УДК 004.91 doi: 10.15827/0236-235X.149.100-107 2025. Т. 38. № 1. С. 100–107

Фреймворк для автоматизации прогнозирования остаточного ресурса оборудования при построении проактивных систем поддержки принятия решений

К.С. Задиран [1], Д.А. Волкова [1]✉, М.В. Щербаков [1]

[1] Волгоградский государственный технический университет, г. Волгоград, 400005, Россия

Ссылка для цитирования
Задиран К.С., Волкова Д.А., Щербаков М.В. Фреймворк для автоматизации прогнозирования остаточного ресурса оборудования при построении проактивных систем поддержки принятия решений // Программные продукты и системы. 2025. Т. 38. № 1. С. 100–107. doi: 10.15827/0236-235X.149.100-107
Информация о статье
Группа специальностей ВАК: 1.2.1
Поступила в редакцию: 08.05.2024 После доработки: 24.06.2024 Принята к публикации: 23.07.2024

Аннотация. В статье описан фреймворк, разработанный для автоматизации исследований в области проектирования проактивных систем поддержки принятия решений. В частности, исследуется задача анализа и прогнозирования временных рядов с целью создания инструментов автоматизации прогнозирования различных процессов в системах управления активами, в том числе технического обслуживания и ремонта. Определена роль процессов автоматизации в управлении активами в этих системах. В работе выделены основные факторы, влияющие на выбор программы для реализации системы предиктивной аналитики. Предложен алгоритм решения задачи прогнозирования оставшегося срока службы на основе анализа данных о производственных активах с использованием компонентов искусственного интеллекта. Предлагаемое программное решение, основанное на CRISP-DM, не является отдельным программным продуктом и может быть встроено в существующее программное обеспечение, поддерживая возможность модификации методов. Фреймворк выполняет следующие функции: загрузка и предобработка данных, построение моделей прогнозирования, прогнозирование временных рядов, оценка прогноза. Разработанный фреймворк обладает гибкой модульной архитектурой для добавления новых методов анализа и прогнозирования. Возможность на основе имеющихся базовых классов переопределить и реализовать собственные источники данных, стадии предобработки, модели прогнозирования и метрики расширяет вариативность и повышает эффективность его функционирования. В статье приведен пример использования фреймворка для решения задачи анализа временных рядов и определения оставшегося срока службы оборудования, демонстрирующий эффективность разработанного продукта в области исследования данных и искусственного интеллекта.

Ключевые слова: программный фреймворк, прогнозирование временных рядов, анализ временных рядов, системы поддержки принятия решений, RUL, модели прогнозирования

Благодарности. Исследование выполнено за счет гранта Российского научного фонда № 24-21-00483, https://rscf.ru/project/24-21-00483/

Введение. В современном мире эффективное управление *техническим обслуживанием, инженерной поддержкой и ремонтом* (ТОИР) является ключевым фактором для обеспечения бесперебойной работы промышленных и технических систем. Вместе с тем управление активами становится все более важным аспектом бизнеса, так как предприятия сталкиваются с растущей необходимостью оптимизации расходов, повышения производительности и увеличения надежности оборудования. Эффективность работы технических систем напрямую зависит от условий эксплуатации и поддержания работоспособного состояния.

Автоматизация процесса планирования технического обслуживания и ремонта оборудования на основе определения его остаточного ресурса дает возможность снизить риск аварийных ситуаций, минимизировать затраты на эксплуатационные расходы [1–3]. Современные подходы и методы анализа позволяют совершенствовать системы поддержки принятия решений путем внедрения механизмов прогнозирования развития событий и оценки риска неблагоприятного исхода, снижая количество отказов оборудования на основе оценки *остаточного ресурса оборудования* (*Remaining Useful Life,* RUL) [4].

Для управления активами в системах ТОИР RUL имеет решающее значение для оптимизации стратегий технического обслуживания, обеспечивая прогностическое планирование задач по ремонту [5]. Создание такого контроля способствует повышению маневренности действий предприятий ТОИР и своевременному реагированию на меняющееся состояние оборудования. Регулярное профилактическое обслуживание упрощается, так как система определяет начало процесса деградации, что приводит к оптимизации трудозатрат и снижению трудоемкости управления процессами.

