

УДК 004

**АВТОМАТИЗАЦИЯ УВЯЗКИ ГЕОЛОГИЧЕСКИХ СКВАЖИН:
ЭВРИСТИЧЕСКАЯ МОДЕЛЬ СОПОСТАВЛЕНИЯ ИНТЕРВАЛОВ С
МЕХАНИЗМОМ НАКОПЛЕНИЯ ОБУЧАЮЩИХ ДАННЫХ**

Золотарёв Н.С., студент гр. ИИМ-241

Научный руководитель: Протоdjяконов А.В, к.н., доцент
Кузбасский государственный технический университет
имени Т.Ф. Горбачева, г. Кемерово

В статье описывается система автоматизации процесса увязки данных кернового анализа и радиоактивного каротажа угольных скважин. Предложен эвристический алгоритм сопоставления интервалов на основе метрики перекрытия глубин, реализованный в виде WPF - приложения с REST API.

Система обеспечивает интерактивное редактирование результатов, визуализацию геологического разреза и накопление подтверждённых экспертом примеров для последующего обучения полноценной ML-модели

Задача заключается в сопоставлении угольных интервалов из каротажа с соответствующими интервалами керна для получения уточнённых глубин пласта и показателей качества угля.

Традиционно эта работа выполняется вручную: инженер сравнивает таблицы из двух Excel-файлов, визуально находит совпадающие интервалы и переносит значения зольности. Процесс трудоёмок, занимает несколько часов на скважину и подвержен субъективным расхождениям между специалистами.

Цель разработки — автоматизировать первичное сопоставление, предоставить инструменты верификации и накапливать подтверждённые экспертом результаты для последующего обучения более точной модели.

Источники данных и их структура

Система работает с двумя входными файлами формата кернового анализа и радиоактивного каротажа.

Файл кернового анализа содержит интервалы отбора образцов:

Таблица 1. Пример данных кернового анализа

Скважина	Глубина от (до увязки)	Глубина до (до увязки)	Зольность
1034з	3.50	4.00	47.9
1034з	4.00	4.50	26.7

Файл радиоактивного каротажа содержит непрерывную литологическую колонку:

Таблица 2. Пример данных радиоактивного каротажа

Номер скв.	Горная порода название	от	до
------------	---------------------------	----	----

1034з	Песчаник мелко-зернистый	0.80	1.85
1034з	Уголь	3.60	4.10

Ключевое различие источников: каротаж даёт более точные глубины пластов и пород, дополнительно может содержать некоторые химические показатели, керн также содержит зольность, но глубины могут незначительно расходиться из-за особенностей отбора образцов.

Архитектура системы.

Система реализована как приложение с разделением на клиентскую и серверную части.

