

УДК 658.5.012.7

БИБЛИОТЕКА SOLARA ДЛЯ АНАЛИТИКИ ПРОИЗВОДСТВЕННЫХ ПРОЦЕССОВ: ОПТИМИЗАЦИЯ И ВИЗУАЛИЗАЦИЯ ДАННЫХ С ИСПОЛЬЗОВАНИЕМ APACHE SUPERSET

Андреева Кристина Александровна, студент, Базовая кафедра «Аналитика больших данных и методы видеоанализа», Уральский федеральный университет имени первого Президента России Б.Н. Ельцина

Солод Анастасия Васильевна, студент, Базовая кафедра «Аналитика больших данных и методы видеоанализа», Уральский федеральный университет имени первого Президента России Б.Н. Ельцина

Балунгу Даниель Мусафири, ассистент, Базовая кафедра «Аналитика больших данных и методы видеоанализа», Уральский федеральный университет имени первого Президента России Б.Н. Ельцина

Аннотация

Статья посвящена совместному использованию Python-фреймворка Solara и BI-платформы Apache Superset для анализа данных промышленных систем (SCADA, MES, ERP, IoT). Solara применяется для создания интерактивных мнемосхем, а Superset — для аналитических дашбордов. Комбинация этих инструментов повышает гибкость обработки данных и эффективность визуализации, что способствует операционной прозрачности и оптимизации решений в условиях Industry 4.0.

КЛЮЧЕВЫЕ СЛОВА: Solara, аналитика, производственные процессы, Apache Superset, визуализация данных.

SOLARA LIBRARY FOR PROCESS ANALYTICS: OPTIMIZATION AND DATA VISUALIZATION USING APACHE SUPERSET

Andreeva Kristina Aleksandrovna, Student, Basic Department «Big Data Analytics and Video Analysis Methods», Ural Federal University named after the first President of Russia B.N. Yeltsin

Solod Anastasia Vasil'evna, Student, Basic Department «Big Data Analytics and Video Analysis Methods», », Ural Federal University named after the first President of Russia B.N. Yeltsin

Balungu Daniel Musafiri, Assistant, Basic Department «Big Data Analytics and Video Analysis Methods», », Ural Federal University named after the first President of Russia B.N. Yeltsin

Abstract

The study is devoted to the joint use of the Solara Python framework and the Apache Superset BI platform for analyzing industrial systems data (SCADA, MES, ERP, IoT). Solara is used to create interactive mnemonic diagrams, and Superset is used for analytical dashboards. The combination of these tools increases the flexibility of data processing and the efficiency of visualization, which contributes to operational transparency and optimization of solutions in the context of Industry 4.0.

KEYWORDS: Solara, analytics, manufacturing processes, Apache Superset, data visualization.

Введение

В условиях технологического прогресса и глобальной конкуренции эффективная обработка и анализ данных становятся ключевыми факторами успеха производственных компаний. Современные предприятия генерируют огромные объемы разнородных данных (телеметрия, ERP, MES), требующие оперативной обработки и интерпретации для принятия управленческих решений.

Интеграция инструментов бизнес-аналитики (BI) позволяет выявлять тенденции, оптимизировать процессы и снижать затраты. Особую актуальность приобретают low-code и Python-ориентированные решения, такие как библиотека Solara, которая упрощает разработку интерактивных дашбордов без глубокого знания JavaScript. В сочетании с Apache Superset (платформой для визуализации данных на основе SQL) Solara позволяет создать эффективное решение для мониторинга производства, повышая прозрачность процессов и прогнозируя сбои.

Цель статьи — исследование возможностей интеграции Solara и Apache Superset для использования в производственной среде, включая применение методов оптимизации и визуализации данных с целью повышения операционной эффективности.

