

Анализ методов интеграции моделей машинного обучения в SCADA- системы

Н.В. Никитин, П.Г. Фрасын, Д.В. Масанов

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

Аннотация: В статье рассмотрены методы интеграции моделей машинного обучения в SCADA-системы. Проанализированы архитектурные подходы, обеспечивающие сопряжение интеллектуальных алгоритмов с автоматизированными системами управления технологическими процессами. Особое внимание уделено практическим аспектам применения открытых протоколов обмена данными — OPC UA и MQTT — для организации двусторонней связи между SCADA и внешними аналитическими модулями. Описана реализованная архитектура, включающая механизмы агрегации телеметрических данных, их трансформации в формат, пригодный для подачи в модель, и обратной интеграции прогнозных результатов в интерфейс операторской визуализации. Представленный подход обеспечивает модульность, масштабируемость и воспроизводимость решений, что позволяет адаптировать его к различным объектам промышленной автоматизации.

Ключевые слова: SCADA, машинное обучение, интеграция, телеметрия, прогнозная аналитика, OPC UA, MQTT, АСУТП, предиктивные модели, агрегация данных.

Введение

В современных автоматизированных системах управления технологическими процессами (АСУТП) существенно возрастают объёмы данных, которые регистрируются и хранятся при помощи телеметрических комплексов. Ярким примером является технологический процесс водоподготовки, где показатели датчиков (давление, расход, мощность, температура и др.) с высокой частотой фиксируются в базе исторических данных. Несмотря на увеличение объёмов доступной информации, комплексный анализ накопленных сведений на практике часто оказывается затруднён или вовсе не выполняется. В результате ценные закономерности в технологических процессах остаются невыявленными, а потенциал оптимизации — нераскрытым [1].

Отсутствие структурированного подхода и унифицированных методологий приводит к тому, что инженерно-технические решения по

эксплуатации и настройке оборудования принимаются на основе локальных наблюдений и эмпирического опыта.

Однако методы машинного обучения (ML) способны выявлять скрытые закономерности и оптимизировать технологические процессы, но их потенциал полностью раскрывается лишь при интеграции в действующие SCADA-системы, позволяя применять более сложную систему поддержки управления в реальном времени. Такая система позволит оперативно визуализировать предсказания, формировать рекомендации для операторов и, в перспективе, автоматизировать управление технологическим оборудованием на основе результатов ансамбля моделей машинного обучения.

Цель данной статьи — исследование существующих подходов и концепций к внедрению сложных алгоритмов предиктивной логики и моделей машинного обучения с последующим внедрением модели анализа эффективности оборудования водоподготовки в SCADA-систему, реализованную на базе фреймворка Flask.

Анализ архитектурных подходов к интеграции

Традиционные системы SCADA и современные инструменты машинного обучения изначально разрабатывались независимо друг от друга, что привело к возникновению ряда системных несовместимостей. Во многих случаях SCADA-системы являются проприетарными решениями с жесткими ограничениями по расширению функционала, закладываемыми разработчиками непосредственно при создании инструмента, что существенно осложняет добавление внешних, самодельных модулей. Дополнительным фактором является отсутствие унифицированных стандартов и методологий для интеграции моделей машинного обучения в АСУТП, из-за чего каждое внедрение приобретает уникальный, индивидуальный характер, в частности,

выбор технологии интеграции. Например, выбор между OPC UA, MQTT или прямым доступом к БД напрямую зависит от особенностей уже установленного оборудования и программного обеспечения на конкретном объекте, что исключает возможность универсального подхода.

Интеграция ML-моделей в SCADA требует гибкой, масштабируемой архитектуры, позволяющей объединить SCADA с ИИ [2]. Как правило, выделяются распределенные схемы, где модель работает во внешнем сервисе, и встроенные, когда алгоритм встраивается непосредственно в функционал SCADA. В обоих случаях необходимы стандартные интерфейсы и протоколы, через которые алгоритмы ML могут получать доступ к данным SCADA и при необходимости передавать команды обратно в систему управления.

Один распространенный подход – обмен данными между SCADA и моделью посредством стандарта OPC UA. SCADA-система может выступать OPC-клиентом, а внешний ML-сервис – OPC-сервером или наоборот, что позволяет передавать измерения на вход модели и получать результаты как OPC-теги.

