Инженерный вестник Дона, №1 (2023)
ivdon.ru/ru/magazine/archive/n1y2023/8123

Система прогнозирования потребления электроэнергии на пищевом производстве на основе потоковых данных

А.А. Абузяров, А.А. Макаров

Российский государственный университет имени А.Н. Косыгина (Технологии. Дизайн. Искусство)

Аннотация: Целью данной работы является реализация системы прогнозирования потребления электроэнергии на пищевом производстве и выбор наиболее подходящего метода обучения модели прогнозирования. В нашем исследовании была реализована система прогнозирования потребления электроэнергии на основе потоковых данных, получающая их в «реальном времени». Система создана по принципу микросервисной архитектуры, где были реализованы сервис сбора данных с счетчиков, сервис агрегации данных и сервисы прогнозирования: с использованием классического подхода к обучению на основе модели ARIMA и онлайн- подхода к обучению с использованием онлайн-модели HATR, результаты работы которых были сравнены с помощью тестов на прогнозирование аномальных значений и прогнозирование в условиях смены концепта данных, или дрейфа концепций.
Ключевые слова: машинное обучение, онлайн-обучение, онлайн-модель, дрейф концепций, дрейф данных.

На данный момент, прогнозирование потоковых данных является одной из важнейших задач. Прогнозирование временных рядов используется, когда необходимо определить, что будет происходить с теми, или иными показателями в ближайшую минуту/час/день. Это актуально как для сервисов IT, так и для производственных предприятий, разница лишь в критичности прогнозирования, так как для производства некорректный прогноз может привести к серьезным потерям.

Для пищевого производства особенно важно быстро и качественно реагировать на любые изменения данных и делать корректный прогноз, на основе которого возможны принятия каких-либо решений. В частности, прогнозирование потребления энергоресурсов на предприятии может позволить увидеть нежелательные перерасходы энергии (аномалии) в «будущем» и помочь на их основе принять решения для их предотвращения. Помимо устранения нежелательных перерасходов энергии, также возможно предотвратить аварийные ситуации такие, как выход из строя оборудования,

благодаря способности отследить дрейф концепции данных. Дрейф концепций означает, что статистические свойства целевой переменной, которую модель пытается предсказать, со временем меняются непредвиденным образом. Прогнозирование в условиях возникновения частых аномальных значений и дрейфах концепций является сложной задачей. Существует множество классических моделей прогнозирования: AR, ARMA, ARIMA, SARIMA, модель линейной регрессии и т.д. [1]. Вышеперечисленные модели нашли применения в разных областях, в т.ч. и производственных, но все же их объединяет один большой недостаток – устаревание модели. Так как реальные данные имеют свойство быстро и хаотично (аномально) меняться, классическая модель не может справиться с этой проблемой без периодического переобучения модели, что приводит к большим как временным, так и ресурсным затратам. Для таких задач существуют онлайн - модели, позволяющие не переобучать модель, а учиться на реальных данных [2]. Это полностью противоположно традиционному способу машинного обучения, который заключается в одновременном обучении модели на всех пакетных данных – датасетах. Устаревание модели также влияет на способность решение проблемы дрейфа концепций. Однако, разница между классической моделью и онлайн-моделью заключается в том, что онлайн - модель способна справиться с дрейфом, так как учится постоянно, а классическая без переобучения на это не способна [3].

В данной работе была реализована система прогнозирования потребления электроэнергии, но с двумя разными моделями прогнозирования: применением классической модели прогнозирования ARIMA и применением онлайн-модели, реализованной на основе пакета River [4].

ARIMA, сокращение от «Авторегрессивная интегрированная скользящая средняя», представляет собой класс моделей, которые

«объясняют» данный временной ряд на основе его собственных прошлых значений, то есть его собственных задержек и запаздывающих ошибок прогноза [5].

Модель ARIMA (p, d, q) для нестационарного временного ряда X_t имеет вид:

$$\Delta^d X_t = c + \sum_{i=1}^{p} a_i \Delta^d X_{t-i} + \sum_{j=1}^{q} b_j \varepsilon_{t-j} + \varepsilon_t \quad (1)$$

где: ε_t — стационарный временной ряд,

c - константа

a_i, b_j — авторегрессионные коэффициенты

Δ^d— оператор разности временного ряда порядка d (последовательное взятие d раз разностей первого порядка — сначала от временного ряда, затем от полученных разностей первого порядка, затем от второго порядка и т. д.);

p — это порядок термина «авторегрессивный» (AR). Это относится к количеству лагов Y, которые будут использоваться в качестве предикторов;

q — это порядок термина «Скользящая средняя» (MA). Это относится к количеству запаздывающих ошибок прогноза, которые должны быть включены в модель ARIMA;

d — количество разностей, необходимых для того, чтобы временной ряд стал стационарным.