В условиях цифровой трансформации промышленных предприятий ключевую роль в повышении эффективности играют потоковая обработка данных, цифровые двойники и непрерывный интеллект (CI), где CI обеспечивает всесторонний анализ и быструю реакцию на изменения. Современные специализированные платформы, такие как DATA-TRACK и EXPERT BASE (разработчик - компания "DATA-ЦЕНТР Автоматика", в сочетании с инструментами визуализации Solara и Apache Superset, предоставляют комплексные решения для аналитики производственных процессов [1]. Аналитика позволяет отслеживать показатели в реальном времени, прогнозировать события, выявлять узкие места и принимать обоснованные решения для оптимизации управления, сокращения времени простоя и затрат, что обеспечивает повышение производительности и конкурентоспособности компаний. В зависимости от целей и глубины анализа, выделяют четыре типа аналитики:

- Описательная аналитика предоставляет сводные данные по ключевым метрикам, таким как объём производства, время простоя, уровень брака;
- Диагностическая аналитика, с помощью корреляционного анализа и визуализации процессов устанавливает взаимосвязи между отклонениями и параметрами оборудования, раскрывая причины выявленных проблем;
- Прогностическая аналитика использует методы машинного обучения и применяется для предсказания отказов оборудования, колебаний спроса, потребностей в сырье;
- Прескриптивная аналитика формирует конкретные рекомендации по оптимизации процессов, использует симуляции и модели принятия решений.

Источниками данных в производственной среде выступают различные цифровые устройства и системы, представленные в таблице 1.

Таблица 1. Ключевые источники данных в производственной среде

Источник данных	Назначение
SCADA-системы	Контроль и сбор данных в реальном времени с производственного оборудования.
MES (Manufacturing Execution System)	Оперативное управление производственными операциями.
ERP (Enterprise Resource Planning)	Интеграция бизнес-процессов, управление ресурсами и финансами.
IoT-устройства	Датчики и сенсоры, передающие данные об условиях работы оборудования и персонала, а также параметрах сырья.

Аналитика производственных процессов представляет собой комплекс методов и технологий, направленных на анализ данных с целью повышения эффективности и устойчивости производственных операций. Процесс аналитики можно разделить на несколько ключевых этапов:

1. Систематический сбор данных из различных источников, таких как сенсоры, MES, ERP-системы и логистические платформы;
2. Удаление дубликатов, обработка пропущенных значений и нормализация данных для обеспечения точности анализа;
3. Применение статистических методов, машинного обучения и моделирования для выявления закономерностей и прогнозирования будущих событий;
4. Представление результатов в наглядной форме для лучшего понимания и интерпретации данных;
5. Разработка стратегий на основе аналитических инсайтов для повышения эффективности производственных процессов.

Цифровизация производства делает аналитику ключевым инструментом роста. По данным McKinney [2], внедрение цифровых решений повышает производительность на 20–30%. Анализ данных (ОЕЕ, датчики, ERP) через BI-системы сокращает простои, снижает потери и улучшает контроль качества, повышая конкурентоспособность предприятий.

Материалы и методы

Данное исследование посвящено применению фреймворка Solara для разработки интерактивных веб-приложений и дашбордов. Solara представляет собой инструмент для создания интерфейсов исключительно средствами языка программирования Python, минуя

необходимость написания клиентского кода на JavaScript. Это существенно снижает порог входа для специалистов в области анализа данных и инженеров данных, предоставляя удобную среду для быстрой разработки прототипов и приложений, ориентированных на взаимодействие с большими объемами данных. Основные компоненты Solara представлены в таблице 2. Фреймворк Solara характеризуется рядом уникальных особенностей, позволяющих эффективно решать широкий спектр задач, связанных с обработкой и визуализацией данных [3]:

- Позволяет создавать динамические графики, таблицы, формы ввода и фильтры для взаимодействия с данными в реальном времени;
- Интегрируется с Pandas и NumPy, что упрощает обработку и визуализацию больших объемов данных;
- Использует декларативный подход, позволяя разработчикам сосредоточиться на логике приложения;
- Поддержка Matplotlib и Plotly позволяет создавать разнообразные настраиваемые графики и диаграммы;
- Использует реактивную архитектуру для автоматического обновления интерфейса при изменении данных.

Таблица 2. Основные компоненты Solara [3]

Компонент	Описание
<code>solara.component</code>	Декоратор для создания компонентов
<code>solara.run</code>	Основной механизм для запуска приложения
<code>solara.data</code>	Инструменты для загрузки, обработки и работы с данными
<code>reactive()</code>	Объект, отслеживающий изменения состояния
<code>use_state()</code>	Сохранение состояния между обновлениями интерфейса
Widgets	Разнообразные элементы управления (кнопки, слайдеры, таблицы и фильтры)
Plotly и Matplotlib	Инструменты для визуализации данных: гистограмм и тепловых карт.