В научной работе “Low-cost SCADA/HMI with Tiny Machine Learning for Monitoring Indoor CO2 Concentration” авторов Wardana I. N. K., Fahmy S. A., Gardner J. W. представлена интеграция алгоритма ML на уровне мини устройств с SCADA через OPC-протокол [3].

В этом проекте три микроконтроллера Arduino служат удаленными устройствами, собирающими данные и сразу вычисляющими прогноз. Обмен с верхним уровнем организован через OPC-сервер: на Arduino загружена библиотека OPC для обмена данными с VTScada. Данный пример демонстрирует, что OPC UA обеспечивает стандартный “канал” для двусторонней связи: SCADA может как читать результаты ML-модели, встроенной в контроллеры, так и при необходимости отправлять исходные данные или команды.

В научной работе “A Study on an IoT-Based SCADA System for Photovoltaic Utility Plants ” [4] представлена интеграция алгоритма ML. Для крупных солнечных электростанций была разработана архитектура, в которой к традиционному подходу добавлен модуль машинного обучения, выполняющий прогнозирование генерации электроэнергии. Данные телеметрии — такие как напряжение, ток, температура панелей, погодные параметры и другие — публикуются контроллерами в MQTT токипк. Оттуда эту информацию получают модуль хранения и технического обслуживания (O&M), ответственный за регистрацию и запись исторических данных и ML-пайплан, осуществляющий прогноз выработки и выявление отклонений.

Результаты анализа, включая прогнозируемую мощность и возможные отклонения, также публикуются обратно в MQTT. SCADA-система, подписанная на соответствующие топики, отображает как текущие параметры, так и выводы ML-модели — например, предупреждение о снижении эффективности из-за загрязнения солнечных панелей.

В качестве брокера используется EMQX для обеспечения отказоустойчивости и масштабируемости. Отмечается, что выбор MQTT обусловлен его стабильной работой в промышленных условиях и широкой поддержкой среди оборудования и ПО, что значительно упрощает интеграцию компонентов.

Несмотря на отсутствие строго унифицированных методик интеграции моделей машинного обучения в SCADA-системы, современные подходы демонстрируют формирование повторяющихся архитектурных решений, которые могут рассматриваться в качестве де-факто стандартов [5]. К ним относятся использование внешних модулей и брокеров сообщений, обеспечивающих обмен данными между компонентами АСУТП и интеллектуальными сервисами. Такие схемы позволяют обеспечить модульность, масштабируемость и переносимость решений.

Наиболее распространёнными способами организации взаимодействия являются применение протокола OPC UA и использование брокеров MQTT [6]. OPC UA позволяет наладить прямой обмен телеметрией и управляющими командами между SCADA-системой и внешними вычислительными сервисами, в том числе встроенными модулями на базе микроконтроллеров. В свою очередь, MQTT демонстрирует высокую устойчивость и гибкость при передаче данных в распределённых промышленных системах, особенно в задачах, связанных с предиктивной аналитикой и мониторингом на уровне крупных промышленных объектов.

Таким образом, можно утверждать, что архитектура, основанная на использовании стандартных протоколов и интерфейсов, в частности OPC UA и MQTT, представляет собой практически применимый и воспроизводимый подход для интеграции ИИ-моделей в SCADA-системы, независимо от специфики объекта автоматизации.

Анализ данных, построение признаков и выбор модели

На этапе построения модели была сформирована единая методика, включающая анализ исходных данных, генерацию признаков, выбор алгоритма и оценку его точности. Исходная выборка включала значения, охватывающие период в шесть месяцев непрерывной регистрации показателей с объектов водозабора [7].

Для обучения модели были выбраны признаки, отражающие поведение оборудования за предыдущие интервалы. В частности, использовались агрегированные значения таких параметров, как давление, температура, расход и мощность за определенные интервалы времени, что позволило модели учитывать краткосрочные и среднесрочные тренды.

