Construction of predictive models of meteorological parameters of the atmospheric surface layer

N A Soltaganov¹, V S Sherstnev², A I Sherstneva², I A Botygin⁴, and VA Krutikov⁴

¹ Undergraduate, National Research Tomsk Polytechnic University, Tomsk, Russia
² Assistant Professor, National Research Tomsk Polytechnic University, Tomsk, Russia
³ Senior Researcher, Institute of Monitoring of Climatic and Ecological Systems SB RAS, Tomsk, Russia
⁴ Chief Researcher, Institute of Monitoring of Climatic and Ecological Systems SB RAS, Tomsk, Russia
E-mail: nasoltaganov@yandex.ru, vss@tpu.ru, sherstneva@tpu.ru, bia@tpu.ru, krutikov@imces.ru

Abstract. This paper considers some approaches to building a regression model and a seasonal autoregressive (moving average) integrated model using the Python programming language. The additive regression model was created by using Facebook’s Prophet library. The seasonal integrated autoregressive model was created by using the StatsModels library. We developed a prognostic time series of the monthly precipitation sum for the next 2 years. Program experiments were conducted by using data acquired on a Tomsk station (station synoptic index 29430) with an observation period from 1996 to 2016. An interactive environment called Jupiter Notebook was used for the initial data processing, mathematical calculations, and graph plotting. The environment in question is a graphical web-interface for Python which expands the idea of console approach for interactive computing. The model prediction accuracy was assessed by finding the absolute and average absolute errors. The maximum values of the studied time series could not be predicted.

1. Introduction
Prediction of time series is currently a fairly popular analytical task. The precision of prediction depends on the qualitative characteristics of the time series, the chosen prediction model, the qualification of the expert, the availability of software and hardware required to implement the model algorithms, and other factors. There are multiple different models for the prediction of time series: autoregressive, regressive, exponential smoothing, Markov, neural network models, etc. [1-10]. According to the estimates of researchers, there are already more than a hundred various prediction methods. Therefore, choosing the best method for adequate prediction of the process being studied is the task that must be solved by a specialist.

Since any time series can be viewed as a selective implementation of an infinite set of values obtained by observing the given processes, statistical research prediction models are of most interest for the task, among other formalized time series research methods. In this kind of models the functional dependences (between the current and predicted values or the external factors) are given analytically.

Regression analysis is one of the widely used approaches for statistical modeling. There are linear and nonlinear regression models of various complexity, multiple regression models, additive regressive models, etc. Many researchers note transparency and flexibility of such models, relative ease of program implementation and, consequently, prompt results. Moreover, autoregressive AR(p) models and methods have proven to be quite popular as well, and are widely used for solving time series prediction tasks in various fields [11]. The residual of AR(p) models can be used to design an autoregressive model with conditional heteroskedasticity (GARCH) that has two parameters: p
characterizes the order of autoregression of the squares of residuals, \( q \) is the number of the previous estimates of residuals. The autoregressive model is often used in the financial field to model volatility. Additionally, there are a number of model modifications such as NGARCH, IGARCH, EGARCH, GARCH-M, and others [12, 13].

Autoregression is frequently paired together with a moving average model with \( q \) order, usually denoted by MA(\( q \)). The moving average model is a low-frequency filter, with a number of variations such as simple, cumulative, or exponential forms. Joint use can allow more flexibility in the model fitting. The general form of the model is denoted by ARMA(\( p \),\( q \)) and it combines a moving average filter with \( q \) order and \( p \)-th order of autoregression of filtered values of the process [14, 15].

In the case where the difference between values is used as input data instead, the model is called autoregressive integrated moving average [16] denoted by ARIMA(\( p \),\( d \),\( q \)). If the autoregressive model for the prediction of a process resulting from the MA(\( q \)) model also contains extra regressors for the external factors, the ARIMAX model is used instead. A seasonal integrated autoregressive moving average model called SARIMA has proven to be quite popular as well [17]. SARIMA’s input parameters are \( p \),\( p_s \),\( q \),\( ps \),\( Ds \), and \( Qs \). The primary parameters are: \( p \) is the autoregression order, \( d \) is the difference order, and \( q \) is the moving average order. The seasonal parameters include: \( p_s \), the seasonal autoregression order, \( Ds \), the seasonal difference order, and \( Qs \), the seasonal moving average order.

