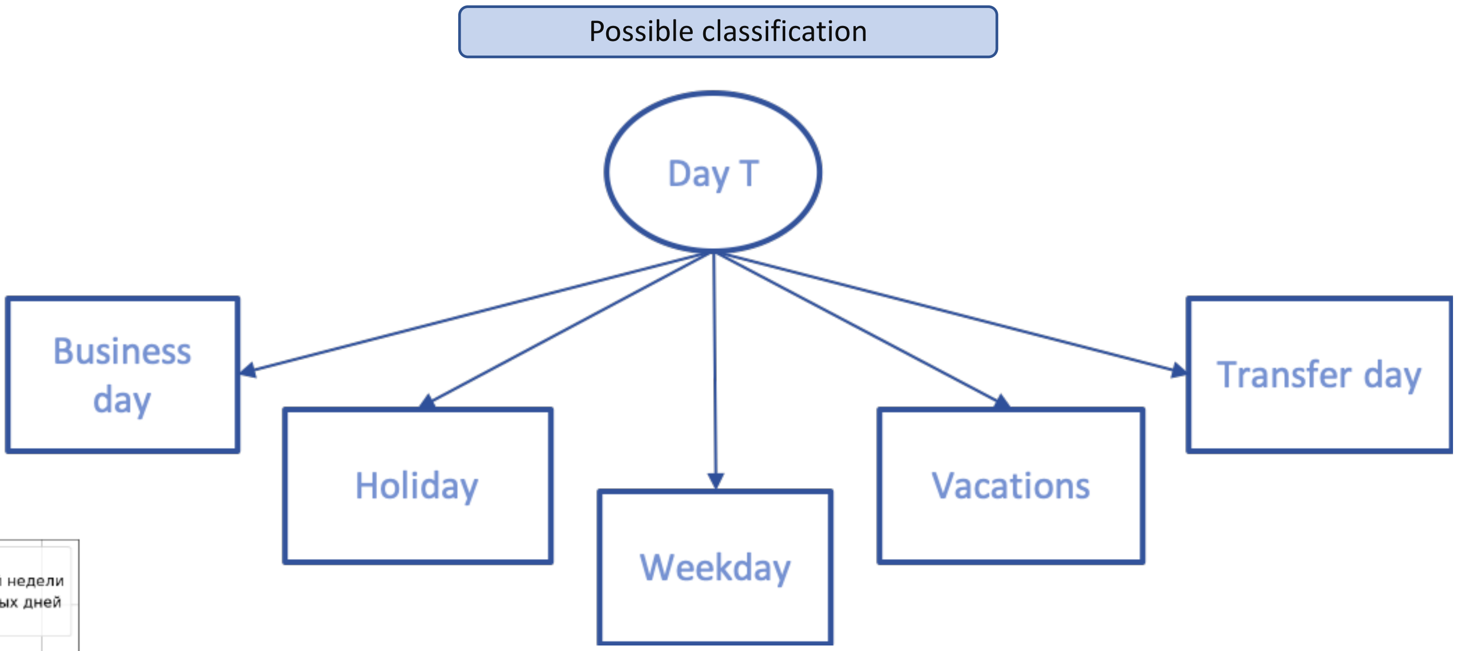


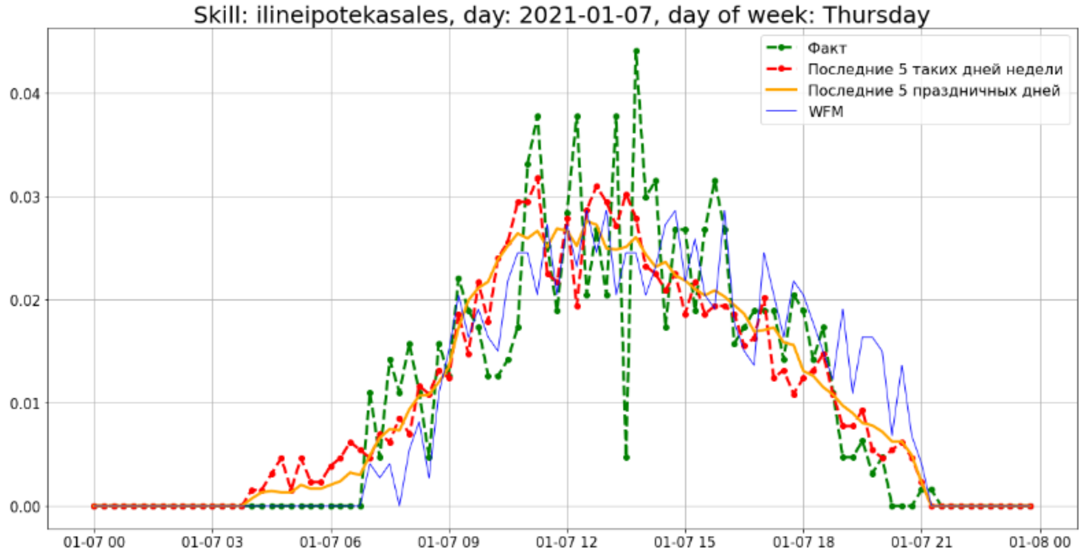
Daily Feature Space

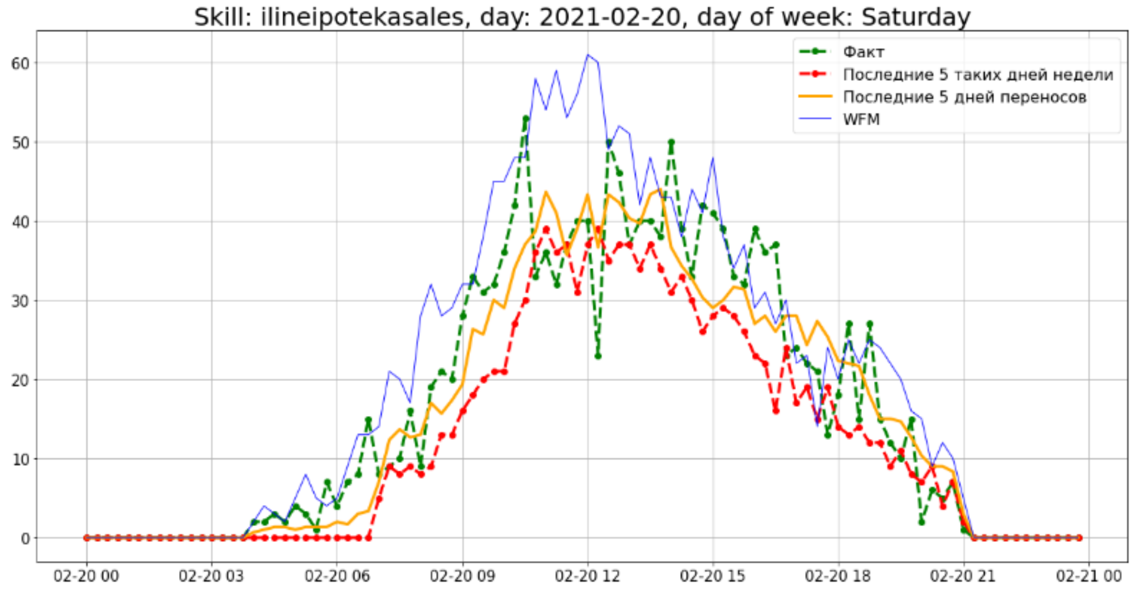


Intraday profiles

To move from daily to intraday call center load, we solve the classification problem. By the nature of intraday load profile, every day can be attributed to one or several clusters



