

**Воронин Павел Александрович** – Московский авиационный институт (национальный исследовательский университет); e-mail: heyholyrampage@yandex.ru; г. Москва, Россия; тел.: +79857048284; ассистент.

**Белевцев Андрей Михайлович** – e-mail: ambelevtsev@yandex.ru; тел.: +79037691788; д.т.н.; профессор.

**Александрова Светлана Сергеевна** – e-mail: sweta.sergeeva@gmail.com; тел.: +79859907055; к.т.н.

**Voronin Pavel Aleksandrovich** – Moscow Aviation Institute (National Research University); e-mail: heyholyrampage@yandex.ru assistant; Moscow, Russia; phone: +79857048284; assistant.

**Belevtsev Andrey Michailovitch** – e-mail: ambelevtsev@yandex.ru; phone: +79037691788; dr. of eng. of sc.; professor.

**Aleksandrova Svetlana Sergeevna** – e-mail: sweta.sergeeva@gmail.com; phone: +79859907055; cand. of eng. sc.

УДК 681.3.068:796.01

DOI 10.18522/2311-3103-2023-2-250-263

**В.В. Альчаков, В.А. Крамарь**

## **ОЦЕНКА МЕТОДОВ МАШИННОГО ОБУЧЕНИЯ ДЛЯ ПРОГНОЗИРОВАНИЯ СЕЗОННЫХ ВРЕМЕННЫХ РЯДОВ**

*Временные ряды с сезонной изменчивостью получили широкое распространение для описания процессов в различных областях, таких как торговля, анализ финансовых рынков, прогнозирование пассажирских авиаперевозок, описание климатических изменений. В последнее время широкое применение данный подход стал применяться и для описания технологических процессов. В связи с чем стало возможным применять прогнозирующие модели в системах управления сложными техническими объектами. Методы машинного обучения могут быть эффективно использованы для построения прогнозирующих моделей рядов такого типа. При этом для построения прогноза в качестве входных данных используются лишь исторические данные, накопленные за несколько периодов сезонных наблюдений, знание других параметров, как правило, не требуется. В статье рассмотрено построение прогнозирующей модели временного ряда с сезонной изменчивостью, описывающего технологический процесс, в качестве которого выбран входной поток завода по очистке сточных вод. Описана общая методология построения модели, требования к входным массивам данных, алгоритмы предварительной обработки для формирования выборок, используемых для обучения и тестирования моделей. Для построения прогнозирующей модели в работе использовались классические методы (SARIMA, Holt-Winters Exponential Smoothing, ETS), а также новые алгоритмы (Facebook Prophet, XGBoost, Long Short Term Memory). Реализация алгоритмов выполнена на языке Python, в работе даны рекомендации по использованию существующих библиотек и функций этого языка. Приводится сравнительный анализ точности полученных моделей на основе расчета набора статистических метрик. Также проведен анализ быстродействия методов, поскольку время, затраченное на создание модели и получение прогноза, играет немаловажную роль при запуске модели в реальных условиях на производстве. По совокупности оценок выбран лучший метод для решения поставленной задачи для применения в системах управления реального времени. В заключении даны рекомендации по повышению точности прогноза и обозначены направления будущих исследований в данной области.*

*Временные ряды с сезонной изменчивостью; машинное обучение; прогнозирующие модели; SARIMA; экспоненциальное сглаживание Хольта-Винтерса; Error Trend Seasonal (ETS); Facebook Prophet; XGBoost; Long Short-Term Memory (LSTM).*

V.V. Alchakov, V.A. Kramar

**THE MACHINE LEARNING TECHNIQUE FOR FORECASTING  
THE SEASONAL TIME SERIES**

*Time series with seasonal variability is widely used to describe processes in various fields, such as trade, analysis of financial markets, forecasting of passenger air transportation, and description of climatic changes. Recently, this approach has been widely used to describe technological processes as well. In this regard, applying predictive models in control systems of complex technical objects has become possible. Machine learning methods can be effectively used to build predictive models of series of this type. In this case, only historical data accumulated over several periods of seasonal observation is used as input data for constructing the forecast. Knowledge of other parameters, as a rule, is not required. The article considers creating a predictive time series model with seasonal variability, describing a technological process, the inlet flow of a wastewater treatment plant being chosen as a model. The general methodology of model building, requirements for the input data sets, and algorithms of preprocessing to form samples used for model training and testing are described. Classical methods (SARIMA, Holt-Winters Exponential Smoothing, ETS), as well as new algorithms (Facebook Prophet, XGBoost, Long Short Term Memory), were used to build the predictive model. The implementation of the algorithms is done in the Python language, and recommendations for the use of existing libraries and functions of this language are given in the work. The comparative analysis of the accuracy of the obtained models is given on the calculation of a set of statistical metrics. Analysis of methods performance is also carried out since the time it takes to create a model and get a forecast plays an important role when running the model in real production conditions. The best method for solving the set task for application in real-time control systems was chosen based on the sum of estimates. In conclusion, recommendations for improving forecast accuracy were given, and future research directions were outlined.*

*Seasonal time series; machine learning; forecasting; SARIMA; Holt-Winters exponential smoothing; Error Trend Seasonal (ETS); Facebook Prophet; XGBoost; Long Short-Term Memory (LSTM).*

**Введение.** В настоящее время анализ и прогнозирование временных рядов с сезонной изменчивостью чаще всего встречается в области финансовых рынков для получения моделей прогноза цен на товары и курса криптовалют и торговли [1–4]. В работах [5, 6] прогнозируется величина туристического потока в зависимости от времени года. В свою очередь туристические потоки оказывают влияние на изменение трафика пассажирских перевозок различными видами транспорта, что показано в работах [7, 8]. Также сезонным изменениям подвержены процессы, оказывающие влияние на климат [9, 10], экологию [11, 12] и медицину [13, 14]. Еще одно применение временные ряды с сезонной изменчивостью нашли в области промышленности и энерго- и ресурсопотребления [15–18].