Авторы исследовали работы, в которых описаны методики для анализа и прогнозирования временных рядов [6–8], и выделили четыре подхода к прогнозированию [9]: на основе моделей, данных, слияния и гибридных моделей. Стоит отметить, что все подходы, кроме моделирования, полагаются на собранные данные и включают методы, основанные на временных рядах, статистике и искусственном интеллекте. Наиболее общая классификация подходов [4] приведена на рисунке 1.

Модели прогнозирования, описывающие процесс деградации оборудования, построены на зависимости между показателями функционирования объекта и его состояниями. На основе анализа временных рядов строится алгоритм машинного обучения, способный для заданного набора параметров функционирования выдать ответ о состоянии устройства [10].

В последние годы все более актуальным становится направление по определению технического состояния с использованием машинного обучения [11]. Применение моделей прогнозирования временных рядов методами предиктивной аналитики [12, 13] с компонентами искусственного интеллекта лежит в основе проактивных систем принятия решений [14]. Для анализа временного ряда и построения прогноза широко применяются различные программные пакеты на основе предиктивной аналитики (https://loginom.ru/, https://www.knime.com/about, https://in-dap.ru/). Значимым критерием выбора, помимо наличия ограничений на бесплатное использование ПО, является тип лицензии, установленной разработчиком и отвечающей за возможность реализации дополнительных методов и интеграции предоставляемого пакета в собственное решение. Также стоит учитывать, что сборка, разворачивание и доработка свободного ПО требуют компетенций и значительных временных ресурсов.

Сложность методики построения моделей оценки RUL в проактивных системах принятия решений связана с ее долгосрочной реализацией. В качестве решения данной проблемы разработан фреймворк оценки и прогнозирования технического состояния оборудования (https://github.com/zadiran/Dataworks) для проведения исследований при анализе и прогнозировании временных рядов на основе анализа данных согласно методологии CRISP-DM.

Описание фреймворка

Разработка фреймворка на языке Python началась с идеи создания интуитивно понятной программной платформы, взаимодействие с которой не требует больших временных затрат для аналитиков. Это достигается за счет применения единого алгоритма решения задачи прогнозирования RUL, представленного на рисунке 2, отдельных параметризуемых и переопределяемых шагов алгоритма, имеющих общий интерфейс, а также единого интерфейса хранения и передачи данных между шагами алгоритма. Структура программной системы приведена на рисунке 3. При получении на вход данных выполняется их подготовка: предобработка исходных данных, преобразование в выходной формат (в виде списка точек) и применение дополнительной обработки. Готовые данные сохраняются в кэше. Следующим шагом является построение модели: инициализация модели, оптимизация гиперпараметров и сохранение обученных моделей. Обученные модели с применением кросс-вали-

