

Способы прогнозирования параметров киберфизических систем

Д. П. Плахотников

Санкт-Петербургский государственный электротехнический университет
«ЛЭТИ» им. В.И. Ульянова (Ленина)
dimapl21@yandex.ru

Аннотация. Киберфизические системы являются передовыми на сегодняшний день. Они представляют собой интеграцию физических объектов и информационных систем с помощью различного рода датчиков и сенсоров. Полученная информация с данных систем может быть использована для прогнозирования будущего состояния системы, что позволяет предугадать нагрузку или выявить её снижение. Это позволяет принимать решение о расширении мощностей системы и выявлять какие-либо проблемы при эксплуатации киберфизической системы.

Ключевые слова: киберфизические системы; прогнозирование; большие данные; аналитические платформы

I. ВВЕДЕНИЕ

В отличие от уже существующих встраиваемых систем, киберфизическая система разрабатывалась как сеть взаимосвязанных элементов с физическими интерфейсами входа и выхода, а не как отдельные автономные устройства. Текущий прогресс науки позволяет улучшить данную связь физических и вычислительных элементов с помощью интеллектуальных механизмов, повышая безопасность, функциональность, эффективность и другие важные параметры киберфизических систем [1].

Киберфизические системы в процессе своей работы генерируют большое количество информации, такой как данные с датчиков, сенсоров, систем контроля и др. Полученные данные в дальнейшем могут быть использованы для их прогнозирования, чтобы предугадать будущую нагрузку на киберфизическую систему или предотвратить возможные технические проблемы.

II. АРХИТЕКТУРА КИБЕРФИЗИЧЕСКОЙ СИСТЕМЫ

Главными частями каждой киберфизической системы являются:

- физический слой (различные объекты реального мира);
- цифровой слой (множество данных о системе – алгоритмы управления реальными объектами, алгоритмы обработки данных и т. д.);
- интерфейс взаимодействия цифрового и физического слоёв (датчики, управляющие механизмы и т. д.);

- интерфейс взаимодействия цифрового и физического слоя с человеком (визуализация, расширенная реальность и т. д.).

Концептуальная схема представлена на рис. 1.



Рис. 1. Концептуальная схема киберфизических систем

Данные части взаимодействуют между собой, образуя единую экосистему, направленную на решение определённой задачи.

Считается, что киберфизическая система является следующей ступенью развития систем с большим масштабом гранулярности. Иными словами, киберфизическая система сама состоит из множества других сложных систем [2].

III. ПОЛУЧЕНИЕ ИНФОРМАЦИИ С КИБЕРФИЗИЧЕСКОЙ СИСТЕМЫ

В качестве киберфизической системы была использована сеть территориально распределённых газозаправочных станций. Каждая станция является киберфизической системой – она содержит датчики утечки, интерфейс взаимодействия с колонкой. Колонка, через которую происходит отпуск топлива, тоже является киберфизической системой, которая содержит датчики плотности, температуры, утечки, объёма и другие. Данные с колонки с помощью специального контроллера передаются на станцию [3].

Между собой данные станции не соединены, а информация с них собирается и передаётся в единый информационный центр через корпоративную частную сеть. Схематически это представлено на рис. 2.