Во всех перечисленных работах используется практически один и тот же набор методов и алгоритмов. Прежде всего это методы для получения авторегрессионных моделей ARIMA (Autoregressive Integrated Moving Average) и SARIMA (Seasonal ARIMA), которые известны достаточно давно и математический аппарат которых хорошо проработан. Первое упоминание метода было сделано еще в 1970-х годах Джорджем Боксом и Гвиллимом Дженкинсом. Методы ARIMA/SARIMA позволяют получить основной класс моделей, которые используются для анализа и прогнозирования временных рядов. В экономических задачах широкое применение находят методы экспоненциального сглаживания Holt-Winters (Holt-Winters Exponential Smoothing), разработанный в 1950-х годах Чарльзом Хольтом, а также его современная модификация – Exponential Smoothing Algorithm (ETS). Помимо перечисленных методов, которые уже можно отнести к классическим, в последние годы появились новые алгоритмы и подходы, которые

также успешно справляются с задачей построения прогнозирующих моделей временных рядов с сезонной изменчивостью. Это методы Facebook Prophet [19], XGboost [20] а также набор методов Artificial Neural Network (ANN), в частности, метод Long Short-Term Memory (LSTM) [21].

В общем случае временной ряд представляет собой набор данных, упорядоченных во времени. Каждому элементу временного ряда ставится в соответствие уникальный индекс – Timestamp ( $TS$ ). Расстояние между индексами может составлять секунды, часы, дни, месяцы и даже годы. При этом значение временного интервала между соседними индексами, как правило, принимается одинаковым и обозначается  $T_s = TS[k] - TS[k - 1]$ . Например, изменение котировок на бирже может быть описано набором вещественных чисел  $Y[k]$  с интервалом  $T_s = 1$  минута, при этом котировки накапливаются в наборе данных от момента начала торгов на бирже, до ее закрытия. Т.е. рассматривается некоторая конечная выборка  $Y(t_k)$  на определенном промежутке времени  $t_k \in [t_0, t_0 + T_s, \dots, t_f]$ .

В общем виде, временной ряд может быть представлен функцией вида:

$$Y(t) = T(t) + S(t) + R(t), \quad (1)$$

где  $T(t)$  – компонент трендового цикла (Trend),  $S(t)$  – сезонный компонент (Seasonal),  $R(t)$  – остаток (Residuals).

Временные ряды, которые содержат сезонный компонент  $S(t)$ , обладают свойством периодичности. Т.е. характер изменчивости временного ряда повторяется время от времени. Примером может служить рост покупательской способности в праздничный сезон или рост пассажирских перевозок в сезон отпусков. Для такого типа временных рядов могут быть построены прогнозирующие модели, которые позволят вычислить оценку (прогноз) параметра на несколько шагов вперед, относительно текущего момента времени (горизонт прогноза). При этом для обучения модели, необходимо лишь знание истории изменения параметра, который описывается временным рядом, т.е. исторические данные на временном интервале, длина которого соответствует нескольким периодам сезонной изменчивости  $m$ . Знание других параметров, как правило, не требуется. Таким образом, возникает задача построения моделей, которые могут генерировать оценку некоторого наблюдаемого параметра с сезонной изменчивостью на требуемом горизонте прогноза, основываясь на информации о предыдущих и настоящих значениях этого параметра (historical data).

В данной статье представлены результаты применения всех перечисленных выше методов для построения модели, предназначенной для предсказания входного потока сточных вод завода очистных сооружений. Данная модель может быть использована для расчета оптимальной загрузки оборудования и обеспечения заданного качества очистки, что в свою очередь позволит значительно сократить потребление электрической энергии оборудованием завода. Описана методика подготовки данных, специфика применения каждого из методов, выполнен статистический анализ полученных результатов и даны рекомендации для повышения качества прогноза. Все численные расчеты и реализация алгоритмов выполнены с помощью языка Python и специализированных библиотек [22].

Методология. В работе использовались реальные данные, полученные от системы мониторинга роботизированного завода очистных сооружений. Датчики системы мониторинга осуществляют сбор основных параметров, таких как величина входного и выходного потока, концентрация кислорода, содержание аммиака и др. Численные значения, соответствующие текущим измерениям параметров, от датчиков поступают на OPC-сервер (Open Platform Communications – семейство программных технологий, предоставляющих единый интерфейс для управления

объектами автоматизации и технологическими процессами), после чего осуществляется их сохранение в базу данных. Регистрация данных осуществляется с частотой 1 раз в секунду в синхронном и асинхронном режимах. Система мониторинга работает в круглосуточном режиме, что позволяет получить данные для исследования и построения моделей за любой интересующий период времени в виде файла в csv-формате. Выбор формата экспортируемых данных обусловлен спецификацией используемых python-библиотек.

Информация от датчиков системы мониторинга хранятся в базе в так называемом «сыром виде» («raw data»). Это означает, что простая выгрузка интересующего параметра за определенный интервал времени не может быть использована как набор входных данных для обучения модели, требуется предварительная обработка и фильтрация. Формат данных, который должен быть использован в подавляющем числе методов может быть представлен в общем виде:

$$YYYY - MM - DD HH:MM:SS \quad VALUE,$$

где *YYYY* – четырехразрядный код года, *MM* – двухразрядный код текущего месяца, *DD* – двухразрядный код дня, *HH* – двухразрядный код часа, *MM* – двухразрядный код минут, *SS* – двухразрядный код секунд, *VALUE* – вещественное число, соответствующее величине параметра, полученное с помощью датчика.