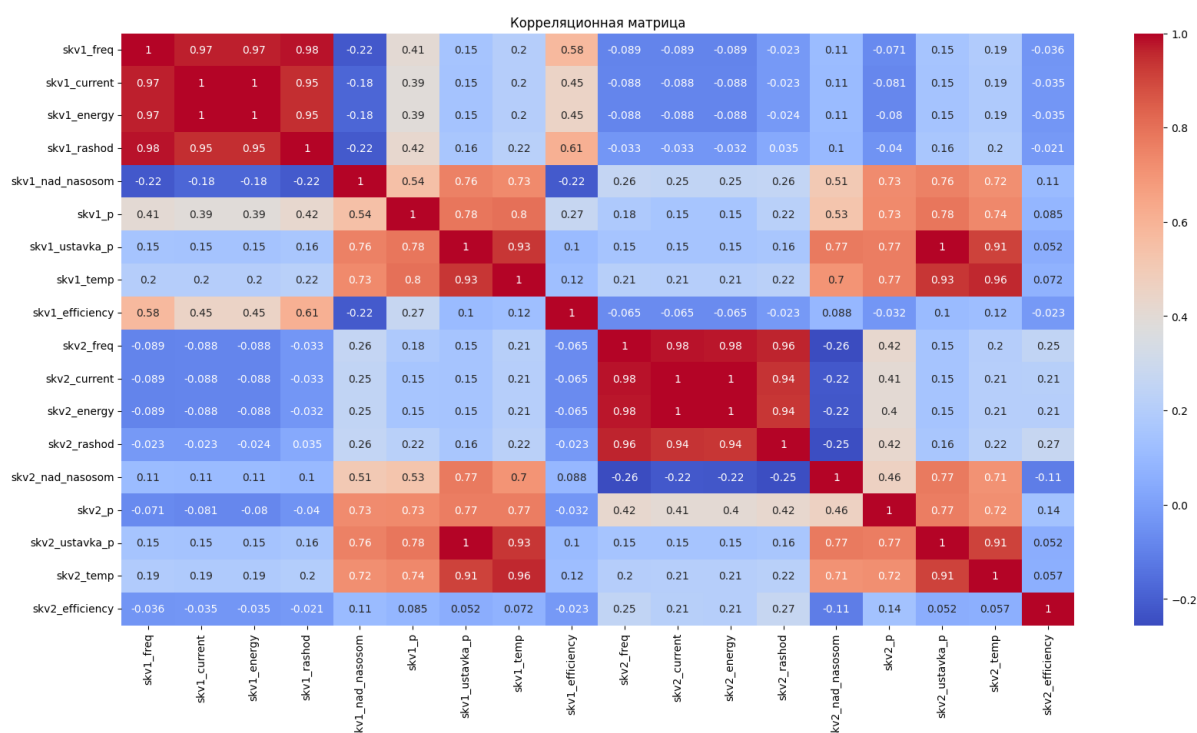


Рис 1. – График Heatmap корреляционной матрицы

Дополнительно применялись скользящие статистики, а также производные признаки, такие как скорость изменения параметров и взаимные соотношения между ними.

Для обработки и подготовки данных использовались инструменты библиотеки pandas, а для визуального анализа — matplotlib и seaborn. Данные инструменты позволили выявить аномалии, выбросы и корреляции между параметрами, что стало основой для выбора наиболее информативных признаков. Особое внимание уделялось качеству данных: отсутствующие значения заполнялись с использованием методов интерполяции [8].

В качестве базовых моделей были протестированы алгоритмы линейной регрессии, градиентного бустинга и рекуррентные нейронные сети для работы с временными рядами. По результатам кросс-валидации на тестовой выборке наиболее стабильные и интерпретируемые результаты показала модель на основе LightGBM, обеспечившая баланс между точностью и скоростью предсказаний, что критично для работы в реальном времени.