Рис. 1. Классификация методов прогнозирования отказов оборудования

Fig. 1. Classification of equipment failure prediction methods

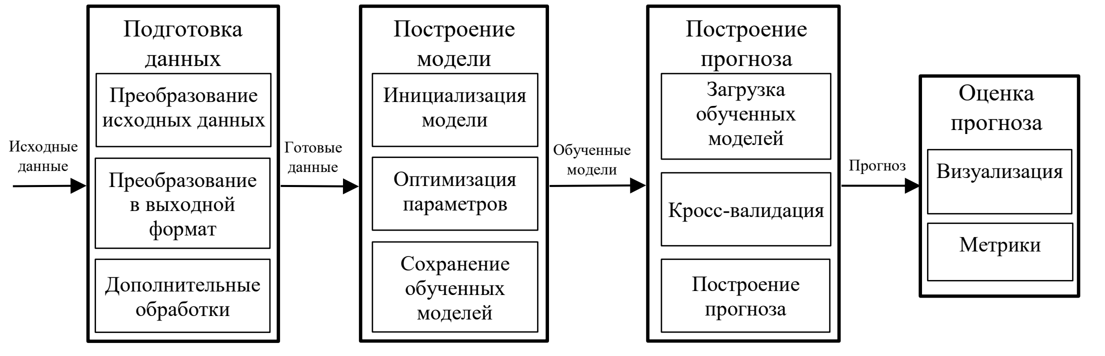

Рис. 2. Алгоритм построения прогноза

Fig. 2. Prediction algorithm

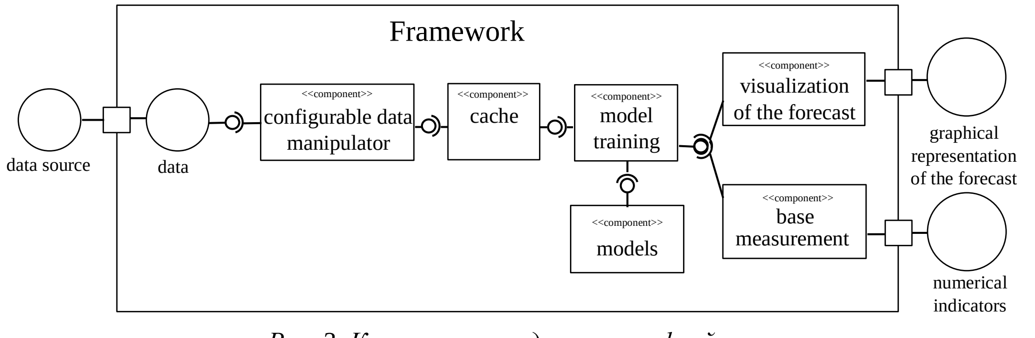

Рис. 3. Компонентная диаграмма фреймворка

Fig. 3. Component diagram of the framework

дации используются для построения прогноза и его оценки в виде расчета метрик и визуализации. Отметим, что можно переопределить каждый из базовых классов и реализовать свои собственные источники данных, стадии предобработки, модели прогнозирования и метрики.

Основная цель фреймворка – оценка технического состояния оборудования и его прогнозирование. Все функции фреймворка можно отнести к одной из следующих групп:

– загрузка и предобработка данных;
– построение моделей прогнозирования;
– прогнозирование временных рядов;
– оценка прогноза.

Рассмотрим их подробнее на примере кода по заданию параметров прогнозирования.

Шаг 1. Загрузка данных и формирование внутреннего фрейма данных.

Импортируем необходимые модули и функции из различных библиотек для обработки данных.

Для загрузки данных пользователь задает значения параметров *csv_data_source*:

‘filename’ – имя файла и путь к нему; ‘separator’ – разделитель.

Для удобства восприятия представим загружаемый исходный датасет в виде таблицы (табл. 1), где unit – экземпляр оборудования, time – порядковый номер записи в рамках unit. Последняя запись в рамках unit – момент отказа, то есть 1 unit – 1 отказ. Остальные параметры – данные о работе оборудования.

Шаг 2. Подбор параметров для методов предварительной обработки данных.

add_pre_point_conversion_stage() – добавление нового этапа предобработки в список.

set_point_conversion_stage() – этап преобразования точек.

add_post_point_conversion_stage() – добавление этапа постобработки в список.

Запускается метод обработки данных *get_processed_data* с установленными этапами.

Покажем выполнение шагов 1 и 2 на примере кода выполнения загрузки данных и настройки этапов их подготовки. Сначала устанавливаем ширину скользящего окна для последующего формирования точек типа point в процессе построения обучающей выборки [15].

Фрагмент кода:

```
   window_size = WS
   raw_data = csv_data_source().
get_data(.\data\DATA.csv', 'SEPARATOR')
   cdm = configurable_data_manipula-
tor(.\cache\
FILE_NAME.pickle')
```

Таблица 1

Представление исходного датасета

Table 1

Input dataset

Unit (идентификатор/ экземпляр оборудования)	Time (порядковый номер рабочего цикла)	COL 1 (номер параметра, снятого с оборудования)	…	COL N (номер параметра, снятого с оборудования)
1	1	value	…	value
1	…	value	…	value
1	192	value	…	value
2	1	value	…	value
2	…	…	…	…
2	287	value	…	value
3	1	value	…	value
…	…	…	…	…
100	200	value	…	value

```
   cdm.add_pre_point_conversion_
stage(drop_columns(['COL_1', 'COL_2',]))
   cdm.set_point_conversion_stage(CONVER-
TING_FORMAT(window_size))
   data = cdm.get_processed_
data(raw_data)
```

Рассмотрим классы, используемые в приведенном фрагменте кода.

data_source – абстрактный базовый класс, предоставляющий доступ к источнику данных:

get_data(*self,**params*) -> *List*[*point*]: – метод принимает произвольное количество именованных параметров (**params) и возвращает список объектов типа point.

csv_data_source – реализация класса data_source для CSV-файлов. В качестве параметров принимает имя файла и разделитель данных в файле CSV. Если в данных не указаны столбцы unit_col и time_col, генерируется исключение, иначе столбцы переименовываются для дальнейшей обработки в датафрейме.

configurable_data_manipulator – базовый класс, инициализирующий имя файла кэша и списки этапов обработки данных:

get_processed_data()->*List*[*point*]: – основной метод, принимающий на вход датафрейм и выполняющий преобразование исходных данных в список объектов point.

point – базовый класс для хранения данных точки временного ряда, содержащей информацию о входных данных, спрогнозированных результатах и дополнительных параметрах.

