**МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РЕСПУБЛИКИ КАЗАХСТАН КОМИТЕТ НАУКИ**

**ИНСТИТУТ МЕХАНИКИ И МАШИНОВЕДЕНИЯ ИМЕНИ АКАДЕМИКА У.А. ДЖОЛДАСБЕКОВА**

**Мухамбетжанов Салтанбек**

# ИНТЕЛЛЕКТУАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ И ЦИФРОВЫЕ ДВОЙНИКИ ДЛЯ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ХАРАКТЕРИСТИК НЕФТЯНЫХ МЕСТОРОЖДЕНИЙ

**Монография**

**Алматы**
**2024**

**УДК**
**ББК**

***Рекомендовано к изданию Ученым советом РГП на ПХВ «Института механики и машиноведения им. академика У.А.Джолдасбекова» КН МНВО РК (Протокол №6 от 06 сентября 2024)***

**Рецензенты:**
**Байшемиров Ж.Д -** PhD кафедры «Математика и математическое моделирование, Профессор Казахского национального педагогического университета им. Абая
**Бурибаев Ж.А**. - и.о. доцента, PhD кафедры Компьютерных Наук, Казахский национальный университет имени Аль-Фараби

**Мухамбетжанов Салтанбек**

**Интеллектуальные вычислительные модели и цифровые двойники Для анализа и прогнозирования характеристик нефтяных месторождений.** Монография. – Алматы: Дарын баспасы, 2024. – 173 бет.

**ISBN**

В монографии представлено обширное исследование, посвященное применению математических моделей и методов машинного обучения в нефтяной отрасли. Основное внимание уделено четырем ключевым направлениям: разработке и совершенствованию математических моделей, внедрению высокопроизводительных вычислений (HPC), применению интеллектуальных моделей и созданию цифровых двойников. Исследование охватывает такие модели, как Бакли-Леверетта и Black Oil, а также композиционные и стохастические модели, что позволяет более точно описать динамику фильтрационных процессов в пористых средах. Особое внимание уделяется применению машинного обучения для оптимизации процессов добычи нефти и газа, включая использование деревьев решений, случайных лесов и градиентного бустинга.

Подробно рассматриваются аспекты использования HPC для решения вычислительных задач, что значительно повышает скорость расчетов и масштабируемость моделей. Важной частью работы является создание цифрового двойника, интегрирующего данные из различных источников, что открывает новые возможности для мониторинга и управления системами в реальном времени.

Монография представляет собой значимый вклад в нефтегазовую инженерию, предлагая практические решения для оптимизации процессов и прогнозирования характеристик месторождений. Работа интересна как для научного сообщества, так и для практиков, стремящихся к внедрению современных технологий и методов в свою деятельность. Таким образом, данная монография является значимым вкладом в область нефтегазовой инженерии и предоставляет актуальные и практические решения для анализа и оптимизации процессов в нефтяных месторождениях.

**УДК**
**ББК**

ISBN 978-601-7849-12-2

© Мухамбетжанов С, 2024
© Дарын баспасы, 2024

## СОДЕРЖАНИЕ

Введение ..... 5
1.Разработка математических моделей для решения задачи неравновесной фильтрации и стохастического моделирование. ..... 10
1.1. Математическая модель Бакли-Леверетта. ..... 11
1.2. Модель Black Oil. ..... 14
1.3.Разработка математических моделей для решения задачи неравновесной фильтрации (композиционная модель). ..... 21
1.4. Стохастическая модель Баклея-Леверетта. ..... 24
2. Разработка интеллектуальных вычислительных моделей для решения задачи неравновесной фильтрации и стохастического моделирования. ..... 28
2.1. Разработка интеллектуальных вычислительных моделей для решения задачи неравновесной фильтрации (композиционная модель). ..... 28
2.2. Разработка интеллектуальных вычислительных моделей для модели Black Oil. ..... 49
2.3. Использование методов машинного обучения для стохастического моделирование. ..... 86
3. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи неравновесной фильтрации и стохастичкеского моделирование. ..... 99
3.1. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи неравновесной фильтрации (композиционная модель). ..... 100
3.2. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи моделирования трехфазного течения Black Oil. ..... 109

3.3. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи стохастичкеского моделирования..... 121
4. Разработка виртуального прототипа нефтяной скважины. ..... 130
4.1. Проектирование архитектуры виртуального прототипа нефтяной скважины. ..... 130
4.2. Разработка виртуального прототипа нефтяной скважины. 134
Заключение ..... 161
Список использованных литератур..... 168

## Введение

В данной монографии рассматриваются современные методы и подходы к использованию интеллектуальных вычислительных моделей и технологий цифровых двойников для анализа и прогнозирования характеристик нефтяных месторождений. Значительный прогресс в области машинного обучения, высокопроизводительных вычислений и искусственного интеллекта открывает новые возможности для решения сложных задач, связанных с гидродинамическим моделированием и управлением добычей нефти. Комплексное применение этих технологий позволяет более точно прогнозировать поведение пластов, учитывать стохастические процессы и неопределенности, а также оптимизировать эксплуатационные процессы.