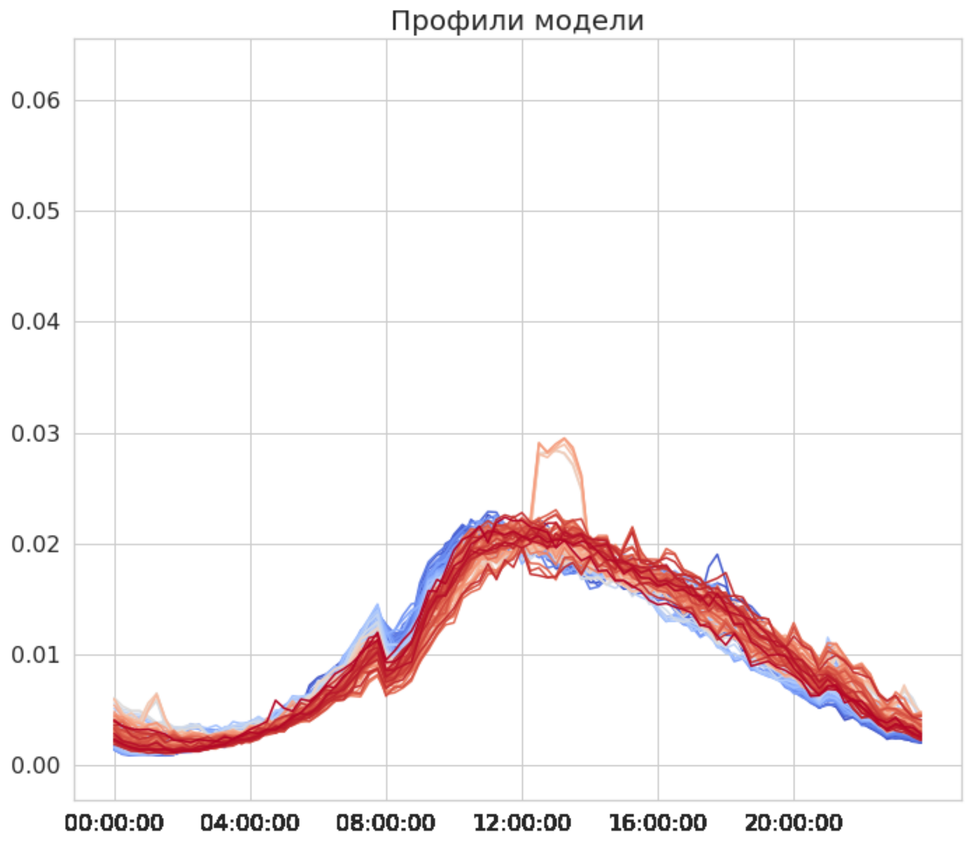




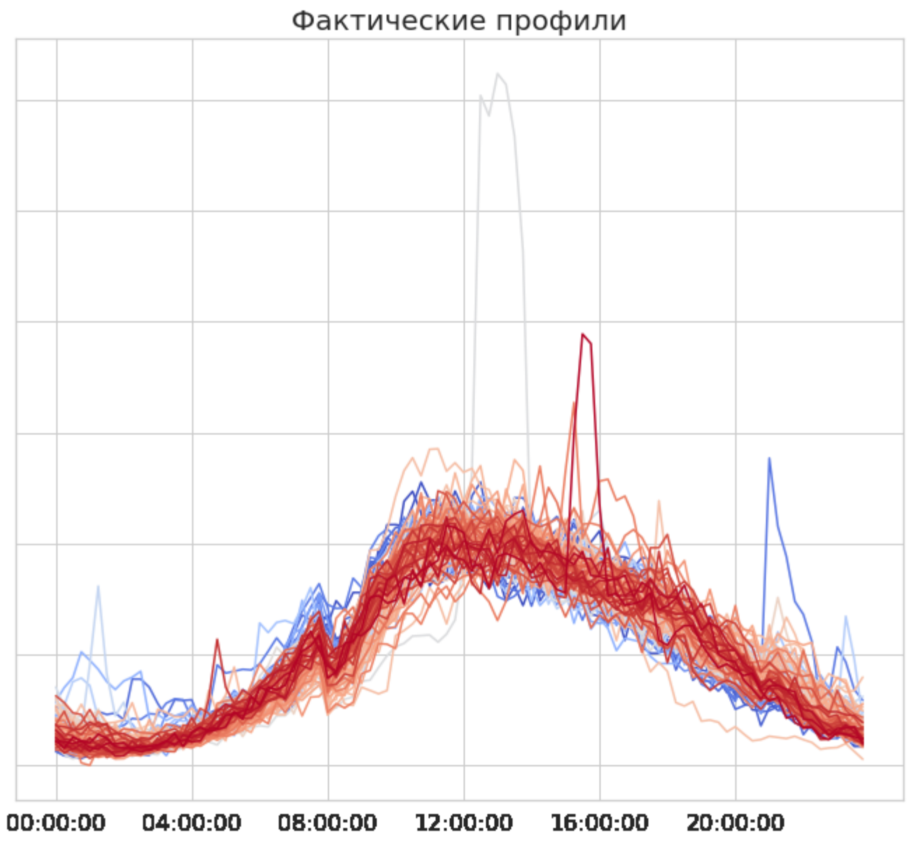
As a forecast for the intraday profile, we use the group's average profile in a sliding window for each call center's selected direction