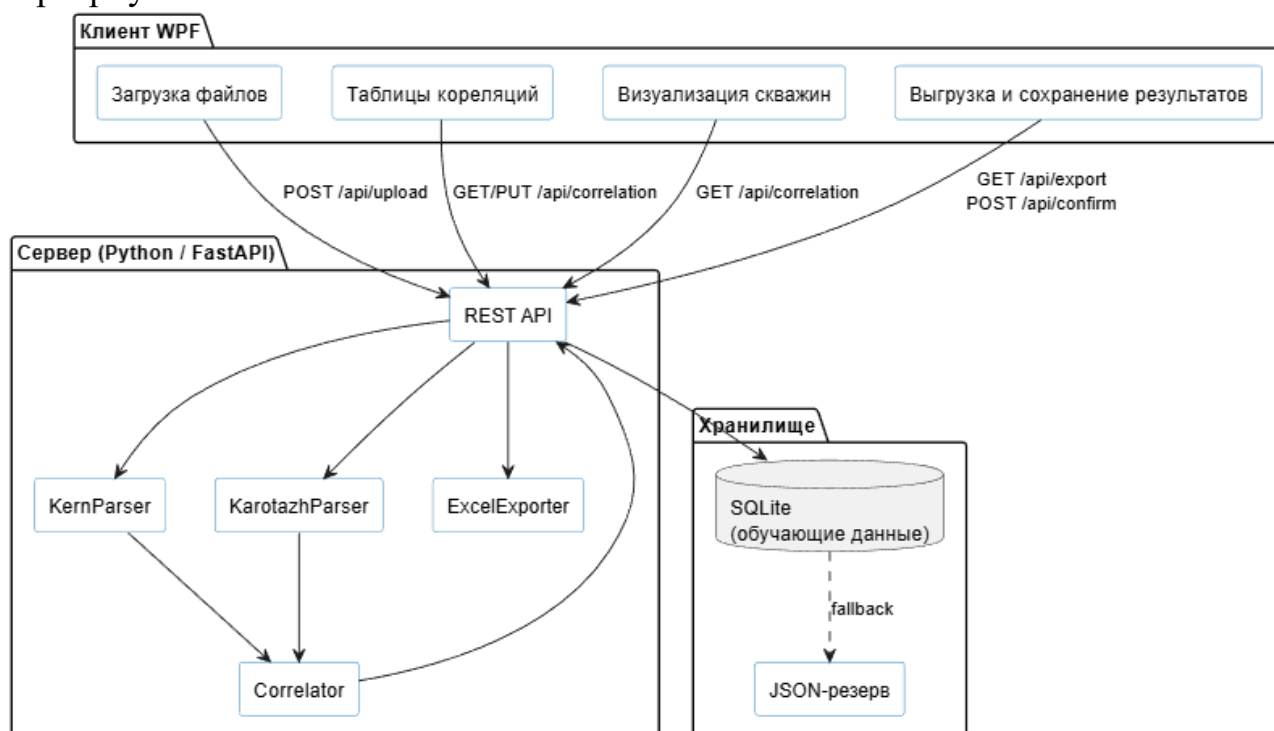


Рисунок 1. Архитектура системы

Серверная часть построена на фреймворке FastAPI (Python). Она отвечает за приём файлов, парсинг, запуск алгоритма увязки, экспорт результатов и хранение обучающих примеров. Сессионное состояние хранится в памяти процесса — это упрощает развёртывание и достаточно для однопользовательского режима работы.

Клиентская часть реализована на WPF. Она обеспечивает загрузку файлов через “drag-and-drop”, отображение и редактирование таблицы результатов, SVG-визуализацию геологического разреза и экспорт.

Хранилище обучающих данных использует SQLite для персистентного хранения подтверждённых результатов. При сбое базы данных система автоматически переключается на резервное сохранение в JSON-файл.

Алгоритм увязки включает в себя:

- Идентификация угольных интервалов

Первый шаг — выделение угольных пластов из каротажной колонки. Критерий прост: нормализованное (приведённое к нижнему регистру) название породы должно содержать подстроку «уголь». Это покрывает все встречающиеся на практике варианты: «Уголь», «УГОЛЬ», «уголь газовый», «уголь коксующийся».

- Метрика перекрытия интервалов

Для каждого угольного интервала каротажа и каждого интервала керна вычисляется длина перекрытия и нормированная уверенность сопоставления.

Выбирается интервал керна с максимальным значением overlap. Если перекрытие равно нулю, но расстояние между ближайшими границами интервалов не превышает допуска, $\delta = 0.3$ м, результат помечается флагом “needs_review = true” — это сигнал инженеру о необходимости ручной проверки.

Каждая скважина обрабатывается независимо — функция “correlate()” принимает данные ровно одной скважины и не имеет доступа к данным других. Это гарантирует изоляцию результатов и упрощает параллельную обработку.

- Предобработка данных

Перед запуском алгоритма данные проходят нормализацию в двух парсерах.

Парсеры решают несколько практических проблем реальных файлов:

- Объединённые ячейки заголовков. В реальном файле кернового анализа заголовок «Глубина (до увязки)» является объединённой ячейкой, охватывающей два столбца (от и до). Pandas читает такую структуру как одну именованную колонку и одну безымянную (“Unnamed: N”). Парсер обнаруживает этот паттерн и переименовывает колонки в явные имена.

- Десятичный разделитель. Числа могут быть записаны как с точкой (“3.14”), так и с запятой (“3,14”). Нормализация выполняется заменой запятой на точку перед преобразованием в float.

- Пустые строки. Строки с отсутствующими значениями глубин пропускаются с записью предупреждения в лог.