Разработка математических моделей для решения задачи неравновесной фильтрации и стохастического моделирования для гидродинамического моделирования процессов в нефтяных пластах является важным процессом для повышения эффективности добычи нефти и газа. В ходе работы была изучена задача неравновесной фильтрации (композиционная модель), что является важной темой для стран и компаний, занимающихся добычей нефти. Для достижения более высокой точности в моделировании сложных гидродинамических явлений, происходящих в нефтяных пластах, необходимо учитывать компонентный состав фаз [1, 2]. Моделирование таких процессов осложняется необходимостью учитывать конвекцию, диффузию, фазовые переходы и химические реакции [3, 4, 5]. Для решения задачи неравновесной фильтрации использовались различные модели. В частности, была интегрирована модель Бакли-Леверетта, известная своим применением для описания двухфазного течения в пористой среде. Модель Бакли-Леверетта способствует

учету процессов вытеснения нефти водой или полимерами, что является важным для оптимизации полимерного заводнения [6, 7]. Также была применена модель Black oil, учитывающая фазовые переходы между нефтью, газом и водой в рамках трехфазного течения [8, 9]. Модель предоставляет более сложное и точное описание процессов в нефтяных пластах, что расширяет возможности прогнозирования гидродинамических явлений при изменении условий в пласте. Стохастическая модель позволяет учитывать неопределенности исходных данных, таких как пористость, проницаемость и вязкость фаз, что делает моделирование более адаптированным к реальным условиям [10, 11]. Данный подход улучшает точность прогнозов и снижает риски, связанные с эксплуатацией нефтяных пластов, что имеет критическое значение для управления процессами добычи нефти [12, 13]. Создание интеллектуальных вычислительных моделей, предназначенных для решения задач неравновесной фильтрации и стохастического моделирования, является важным шагом в развитии методов прогнозирования и оптимизации в сложных системах. Исследование сосредоточено на внедрении методов машинного обучения (ML) для решения проблемы неравновесной фильтрации в нефтяной отрасли. В нефтяной отрасли машинное обучение применяется для точной оценки запасов нефти (OOIP) с помощью искусственных нейронных сетей (ИНС) [14]. Методы ML оптимизируют добычу, прогнозируя размещение скважин и механические свойства горных пород [15,16]. Они также помогают в закачке полимера и подземном хранении газа, минимизируя CO2 [17,18]. Суррогатные модели, использующие XGBoost и MLP, оценивают чистую приведенную стоимость (NPV) резервуаров, способствуя принятию решений [19]. Интеграция ML с численными методами повышает

точность моделирования и сокращает время вычислений [20,21]. Наконец, модели глубокого обучения с нейронными операторами ускоряют численное моделирование, обеспечивая более точные прогнозы для управления резервуарами [22]. Для этой цели был использован синтетический набор данных, полученный с помощью математических моделей, таких как модель Бакли-Леверетта, модель Black oil и стохастическая модель. На основе обработанных данных были использованы различные алгоритмы машинного обучения, включая деревья решений (DT), случайные леса (RF), градиентный бустинг (GB), экстра-градиентный бустинг (ExGB), метод опорных векторов (SVM), стекинг-регрессию (Stacking Regression)[23], нейронные сети (NN), LSTM-сети и глубокие нейронные сети (DNN)[24]. После методов машинного обучения рассматривается распараллеливание алгоритмов машинного обучения как важная задача для обработки больших объемов данных и повышения производительности в нефтяной отрасли. Для решения проблемы добычи нефти можно использовать подход параллельного распределенного обучения с применением различных математических моделей и алгоритмов машинного обучения. Разработка параллельного алгоритма обучения регрессионных моделей является актуальной задачей. В статье [25] рассматривается развитие параллельных и распределенных алгоритмов машинного обучения для улучшения процессов машинного обучения на больших данных. В работе делается акцент на использовании технологий GPU и MapReduce, которые с 1995 года значительно ускорили машинное обучение. В статье [26] обсуждается распределенное машинное обучение, подчеркивая необходимость распределения задач между несколькими системами для управления большими объемами данных и улучшения параллелизма. В статье [27]