Онлайн – модель представляет собой регрессионную версию адаптивного древовидного классификатора Хёффдинга (HATR) [6] с моделью Байесовской линейной регрессии [7]:

$$y_i = X_i^T \beta + \varepsilon_i \quad (2)$$

где: β является k x 1 вектором, а ε_i являются независимыми и одинаково распределёнными нормально случайными величинами

HATR использует экземпляр детектора концептуального дрейфа ADWIN на каждом узле принятия решений для отслеживания возможных изменений в распределении данных. Если в узле обнаруживается дрейф, то в фоновом режиме начинает индуцироваться альтернативное дерево. Когда собрано достаточно информации, HATR меняет местами узел, в котором было обнаружено изменение, своим альтернативным деревом. Для реализации дерева был унаследован класс библиотеки Python River:

```
tree.HoeffdingAdaptiveTreeRegressor(
grace_period=1,
model_selector_decay=0.95,
leaf_prediction='adaptive',
drift_window_threshold=1,
drift_detector=adwin,
leaf_model=linear_model.BayesianLinearRegression()
)
```

Описание аргументов класса:

- grace_period=1 - количество экземпляров между попытками разделения;
- model_selector_decay=0.95 - коэффициент экспоненциального затухания применяется к квадратам ошибок моделей обучения;
- leaf_prediction='adaptive' - механизм прогнозирования;
- drift_window_threshold=1 - минимальное количество примеров, которое должно соблюдаться альтернативным деревом, прежде чем оно будет рассматриваться как потенциальная замена текущему;
- drift_detector=adwin - детектор дрейфа, используемый для построения дерева;
- leaf_model=linear_model.BayesianLinearRegression() - модель регрессии, используемая для предоставления прогнозов.

В общем виде система построена по принципу микросервисной архитектуры. Микросервисная архитектура - вариант сервис-ориентированной архитектуры программного обеспечения, направленный на взаимодействие насколько это возможно небольших, слабо связанных и легко изменяемых модулей — микросервисов. Система разделена на следующую микросервисы:

- OPC_Service – сервис сбора данных показаний с счётчиков электроэнергии;
- ARIMA_Service – сервис прогнозирования полученных данных на основе модели ARIMA;
- HATR_Service – сервис прогнозирования полученных данных на основе онлайн - модели HATR;
- Nats_Service – потоковый брокер – сообщений, предназначенный для передачи данных от OPC_Service к ARIMA_Service и HATR_Service

Ниже приведена архитектура системы.

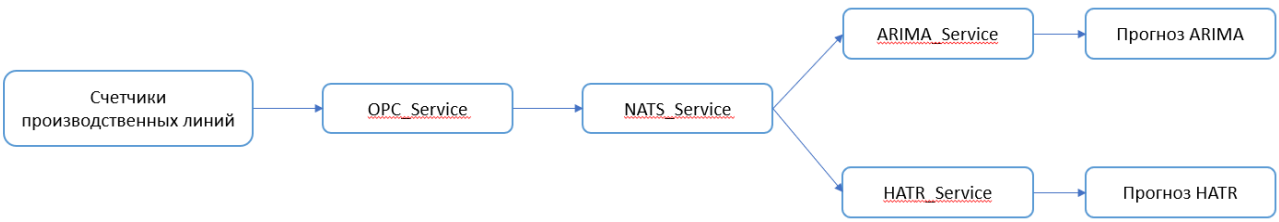

Рис. 1 - Схема тестирования моделей прогнозирования

Система функционирует следующим образом:

- OPC_Service осуществляет сбор показаний счетчиков электропотребления по протоколу OPC UA и создает «тему» в потоковом брокере сообщений NATS [8], куда записывает поочередно потоковые данные с счетчиков по мере их изменения, при этом в случайный момент времени счетчик переводится в

режим симуляции с целью генерации аномальных значений и смены концепта данных.