Исследуемый случай касается реализации интерактивной визуализации перемещения изделий («заготовок») вдоль технологических потоков («нитей»). Для решения поставленной задачи применяется следующий алгоритм действий:

1. Получение данных (рисунок 1). Через REST API метода `get_current_state`

формируется запрос на получение текущих координат всех заготовок и характеристик нитей (длина, статус). Данные поступают в структуру данных, представляющую объекты «нити» (threads) и «заготовки» (ingots), содержащие подробную информацию о позиции каждого изделия относительно определенной нити.

```
def get_current_state():
    response = requests.get(STATE_API_URL, params=API_PARAMS)
    data = response.json()
    ...
```

Рис. 1. Получение данных

2. Обработка состояний (рисунок 2). Класс IngotTracker служит центральным механизмом хранения и обновления актуальной информации о положении заготовок. Он реализует методы добавления и удаления позиций заготовок, поддерживает синхронизацию состояния между различными элементами интерфейса.

```
class IngotTracker:
    def update_state(self, state_data):
        # Обновление списка заготовок
        for ingot_id, ingot_data in current_ingots.items():
            self.ingots[ingot_id] = ingot_data
        # Удаление отсутствующих заготовок
        removed_ids = set(self.ingots.keys()) - set(current_ingots.keys())
        for ingot_id in removed_ids:
            del self.ingots[ingot_id]
```

Рис. 2. Управление состоянием

3. Отображение результатов (рисунок 3). Интерфейс построен на основе компонента MnemoScheme, разработанном в среде Solara. Панель управления предоставляет пользователю доступ к инструментам автообновления, настройки частоты обновления данных и кнопки для ручного обновления страницы. Осевые линии нитей размечены в процентах (от 0% до 100%), отображающие положение заготовок относительно заданных значений длины нити.

```
@solara.component
def MnemoScheme(tracker: IngotTracker):
    auto_update = solara.use_reactive(True) # Флаг автообновления
    update_interval = solara.use_reactive(5) # Интервал (сек)
    last_update_time = solara.use_reactive(None)
```

Рис. 3. Визуализация

4. Формирование визуальной модели (рисунок 4). Процесс рендеринга основывается на создании контейнеров для каждой нити посредством функции `render_thread`. Каждая заготовка представляется прямоугольником, положение которого рассчитывается в зависимости от относительных координат её начала и конца (`percentHead`, `percentTail`).

```
def render_thread(thread_id: int, thread_info: Dict, tracker: IngotTracker):
    # Отрисовка оси нити
    solara.HTML("div", style={
        "position": "absolute",
        "left": "0", "right": "0", "height": "2px",
        "background-color": "#666", "top": "50%"
    })
    # Отрисовка заготовки
    solara.HTML("div", style={
        "left": f"{x_tail}%", "width": f"{width_percent}%",
        "background-color": "#4CAF50", "height": "40px"
    }, unsafe_innerHTML=f"{ingot['id']}")
```

Рис. 4. Логика отрисовки

5. Настройка производительности (рисунок 5). Чтобы обеспечить плавную анимацию и точность представления данных, интерфейс позволяет изменять частоту обновления данных вручную или автоматически с возможностью установки фиксированного временного интервала (от 1 до 30 секунд).

```
with solara.Card("Управление"):
    solara.Checkbox(label="Автообновление", value=auto_update)
    solara.SliderInt("Интервал обновления (сек)", value=update_interval, min=1, max=30)
    solara.Button("Обновить вручную", on_click=update_data)
```

Рис. 5. Интерфейс управления

Результаты и обсуждение

Исследование показало эффективность использования двух основных подходов к созданию интерактивных веб-приложений и визуализации данных — библиотеки Solara и платформы Apache Superset, каждый из которых обладает уникальными характеристиками и возможностями.

Одним из примеров успешного применения Solara является реализация интерактивной мнемосхемы производственного процесса, где каждая заготовка представлена зеленым прямоугольником (рисунок 6), перемещающимся вдоль технологического потока (нити). Такой подход обеспечивает операторов удобными средствами мониторинга текущего положения изделий, позволяя оперативно реагировать на возможные отклонения или сбои.