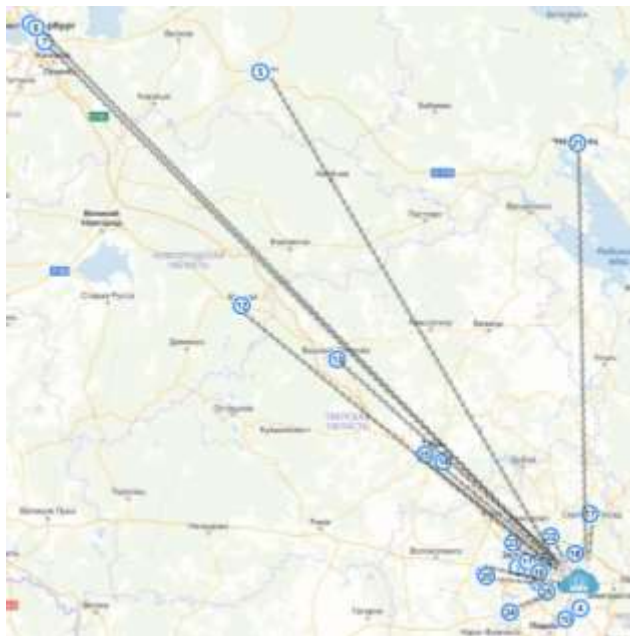


Рис. 2. Схема получения информации со станций

В едином информационном центре развёрнута база данных. Для получения данных из неё был написан запрос в аналитическую систему. Пример такого запроса представлен ниже:

```
TRANSINF:
SELECT
    CAST(TRANSID AS varchar2(50)) as TRANSID,
    TRANSDATETIME,
    OILSUPPLIED,
    CAST(PAYMENTTYPE AS varchar2(50)) as PAYMENTTYPE
FROM IC.TRANSINF
WHERE TRANSDATETIME>=TO_TIMESTAMP('01.01.2021
00:00:00', 'DD.MM.YYYY HH24:MI:SS');
```

```
TRANSADDDATA:
SELECT
    CAST(a1.TRANSID AS varchar2(50)) as TRANSID,
    a1.STARTPISTOLCOUNTER as STARTPISTOLCOUNTER,
    a1.ENDPISTOLCOUNTER as ENDPISOTLCOUNTER,
    a1.STARTFILLINGDT as STARTFILLINGDATETIME,
    a1.ENDFILLINGDT as ENDFILLINGDATETIME,
FROM IC.TRANSADDDATA a1 INNER JOIN IC.TRANSINF a2 on
a1.TRANSID=a2.TRANSID
WHERE a2.TRANSDATETIME>=TO_TIMESTAMP('01.01.2021
00:00:00', 'DD.MM.YYYY HH24:MI:SS');
```

С помощью данного запроса можно получить информацию об отпуске топлива за 2021 год, содержащую идентификатор транзакции (TRANSID), время транзакции (TRANSDATETIME), объём отпуска топлива (OILSUPPLIED), тип оплаты (PAYMENTTYPE). В отдельной таблице дополнительной информации о транзакции (TRANSADDDATA) представлена информация о идентификаторе транзакции (TRANSID) который используется для связи этих двух таблиц, данные о стартовой позиции счётчика (STARTPISTOLCOUNTER), конечной позиции счётчика (ENDPISTOLCOUNTER), времени начала заправки (STARTFILLINGDATETIME) и времени конца заправки (ENDFILLINGDATETIME).

Данные разбиты на разные таблицы, поскольку в первую очередь отправляется оперативная информация для бухгалтерского и налогового учета, а после, при отсутствии загрузки линии передачи данных, техническая информация.

После загрузки информации она была обработана и преобразована в текстовый формат для хранения табличных данных csv. Способы прогнозирования полученных данных, представлены в следующей главе.

IV. ПРОГНОЗИРОВАНИЕ ПАРАМЕТРОВ С ПОМОЩЬЮ АНАЛИТИЧЕСКОЙ СИСТЕМЫ

Одними из инструментов обработки больших данных являются системы анализа данных Business Intelligence. Для выбора системы, был изучен отчет исследовательской и консалтинговой компании в сфере информационных технологий Gartner. Она каждый год выпускает актуальные требования для аналитических систем, анализирует их сильные и слабые стороны и издаёт свой рейтинг таких компаний. Согласно ее анализу, лидерами аналитических систем являются три компании – Qlik, Microsoft и Tableau [4]. Из данных лидеров была выбрана компания Qlik и её продукт Qlik Sense, поскольку он имеет возможность бесплатного пробного доступа.

После установки платформы было установлено расширение на стороне сервера (SSE), построенное с использованием Python. Его цель состоит в том, чтобы предоставить набор функций прогнозирования для обработки данных, которые можно использовать в качестве выражений в аналитической платформе.

После установки расширения было настроено аналитическое соединение. Настройки представлены на рис. 3.

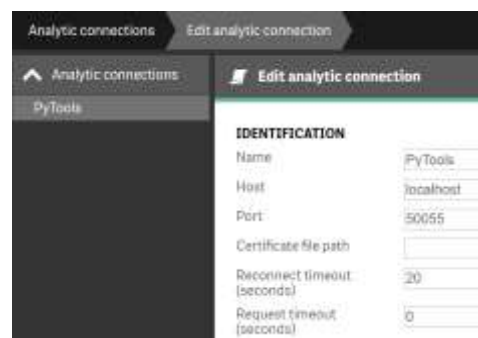


Рис. 3. Настройка аналитического соединения

В качестве прогнозных параметров были выбраны следующие: количество транзакций за день, средний объём отпуска и медианная длительность отпуска. Для этой задачи был использован компонент Facebook Prophet.