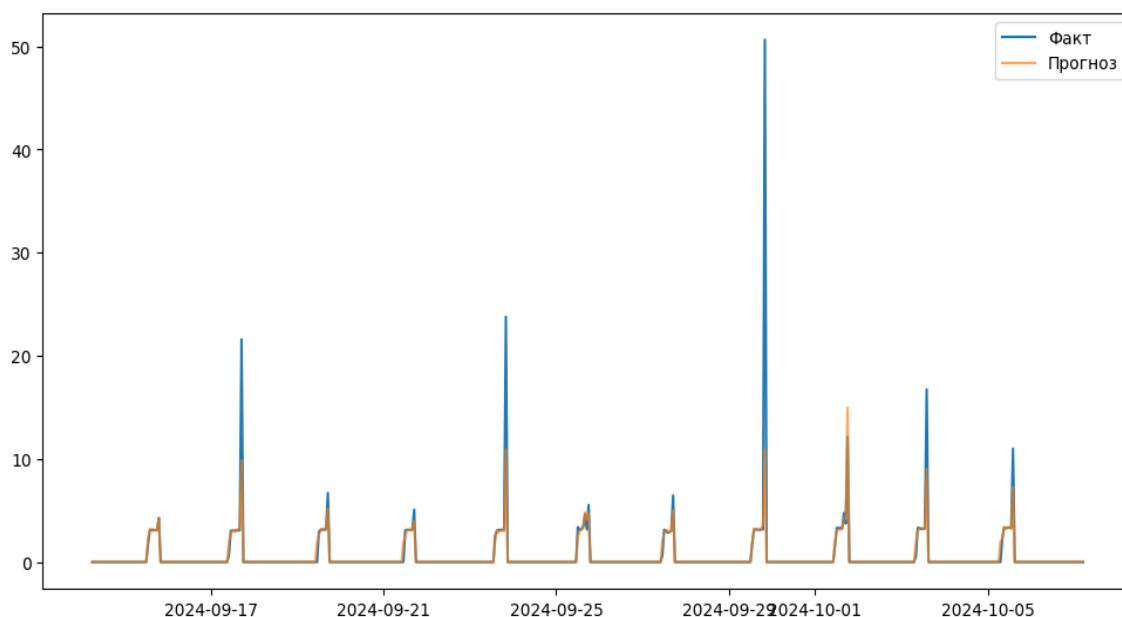


Рис 2. – График фактических и предсказанных значений модели

После выбора модели была выполнена её оптимизация и сериализация с использованием библиотеки `joblib`, что обеспечит более надежное хранение результатов возможность дообучить модель и удобно интегрировать ее в систему мониторинга.

Интеграция модели машинного обучения в веб-интерфейс SCADA

Для реализации прогнозной поддержки принятия решений в рамках SCADA-системы была разработана архитектура, обеспечивающая бесшовную интеграцию модели машинного обучения в веб-интерфейс мониторинга [9].

В данном исполнении Web SCADA данные поступают на мониторинг с помощью `mqtt` топика.



Рис 3. – Старая структура перемещения данных

Базовая архитектура Web SCADA предполагает поступление телеметрических данных в реальном времени через MQTT-протокол. Основной поток «сырых» данных, публикуемый в топик `raw/data`, содержит

мгновенные значения параметров оборудования. Эти данные визуализируются на стороне клиента через SocketIO в виде цифровых индикаторов и графиков [10].

Для обеспечения корректной работы модели, входные данные должны представлять собой агрегированные метрики за заданный временной интервал. Для этого классический пайплайн будет заменен на предоставленный на рисунке 4.

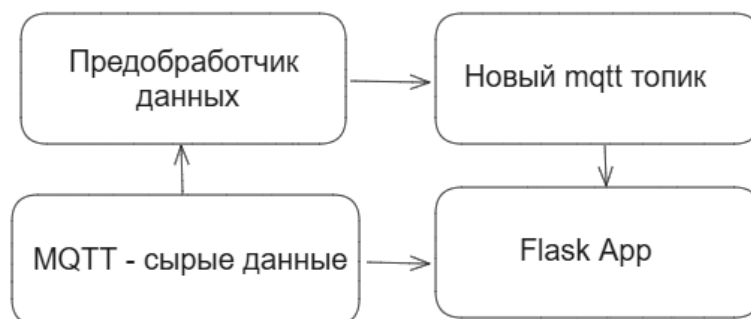


Рис 4. – Новая структура перемещения данных

В новую архитектуру был добавлен специализированный компонент — автономный Python-скрипт, выполняющий агрегацию телеметрических данных. Этот скрипт подписывается на топик `raw/data`, накапливает значения и рассчитывает статистические признаки: средние значения, максимумы, стандартные отклонения и производные. После вычисления признаки объединяются в структурированный JSON-объект и публикуются во второй топик — `ml/aggregated`, который предназначен исключительно для подачи входных данных в модель.