Intraday profiles

Model profiles

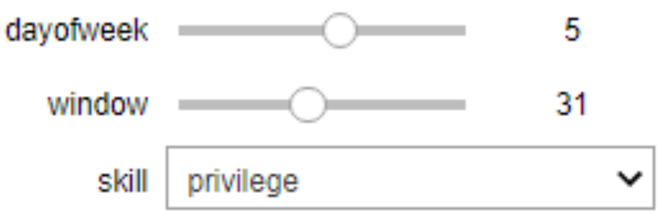


Actual profiles

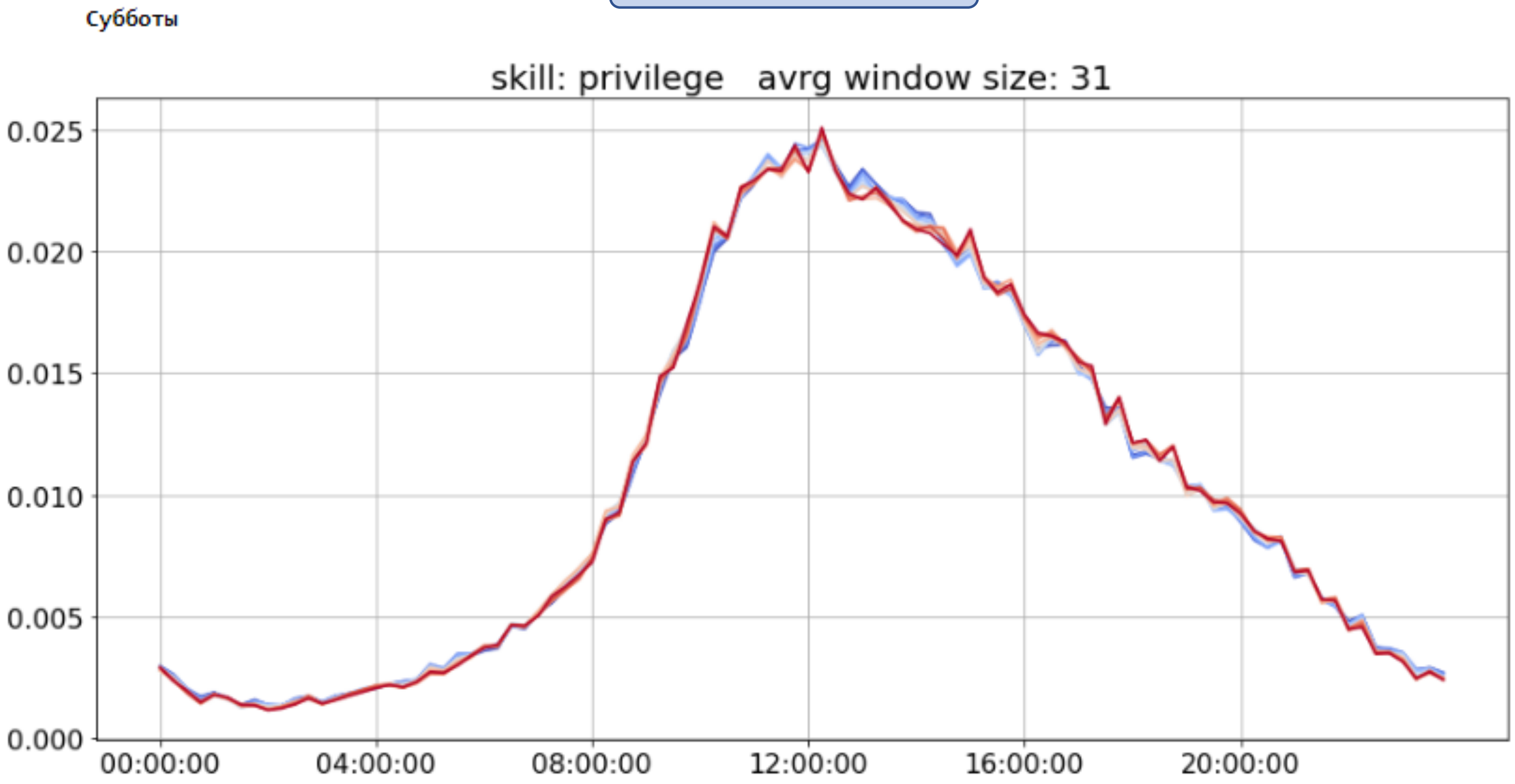


Intraday profiles

Selecting groups of intraday profiles and corresponding window sizes



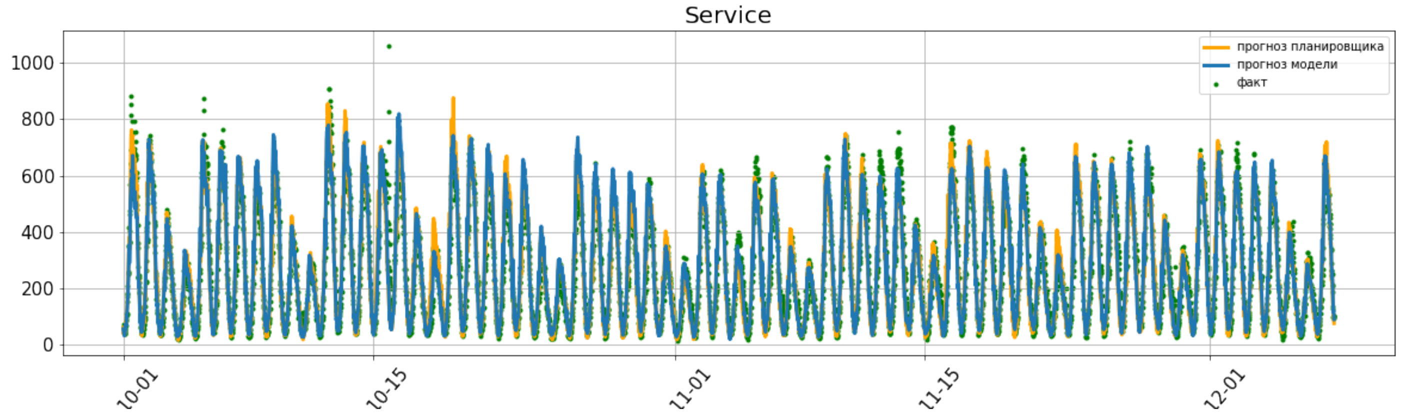
Profile evolution



Results

As mentioned above, we obtain the final forecast by multiplying the daily call center load forecast by the corresponding intraday profile

Final prediction



Comparison of errors (MAE - mean absolute error, MAPE - mean absolute percentage error, WAPE - weighted absolute percentage error) of model forecast and planner forecast errors

Точность планирования нагрузки = $100\% - WAPE_{VOL}$, *где*

$WAPE_{VOL}$ - Weighted Absolute Percent Error (взвешенная абсолютная процентная ошибка прогноза нагрузки)

$$WAPE_{VOL} = \frac{AE_1 + AE_2 + \cdots AE_n}{\sum V_{ACT}} * 100\%$$

$\sum V_{ACT}$– суммарная фактическая нагрузка за оцениваемый период

	skill	MAE	MAE исходной модели	MAPE	MAPE исходной модели	WAPE	WAPE исходной модели	изменение MAPE	изменение WAPE
0	service	76.255	79.594	0.153	0.125	0.11	0.115	0.028	-0.005