Как было сказано выше, регистрация параметров осуществляется с частотой 1 раз в секунду. Поскольку ресурсы системы мониторинга являются ограниченными, в базу данных записываются данные, усредненные за 1 минуту, по каждому наблюдаемому параметру. Для задачи построения модели прогнозирования входного потока такая частота также является избыточной, поэтому сырые данные, полученные из базы данных, проходят процедуру предварительной обработки. Прежде всего, отбрасываются значения параметра, которые содержат явные выбросы и аномальные значения. На следующем этапе обработки производится усреднение данных за интервал времени, соответствующий 1 часу (data resampling). При проведении эксперимента, была сформирована выборка данных за 19 дней наблюдений, в которой каждый день был представлен набором из 24 значений наблюдаемого параметра. Набор данных, соответствующий первым 17 дням наблюдений, использовался в качестве массива обучающих данных, оставшиеся данные использовались для проверки качества моделей. Вся предварительная обработка выполнена с помощью библиотеки Pandas (Python Data Analysis Library) [24].

Ниже представлен пример таблицы итогового набора данных, полученного после завершения процедуры предобработки.

Таблица 1

**Таблица итогового набора данных**

Full dataset (457 rows)			
	YYYY-MM-DD	HH:MM:SS	Y
	timestamp		
0	2022-12-12	00:00:00	6.305967
1	2022-12-12	01:00:00	5.355895
2	2022-12-12	02:00:00	4.122726
...	...	...	...
456	2022-12-31	00:00:00	6.322228

На рис. 1 четко видна периодичность процесса, период повторения составляет 24 часа. Данные, находящиеся в зеленой зоне (Тест), использовались для тестирования качества модели, данные слева от зеленой зоны – для обучения модели. Задача формулируется следующим образом: по имеющемуся набору данных для обучения построить прогнозирующую модель и с помощью модели построить

прогноз входного потока завода на ближайшие 48 часов. Дальнейшие шаги сводятся к построению и проверке полученных моделей с помощью методов машинного обучения, о которых шла речь в начале статьи.

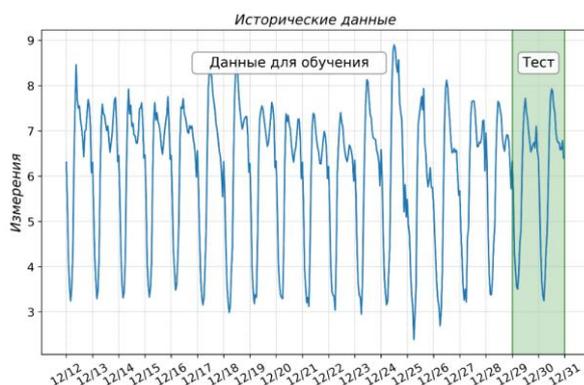


Рис. 1. Исходный набор данных с разбивкой на обучающую и тестовую выборки

В общем случае процесс разработки модели может быть представлен в виде технической дорожной карты, изображенной на рис. 2. Выделяется два уровня: уровень данных и уровень модели. Каждый уровень условно разбит на три шага. Уровень данных (шаги 1-3) реализует предварительную обработку и подготовку данных как это описано выше.

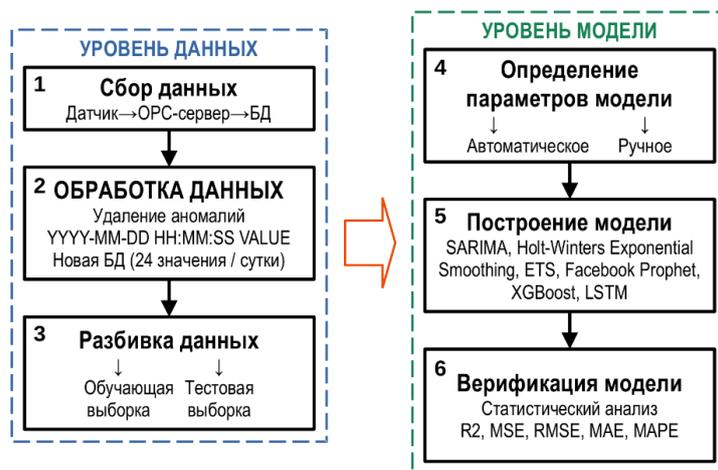


Рис. 2. Техническая дорожная карта построения прогнозирующей модели

Уровень модели подразумевает процедуру выбора параметров модели (шаг 4), который может быть реализован двумя способами: автоматически и вручную. Способ выбора зависит от возможности библиотеки, которая используется для построения модели. На следующем шаге (шаг 5) происходит построение модели, включающее задание параметров, определенных на предыдущем шаге и обучение модели на основе данных из обучающей выборки. В общем случае все действия с моделью на данном шаге можно назвать построением модели. На последнем шаге (шаг 6) происходит валидация модели. Под валидацией будем понимать процесс получения прогноза и сравнение оценок параметра с их истинными значениями.

Для оценки качества модели будет использован набор статистических метрик, описание которых будет приведено ниже. В случае, если метрики модели удовлетворяют заданным критериям, то модель сохраняется и может быть использована для построения прогноза. Если метрики не удовлетворяют заданным, необходимо вернуться на шаг 4 и выполнить все последующие шаги, находящиеся на уровне модели.

Далее будут изложены основные теоретические положения и особенности реализации каждого из перечисленных методов.

**Разработка модели SARIMA.** SARIMA – метод построения модели на основе использования сезонного авторегрессионного интегрированного значения скользящего среднего. SARIMA или Seasonal ARIMA, является расширением ARIMA метода, явно поддерживающим одномерные данные временных рядов с сезонным компонентом. В соответствии с [25] сезонная модель ARIMA включает авторегрессию и скользящее среднее с запаздыванием  $s$ . Сезонная модель ARIMA  $(p, d, q)(P, D, Q)_s$  может быть наиболее кратко выражена с использованием оператора обратного сдвига

$$\theta_p(B^s)\theta_p(B)(1 - B^s)^D(1 - B)^d x_i = \Phi_Q(B^s)\phi_q(B)\omega_t, \quad (1)$$

где  $\theta_p, \theta_p, \Phi_Q$  и  $\phi_q$  – полиномы порядков  $P, p, Q$  и  $q$  соответственно.