- Лишние колонки. Файлы могут содержать дополнительные колонки (комментарии, служебные поля). Парсер явно выбирает только обязательные колонки, игнорируя остальные.

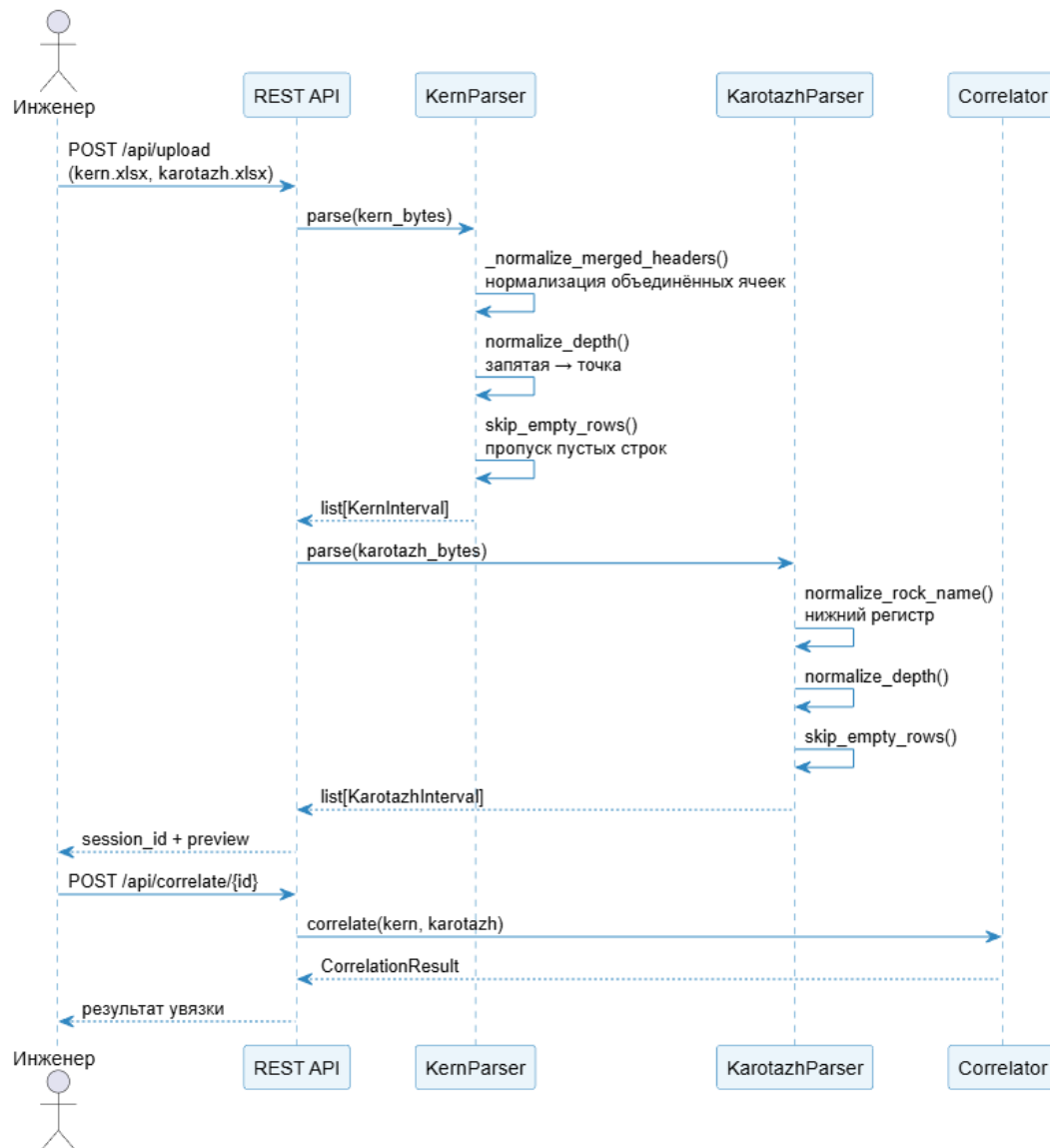


Рисунок 2. Диаграмма последовательности работы парсеров

Путь к машинному обучению и построению полноценной ML-системы.