подчеркивается необходимость адаптации машинного обучения к возрастающим требованиям обработки больших данных через распределенный параллелизм, с учетом новых алгоритмов и систем, специально разработанных для этой цели. Данные исследования демонстрируют значимость использования параллельных и распределённых методов в машинном обучении, акцентируя внимание на необходимости приспособления к увеличению объема и сложности данных в разнообразных сферах, включая бизнес и научные разработки. Все три исследования подчеркивают важность инновационных технологических решений и создают основу для дальнейших разработок в этой быстро развивающейся области. Кроме того, реализация алгоритмов машинного обучения на графических процессорах (GPU) и использование различных подходов к параллелизации, как указано в работах [28-32], подтверждают актуальность и значимость данной темы. Применение HPC (высокопроизводительных вычислений) в этих контекстах позволяет значительно увеличить производительность и точность прогнозов в нефтяной отрасли, что, в свою очередь, способствует более эффективному принятию решений и оптимизации процессов добычи.

В заключительной части представлен проект и разработка испытательного стенда цифрового двойника для мониторинга и оптимизации нефтяных скважин. Технология цифрового двойника (DT) в нефтегазовой отрасли в основном используется для мониторинга активов, обслуживания и планирования проектов, включая такие технологии, как 3D/4D моделирование, VR, AR, IoT и машинное обучение, а также вспомогательные технологии, такие как рентгеновское излучение и LiDAR [33,34]. В работе [35] авторы предложили структуру DT для резервуара с использованием схем Навье-Стокса и LBM, в

то время как в работе [36] авторы разработали DT для прогнозирования добычи нефти и газа путем интеграции ИИ и слияния данных в реальном времени без обращения к пространственной науке. Авторы в работах [37,38] предложили модели DT для мониторинга рисков и управления активами, но они не были проверены для полевого инжиниринга. Для решения проблем мониторинга и оптимизации производительности нефтяных скважин была инициирована разработка испытательного стенда цифрового двойника. Цифровой двойник может не только выявлять узкие места в добыче, но и прогнозировать потенциальные проблемы, что приводит к принятию более обоснованных решений на основе данных и повышению общей производительности скважины. Все исследования и разработки, описанные в монографии, имеют высокую актуальность и могут принести значительную пользу как научной, так и промышленной сфере.

## 1.Разработка математических моделей для решения задачи неравновесной фильтрации и стохастического моделирование.

Моделирование неравновесной фильтрации и стохастического моделирование является ключевым этапом в анализе множества инженерных и геолого-геофизических задач. В рамках данного исследования рассмотрены 4 основные модели: модель Бакли-Леверетта, Black oil, композиционная модель и стохастическая модель. Модель Бакли-Леверетта, предполагающая равновесие фаз в пласте, предоставляет аналитическое решение для динамики распределения давления и насыщенности. Модель Black oil использует упрощённые уравнения состояния для анализа поведения смеси нефти и газа, что особенно важно при учёте фазовых переходов. Композиционная модель представляет собой набор уравнений, описывающих влияние различных компонентов (нефть, газ, вода) и их взаимодействие с пластом во времени. Стохастическая модель Бакли-Леверетта расширяет классическую модель для описания двухфазного потока (например, нефть и вода) в пористой среде, вводя случайные переменные, чтобы учитывать неопределенность в физических параметрах системы, таких как относительная проницаемость. Это позволяет лучше моделировать реальное поведение потока, которое может быть подвержено флуктуациям из-за неоднородностей среды или других факторов. Каждая модель имеет свои специфические области применения и предлагает различные уровни сложности при решении задач неравновесной фильтрации и стохастического моделирование в разнообразных геологических условиях.

В данном разделе будет подробно рассмотрена каждая из математических моделей, используемых для описания динамики многокомпонентных систем.