В общем случае модель является нестационарной, хотя, если  $D = d = 0$  и все корни характеристического уравнения (многочлены в левой части уравнения (2)) превышают единицу по модулю, результирующая модель будет быть стационарной.

Таким образом, перед использованием метода необходимо проверить исходный набор данных на стационарность. Сделать это можно с помощью Augmented Dickey–Fuller (ADF) теста. Этот тест основан на гипотезе, согласно которой, если значение параметра  $p$  меньше 0,05, то можно считать, что временной ряд является стационарным, а если значение  $p$  больше 0,05, то временной ряд является нестационарным.

Для выполнения ADF теста и реализации метода SARIMA была использована библиотека *statmodels* [26]. Результат ADF теста имеет вид:

```

1. ADF: -6.162488723252326
2. P-Value: 7.118446442706881e-08
3. Num Of Lags: 18
4. Num Of Observations Used For ADF Regression: 389
5. Critical Values:
1%: -3.447272819026727
5%: -2.868998737588248
10%: -2.5707433189709294
    
```

Таким образом, по результатам ADF теста исходный временной ряд является стационарным.

Подбор параметров  $(p, d, q)(P, D, Q)$  может быть выполнен в автоматическом режиме с помощью функции *auto\_arima* библиотеки *pmdarima* [27].

Процедура поиска оптимальной комбинации параметров занимает достаточно продолжительный промежуток времени, и это необходимо учитывать при использовании метода в системах реального времени.

При поиске параметров, есть возможность выводить лог операций, в соответствии с которым видно, что некоторые комбинации параметров в конце концов отбрасываются алгоритмом. Выбор оптимальной комбинации осуществляется по минимальному значению параметра AIC:

ARIMA (3, 0, 1) (0, 1, 2) [24] intercept: AIC=177.139, T=63.80 sec  
 ARIMA (3, 0, 3) (0, 1, 2) [24] intercept: AIC=181.003, T=78.64 sec  
 ARIMA (2, 0, 3) (0, 1, 2) [24] intercept: AIC=177.237, T=77.82 sec  
**ARIMA (3, 0, 2) (0, 1, 2) [24] : AIC=169.442, T=72.05 sec**  
 ARIMA (3, 0, 2) (0, 1, 1) [24] : AIC=185.525, T=8.53 sec  
 ARIMA (3, 0, 2) (1, 1, 2) [24] : AIC=inf, T=82.20 sec  
 ARIMA (3, 0, 2) (1, 1, 1) [24] : AIC=inf, T=9.30 sec

Результат работы функции имеет вид:

Best model: ARIMA (3, 0, 2) (0, 1, 2) [24]

где  $s = 24$  – период сезонного временного ряда в отсчетах.

После построения модели с учетом полученных оптимальных параметров, для построения прогноза достаточно воспользоваться методом *get\_forecast* объекта модели, передав в качестве параметра количество временных отсчетов, для которых необходимо получить предсказание. Поскольку интервал между отсчетами в исходном датасете равен одному часу, а прогноз составляется на 48 часов, то необходимо определить входной параметр  $steps = 48$ .

На рис. 3. представлен график временного ряда, полученный с помощью модели SARIMA (сплошная линия на графике) и истинные значения измерений из тестового набора данных (dark bullet marker).

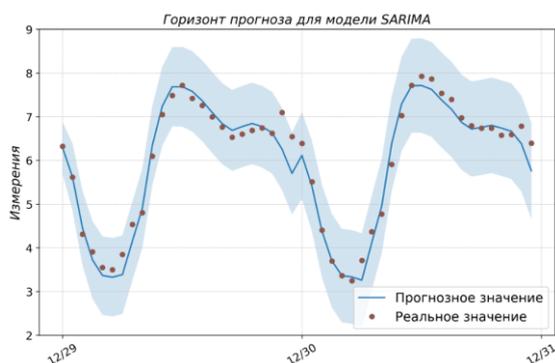


Рис. 3. Результат предсказания с помощью SARIMA модели и истинные значения из тестового набора данных

Количественная оценка качества этой и последующих прогнозирующих моделей будет дана в разделе анализа результатов.

**Разработка модели экспоненциального сглаживания Holt-Winters.** Модель *Holt-Winters* включает уравнение прогноза и три уравнения сглаживания – одно для уровня  $\ell_t$ , одно для тренда  $b_t$  и одно для сезонного компонента  $s_t$  с соответствующими параметрами сглаживания  $\alpha$ ,  $\beta^*$  и  $\gamma$ . Параметр  $m$  используется для обозначения периода сезонности [28]. Существует два варианта этого метода, отличающихся характером сезонной составляющей – аддитивный метод и мультипликативный метод.

Компонентная форма для аддитивного метода имеет вид:

$$\begin{aligned}
 \hat{y}_{t+h|t} &= \ell_t + hb_t + s_{t+h-m(k+1)} \\
 \ell_t &= \alpha(y_t - s_{t-m}) + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \\
 b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \\
 s_t &= \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1 - \gamma)s_{t-m} .
 \end{aligned} \tag{3}$$

Компонентная форма для мультипликативного метода имеет вид:

$$\begin{aligned} \hat{y}_{t+h|t} &= (\ell_t + hb_t)s_{t+h-m(k+1)} \\ \ell_t &= \alpha \frac{y_t}{s_{t-m}} + (1 - \alpha)(\ell_{t-1} + b_{t-1}) \\ b_t &= \beta^*(\ell_t - \ell_{t-1}) + (1 - \beta^*)b_{t-1} \\ s_t &= \gamma \frac{y_t}{\ell_{t-1} + b_{t-1}} + (1 - \gamma)s_{t-m}. \end{aligned} \quad (4)$$

В соотношениях (3) и (4)  $k$  это целая часть  $(h - 1)/m$ .  $\ell_t$  показывает средневзвешенное значение между сезонно скорректированным наблюдением  $(y_t - s_{t-m})$  и несезонным прогнозом  $(\ell_t - \ell_{t-1})$  для  $t$ . Уравнение тренда  $b_t$  идентично линейному методу Холта. Сезонное уравнение показывает средневзвешенное значение между текущим сезонным индексом  $(y_t - \ell_{t-1} - b_{t-1})$  и сезонным индексом того же сезона  $m$  периодов назад.  $s_t$  — это уравнение для сезонной составляющей.