Since the experiments utilize series of yearly observations of the atmospheric precipitation, the seasonal component is the most significant component for modeling, directly influencing the prediction accuracy. In our case the studied data series have a seasonality index of 12 (yearly).

The following work is devoted to design and research of prediction models based on an additive regressive model and a seasonal integrated moving average model.

2. Research and program experiments on construction of prediction models

The goal of the experiments was to study and develop an algorithm for predicting meteorological fields. In particular, we made a prognosis of a time series of the monthly precipitation sum for the next 2 years. The data files were acquired from the federal state budget institution “RIHMI-WDC” [18]. The program experiments were conducted for data gathered at the Tomsk station (synoptic index: 29430), observation period: from 1966 to 2016. The programming language Python [19] and a statsmodels library [20] were used in the software development. The interactive environment Jupyter Notebook [21] was used for writing the source code and its testing. It acts as a web-interface for Python and expands the idea of a console approach to interactive computation.

2.1. Stationarity test

Work with time models requires knowing whether the studied time series are stationary or not. It is important, since stationary and nonstationary time series display different statistical properties and should be assessed with different approaches.

For the stationarity test we used the DF-test (Dickey-Fuller test) that checks the existence of a unit root. The following program code was used:

```python
sm.tsa.stattools.adfuller(data['num_data'])[1]
sm.tsa.seasonal_decompose(data['num_data']).plot()
print("Критерий Дики-Фуллера: p=%f" % sm.tsa.stattools.adfuller(data['num_data'])[1])
```

The row is stationary, since the Dickey-Fuller criterion rejects the nonstationarity hypothesis. Figure 1 shows the Dickey-Fuller test graphs, and all of them prove the nonstationarity. In particular, there is no tendency of time series change, the values fluctuate around zero, and the Dickey-Fuller criterion \( p=0.000000 \)
2.2. Additive regressive model

It is a generalized linear model with a specific set of extra factors to calculate the seasonality and account for important short-term changes (anomalies):

\[ y(t) = g(t) + s(t) + h(t) + \epsilon_t, \]

where \( y(t) \) is the predicted time series value; \( g(t) \) is the piecewise linear or logistic function (trend); \( s(t) \) is the component that reflects the modeling of periodic changes (seasonal component); \( h(t) \) is the component that reflects the modeling of anomalous days and other irregular events; and \( \epsilon_t \) is the component that is not accounted for in the model (error).

Figure 2 below shows the modeled results for the precipitation time series from 2013 to 2016. In the experiments on constructing the prediction model we used arrays containing 8-period data of atmospheric precipitation observations. In particular, the models used the precipitation sum for the period between standard synoptic times with a 3-hour interval.

2016 was chosen as the year for prediction. The initial data series was processed; emissions not relevant to seasonal fluctuations were removed. Additionally, for the observation station with index 29430, we modified the maximum precipitation sum between the dates according to the values recommended by the All-Russian Research Institute of Hydrometeorological Information - World Data Center (RIHMI-WDC) [18].
The graph shows the source data, predicted values, and prediction boundaries (upper and lower).

2.3. Seasonal integrated model

The autoregressive moving average model (ARIMA) of p-th order and q-th order of moving average is defined as follows:

\[ y(t) = c + \alpha_1 y_{t-1} + \ldots + \alpha_p y_{t-p} + \beta_1 e_{t-1} + \ldots + \beta_q e_{t-q} + \epsilon_t, \]

where \( y(t) \) is the predicted time series value; \( c \) is the constant value; \( \alpha_1, \ldots, \alpha_p \) are the parameters of the autoregression order; \( \beta_1, \ldots, \beta_q \) are the parameters of the moving average order; \( y_{t-1}, \ldots, y_{t-p} \) are the previous values of the time series offset by lag ranging from 1 to \( p \); and \( e_{t-1}, \ldots, e_{t-q} \) are the previous errors in the model offset by lag from 1 to \( q \). While ARIMA-modeling requires finding a correct ARIMA(p,d,q) specification, constructing the seasonal integrated autoregressive moving average model (SARIMA) requires finding a specification that accounts for the seasonal components SARIMA(p,d,q) (Ps,Ds,Qs), where \( s \) is the seasonality size.