- Сервисы прогнозирования:
    1. ARIMA_Service считывает данные из сервиса NATS и обучает модель ARIMA на заготовленном датасете, динамически подбирая параметры p, q, d. При оценке (выборе параметров) и сравнении статистических моделей, соответствующих различным параметрам, учитывается, насколько та или иная модель соответствует данным и насколько точно она способна прогнозировать будущие точки данных. Для этого используется значение AIC (Akaike Information Criterion). На основе AIC оценивается, насколько хорошо модель соответствует данным, принимая во внимание общую сложность модели [9, 10]. Чем меньше функций использует модель, чтобы достичь соответствия данным, тем выше её показатель AIC. Поэтому нужно найти модель с наименьшим значением AIC. После подбора необходимых параметров p, q, d, сервис начинает прогнозирование.
    2. HATR_Service считывает значения из сервиса NATS, делает прогноз и выполняет дообучение модели с учетом новых данных в режиме «реального времени».

Для оценки сервисов прогнозирования были проведены следующие тесты:

- Тест на прогнозирование с учетом аномальных значений;
- Тест на дрейф концепций.

Тест на прогнозирование с учетом аномальных значений.

Счетчик электроэнергии переводится в режим симуляции, который генерирует потоковые данные и периодически делает вброс аномальных значений, которые модель должна спрогнозировать.

Модель ARIMA с параметрами (p = 5, d =2, q=1):

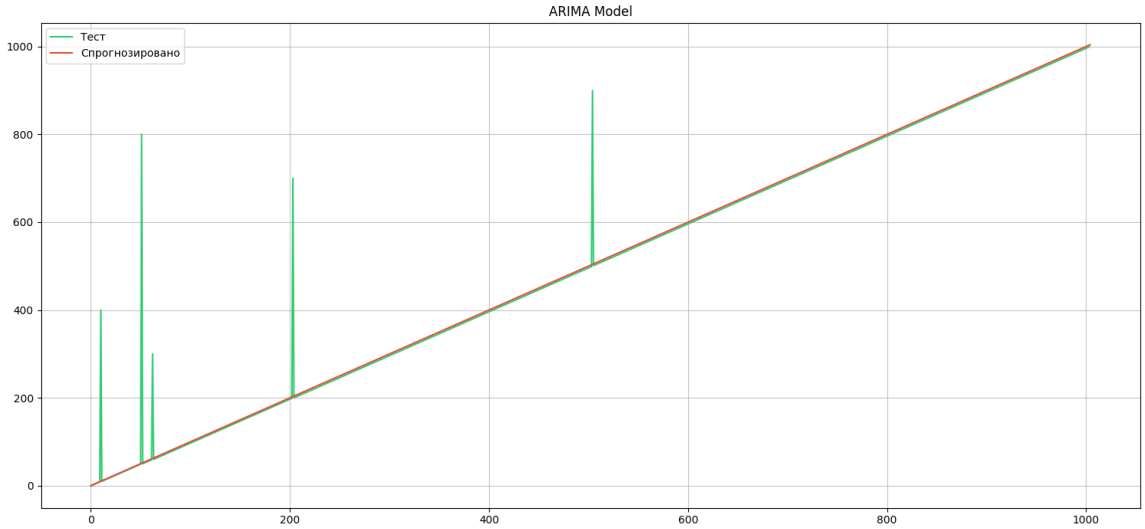

Рис. 2 - Прогноз модели ARIMA

Как видно из рис.2, модель ARIMA не справляется с наличием аномальных пиков значений. При наличии пиковых вбросов, модель их игнорирует и делает некорректный прогноз.

HATR с параметрами ($\beta = 1$):