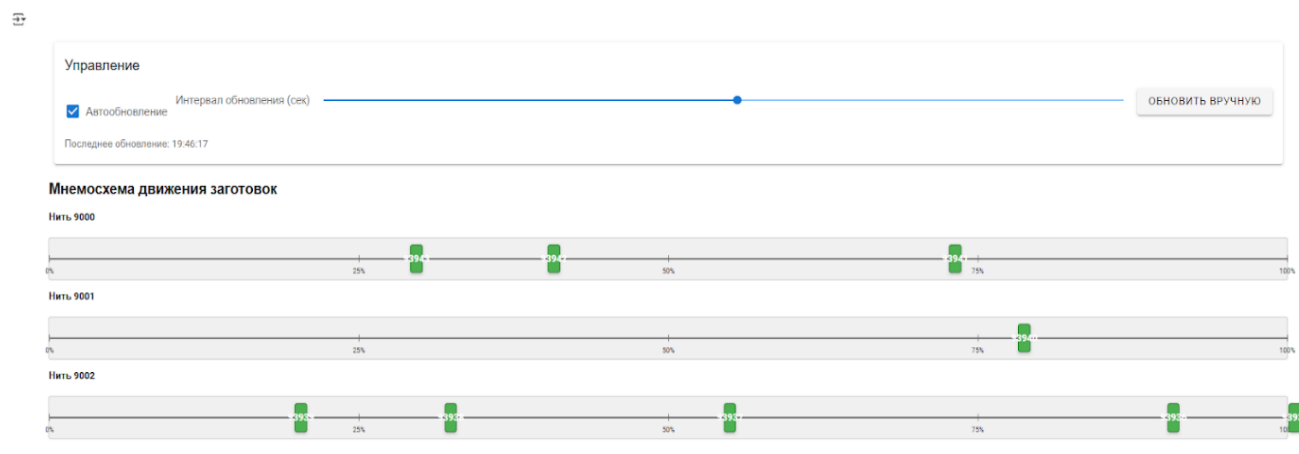


Рис. 6. Мнемосхема движения заготовок

Таким образом, Solara идеально подходит как для быстрого прототипирования, так и для создания полноценных аналитических систем. Это решение может быть использовано для разработки дашбордов, визуализации производственных данных, мониторинга показателей бизнеса и в других областях, где необходимы интерактивные веб-приложения для работы с данными [4].

Apache Superset — это мощная open-source платформа для визуализации данных и бизнес-аналитики. Она позволяет создавать интерактивные дашборды и графики с минимальным объемом кодирования. Apache Superset поддерживает различные базы данных, предоставляет инструменты для SQL-запросов и анализа данных, а также используется в инфраструктуре ORT/CI в качестве визуального уровня. Необходимо отметить, что метод объектных отношений (ORT – object relations technique) используется для создания единой модели данных, что позволяет строить цифровые двойники материальных потоков. Выделяют следующие ключевые характеристики и возможности Apache Superset [5]:

1. Поддержка PostgreSQL, MySQL, SQLite, Redshift, ClickHouse, Snowflake и других СУБД позволяет работать с большими объемами данных из разных источников;
2. Графики, карты, таблицы и диаграммы (круговые, линейные, столбчатые, географические карты) помогают представлять данные в различных форматах;
3. Пользователи могут создавать дашборды с фильтрами и настраиваемыми параметрами отображения, что обеспечивает интерактивный анализ данных в реальном времени.

Основные компоненты Apache Superset представлены в таблице 3 [5].

Таблица 3. Основные компоненты Apache Superset

Компонент	Описание
SQL Lab	Интерфейс для написания и отладки SQL-запросов для извлечения

Компонент	Описание
	данных.
Charts	Инструменты для создания графиков и различных типов визуализаций.
Dashboards	Средства для создания дашбордов для комбинации различных визуализаций и метрик.
Filters	Настройка фильтров для визуализаций для быстрого изменения параметров данных.
User Access	Управление правами доступа для пользователей, интеграция с LDAP и SSO для организации безопасного доступа.

Рассмотрим пример создания дашборда с использованием Apache Superset, встроенного в платформу Expert Base, который выводит партию заготовок и график температуры заготовок в данной партии.

Рекомендуемый порядок действий приведен ниже.