На стороне Flask-приложения реализован отдельный фоновый MQTT-клиент, подписанный на `ml/aggregated`.

```
1  def aggregate_and_publish(client):
2      df = pd.DataFrame(buffer)
3  if df.empty:
4      return
5
6  result = {
7      "avg_pressure": df["pressure"].mean(),
8      "max_pressure": df["pressure"].max(),
9      "std_pressure": df["pressure"].std(),
10     "delta_pressure": df["pressure"].iloc[-1] - df["pressure"].iloc[0],
11     "pressure_trend": np.polyfit(range(len(df)), df["pressure"], 1)[0],
12
13     "avg_temp": df["temperature"].mean(),
14     "std_temp": df["temperature"].std(),
15     "delta_temp": df["temperature"].iloc[-1] - df["temperature"].iloc[0],
16
17     "avg_flow": df["flow"].mean(),
18     "std_flow": df["flow"].std(),
19     "delta_flow": df["flow"].iloc[-1] - df["flow"].iloc[0],
20     "flow_lag_1": df["flow"].iloc[-2] if len(df) >= 2 else None,
21     "flow_rel_change": (
22         (df["flow"].iloc[-1] - df["flow"].iloc[0]) / df["flow"].iloc[0]
23         if df["flow"].iloc[0] != 0 else 0
24     ),
25
26     "timestamp": time.time(),
27     "hour_of_day": time.localtime().tm_hour,
28     "weekday": time.localtime().tm_wday
29 }
30
31 client.publish(AGG_TOPIC, json.dumps(result))
32 print("Отправлены агрегированные данные:", result)
```

Рис 5. – Код алгоритма агрегации данных

При поступлении новых данных происходит формирование вектора признаков и модели машинного обучения, предварительно загруженной в память. Результат предсказания, включая категориальную оценку состояния и сгенерированные рекомендации. Они передаются через WebSocket на клиентскую часть SCADA-интерфейса. Такой подход позволяет в реальном времени обновлять данные на сайте, без использования авто обновлений страницы, а также позволит интегрировать обновления без перезапуска самого алгоритма SCADA-системы.

```
# ml_inference.py
import json
import paho.mqtt.client as mqtt
import joblib

from flask_socketio import SocketIO

BROKER = "localhost"
PORT = 1883
TOPIC = "ml/aggregated"

model = joblib.load("models/model.pkl")

socketio: SocketIO = None

def on_connect(client, userdata, flags, rc):
    print("Инференс-модуль подключён к MQTT")
    client.subscribe(TOPIC)

def on_message(client, userdata, msg):
    try:
        payload = json.loads(msg.payload.decode())
        features = preprocess(payload)
        prediction = model.predict([features])[0]

        result = {
            "prediction": prediction,
            "status": map_prediction(prediction),
            "input": payload
        }

        print("● Предсказание:", result)
        socketio.emit("prediction", result)

    except Exception as e:
        print("Ошибка инференса:", e)

def map_prediction(pred):
    mapping = {
        0: "Норма",
        1: "Отклонение",
        2: "Риск сбоя"
    }
    return mapping.get(pred, "Неизвестно")

def start_inference_mqtt(sock: SocketIO):
    global socketio
    socketio = sock

    client = mqtt.Client()
    client.on_connect = on_connect
    client.on_message = on_message
    client.connect(BROKER, PORT, 60)
    client.loop_forever()
```

Рис 6. – Код основного алгоритма работы модели

Клиентская часть SCADA-системы обеспечивает визуализацию результатов модели машинного обучения в режиме реального времени. Результаты данной модели так же записываются в базу данных, чтобы можно было сравнить результат и внедрить последующую корректировку модели.



Статус	Класс	Рекомендация
Ожидание...		Ожидание данных от модели...

Рис 6. – Визуальное отображение результата модели в SCADA

Прогнозируемое состояние оборудования представляется в виде цифровых индикаторов, дополненных цветовой индикацией уровня риска (нормальное состояние, предупреждение, критическое отклонение), а также текстовыми пояснениями, основанными на классификационном или вероятностном выходе модели.

Вывод

В данной работе представлена комплексная реализация подхода к интеграции моделей машинного обучения в SCADA-систему с использованием открытых и воспроизводимых решений. Особое внимание уделено анализу возможных архитектурных схем, среди которых выделены два наиболее жизнеспособных направления — внедрение через протокол OPC UA и публикация агрегированных данных через MQTT.

Сформированная модель, базирующаяся на алгоритме градиентного бустинга LightGBM, получила возможность функционировать в режиме реального времени благодаря внедрению специализированного механизма агрегации данных. Этот механизм реализован в виде автономного Python-скрипта, который подписывается на поток сырых телеметрических данных, рассчитывает агрегированные признаки и публикует их во внутренний MQTT-канал, предназначенный для подачи данных на вход модели. Такая архитектура позволила сформировать устойчивую инфраструктуру для прогнозной аналитики без необходимости модификации основной SCADA-системы.