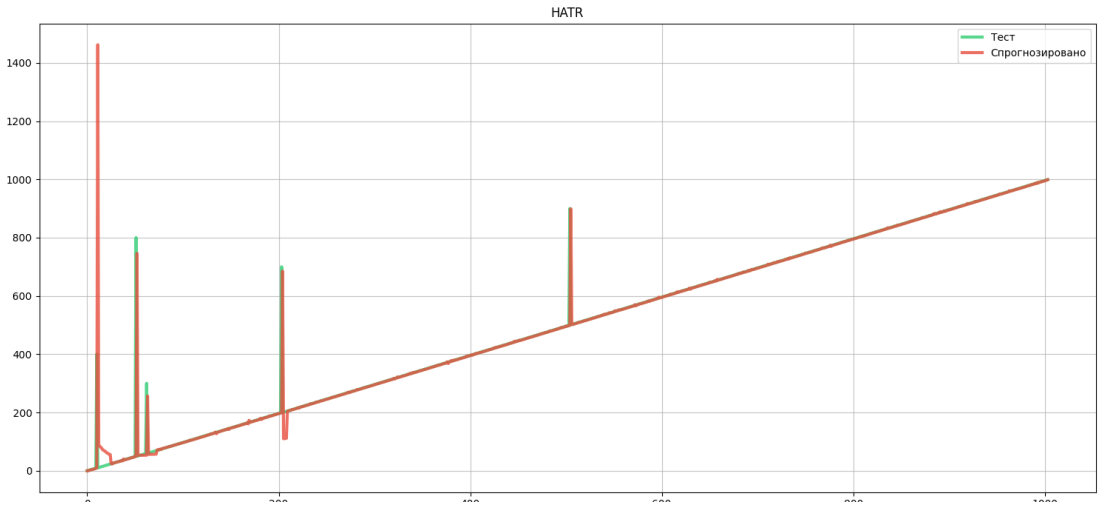

Рис. 3 - Прогноз модели HATR

Как видно из рис. 3, HATR значительно лучше справляется с наличием аномальных пиков значений. Когда случается пиковый вброс, модель более корректно осуществляет свой прогноз.

Тест на прогнозирование с учетом дрейфа концепций.

Изначально счетчик также как и при тесте на аномалии генерирует данные, в которых прослеживается концепт на рост, но через определенный момент времени происходит смена концепта данных, которую модель должна распознать и спрогнозировать.

Модель ARIMA с параметрами (p = 5, d =2, q=1):

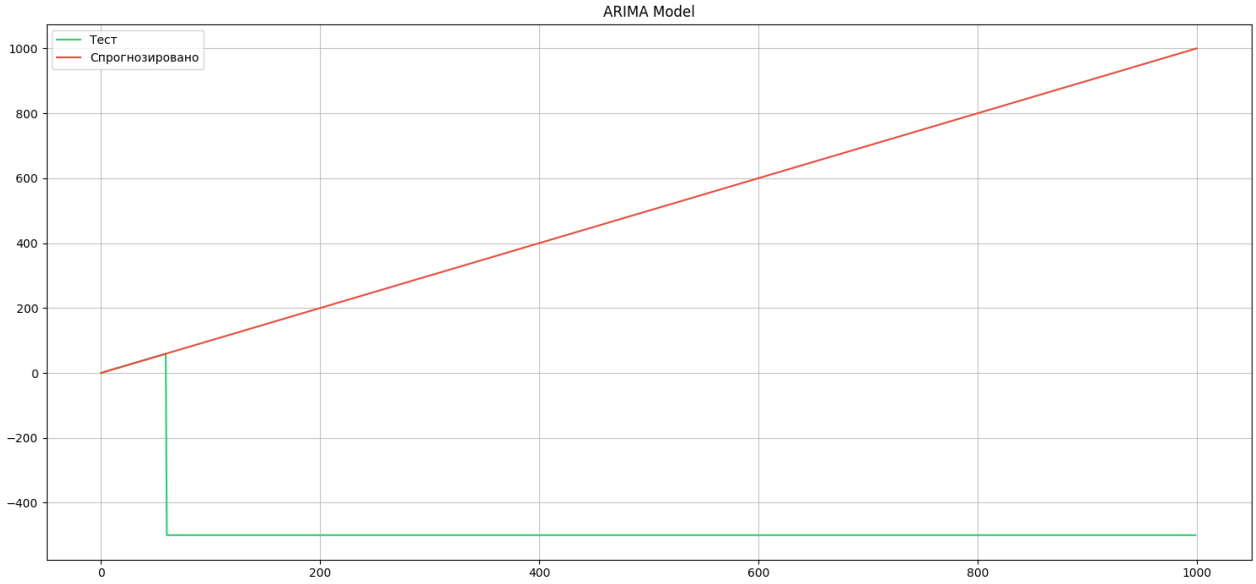

Рис. 4 - Прогноз модели ARIMA (дрейф)

Исходя из рис. 4, видно, что модель ARIMA не справляется с наличием дрейфа. В момент, когда происходит падение, модель прогнозирует рост.

Как видно из рис.5, HATR обнаружил дрейф, и смог осуществить прогноз на основе нового концепта данных.