Шаг 3. Моделирование и прогнозирование. Фреймворк включает в себя единый базовый интерфейс для методов построения моделей прогнозирования временных рядов. На данный момент реализованы три метода: линейная регрессия (linear_regression_point_forecast_model), градиентный бустинг XGBoost (xgboost_point_forecast_model), сверточные нейронные сети (cnn_point_forecast_model).

Пример выполнения шага 3 – код обучения модели и прогнозирования с использованием кросс-валидации:

```
   def split_func(training_set, verifica-
tion_set):
        bbpfm = LEARNING_MODEL()
        bbpfm.fit(training_set)
        bbpfm.predict_points(verifica-
tion_set)
   splitter().run(split_func,
LIST[POINT], NUM_OF_PARTITIONS,
EXCLUSIONS)
```

splitter – базовый класс для применения кросс-валидации.

run (*func, data, num_of_partitions, exclusions*) – метод, принимающий на вход функцию (*func*), которую нужно выполнить на наборе данных (*data*), количество итераций и количество частей разбития (*num_of_partitions*) и список исключений (*exclusions*), которые не должны участвовать в процессе кросс-валидации, возвращает список результатов после применения функции *func* на каждом разбиении данных.

remove_exclusions () – метод для удаления объектов (unit) из списка исключений.

split() – метод, используемый для разделения данных на равные части, который возвращает список частей. Если разделение невозможно, возникает исключение.

join_df () – метод для каждой части данных, объединяющий остальные части в список после разделения.

hm_hm(): – метод, представляющий стратегию анализа временных рядов для выявления процесса деградации, используя критерий Краскела–Уоллиса.

Шаг 4. Задание мер точности.

Меры точности задаются списком измерений, фактическими и спрогнозированными значениями, которые передаются в *calculate_measurements_for_points*(). На данный момент реализованы следующие меры точности: MAE, RMSE, MSE, MdAE, MAPE, MdAPE, RMSPE, RMdSPE, MASE, MdASE, RMSSE, SMAPE, SMdAPE. Пример:

```
measurements = calculate_measure-
ments_for_points(LIST[BASE_MEASUREMENT],
LIST[POINT])
  for m in measurements:
      print(f '{m.name}: {m.value}')
```

base_measurement () – базовый абстрактный класс, используемый для вычисления метрик оценки качества прогнозирования между фактическими и прогнозными значениями временных рядов.

calculate (*self*, *actual_set*, *forecasted_set*): – метод этого класса принимает два набора данных: actual_set (фактические значения) и forecasted_set (прогнозные значения).

За визуализацию данных во фреймворке отвечает метод plot из библиотеки matplotlib.pyplot – стандартная функция построения графика. Пример визуализации:

```
  import matplotlib.pyplot as plt
  plt.plot(list(map(lambda x: x.train-
ing_output, List[point])))
  plt.plot(list(map(lambda x: x.fore-
casted_output, List[point])))
  plt.show()
```

Пример использования

Настоящий фреймворк применялся для анализа временных рядов на примере задач NASA (https://www.kaggle.com/datasets/behrad3d/nasa-cmaps) для определения оставшегося срока службы (RUL) в наборе тестовых данных. В частности, для решения задачи на данных NASA были установлены значения, представленные в таблице 2, в соответствии с функциями фреймворка и выделенными ранее шагами. Сначала производятся импорт необходимых модулей и функций из различных библиотек и загрузка данных из файла train_FD001.csv с агрегированными данными рабочих циклов [19]. Затем в качестве этапа предобработки данных удаляются малоинформативные для анализа столбцы и исходный датафрейм raw_data преобразуется в список точек data с использованием метода скользящего окна, при этом размер окна задан значением параметра, равным 10. Обучается модель линейной регрессии с применением кросс-валидации (разделение имеющегося набора данных на пять частей). Для оценки качества прогноза использованы метрики точности и средняя величина ошибки – RMSE и MAE соответственно. Результаты оценки и график с прогнозными значениями данных NASA приведены на рисунке 4 (синим цветом отмечены реальные значения, оранжевым – полученные в результате прогнозирования).