Для создания прогнозирующей модели на основе метода экспоненциального сглаживания Holt-Winters, была использована функция *ExponentialSmoothing* библиотеки *statmodels* [26]. На вход функция принимает три параметра: набор данных для обучения, тип сезонной компоненты *add/mul*, и значение параметра  $m$  из уравнений (3) и (4) — значение периода сезонности выборки. Результат прогнозирования с помощью модели Holt-Winters представлен на рис. 4.

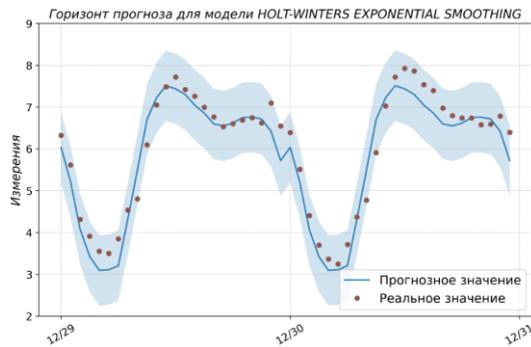


Рис. 4. Результат предсказания с помощью модели экспоненциального сглаживания Holt-Winters и истинных значений из тестового набора данных

**Разработка ETS модели.** Модель *ETS* представляют собой семейство моделей временных рядов с базовой моделью пространства состояний, состоящей из компонента уровня, компонента тренда (Т), сезонного компонента (S) и ошибки (E). Точечные прогнозы можно получить из моделей путем повторения уравнений для  $t = T + 1, \dots, T + h$  и установки всех  $\varepsilon_t = 0$  для  $t > T$ . Например, для модели ETS (M, A, N)  $y_{T+1} = (\ell_T + b_T)(1 + \varepsilon_{T+1})$ . Поэтому  $\hat{y}_{T+1|T} = \ell_T + b_T$ . Таким образом, можно записать:

$$\begin{aligned} y_{T+2} &= (\ell_{T+1} + b_{T+1})(1 + \varepsilon_{T+2}) = \\ &[(\ell_T + b_T)(1 + \alpha\varepsilon_{T+1}) + b_T + \beta(\ell_T + b_T)\varepsilon_{T+1}](1 + \varepsilon_{T+2}). \end{aligned} \quad (5)$$

Следовательно,  $\hat{y}_{T+2|T} = \ell_T + 2b_T$  и так далее. Эти прогнозы идентичны прогнозам линейного метода Холта, а также прогнозам модели ETS (A, A, N). Таким образом, точечные прогнозы, полученные с помощью метода и двух моделей, лежащих в основе метода, идентичны (при условии, что используются одни и те же значения параметров). Построенные таким образом точечные прогнозы ETS равны средним значениям прогнозных распределений, за исключением моделей с мультипликативной сезонностью [28].

Аналогично модели Holt-Winters, ETS модель может быть получена с помощью библиотеки *statmodels*. Функция *ETSModel* принимает в качестве входных параметров массив данных с отсчетами временного ряда, флаг, определяющий тип сезонной компоненты *add/mul* и значение периода сезонности (для рассматриваемого примера *seasonal\_periods = 24*). Для получения прогноза используется метод *get\_prediction* объекта модели, который получает в качестве входных параметров два значения *timestamp*, соответствующие начальному и конечному отсчету времени для желаемого горизонта прогноза. При использовании *pandas*, эти значения могут быть найдены как начальное и конечное значение индекса датафрейма, в котором храниться тестовая выборка временного ряда. Результат прогнозирования с помощью модели ETS представлен на рис. 5.

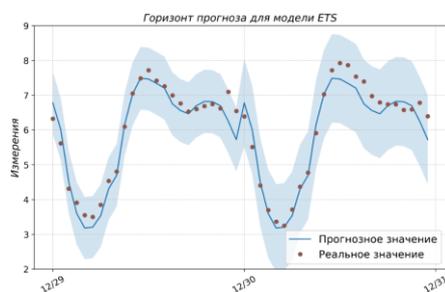


Рис. 5. Результат предсказания с помощью ETS модели и истинные значения из тестового набора данных

**Разработка модели Facebook Prophet.** Модель *Prophet* – это процедура прогнозирования данных временных рядов, основанная на модели аддитивной регрессии, в которой нелинейные тренды соответствуют годовой, еженедельной и ежедневной сезонности, а также праздничным эффектам. Процедура лучше всего работает с временными рядами, которые имеют сильные сезонные эффекты и несколько сезонов исторических данных. *Prophet* устойчив к отсутствующим данным и сдвигам в тренде и обычно хорошо обрабатывает выбросы [29].

Основные положения метода представлены в работе [19]. *Prophet* представляет собой open-source библиотеку, которая распространяется на языках R и Python. Подбор необходимых параметров модели полностью автоматизирован, поэтому для построения модели достаточно передать входной набор данных для обучения. Для создания модели используется функция *Prophet* библиотеки *prophet*. Прогнозные значения параметра могут быть найдены с помощью метода *predict* модели, на вход которого подается датафрейм из временных меток, соответствующих горизонту прогноза. Результат прогнозирования с помощью модели *Prophet* представлен на рис. 6.