The SARIMA model type is determined by a correlogram. A correlogram is a graphical representation of the autocorrelation function. Creating correlograms required using plot_acf() and plot_pacf() functions in the statsmodels library.

```python
ax = plt.subplot(211)
sm.graphics.tsa.plot_acf(data['num_data'][13:].values.squeeze(), lags=58, ax=ax)
ax = plt.subplot(212)
sm.graphics.tsa.plot_pacf(data['num_data'][13:].values.squeeze(), lags=58, ax=ax);
```

In Figure 3, light-blue color in the autocorrelation graphs denotes the critical interval where ACF and PACF values are considered equal to zero. The autoregression model is indicated by emissions during first few lags in the private autocorrelation function, PACF. Emission during the 12-th lag in the private autocorrelation function shows the 12-season seasonality.
Therefore, it is required to build a seasonal integrated autoregressive moving average model with the following input parameters: p, autoregression order, d, initial row differences order (integration order), q, moving average order (model order), Ps, the order of the seasonal component of autoregression, Ds, the order of the seasonal difference (the seasonal component integration order), Qs, the seasonal parameter of moving average. Let us define the initial approximations to build our model according to the correlogram: Qs = 5, q = 12, Ps = 5, p = 2. The integration order d and the integration order of the seasonal component Ds will be equal to 1.

```python
%time
results = []
best_aic = float("inf")
for param in parameters_list:
    try:
        model = sm.tsa.statespace.SARIMAX(data['num_data'], order=(param[0], d, param[1]),
                                          seasonal_order=(param[2], D, param[3], 12)).fit(disp=-1)
    except ValueError:
        print('wrong parameters:', param)
        continue
    aic = model.aic
    if aic < best_aic:
        best_model = model
        best_aic = aic
        best_param = param
    results.append((param, model.aic))
warnings.filterwarnings('default')
```

Figure 3. ACF and PACF graphs.
Figure 4 shows the best prediction model over the period of time. The blue line shows the actual values, and the red line shows the modeled values. It is evident how well the model describes the data. However, maximum values could not be predicted.

To build a prediction graph for the next two years (without any factual data for this period of time), it is necessary to execute the following program code:

```python
from dateutil.relativedelta import relativedelta

data2 = data[['num_data']]

date_list = [pd.datetime.strptime("2016-01-01", "%Y-%m-%d") + relativedelta(months=x) for x in range(0,24)]

future = pd.DataFrame(index=date_list, columns=data2.columns)
data2 = pd.concat([data2, future])
data2['forecast'] = best_model.predict(start=600, end=700)
data2['num_data'].plot(color='b')
data2['forecast'].plot(color='r')
plt.ylabel('осадки');
```
The result is shown in Figure 5.

![Figure 5. Forecast for the next two years.](image)

As with the previous graphs, we can see how close are the modeled values to the expected value range.

To analyze the results of the forecast calculation, the following errors were calculated:

MAPE (mean absolute percentage error), this is the average absolute error of the forecast.

\[
t_{\text{MAPE}} = \frac{\sum |p_t - y_t|}{n}
\]

where: \(y_t\) is an indicator, and \(\hat{y}_t\) is the forecast of the model corresponding to this value. Then \(e_t = \hat{y}_t - y_t\) is the forecast error, and \(p_t = \frac{\hat{y}_t}{y_t}\) is the relative forecast error.

It is also useful to know the absolute error MAE (mean absolute error) in order to understand how much the model is mistaken in the absolute values.

\[
t_{\text{MAE}} = \frac{\sum |e_t|}{n}
\]

Indicators of the prediction errors in the time series of the model SARIMA: MAPE = 26.79%, MAE = 8.49. Prophet: MAPE = 37.35%, MAE = 10.62.