Пользовательская часть системы обеспечивает визуализацию текущего состояния оборудования, отображая предсказания модели в виде цифровых индикаторов и цветowych меток уровней риска, сопровождаемых текстовыми рекомендациями.

В результате предложенная система, позволяет с легкостью интегрировать инструменты статистической обработки, алгоритмы машинного обучения в современные средства визуализации, адаптируясь под нее, не меняя логику программы.

Литература

1. Макаров Д.А., Шибанова А.Д. Алгоритмы машинного обучения // Теория и практика современной науки. 2017. № 1. С. 12–20. URL: cyberleninka.ru/article/n/algoritmy-mashinnogo-obucheniya (дата обращения: 22.04.2025)
 2. Garbaruk V.V., Fomenko V.N., Kupriyanov A.S. Statistical model of time series // Intellectual Technologies in Transport. 2017. № 2. С. 35–42. URL: cyberleninka.ru/article/n/statistical-model-of-time-series (дата обращения: 22.04.2025)
 3. Wardana I. N. K., Fahmy S. A., Gardner J. W. Low-cost SCADA/HMI with Tiny Machine Learning for Monitoring Indoor CO₂ Concentration // Proc. 2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2024). 2024. URL: warwick.ac.uk/fac/sci/eng/people/suhaib_fahmy/publications/i2mtc2024-wardana.pdf#:~:text=We%20utilized%20the%20Arduino%20OPC,19 (дата обращения: 22.04.2025)
 4. Ferlito S., Ippolito S., Santagata C., Schiattarella P., Di Francia G. A Study on an IoT-Based SCADA System for Photovoltaic Utility Plants // Electronics. 2024. Vol. 13. № 11. Art. 2065. DOI: 10.3390/electronics13112065. URL: mdpi.com/2079-9292/13/11/2065#:~:text=the%20MQTT%20broker,pursued%20using%20a%20monolithic%20solution (дата обращения: 22.04.2025)
 5. Rustamova S., Rustamov F. Problems of integrating artificial intelligence with SCADA systems // ScienceRise. 2024. № 2. С. 106–117. DOI: 10.21303/2313-8416.2024.003691. URL: journal.eu-jr.eu/sciencerise/article/view/3691#:~:text=The%20main%20scientific%20results%3A%20AI, costs%20while%20improving%20system%20performance (дата обращения: 22.04.2025)
-

6. Аунг Зеар, Михайлов И.С. Применение машинного обучения для повышения эффективности управления нефтяными скважинами // Евразийский Союз Ученых. 2019. № 3. С. 50–57. URL: cyberleninka.ru/article/n/primenenie-mashinnogo-obucheniya-dlya-povysheniya-effektivnosti-upravleniya-neftyanyimi-skvazhinami (дата обращения: 22.04.2025)
 7. Макаров Д.А., Шибанова А.Д. Принципы машинного обучения // Теория и практика современной науки. 2018. № 1. С. 78–85. URL: cyberleninka.ru/article/n/printsipy-mashinnogo-obucheniya (дата обращения: 22.04.2025)
 8. Бушмелева К.И., Васильчук А.Н. Аспекты машинного обучения в крупной компании нефтегазовой отрасли // Вестник кибернетики. 2018. № 5. С. 92–99. URL: cyberleninka.ru/article/n/aspekty-mashinnogo-obucheniya-v-krupnoy-kompanii-neftegazovoy-otrasli (дата обращения: 22.04.2025)
 9. Дмитриевский А. Н., Сбоев А. Г., Еремин Н. А., Черников А. Д., Наумов А. В. Об увеличении продуктивного времени бурения нефтегазовых скважин с использованием методов машинного обучения // Георесурсы. 2020. № 2. С. 101–109. URL: cyberleninka.ru/article/n/ob-uvelichenii-produktivnogo-vremeni-bureniya-neftegazovyh-skvazhin-s-ispolzovaniem-metodov-mashinnogo-obucheniya (дата обращения: 22.04.2025)
 10. Shah Rohin, Irpan Alex, Turner Alexander Matt, Wang Anna, Conmy Arthur, Lindner David, Brown-Cohen Jonah, Ho Lewis, Nanda Neel, Popa Raluca Ada, Jain Rishub Rishub, Greig Rory, Albanie Samuel, Emmons Scott, Farquhar Sebastian, Krier Sébastien, Rajamanoharan Senthooan, Bridgers Sophie, Ijitoye Tobi, Everitt Tom, Krakovna Victoria, Varma Vikrant, Mikulik Vladimir, Kenton Zachary, Orr Dave, Legg Shane, Goodman Noah, Dafoe Allan, Flynn Four, Dragan Anca An Approach to Technical AGI Safety
-

and Security // arXiv. 2025. URL: arxiv.org/abs/2504.01849 (дата обращения: 22.04.2025)