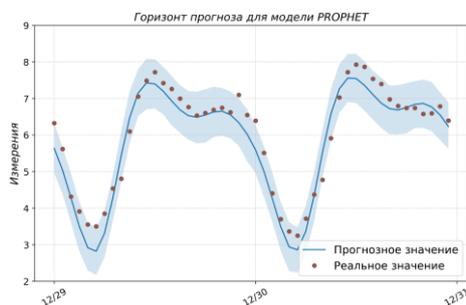


Рис. 6. Результат предсказания с помощью Prophet модели и истинные значения из тестового набора данных

**Разработка модели XGBoost.** Модель *Extreme Gradient Boosting* (XGBoost) – это оптимизированная распределенная гибкая и портативная библиотека повышения градиента. Библиотека реализует алгоритмы машинного обучения в рамках фреймворка Gradient Boosting. XGBoost предоставляет оптимизированный алгоритм параллельного бустинга, также известный как Parallel Gradient Boosting Decision Trees (GBDT), который позволяет получить быстрое и точное решение большого количества задач, которые возникают в области науки о данных. Один и тот же код работает в основных распределенных средах (Hadoop, SGE, MPI) и может решать большое количество проблем различного характера [30].

Основные положения метода представлены в работе [20]. Модель создается с помощью функции *XGBRegressor* библиотеки *xgboost*. Функция содержит достаточно большое количество параметров, с полным списком можно ознакомиться в разделе документации к библиотеке [31]. При построении модели использовался следующий набор параметров:

- ◆  $\text{max\_depth} = 6$  – максимальная глубина дерева решений;
- ◆  $\text{learning\_rate} = 0,05$  – повышение скорости обучения;
- ◆  $\text{n\_estimators} = 5000$  – количество деревьев;
- ◆  $\text{gamma} = 0,1$  – параметр алгоритма;

Результат прогнозирования с помощью модели XGBoost представлен на рис. 7.

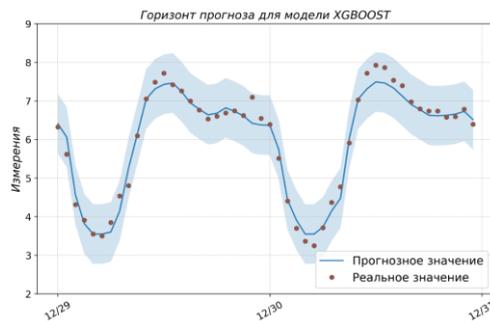


Рис. 7. Результат предсказания с помощью XGBoost модели и истинные значения из тестового набора данных

**Разработка модели LSTM.** Модель *Long Short-Term Memory* (LSTM) – это искусственная нейронная сеть, используемая в области искусственного интеллекта и глубокого обучения. В отличие от стандартных нейронных сетей с прямой связью, LSTM имеет обратные связи. Такая рекуррентная нейронная сеть может обрабатывать не только отдельные точки данных, но и целые последовательности данных. Эта характеристика делает сети LSTM идеальными для обработки и прогнозирования данных, включая временные ряды. Основные положения метода представлены в работе [21].

Для построения LSTM модели была использована библиотека *keras* и набор функций *Sequential*, *Dense*, *LSTM* которые использовались для задания структуры нейронной сети. Для сборки модели были выбраны следующие параметры:

- ◆  $\text{batch\_size} = 16$  – количество выборок на обновление градиента;
- ◆  $\text{epochs} = 200$  – количество эпох для обучения модели;
- ◆  $\text{units} = 32$  – размерность выходного пространства.

В результате применения библиотеки получена модель вида:

LSTM model summary  
Model: "sequential"

Layer (type)	Output Shape	Param #
lstm (LSTM)	(None, 32)	4352
dense (Dense)	(None, 1)	33

Total params: 4,385  
Trainable params: 4,385  
Non-trainable params: 0

Результат прогнозирования с помощью модели LSTM представлен на рис. 8.

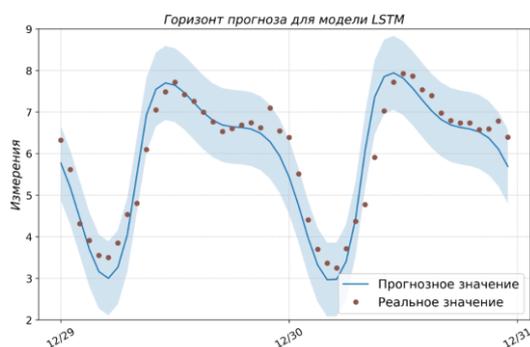


Рис. 8. Результат предсказания с помощью LSTM модели и истинные значения из тестового набора данных

**Результаты.** В этой работе для оценки производительности модели использовались пять статистических метрик: коэффициент корреляции (R2), среднеквадратическая ошибка (MSE), среднеквадратическая ошибка (RMSE), средняя абсолютная ошибка (MAE) и средняя абсолютная ошибка в процентах (MAPE) [23]. Коэффициент корреляции (R2) использовался для определения степени соответствия прогнозируемых значений общим фактическим значениям. MSE – это среднее изменение фактических и прогнозируемых значений, а RMSE – это квадратный корень из MSE, тогда как MAE предлагается как среднее изменение между фактическими и прогнозируемыми значениями. По сравнению с MAE, RMSE придает большее значение дисперсии между выбросами данных. Эти показатели рассчитываются по следующим формулам

$$\begin{aligned}
 R^2 &= 1 - \frac{\sum_{i=1}^N |\hat{y}_i - y_i|^2}{\sum_{i=1}^N \left| \frac{1}{N} \sum_{i=1}^N y_i - y_i \right|^2}, \\
 MSE &= \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|^2 \\
 RMSE &= \sqrt{\frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i|^2}, \\
 MAE &= \frac{1}{N} \sum_{i=1}^N |\hat{y}_i - y_i| \\
 MAPE &= 100\% \frac{1}{N} \sum_{i=1}^N \frac{|\hat{y}_i - y_i|}{|y_i|},
 \end{aligned} \tag{6}$$

где  $y_i$  и  $\hat{y}_i$  – реальное и предсказанное значения, а параметр  $N$  представляет число элементов в выборке.