### 1.1. Математическая модель Бакли-Леверетта.

В потоке пористой среды в резервуаре действуют различные масштабы, каждый из которых подчеркивает различные физические процессы. На микроскопическом уровне преобладают капиллярные силы. В более крупных масштабах преобладают вязкие и гравитационные силы. Хотя некоторые неопределенности можно смягчить путем обширного отбора проб и включения в модель разнообразных геологических данных, общая структура двухфазной модели выглядит следующим образом:

$$\frac{\partial S_1}{\partial t} + \frac{u}{\emptyset}\left[\frac{dB(s_1)}{dS_1}\frac{\partial S_1}{\partial x} + \frac{\partial}{\partial x}\left(C(S_1)\frac{\partial S_1}{\partial x}\right)\right] = 0\ ; \qquad (1)$$

$$\text{где, } B(S_1) = \frac{\frac{\mu_1}{k_{r1(S_1)}} + \frac{k(\rho_1-\rho_2)gcos\alpha}{u}}{\frac{\mu_1}{k_{r1(S_1)}} + \frac{\mu_2}{k_{r2}}}\ ; \qquad (2)$$

$$\mathrm{C}(S_1) = \frac{k}{u}\frac{1}{\frac{\mu_1}{k_{r1(S_1)}} + \frac{\mu_2}{\mu_2}}\frac{\partial p_c(S_1)}{\partial S_1}\ ; \qquad (3)$$

Фракционный поток жидкости:

$$f_1(S_1) = \frac{\frac{\mu_1}{k_{r1}(S_1)} + \frac{k(\rho_1-\rho_2)gcos\alpha}{u}}{\frac{\mu_1}{k_{r1(S_1)}} + \frac{\mu_2}{k_{r2}}} + \frac{k}{u}\frac{1}{\frac{\mu_1}{k_{r1(S_1)}} + \frac{\mu_2}{\mu_2}}\frac{\partial p_c(S_1)}{\partial S_1} \qquad (4)$$

Где u — сумма скоростей фильтрации u1 и u2, как показано в следующем уравнении

$$u_1 = -\frac{kk_{r1}}{\mu_1}\left(\frac{\partial p_1}{\partial x} - \rho_1 gcos\alpha\right); \qquad (5)$$

$$u_2 = -\frac{kk_{r2}}{\mu_2}\left(\frac{\partial p_2}{\partial x} - \rho_2 gcos\alpha\right) \qquad (6)$$

Поскольку пористое пространство занято либо нефтью, либо водой, оно, естественно, должно иметь:

$$S_1 + S_2 = 1; \tag{7}$$

Предположим, что капиллярное давление является только функцией насыщения, так что $p_1 + p_2 = p_c(S_1)$.

Уравнение Бакли-Леверетта (BL) является наиболее простым выражением материального баланса для систем с водонапорным режимом. Несмотря на свою простоту, модель продемонстрировала замечательную точность. Эта простота вытекает из уравнения (1), где определенные допущения позволяют упростить общую форму двухфазной модели. Например, если предположить, что капиллярные силы пренебрежимо малы, система упрощается до уравнения Бакли-Леверетта.

$$\frac{\partial S_1}{\partial t} + \frac{u}{\emptyset}\left[\frac{df_1(s_1)}{dS_1}\frac{\partial S_1}{\partial x}\right] = 0, \;\; or \;\; \frac{\partial S_1}{\partial t} + v\left[\frac{\partial f_1(s_1)}{\partial x}\right] = 0 \tag{8}$$

где, $u$ является функцией $S_1$, как $u(S_1)$

$$f_1(S_1) = \frac{\frac{\mu_1}{k_{r1}(S_1)} + \frac{k(\rho_1 - \rho_2)gcos\alpha}{u}}{\frac{\mu_1}{k_{r1(S_1)}} + \frac{\mu_2}{k_{r2(S_1)}}}; \tag{9}$$

Если рассматривать только горизонтальный поток, то гравитационным членом можно пренебречь, дробный поток можно записать,

$$f_1(S_1) = \frac{1}{1 + \frac{\mu_1 k_{r2}(S_1)}{\mu_2 k_{r1}(S_1)}}; \tag{10}$$

Хотя уравнение Бакли-Леверетта (BL) обеспечивает надежное приближение для однородных кернов, точно определяя скорости флюидов, абсолютную проницаемость,

скорости закачки и перепады давления, оно сталкивается с ограничениями при применении к более крупным формациям. Такие факторы, как дисперсия, капиллярные эффекты и локальные неоднородности, становятся более сложными для оценки. Кроме того, ошибки выборки возникают из-за обширной площади резервуаров, где можно отобрать только несколько выбранных точек, что потенциально приводит к нерепрезентативным данным. Высокая стоимость сбора данных еще больше ограничивает доступность данных выборки.