Prophet – это инструмент прогнозирования параметров временных рядов на основе аддитивной модели, в которой нелинейные тенденции соответствуют годовой, еженедельной и ежедневной сезонности, а также эффектам праздников или выходных дней. Он лучше всего работает с временными рядами, которые имеют сильные сезонные эффекты и несколько сезонов исторических данных.

Prophet устойчив к отсутствующим данным и изменениям в тренде и, как правило, хорошо справляется с выбросами [5].

Этот инструмент отлично подходит, поскольку отпуск топлива сильно зависит от сезона и типа дня (рабочий или выходной). Для использования данного инструмента, вначале в аналитическое приложение загружаются данные, содержащие дату (Date) счётчик количества (Count), время отпуска (Duration), объём отпуска (OilSupplied). Далее эти данные группируются по дням в соответствии с нужной функцией (количество, среднее или медиана). После загрузки данных для прогноза используется специальная формула:

```
PyTools.Prophet(Date,Parameter,'freq=D,
take_log=true')
```

Формула состоит из:

- PyTools – название аналитического соединения;
- Prophet – название инструмента;
- Date – дата;
- Parameter – параметр для прогнозирования;
- freq=D – частота данных (frequency) равняется дню (day);
- take_log=true – включается логгирование данных.

В результате были полученные прогнозируемые значения до конца текущего года. Таблица прогнозируемых значений представлена на рис. 4.

Дата	Q	Прогнозируемые значения
07.12.2021		77 341
08.12.2021		77 818
09.12.2021		78 343
10.12.2021		78 838
11.12.2021		68 234

Рис. 4. Таблица прогнозируемых значений транзакций

По этим данным был построен график прогноза количества транзакций за день, представленный на рис. 5. Ось X – дата, ось Y – количество транзакций, первая линия – реальные данные, вторая линия – прогнозируемые данные.



Рис. 5. График прогноза количества транзакций в день

Так же были выполнены прогнозы остальных параметров. Результат представлен на рис. 6.



Рис. 6. График прогноза среднего отпуска и таблица прогнозируемых значений медианных значений длительности отпуска

Благодаря проведённому прогнозированию параметров киберфизической системы, можно сделать следующие выводы:

- прогнозируется повышение среднего количества транзакций с 68921 в день в первой половине 2021 года до 76163 во второй половине 2021 (на 10,5 %);
- ожидается повышение максимального количества транзакций за день с 81326 транзакций в день (в 1 половине 2021) до 85806 транзакций в день (во 2 половине 2021);
- медиана длительности отпуска повысится с 4 минут 15 секунд в первой половине 2021 до 4 минут 30 секунд во второй половине 2021;
- средний объём отпуска повысится с 35,24 м³ во 2 половину 2020 года до 36,72 м³ во 2 половину 2021 года.

Помимо этого, в аналитическое приложение были добавлены фильтры по станции и колонке, что позволяет провести анализ не только по всей сети заправочных станций, но и на конкретной станции и на конкретной колонке (или нескольких колонках). Демонстрация работы фильтров представлена на рис. 7.



Рис. 7. Дополнительные фильтры по станции и колонке

V. ПРОГНОЗИРОВАНИЕ ПАРАМЕТРОВ С ПОМОЩЬЮ PYTHON

Ещё одним способом прогнозирования параметров является использование программ, написанных на языке программирования Python. Согласно portalу Data Science Central, Python является лучшим языком для науки о данных и в машинном обучении [6].

На Python существует огромное множество методов прогнозирования данных, в качестве примера будет использована модель ARIMA («Autoregressive Integrated Moving average»).

Это распространенная модель временных рядов, которая очень популярна среди ученых. В то время как модели экспоненциального сглаживания основываются на описании тренда и сезонности в данных, модели ARIMA пытаются описать корреляции в данных между собой. Есть ещё более совершенная модель – это Seasonal ARIMA, поскольку она учитывает сезонность данных [7]. Именно данная модель и будет использована для прогнозирования.