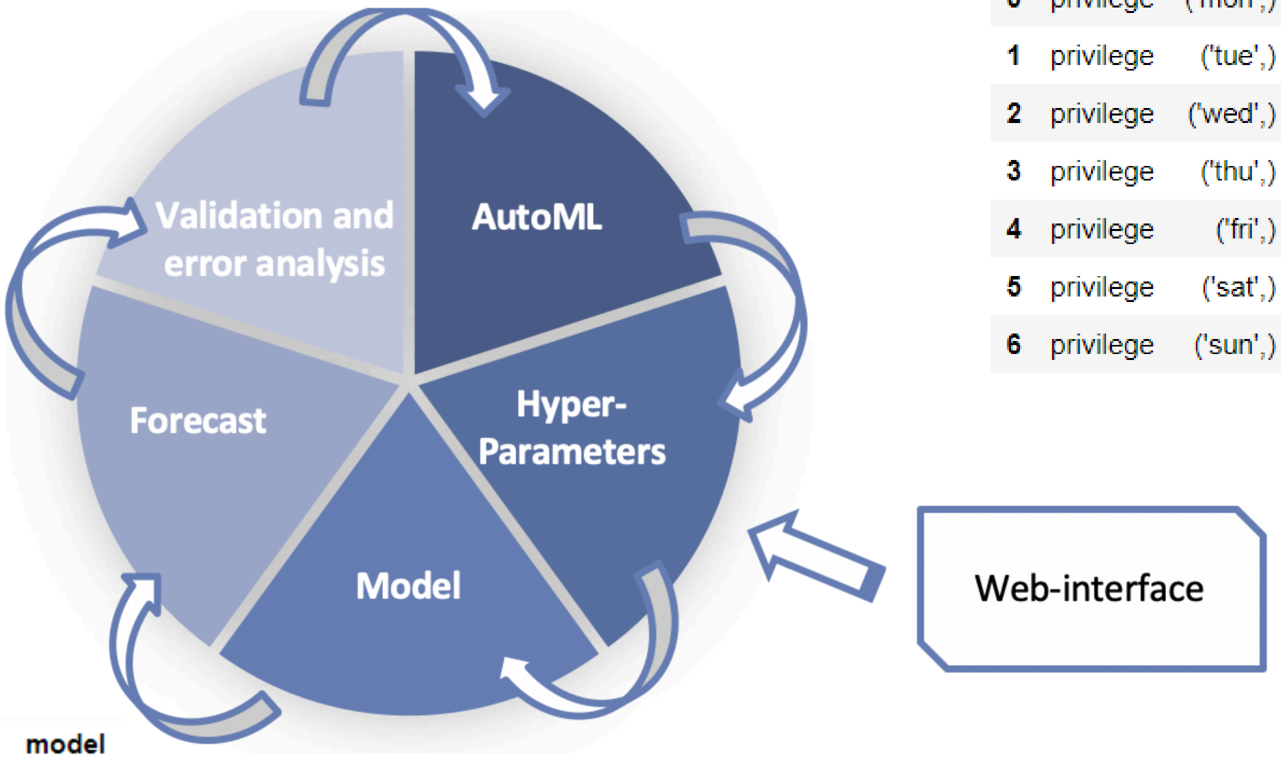
Results

Model vs Planner by WAPE Error

Скилл	Точность	Октябрь	Ноябрь	Декабрь	Квартал	Январь	Февраль
ilineservicebonus	модель - планировщик	1,59	2,45	-0,17	1,26	2,14	2,28
ilineipotekasales	модель - планировщик	5,13	1,82	-1,05	1,97	1,02	4,84
ilineavto	модель - планировщик	NaN	NaN	2,72	2,72	4,63	5,47
ilineipotekasuport	модель - планировщик	0,25	1,75	-1,73	0,074	0,31	11,32
service	модель - планировщик	0,52	4,79	-0,43	1,57	-0,1	0,7
ilinesales	модель - планировщик	-1,23	1,45	1,31	0,48	-2,47	1,45
ilineservicebm	модель - планировщик	2,34	3,6	0,3	2,94	2,73	-0,95
vtbmobile	модель - планировщик	1,05	5,81	1,62	2,75	4,18	0,46
1st line	**модель - планировщик**	0,53	3,73	-0,41	1,24	0,26	1,45
iilineassist1line	модель - планировщик	4,12	3,26	3,35	3,55	2,23	3,8
iilinetelebank	модель - планировщик	2,17	1,18	-0,29	1,02	2,4	3,31
iilineservice	модель - планировщик	8,65	0,93	1,89	3,85	2,05	2,27
2nd line	**модель - планировщик**	3,9	2,08	1,41	2,47	2,23	3,34
vtbplatinum	модель - планировщик	NaN	NaN	NaN	NaN	NaN	NaN
pmsupervisor	модель - планировщик	NaN	NaN	NaN	NaN	NaN	NaN
assistoopk	модель - планировщик	1,23	2,58	3,73	2,48	1,56	3,08
prime	модель - планировщик	0,37	11,78	0,1	3,93	4,91	1,11
privilege	модель - планировщик	-0,69	3,46	1,73	1,47	1,02	2,28
line privilege	**модель - планировщик**	-0,5	4,98	1,69	1,98	2,28	2,27
vtbbroker	**модель - планировщик**	4,58	NaN	6,98	5,78	1,65	-2,6

AutoML

Any machine learning models tend to become outdated since the processes they predict are constantly changing (for example, the coronavirus has seriously affected the load on the call center).



Intraday profiles params

	skill	group	days
0	privilege	('mon',)	10
1	privilege	('tue',)	10
2	privilege	('wed',)	10
3	privilege	('thu',)	10
4	privilege	('fri',)	10
5	privilege	('sat',)	10
6	privilege	('sun',)	10

Hyperparameters

	mod_param_name	param_values	skill	model
0	bootstrap	True	privilege	rfr
1	criterion	mae	privilege	rfr
2	max_depth	NaN	privilege	rfr
3	max_features	auto	privilege	rfr
4	max_leaf_nodes	NaN	privilege	rfr
5	min_impurity_decrease	0	privilege	rfr
6	min_impurity_split	NaN	privilege	rfr

For the models to remain up-to-date, we use an automatic selection of profile groups and averaging windows and model hyperparameters

Thank you for your attention!