References

1. Makarov D.A., Shibanova A.D. Teoriya i praktika sovremennoj nauki. 2017. No. 1. pp. 12–20. URL: cyberleninka.ru/article/n/algoritmy-mashinnogo-obucheniya (accessed: 22.04.2025)
2. Garbaruk V.V., Fomenko V.N., Kupriyanov A.S. Intellectual Technologies in Transport. 2017 No. 2. pp. 35–42. URL: cyberleninka.ru/article/n/statistical-model-of-time-series (accessed: 22.04.2025)
3. Wardana I.N.K., Fahmy S.A., Gardner J.W. Proceedings of the 2024 IEEE International Instrumentation and Measurement Technology Conference (I2MTC 2024). 2024. URL: warwick.ac.uk/fac/sci/eng/people/suhaib_fahmy/publications/i2mtc2024-wardana.pdf (accessed: 22.04.2025)
4. Ferlito S., Ippolito S., Santagata C., Schiattarella P., Di Francia G. Electronics. 2024. Vol. 13. No. 11. Article 2065. DOI: 10.3390/electronics13112065. URL: mdpi.com/2079-9292/13/11/2065 (accessed: 22.04.2025)
5. Rustamova S., Rustamov F. ScienceRise. 2024. No. 2. pp. 106–117. DOI: 10.21303/2313-8416.2024.003691. URL: journal.eu-jr.eu/sciencerise/article/view/3691 (accessed: 22.04.2025)
6. Aung Zear, Mikhaylov I.S. Evrazijskij soyuz uchyonykh. 2019 No. 3. pp. 50–57. URL: cyberleninka.ru/article/n/primenenie-mashinnogo-obucheniya-dlya-povysheniya-effektivnosti-upravleniya-neftyanyimi-skvazhinami (accessed: 22.04.2025)
7. Makarov D.A., Shibanova A.D. Teoriya i praktika sovremennoj nauki. 2018 No. 1. pp. 78–85. URL: cyberleninka.ru/article/n/printsiipy-mashinnogo-obucheniya (accessed: 22.04.2025)

8. Bushmeleva K.I., Vasil'chuk A.N. Vestnik kibernetiki. 2018 No. 5. pp. 92–99. URL: cyberleninka.ru/article/n/aspekty-mashinnogo-obucheniya-v-krupnoy-kompanii-neftegazovoy-otrasli (accessed: 22.04.2025)
9. Dmitrievskij A.N., Sboev A.G., Eremin N.A., Chernikov A.D., Naumov A.V. Georesursy. 2020 No. 2. pp. 101–109. URL: cyberleninka.ru/article/n/ob-uvvelichenii-produktivnogo-vremeni-bureniya-neftegazovyh-skvazhin-s-ispolzovaniem-metodov-mashinnogo-obucheniya (accessed: 22.04.2025)
10. Shah Rohin, Irpan Alex, Turner Alexander Matt, Wang Anna, Conmy Arthur, Lindner David, Brown-Cohen Jonah, Ho Lewis, Nanda Neel, Popa Raluca Ada, Jain Rishub Rishub, Greig Rory, Albanie Samuel, Emmons Scott, Farquhar Sebastian, Krier Sébastien, Rajamanoharan Senthoooran, Bridgers Sophie, Ijitoeye Tobi, Everitt Tom, Krakovna Victoria, Varma Vikrant, Mikulik Vladimir, Kenton Zachary, Orr Dave, Legg Shane, Goodman Noah, Dafoe Allan, Flynn Four, Dragan Anca An Approach to Technical AGI Safety and Security. 2025. URL: arxiv.org/abs/2504.01849 (accessed: 22.04.2025)

Дата поступления: 4.04.2025

Дата публикации: 25.05.2025