Представленная реализация — эвристическая. Это осознанное архитектурное решение: на начальном этапе недостаточно размеченных данных для обучения статистической модели, а эвристика уже даёт приемлемое качество и позволяет накапливать обучающие примеры.



Рисунок 3. Путь построения ML-модели

Механизм накопления. Когда инженер подтверждает результат увязки, система сохраняет тройку - данные керна, данные каротажа, подтверждённый результат в SQLite-хранилище с GUID и меткой времени. При достижении 50 подтверждённых примеров система уведомляет администратора о возможности дообучения.

Планируемая ML-модель. По мере накопления данных эвристику планируется заменить на Gradient Boosting (XGBoost или LightGBM). Признаки для модели: нормированное перекрытие интервалов, разница глубин, длина угольного пласта, тип ближайших пород. Альтернативный подход — рассмотрение последовательности интервалов как задачи выравнивания последовательностей (sequence alignment), аналогичной методам биоинформатики.

Интерфейс “Correlator” спроектирован так, что замена реализации эвристического подхода на полноценный модуль ML-модели не потребует изменений в остальных компонентах системы.

Формальные свойства корректности

Система разработана с применением методологии Property-Based Testing (PBT) - тестирования на основе свойств. Вместо проверки конкретных примеров формулируются универсальные утверждения, которые должны выполняться для любых допустимых входных данных. Для генерации случайных тестовых данных используется библиотека “Hypothesis”.

Ключевые свойства системы:

1. Принятие валидных файлов. Любой корректный .xlsx принимается без ошибки
2. Полнота парсинга. Все обязательные поля присутствуют в результате
3. Нормализация глубин. Глубины представлены с точностью ≤ 3 знаков
4. Группировка по скважинам. N уникальных скважин равняется N групп
5. Трассируемость результата.
6. Изоляция скважин. Результат скважины A не зависит от данных скважины B
7. Валидация глубин.
8. Путь экспорта. Экспорт - импорт возвращает исходные данные
9. Путь JSON. Сериализация - десериализация сохраняет эквивалентность

Свойство 6 (изоляция скважин) особенно важно с точки зрения корректности: оно гарантирует, что добавление новых данных не изменит уже вычисленные результаты. Свойство 5 (трассируемость) формализует основной инвариант алгоритма — происхождение каждого поля результата.

Разработанная система решает практическую задачу автоматизации рутинного геологического процесса. Эвристический алгоритм на основе метрики перекрытия интервалов обеспечивает детерминированный и интерпретируемый результат, который инженер может проверить и скорректировать. Механизм накопления подтверждённых примеров создаёт фундамент для перехода к статистической модели по мере роста базы данных.

Применение методологии “Property - Based Testing” позволило формализовать ключевые инварианты системы и верифицировать их на случайных входных данных, что повышает уверенность в корректности реализации за пределами тестовых примеров.

Дальнейшее развитие системы предполагает: замену эвристики на Gradient Boosting после накопления ≥ 50 обучающих примеров; расширение поддерживаемых форматов входных файлов; добавление многопользовательского режима с разграничением доступа.

Список литературы:

1. Большакова, Е. И. Автоматическая обработка текстов на естественном языке и анализ данных / Е. И. Большакова, К. В. Воронцов, Н. Э. Ефремова [и др.]. — Москва : Изд. дом Высшей школы экономики, 2017. — 269 с.
2. Flask. Web Development, One Drop at a Time : документация API [Электронный ресурс]. — URL: <https://flask.palletsprojects.com/en/latest/api>
3. Васильев, Ю. Обработка естественного языка. Python и spaCy на практике / Ю. Васильев. — Санкт-Петербург : Питер, 2021. — 256 с. : ил.
4. Примеры внедрения искусственного интеллекта в разных отраслях [Электронный ресурс] // КТ.Team. — 2025. — URL: <https://kt.team>
5. Внедрение AI в промышленное производство [Электронный ресурс] // SberPro. — 2025. — URL: <https://sber.pro>
6. Гулин, А. Б. Применение методов машинного обучения для интерпретации геофизических данных скважин / А. Б. Гулин. — 2023.