3. Conclusions
The experiments performed above have shown that in the case of stationary time series the SARIMA prediction model is more accurate compared to the Prophet prediction model. Since the error values can be calculated, these prediction models should only be used for long-term predictions; currently their accuracy is approximately 70%. However, using autoregressive models, as well as their generalizations, may cause some difficulties due to daily, weekly, or other cycles and some internal effects due to the dependences between the members in the data series. Increasing the averaging interval does not produce acceptable results, since a dependence present in one sample might not be present in another sample of the same size that belongs to a different period of time.
averaging over a certain time interval and using average correlograms for the observation period decreases the prediction accuracy even in short periods of time. Consequently, such a task solution is completely impractical for a long period of time. The thus obtained predictions can be used as a new attribute for other processing algorithms. The methods studied for time series value prediction can be used to supplement the existing standard methods (synoptic method, hydrodynamic method). This helps to determine the anomalies in the prediction models by comparison with the SARIMA and Prophet mathematical models.

Acknowledgments
This research was funded by the Russian Foundation for Basic Research and the government of the Tomsk region of the Russian Federation, grant no. 18-47-700005p_a.

References
[1] Khromov S P, Petrosyants M A 2001 Meteorology and Climatology: a textbook for universities (Moscow: Moscow State University Press) p 528
[2] Draper N, Smith G 2016 Applied regression analysis (Moscow: Finance and Statistics) p 912
[3] Shashkov V B 2003 Applied regression analysis. Multifactorial regression (Orenburg: Orenburg State University Press) p 363
[4] Sadovnikova N A, Shmoylova R A 2016 Time series analysis and forecasting (Moscow: Publishing house of the University "Synergy") p 152
[5] Terekhin D A, Botygin I A, Sherstneva A I, and Sherstnev V S 2017 Journal of Physics: Conference Series 803 4
[6] Afanasyev V N, Yuzbashev M M 2010 Time series analysis and forecasting (Moscow: Finance and Statistics, Infra-M) p 320
[7] Lukashin Yu P 2003 Adaptive methods of short-term forecasting of time series (Moscow: Finance and Statistics) p 416
[8] Tartakovsky V A, Botygin I A, Sherstneva A I 2017 Proc. 23rd Int. Symp. Atmospheric and Ocean Optics: Atmospheric Physics: Proc. of SPIE 10466 104665J
[9] Sergienko A B 2006 Digital signal processing (St. Petersburg: Peter) p 751
[10] Kizbikenov K O 2017 Forecasting and time series: a textbook (Barnaul: Altai State Pedagogical University Press) p 113
[11] Time series models: AR (P), MA (Q), ARIMA (P, D, Q) http://economy.bsu.by/wp-content/uploads/2016/10/nafi_ar_ma.pdf Accessed: May 2018
[12] Garcia F, Guijarro F, Moya I, Oliver J 2012 Int. J. Complex Systems in Science 21-26
[13] Model GARCH https://basegroup.ru/community/glossary/garch-mod Accessed: May 2018
[14] ARMA Models http://www.econ.ohio-state.edu/dejong/note2.pdf Accessed: May 2018
[15] Introduction to ARMA Models http://www-stat.wharton.upenn.edu/~stine/stat910/lectures/08_intro_arima.pdf Accessed: May 2018
[16] Applied problems of data analysis https://www.coursera.org/learn/data-analysis-applications/lecture/8yR4G/arima Accessed: May 2018
[17] Seasonal ARIMA with Python http://www.seanabu.com/2016/03/22/time-series-seasonal-ARIMA-model-in-python/ Accessed: May 2018
[18] All-Russian Research Institute of Hydrometeorological Information - World Data Center (RIHMI-WDC) http://meteo.ru/data Accessed: May 2018
[19] Python https://www.python.org/ Accessed: May 2018
[20] StatsModels http://www.statsmodels.org/stable/ Accessed: May 2018
[21] Project Jupyter http://jupyter.org/ Accessed: May 2018