Проверка работоспособности фреймворка для анализа временных рядов показала его высокую эффективность и функциональность.

Таблица 2

Пример параметров задачи прогнозирования

Table 2

Example of prediction task parameters

Шаг	Параметр	Значение
1	window_size	10
	filename	train_FD001.csv
	separator	;
2	cdm	configurable_data_manipulator (.\cache\ all_input_continuous.pickle')
	cdm.add_pre_point_conversion_stage()	(drop_columns(['s3', 's4', 's8', 's9', 's13', 's19', 's21', 's22'])
	cdm.set_point_conversion_stage()	(convert_to_1d_input(window_size))
3	bbpfm	linear_regression_point_forecast_model()
	List[point]	data
	num_of_partitions	5
	exclusions	[]
4	List[base_measurement]	[rmse(), mae()]
	List[point]	data

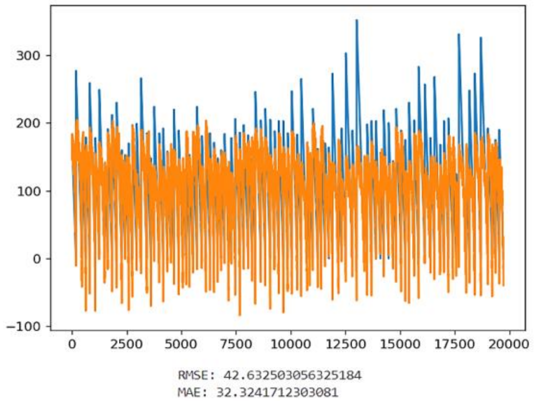

Рис. 4. Результаты прогнозирования на данных NASA

Fig. 4. Prediction results based on NASA data

Таким образом, фреймворк успешно загружает данные из источника, предоставляет широкий спектр инструментов для анализа и визуализации временных рядов, а также обладает гибкой архитектурой для добавления новых методов анализа и прогнозирования.

Заключение

Предложено программное решение на Python для анализа временных рядов, где данные представлены в виде таблиц с разделителями. Данный фреймворк построен на модульной архитектуре, что позволяет расширять его возможности. Исследователи могут легко работать с различными форматами данных и добавлять новые методы анализа и прогнозирования временных рядов по своему усмотрению.

Список литературы

1. Syreyshchikova N.V., Pimenov D.Yu., Mikolajczyk T., Moldovan L. Automation of production activities of an industrial enterprise based on the ERP system. Procedia Manufacturing, 2020, vol. 46, pp. 525–532. doi: 10.1016/j.promfg.2020.03.075.

2. Gómez-Pau Á., Riba J., Moreno-Eguilaz M. Time series RUL estimation of medium voltage connectors to ease predictive maintenance plans. Appl. Sci., 2020, vol. 10, no. 24, art. 9041. doi: 10.3390/app10249041.

3. Safoklov B., Prokopenko D., Deniskin Y., Kostyshak M. Model of aircraft maintenance repair and overhaul using artificial neural networks. Transportation Research Procedia, 2022, vol. 63, pp. 1534–1543. doi: 10.1016/j.trpro.2022.06.165.

4. Cheng J.C., Chen W., Chen K., Wang Q. Data-driven predictive maintenance planning frame-work for MEP components based on BIM and IoT using machine learning algorithms. Automation in Construction, 2020, vol. 112, art. 103087.

5. Wahid A., Breslin J.G., Intizar M.A. TCRSCANet: Harnessing temporal convolutions and recurrent skip component for enhanced RUL estimation in mechanical systems. Human-Centric Intelligent Systems, 2024, vol. 4, pp. 1–24. doi: 10.1007/s44230-023-00060-0.

6. Berghout T., Benbouzid M. A systematic guide for predicting remaining useful life with machine learning. Electronics, 2022, vol. 11, no. 7, art. 1125. doi: 10.3390/electronics11071125.

7. Pérez-Chacón R., Asencio-Cortés G., Troncoso A., Martínez-Álvarez F. Pattern sequence-based algorithm for multivariate big data time series forecasting: Application to electricity consumption. Future Generation Computer Systems, 2024, vol. 154, pp. 397–412. doi: 10.1016/j.future.2023.12.021.