Для расчета метрик (6), использовались функции библиотеки *sklearn.metrics*. Кроме статистических метрик для каждой модели было рассчитано время, которое потребовался python-скрипту для построения модели и время, которое потребовалось для построения прогноза. Это связано с тем, что при работе с моделями в реальных условиях, когда прогноз модели используется для расчета управляющих сигналов или принятия управленческих решений, точность модели со временем может падать, поскольку условия функционирования объекта управления постоянно меняются. Например, происходит замена оборудования, изменяется количество потребителей, характер сезонности для небольших периодов времени меняется в течение года и др. Поэтому, для сохранения точности прогноза, необходимо переобучать модель время от времени. Это означает, что нужно вновь формировать обучающую и тестовую выборки, осуществлять подбор параметров и собирать новую модель. Если переобучение модели будет происходить раз в месяц, то этим вопросом можно пренебречь. Если переобучение будет происходить раз в сутки, при дискретности шага управления менее 10 секунд, то критерий времени становится существенным и играет такую же роль, как и точность, для выбора лучшей модели.

В табл. 2 представлены результаты расчета статистических метрик для оценки точности созданных моделей и затраты времени на создание модели и получение прогноза.

Таблица 2

Статистический анализ и анализ затрат времени

Модель	Метрики					Время (сек.)	
	<i>R2</i>	<i>MSE</i>	<i>RMSE</i>	<i>MAE</i>	<i>MAPE</i>	<i>Training</i>	<i>Prediction</i>
SARIMA	0,961	0,076	0,276	0,198	0,035	369,717	0,008
Holt-Winters ES	0,921	0,156	0,396	0,324	0,059	0,058	0,002
ETS	0,945	0,109	0,329	0,254	0,043	0,353	0,004
Prophet	0,918	0,162	0,402	0,331	0,062	0,252	0,324
XGBoost	0,975	0,050	0,224	0,163	0,029	7,408	0,006
LSTM	0,962	0,076	0,275	0,212	0,040	86,738	0,469

Явными лидерами по точности являются три модели: SARIMA, LSTM, XGBoost. Однако, модели SARIMA и LSTM значительно проигрывают по времени, необходимому для создания новой модели, что вызвано работой автоматической процедуры поиска оптимальных значений параметров модели. Поэтому оптимальным выбором для решения поставленной задачи является выбор метода XGBoost, обеспечивающего максимальную точность и минимальное время (в группе лидеров) на обучение и получение прогнозного значения.

**Заключение.** В статье рассмотрены различные методы прогнозирования временных рядов с сезонной изменчивостью. Представлены классические и более новые подходы на базе алгоритмов машинного обучения и нейронных сетей. На основании имеющихся в распоряжении реальных данных, для каждого из рассмотренных методов проведен тест точности прогноза с использованием статистических метрик. Также для выбора лучшего метода использовался критерий быстроты построения модели и получения прогноза, что обусловлено необходимостью переобучения модели время от времени. Лучшим методом по совокупности признаков оказался метод XGBoost. Тем не менее, оптимизация алгоритмов

поиска параметров моделей, может вывести в лидеры и другие алгоритмы, например LSTM. Также можно предположить, что на качество модели будет влиять и структура нейронной сети, которая задается перед обучением модели. Вопросы оптимизации процедуры обучения моделей и поиску оптимальных структур нейронных сетей и будут посвящены дальнейшие исследования.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Pongdatu G.A.N., Putra Y.H. Time Series Forecasting using SARIMA and Holt Winter's Exponential Smoothing, *IOP Conf. Ser.: Mater. Sci. Eng.*, 2018, 407 012153.
2. Huang W., Li Y., Zhao Y., Zheng L. Time Series Analysis and Prediction on Bitcoin, *BCP Business & Management*, 2022, Vol. 34, pp. 1223-1234.
3. Kemalbay G., B. Korkmazoglu O. Sarima-arch versus genetic programming in stock price prediction, *Sigma J Eng Nat Sci.*, 2021, Vol. 39, No. 2, pp. 110-122.
4. Paliari I., Karanikola A., Kotsiantis S. A comparison of the optimized LSTM, XGBOOST and ARIMA in Time Series forecasting, *Proceedings of 12th International Conference on Information, Intelligence, Systems & Applications, Chania Crete, Greece, 2021*.
5. Andreeski C., Mechkaroska D. Modelling, Forecasting and Testing Decisions for Seasonal Time Series in Tourism, *Acta Polytechnica Hungarica*, 2020, Vol. 17, No. 10, pp. 149-171.
6. Uğuz, Büyükgökoğlan E. A Hybrid CNN-LSTM Model for Traffic Accident Frequency Forecasting During the Tourist Season, *Technical Gazette*, 2022, Vol. 29, pp. 2083-2089.
7. Etuk E. A seasonal time series model for Nigerian monthly air traffic data, *IJRRAS*, 2013, 4 (3), pp. 596-602.
8. Feng T. Tianyu Z., Zheng Y., Jianxing Y. The comparative analysis of SARIMA, Facebook Prophet, and LSTM for road traffic injury prediction in Northeast China // *Frontiers in Public Health*. – July 2022, Vol. 10. DOI: 10.3389/fpubh.2022.946563.
9. Zhu X., Helmer E.H., Gwenzl D., Collin M. Characterization of Dry-Season Phenology in Tropical Forests by Reconstructing Cloud-Free Landsat Time Series, *Remote Sens*, 2021, 13, 4736.
10. Figueiredo N., Blanco C. Water level forecasting and navigability conditions of the Tapajós River - Amazon – Brazil, *La Houille Blanche*, 2016. La Houille Blanche, Vol. 102 (3), pp. 53-64.
11. Shen J., Valagolam D., McCalla S. Prophet forecasting model: a machine learning approach to predict the concentration of air pollutants (PM2.5, PM10, O3, NO2, SO2, CO) in Seoul, South Korea, *PeerJ*, 2020, 8.
12. Hasnain A., Sheng Y., Hashmi M.Z. Time Series Analysis and Forecasting of Air Pollutants Based on Prophet Forecasting Model in Jiangsu Province, China Citation, *Frontiers in Environmental Science*, 2022, 10:945628.
13. Luo Z., Jia X., Bao J. A Combined Model of SARIMA and Prophet Models in Forecasting AIDS Incidence in Henan Province, China, *International Journal of Environmental Research and Public Health*, 2022, 19, 5910.
14. Pandit A., Khan D. Z., Hanrahan J. G. Historical and future trends in emergency pituitary referrals: a machine learning analysis, *Pituitary*, 2022, 25 (6), pp. 927-937.
15. Benkachcha S., Benhra J., El Hassani H. Seasonal Time Series Forecasting Models based on Artificial Neural Network, *International Journal of Computer Applications*, 2015, Vol. 116, No. 20, pp. 9-14.
16. Palmroos C., Gieseler J., Morosan N. Solar energetic particle time series analysis with Python, *Frontiers in Astronomy and Space Sciences*, 2022, 9:1073578.
17. Wan X., Zou Y., Wang J., Wang W. Prediction of shale oil production based on Prophet algorithm, *Journal of Physics: Conference Series*, 2009, Vol. No. 1.
18. El-Rawy M., Abd-Ellah M.K., Fathi H., Abdella Ahmed A. K. Forecasting effluent and performance of wastewater treatment plant using different machine learning techniques, *Journal of Water Process Engineering*, 2021, Vol. 44.
19. Taylor S., Letham B. Forecasting at scale, *The American statistician*, 2017, Vol. 72, No. 1, pp.37-45.
20. Chen T., Guestrin C. XGBoost: A Scalable Tree Boosting System, *Proceedings of the 22nd ACM SIGKDD International Conference, San Francisco California*, 2016.
21. Hochreiter S., Schmidhuber J. Long Short-term Memory, *Neural computation*, 1997, Vol. 9, pp. 1735-80.