HATR с параметрами ($\beta = 1$):

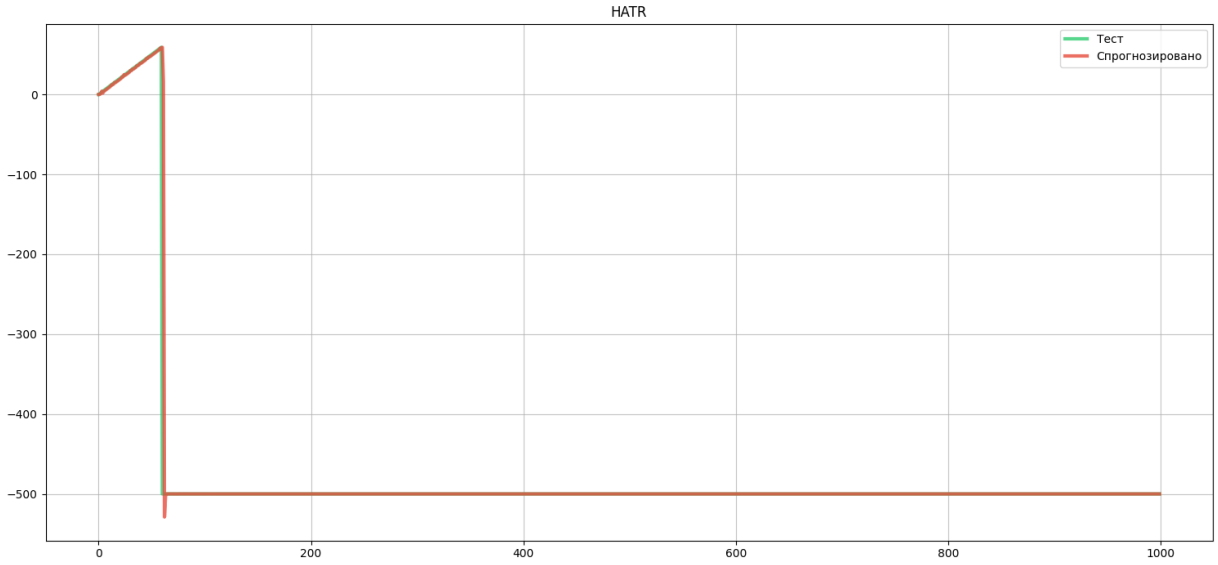

Рис. 5 - Прогноз модели HATR (дрейф)

Из тестов видно, что для разработанной системы больше подходит онлайн - модель HATR, так как результаты показали, что при аномальных вбросах значений и сменах концепций данных, классическая модель ARIMA не справляется с задачей прогнозирования, так как аномалии и дрейф не вписываются в обучающую выборку и носят непостоянный характер. Онлайн- модель HATR показала свою устойчивость к вбросам и дрейфам, так как обучается в режиме реального времени.

Литература (References)

1. Kumar Ajitesh. Different types of Time-series Forecasting Models // Data Analytics. URL: vitalflux.com/different-types-of-time-series-forecasting-models/.
2. Aldweesh A., Derhab A. Deep learning approaches for anomaly-based intrusion detection systems: A survey, taxonomy, and open issues // Knowledge Based Systems. URL: sciencedirect.com/science/article/abs/pii/S0950705119304897.

3. Shumway Robert H., Stoffer David S. Time Series Analysis and Its Applications. Springer, 2011. 596 p.
4. Halford Max, Mastelini Saulo Martiello. River Documentation. River. URL: riverml.xyz/0.13.0/.
5. Warren A. Smith. Static Model Documentation. Static Model. URL: staticmodel.readthedocs.io/en/latest/.
6. Halford Max, Mastelini Saulo Martiello. Hoeffding Adaptive Tree Regressor. River Documentation. URL: riverml.xyz/0.13.0/api/tree/HoeffdingAdaptiveTreeRegressor/#fnref:1.
7. Halford Max. Bayesian linear regression for practitioners. River Documentation. URL: maxhalford.github.io/blog/bayesian-linear-regression/.
8. Collison Derek. JetStream. NATS Documentation. URL: docs.nats.io/nats-concepts/jetstream.
9. Ajitesh Kumar. Machine Learning Models Evaluation Techniques. Data Analytics. URL: vitalflux.com/key-techniques-evaluating-machine-learning-models-performance/.
10. Brockwell Peter J., Davis Richard A. Introduction to time series and forecasting. Springer, 2016. 439 p.