8. Cardoso F.C., Berri R.A., Borges E.N. et al. Echo state network and classical statistical techniques for time series forecasting: A review. Knowledge-Based Systems, 2024, vol. 293, art. 111639. doi: 10.1016/j.knosys.2024.111639.

9. Gómez-Pau Á., Riba J.-R., Moreno-Eguilaz M. Time series RUL estimation of medium voltage connectors to ease predictive maintenance plans. Appl. Sci., 2020, vol. 10, no. 24, art. 9041. doi: 10.3390/app10249041.

10. Клячкин В.Н., Жуков Д.А. Прогнозирование состояния технического объекта с применением методов машинного обучения // Программные продукты и системы. 2019. Т. 32. № 2. С. 244–250. doi: 10.15827/0236-235X.126.244-250.

11. Sharma J., Mittal M.L., Soni G. Condition-based maintenance using machine learning and role of interpretability: A review. IJSAEM, 2024, vol. 15, pp. 1345–1360. doi: 10.1007/s13198-022-01843-7.

12. Соборнов Т.И., Ковалев И.О. Понятие и базовые методы предиктивной аналитики // Инновации. Наука. Образование. 2022. № 49. С. 1204–1207.

13. Терешина В.В. Применение систем предиктивной аналитики и предикативного моделирования // Инновационное развитие экономики. 2022. № 5. С. 243–246. doi: 10.51832/2223798420225243.

14. Щербаков М.В., Задиран К.С., Голубев А.В. и др. Фреймворк для анализа и прогнозирования временных рядов при разработке компонент проактивных систем поддержки принятия решений // Программные продукты и системы. 2017. Т. 30. № 3. С. 439–446. doi: 10.15827/0236-235X.119.439-446.

15. Задиран К.С. Новый метод прогнозирования остаточного ресурса оборудования для высокочастотных данных с неоднородной длительностью рабочих циклов // Изв. ЮФУ. Технические науки. 2023. № 4. С. 65–74.

Software & Systems doi: 10.15827/0236-235X.149.100-107 2025, 38(1), pp. 100–107

A framework for automating equipment remaining life prediction when building proactive decision support systems

Konstantin S. Zadiran [1], Darya A. Volkova [1]✉, Maksim V. Shcherbakov [1]

[1] Volgograd State Technical University, Volgograd, 400005, Russian Federation

For citation

Zadiran, K.S., Volkova, D.A., Shcherbakov, M.V. (2025) ‘A framework for automating equipment remaining life prediction when building proactive decision support systems’, *Software & Systems*, 38(1), pp. 100–107 (in Russ.). doi: 10.15827/0236-235X.149.100-107

Article info

Received: 08.05.2024 After revision: 24.06.2024 Accepted: 23.07.2024

Abstract. The paper describes a framework for automating research in designing proactive decision support systems. In particular, it investigates the problem of time series analysis and prediction in order to create tools for automating prediction of various processes in asset management systems, including maintenance and repair. The authors identify the role of automation processes in asset management in these systems. They highlight the main factors influencing the choice of a program to implement a predictive analytics system. The authors propose an algorithm for solving the problem of predicting remaining useful life based on analyzing production asset data using artificial intelligence components. The proposed software solution is based on CRISP-DM. It is not a separate software product and can be embedded in existing software to support the possibility of modifying methods. The framework loads and preprocesses data, builds predicting models, predicts time series and evaluates the forecast. The developed framework has a flexible modular architecture for adding new methods of analysis and predicting. The possibility to redefine and implement own data sources, preprocessing stages, forecasting models and metrics on the basis of existing base classes extends the variability and increases the efficiency of its functioning. There is the example of using the framework to solve the problem of analyzing time series and determining equipment remaining useful life. It demonstrates the efficiency of the developed product in the field of data exploration and artificial intelligence.

Keywords: software framework, time series forecasting, time series analysis, decision support systems, RUL, predicting models

Acknowledgements. The research was funded by a grant from the Russian Science Foundation no. 24-21-00483, https://rscf.ru/project/24-21-00483/