22. *Zemkoho A.* A Basic Time Series Forecasting Course with Python, *Oper. Res. Forum.*, 2023, 4:2.
23. *Plevris V., Solorzano G., Bakas N., Ben Seghier M.* Investigation of performance metrics in regression analysis and machine learning-based prediction models, *Proceedings of the 8th European Congress on Computational Methods in Applied Sciences and Engineering Oslo, Norway, 2022.*
24. Pandas - Python Data Analysis Library. Available at: <https://pandas.pydata.org>.
25. *Cowpertwait P.S.P., Metcalfe A. V.* Introductory Time Series with R. Springer, London, 2009.
26. Introduction – statmodels. Available at: <https://www.statsmodels.org/stable/index.html/>.
27. Pmdarima: ARIMA estimators for Python. Available at: <https://alkaline-ml.com>.
28. *Hyndman R.J., Athanasopoulos G.* Forecasting: Principles and Practice, Otexts, Monash University, Australia, 2021.
29. Prophet | Forecasting at scale. Available at: <https://facebook.github.io/prophet/>.
30. XGBoost. Available at: <https://xgboost.ai/about>.
31. Python API Reference – xgboost documentation. Available at: [xgboost.readthedocs.io](https://xgboost.readthedocs.io).

Статью рекомендовала к опубликованию к.ф.-м.н. Н.М. Твердохлеб.

**Альчаков Василий Викторович** – Севастопольский государственный университет; e-mail: [alchakov@mail.ru](mailto:alchakov@mail.ru); г. Севастополь, Россия; тел.: +79787489941; к.т.н.; доцент кафедры «Информатика и управление в технических системах».

**Крамарь Вадим Александрович** – e-mail: [kramarv@mail.ru](mailto:kramarv@mail.ru); тел.: +79787927340; д.т.н.; профессор кафедры «Информатика и управление в технических системах».

**Alchakov Vasiliy Viktorovich** – Sevastopol State University; e-mail: [alchakov@mail.ru](mailto:alchakov@mail.ru); Sevastopol, Russia; phone: +79787489941; cand. of eng. sc.; associate professor of the department «Informatics and Control in Technical Systems».

**Kramar Vadim Aleksandrovich** – e-mail: [kramarv@mail.ru](mailto:kramarv@mail.ru); phone: +79787927340; dr. of eng. sc.; associate professor of the department «Informatics and Control in Technical Systems».

УДК 004.81

DOI 10.18522/2311-3103-2023-2-263-273

**А.Д. Ульев, Ю.А. Орлова, В.Л. Розалиев, А.Р. Донская**

### **МЕТОДЫ И СРЕДСТВА СЛЕЖЕНИЯ ЗА ПЕРЕМЕЩЕНИЕМ И ВЗАИМОДЕЙСТВИЕМ СОТРУДНИКОВ И ПОКУПАТЕЛЕЙ ПО ВИДЕОИЗОБРАЖЕНИЮ\***

*Из-за стремительного развития сферы торговли особую популярность набирают средства автоматического контроля работы сотрудников, оказывающих услуги клиентам. На данный момент существует множество современных подходов, методов и алгоритмов для автоматического отслеживания покупателей и продавцов в помещении магазина. Современные компании стараются решить подобную проблему разными путями: подсчетом посетителей, приборами контроля, различными нейросетевыми решениями и так далее. Проведя обзор решений с необходимым функционалом, были выявлены основные недостатки, такие как, например, дороговизна, неудобство в использовании и так далее. В результате авторами была поставлена цель: повысить качество отслеживания перемещения сотрудников/покупателей за счет разработанных автоматизированных средств и методов контроля перемещения, межкамерного трекинга и идентификации личности. В статье рассмотрен метод для автоматического распознавания и слежения за сотрудниками магазинов и фирм. Метод основан на каскаде нейросетей и алгоритмов, позволяющих распознавать покупателей и работников в униформе, а также оценивать качество работы сотрудников и удовлетворенность клиентов по голосу. Как результаты работы*

\* Исследование выполнено при финансовой поддержке ВолГТУ в рамках научного проекта № 60/478-22.