Для создания дашборда убедитесь, что данные загружены в хранилище Expert Base. Проверьте наличие таблиц в разделе «Данные» → «Хранилище». Если данных нет, импортируйте их или дождитесь синхронизации. Затем перейдите в «Запросы» → «+ Запрос», выберите хранилище Expert Base и тип запроса «Объекты». Настройте временной интервал, основной объект (например, «Заготовки») и параметры (температура, ID, время измерения). Добавьте связанный родительский объект (например, «Партии») и его атрибуты (номер партии, дата производства). Примените фильтры при необходимости и сохраните запрос под уникальным именем.

После сохранения запроса перейдите в «Визуализация» → «Графики» → «+ График», выберите запрос как датасет и тип визуализации (например, Line Chart для температуры). Синхронизируйте столбцы через меню графика: «Редактировать датасет» → «СТОЛБЦЫ» → «Синхронизировать столбцы из источника», сохраните изменения. Настройте оси: метрика (Y) — температура, группировка (X) — ID заготовки или время. В «Кастомизации» измените цвета и подписи при необходимости. Сохраните график с понятным названием для добавления на дашборд.

Для создания дашборда перейдите в «Визуализация» → «Дашборды» → «+ Дашборд», назовите его и добавьте графики перетаскиванием из правой панели. Настройте размер и расположение элементов. Добавьте общие фильтры через значок воронки, выбрав параметр (например, «Номер партии»). Сохраните дашборд после завершения настройки.

Готовый дашборд (рисунок 7) появится в списке «Дашборды», где его можно открыть, отредактировать или скопировать URL для общего доступа. Для доступа без входа используйте «Публичный доступ». Вкладка «Оформление» позволяет добавить текстовые блоки, разделители или изображения. Дашборд обновляется автоматически при изменении данных, а фильтры позволяют анализировать информацию динамически.

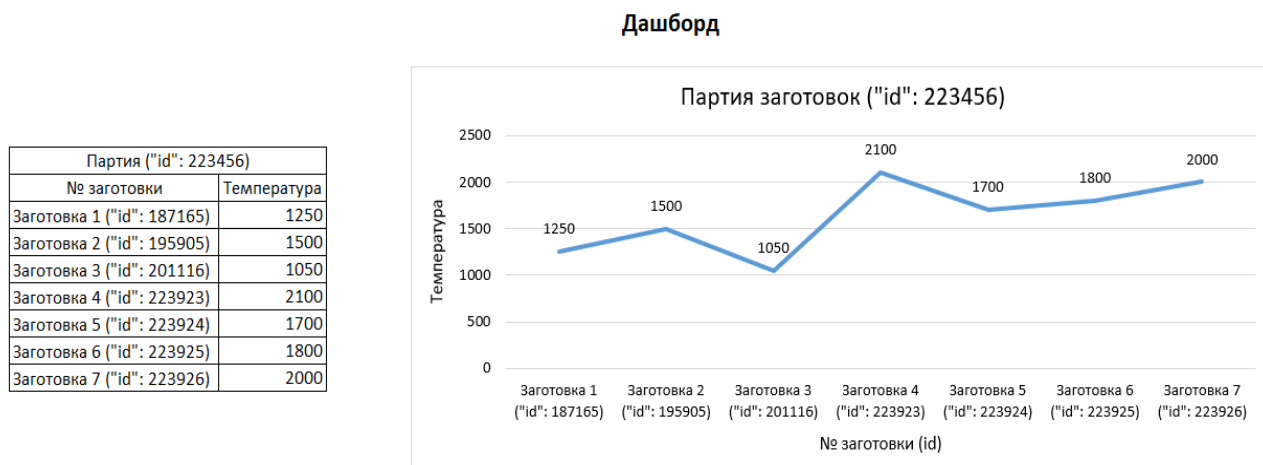


Рис. 7. Данные и график температуры заготовок в партии

Таким образом, Apache Superset представляет собой мощную и гибкую платформу для анализа данных и создания дашбордов с открытым исходным кодом. Она идеально подходит для организаций, которым необходимы высококачественные визуализации и интерактивные отчеты с возможностью работы с большими данными. Легкость в использовании, поддержка множества СУБД и расширенная кастомизация делают её отличным инструментом для бизнес-анализа, повышения прозрачности процессов и принятия обоснованных решений.