References

1. Syreyshchikova, N.V., Pimenov, D.Yu., Mikolajczyk, T., Moldovan, L. (2020) ‘Automation of production activities of an industrial enterprise based on the ERP system’, *Procedia Manufacturing*, 46, pp. 525–532. doi: 10.1016/j.promfg.2020.03.075.
2. Gómez-Pau, Á., Riba, J., Moreno-Eguilaz, M. (2020) ‘Time series RUL estimation of medium voltage connectors to ease predictive maintenance plans’, *Appl. Sci*., 10(24), art. 9041. doi: 10.3390/app10249041.
3. Safoklov, B., Prokopenko, D., Deniskin, Y., Kostyshak, M. (2022) ‘Model of aircraft maintenance repair and overhaul using artificial neural networks’, *Transportation Research Procedia*, 63, pp. 1534–1543. doi: 10.1016/j.trpro.2022.06.165.
4. Cheng, J.C., Chen, W., Chen, K., Wang, Q. (2020) ‘Data-driven predictive maintenance planning framework for MEP components based on BIM and IoT using machine learning algorithms’, *Automation in Construction*, 112, art. 103087.
5. Wahid, A., Breslin, J.G., Intizar, M.A. (2024) ‘TCRSCANet: Harnessing temporal convolutions and recurrent skip component for enhanced RUL estimation in mechanical systems’, *Human-Centric Intelligent Systems*, 4, pp. 1–24. doi: 10.1007/s44230-023-00060-0.
6. Berghout, T., Benbouzid, M. (2022) ‘A systematic guide for predicting remaining useful life with machine learning’, *Electronics*, 11(7), art. 1125. doi: 10.3390/electronics11071125.
7. Pérez-Chacón, R., Asencio-Cortés, G., Troncoso, A., Martínez-Álvarez, F. (2024) ‘Pattern sequence-based algorithm for multivariate big data time series forecasting: Application to electricity consumption’, *Future Generation Computer Systems*, 154, pp. 397–412. doi: 10.1016/j.future.2023.12.021.
8. Cardoso, F.C., Berri, R.A., Borges, E.N. et al. (2024) ‘Echo state network and classical statistical techniques for time series forecasting: A review’, *Knowledge-Based Systems*, 293, art. 111639. doi: 10.1016/j.knosys.2024.111639.
9. Gómez-Pau, Á., Riba, J.-R., Moreno-Eguilaz, M. (2020) ‘Time series RUL estimation of medium voltage connectors to ease predictive maintenance plans’, *Appl. Sci*., 10(24), art. 9041. doi: 10.3390/app10249041.
10. Klyachkin, V.N., Zhukov, D.A. (2019) ‘Forecasting the state of a technical object using machine learning methods’, *Software & Systems*, 32(2), pp. 244–250 (in Russ.). doi: 10.15827/0236-235X.126.244-250.
11. Sharma, J., Mittal, M.L., Soni, G. (2024) ‘Condition-based maintenance using machine learning and role of interpretability: A review’, *IJSAEM*, 15, pp. 1345–1360. doi: 10.1007/s13198-022-01843-7.
12. Sobornov, T.I., Kovalev, I.O. (2022) ‘The concept and basic methods of predictive analytics’, *Innovation. Science. Education*, (49), pp. 1204–1207 (in Russ.).

13. Tereshina, V.V. (2022) ‘Application of predictive analytics and predicative modeling systems’, *Innovative Economic Development*, (5), pp. 243–246 (in Russ.). doi: 10.51832/2223798420225243.

14. Shcherbakov, M.V., Zadiran, K.S., Golubev, A.V. et al. (2017) ‘A framework for analysis and forecasting of time series in the development of proactive decision support system components’, *Software & Systems*, 30(3), pp. 439–446 (in Russ.). doi: 10.15827/0236-235X.119.439-446.

15. Zadiran, K.S. (2023) ‘A new method for predicting the remaining equipment life for high-frequency data with non-uniform duty cycles’, *Izv. SFedU. Eng. Sci.*, (4), pp. 65–74 (in Russ.).

Авторы
Задиран Константин Сергеевич [1],
аспирант, konstantin.zadiran@gmail.com
Волкова Дарья Александровна [1],
студент, d_volkova@vstu.ru
Щербаков Максим Владимирович [1], д.т.н.,
профессор, зав. кафедрой,
maxim.shcherbakov@vstu.ru

[1] Волгоградский государственный технический университет, г. Волгоград, 400005, Россия

Authors
Konstantin S. Zadiran [1], Postgraduate Student,
konstantin.zadiran@gmail.com
Darya A. Volkova [1],
Student, d_volkova@vstu.ru
Maksim V. Shcherbakov [1], Dr.Sci. (Engineering),
Professor, Head of Chair,
maxim.shcherbakov@vstu.ru

[1] Volgograd State Technical University, Volgograd, 400005, Russian Federation