Вначале необходимо загрузить необходимые библиотеки:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import statsmodels.api as sm
```

Библиотека «pandas» нужна для обработки и анализа данных, «numpy» для математических и числовых операций, «matplotlib» для построения графиков, а «statsmodels» содержит необходимую модель.

Далее необходимо загрузить набор данных с помощью формулы: `df = pd.read_csv(r'dataset.csv', delimiter=',')`

После загрузки данных выполняется создание обучающего набора и агрегирование набора данных по дню следующими строками:

```
train.Timestamp =
pd.to_datetime(train.Datetime, format='%d-%m-%Y
%H:%M')
train.index = train.Timestamp
train = train.resample('D').mean()
```

Далее выполняется прогнозирование и строится график:

```
y_hat_avg = test.copy()
fit1 = sm.tsa.statespace.SARIMAX(train.Count,
order = (2,1,4), seasonal_order
=(0,1,1,7)).fit()
y_hat_avg['SARIMA'] =
fit1.predict(start="2021-07-01", end="2021-12-
31", dynamic=True)
plt.figure(figsize=(16,12))
plt.plot(train['Count'], label = "Фактические
данные")
plt.plot(y_hat_avg['SARIMA'], label = "SARIMA")
plt.legend(loc = "best")
plt.title("Прогнозирование количества
транзакций в день")
```

Построенный график представлен на рис. 8.

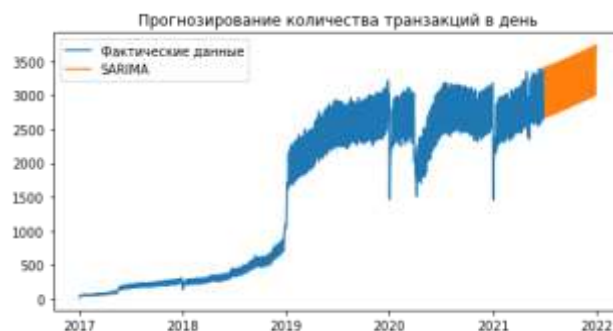


Рис. 8. График прогнозирования количества транзакций в день

VI. ЗАКЛЮЧЕНИЕ

Прогнозирование параметров киберфизических систем позволяет предугадать будущее состояние киберфизической системы, а полученные данные могут служить основой для принятия решений. В докладе было продемонстрировано два различных способа прогнозирования – с использованием аналитической системы и с написанием программы на Python.

Преимущество использования аналитической системы заключается в простоте получения данных, их обработки, построения визуализаций, использования фильтров. Недостатки данного способа заключаются в том, что необходимо приобретать лицензии для использования в коммерческих целях и отсутствие гибкости в выборе инструмента прогнозирования. В свою очередь, преимущество разработки программы для прогнозирования параметров с помощью Python заключается в возможности широкого выбора методов прогнозирования, отсутствии дополнительных затрат, но данный способ требует более высоких компетенций и трудозатрат.

СПИСОК ЛИТЕРАТУРЫ

- [1] C. Alippi: Intelligence for Embedded Systems. Springer Verlag, 2014, 283pp, ISBN 978-3-319-05278-6.
- [2] Plakhotnikov D.P. and Kotova E.E. "The Use of Artificial Intelligence in Cyber-Physical Systems," // 2020 XXIII International Conference on Soft Computing and Measurements (SCM), 2020, pp. 238-241, doi: 10.1109/SCM50615.2020.9198749.
- [3] Plakhotnikov D.P. and Kotova E.E. "Design and Analysis of Cyber-Physical Systems," 2021 IEEE Conference of Russian Young Researchers in Electrical and Electronic Engineering (ElConRus), 2021, pp. 589-593, doi: 10.1109/ElConRus51938.2021.9396364.
- [4] 2021 Gartner Magic Quadrant for analytics and business intelligence platforms // Sisense. URL: <https://www.sisense.com/gartner-magic-quadrant-business-intelligence/> (дата обращения: 03.07.2021).
- [5] Prophet: Automatic Forecasting Procedure [Электронный ресурс]. – Интернет-сайт. – URL: <https://github.com/facebook/prophet> (дата обращения: 03.07.2021).
- [6] Top-Notch Programming Languages for Data Science & Machine Learning // Data Science Central. URL: <https://www.datasciencecentral.com/profiles/blogs/9-top-notch-programming-languages-for-data-science-amp-machine> (дата обращения: 03.07.2021).
- [7] A comprehensive beginner's guide to create a Time Series Forecast [Электронный ресурс]. Интернет-сайт. URL: <https://www.analyticsvidhya.com/blog/2016/02/time-series-forecasting-codes-python/> (дата обращения: 03.07.2021).