Интеграция Solara и Apache Superset объединяет возможности Python-аналитики и BI-платформы, упрощая создание дашбордов и анализ производственных данных [6]. Рассмотрим ключевые этапы интеграции:

1. Solara собирает данные из различных источников (CSV, базы данных и др.) и интегрирует их с инструментами Python;
2. Solara очищает и преобразует данные с использованием Python-кода, обеспечивая гибкость настройки;
3. Подготовленные данные передаются в Apache Superset через SQLAlchemy или прямое подключение к базе данных;
4. В Apache Superset создаются интерактивные графики и дашборды для наглядного представления результатов анализа, включая различные типы диаграмм и карт.

Интеграция Solara и Apache Superset упрощает анализ данных, позволяя аналитикам и

разработчикам сосредоточиться на выявлении трендов и принятии решений. Apache Superset предоставляет визуализацию данных через дашборды и графики, а Solara обеспечивает гибкость интерфейсов на Python. Разработчики могут обрабатывать данные в Solara, а аналитики визуализировать результаты в Apache Superset, что значительно ускоряет анализ данных.

Кроме того, представим сравнение Solara и Apache Superset в таблице 4.

Таблица 4. Сравнение Solara и Apache Superset

Характеристика	Solara	Apache Superset
Язык разработки	Python	Python + SQL
Уровень кастомизации	Высокий (через Python-код)	Средний (через UI и SQL)
Поддержка big data	Средняя (зависит от бэкенда)	Высокая (поддержка кластеров)
Простота освоения	Прост для Python-разработчиков	Прост для SQL-аналитиков
Интерактивность	Очень высокая	Средняя (ограничена компонентами)
Установка и настройка	Минимальная	Более сложная

Заключение

Современная цифровая экономика требует эффективных методов обработки и визуализации данных для поддержки оперативного принятия решений. Исследование продемонстрировало важность объединения технологий, представленных библиотекой Solara и платформой Apache Superset, которые предлагают уникальные подходы к решению задач анализа и визуализации больших объёмов данных. Библиотека Solara зарекомендовала себя как удобное средство для быстрого прототипирования и разработки интерактивных веб-приложений, сочетая простоту Python-разработки с высокой степенью интерактивности. Её главная ценность заключается в снижении порога входа для аналитиков и инженеров данных, что способствует ускоренному внедрению инновационных решений. Платформа Apache Superset предлагает широкие возможности для визуализации данных и построения интерактивных дашбордов, предоставляя аналитикам удобный интерфейс для комплексного анализа и мониторинга ключевых показателей эффективности предприятия. Благодаря поддержке многочисленных баз данных и широких функциональных возможностей, эта

платформа становится незаменимым инструментом для компаний любого масштаба. Однако наибольший потенциал открывается при совместной работе обоих инструментов. Объединение Solara и Apache Superset позволяет создать целостную систему, объединяющую возможности Python-разработки и высокоуровневой визуализации. Такое сочетание оптимизирует процессы сбора, очистки и анализа данных, сокращает временные затраты на разработку и внедрение решений, улучшает контроль над производством и повышает общую конкурентоспособность предприятия. Эффективная интеграция этих технологий создаёт условия для улучшения операционной эффективности, снижения издержек и увеличения прибыли. Предприятия, активно внедряющие подобные технологии, получают конкурентные преимущества в современных условиях цифровой трансформации и глобализации экономики.

Литература

1. Платформа Expert Base. — Текст: электронный // Официальный сайт платформы Expert Base: [сайт]. — 2025. — URL: <https://educational-center.dc.ru> (дата обращения: 02.03.2025).
2. McKinney, W. (2022). Python for Data Analysis. O'Reilly Media.
3. Документация Solara. — Текст: электронный // Официальный репозиторий Solara: [сайт]. — 2025. — URL: <https://github.com/widgetti/solara> (дата обращения: 21.03.2025).
4. Solara Docs. Solara: [Electronic resource]. — 2025. — URL: <https://solara.dev> (accessed: 21.03.2025).
5. Документация Apache Superset. — Текст: электронный // Официальная документация Apache Superset: [сайт]. — 2025. — URL: <https://superset.apache.org/> (дата обращения: 28.04.2025).
6. Kalil, R. (2022). Building Dashboards with Superset and Python. O'Reilly Media.