Усреднение информации по разным масштабам представляет собой еще одну проблему. В пределах резервуара в образцах отражается различное поведение горных пород, что требует выбора репрезентативного объема элемента для получения времени прорыва или других результатов, представляющих интерес. Этот процесс часто включает отказ от разброса значений в пользу более репрезентативного значения, что вносит потенциальные ошибки в модель. Следовательно, существует внутренняя неопределенность в количественной оценке входных данных модели.

Можно утверждать, что кривая дробного потока более фундаментальна, чем относительная проницаемость. Кривая дробного потока служит входным механизмом для различных физических ситуаций, тем самым выступая в качестве канала, через который неопределенность проникает в модель. Она также предлагает ясную и логичную физическую интерпретацию.

## 1.2. Модель Black Oil.

Была использована модель Black Oil, которая является основой для синтеза реалистичных данных для моделирования. Путем численного решения этой модели создается большой синтетический набор данных, который служит основой для последующего машинного обучения. Этот процесс гарантирует, что наши данные отражают реалистичные характеристики нефтяных месторождений и тщательно настроены для высокой точности прогнозирования добычи нефти.

Модель Black Oil является фундаментальным инструментом в инженерии нефтяных месторождений для симуляции поведения нефтяных резервуаров. Модель упрощает сложную динамику жидкостей внутри резервуара до более управляемой структуры, при этом сохраняя важные характеристики системы.

В данном пункте рассматривается одномерная модель Black Oil. Поток в резервуаре состоит из трех фаз и трех компонентов (вода, нефть и газ). Модель предполагает, что жидкости несжимаемы, эффект температуры, силы тяжести и капиллярного давления незначителен, а поток подчиняется закону Дарси и закону сохранения массы. Математическая модель Black Oil записывается следующим образом:

Фаза воды:

$$\frac{\partial}{\partial t}\left(\emptyset \frac{\partial S_w}{\partial t}\right) = \frac{\partial}{\partial x}\left(\frac{k k_{rw}}{\mu_w B_w}\frac{\partial P}{\partial x}\right) + q_w \qquad (11)$$

Фаза нефти:

$$\frac{\partial}{\partial t}\left(\emptyset \frac{S_o}{B_o} + \emptyset \frac{R_v S_g}{B_g}\right) = \frac{\partial}{\partial x}\left(\frac{k k_{ro}}{\mu_o B_o}\frac{\partial P}{\partial x} + \frac{k k_{rg} R_v}{\mu_g B_g}\frac{\partial P}{\partial x}\right) + q_o \qquad (12)$$

Фаза газа:

$$\frac{\partial}{\partial t}\left(\emptyset\frac{S_g}{B_g}+\ \emptyset\frac{R_S S_o}{B_o}\right)=\frac{\partial}{\partial x}\left(\frac{kk_{rg}}{\mu_g B_g}\frac{\partial P}{\partial x}+\ \frac{kk_{rg}R_s}{\mu_g B_o}\frac{\partial P}{\partial x}\right)+q_g \qquad (13)$$

В этой модели предполагается, что нефть и газ несмешиваемы. Таким образом, математические модели фаз нефти и газа упрощаются следующим уравнением:
Для фазы нефти:

$$\frac{\partial}{\partial t}\left(\emptyset\frac{S_o}{B_o}\right)=\frac{\partial}{\partial x}\left(\frac{kk_{ro}}{\mu_o B_o}\frac{\partial P}{\partial x}\right)+q_o \qquad (14)$$

Для фазы газа:

$$\frac{\partial}{\partial t}\left(\emptyset\frac{S_g}{B_g}\right)=\frac{\partial}{\partial x}\left(\frac{kk_{rg}}{\mu_g B_g}\frac{\partial P}{\partial x}\right)+q_g \qquad (15)$$

Применив правило к левой части уравнения (15), уравнение для газовой фазы было получено следующим образом:

$$\frac{\emptyset}{B_g}\frac{\partial S_g}{\partial t}+\left(\frac{\emptyset S_g}{B_g}C_g+\frac{\emptyset S_g}{B_g}C_r\right)\frac{\partial p}{\partial t}=\frac{\partial}{\partial x}\left(\frac{kk_{rg}}{\mu_g B_g}\frac{\partial P}{\partial x}\right)+q_g \qquad (16)$$

где, $C_g=B_g\frac{\partial}{\partial p}\left(\frac{1}{B_g}\right)$ и $C_r=\frac{1}{\emptyset}\frac{\partial\emptyset}{\partial p}$

Обе стороны уравнения (6) умножаются на $\frac{B_g}{B_w}$, в результате получаем следующее:

$$\frac{\emptyset}{B_w}\frac{\partial S_g}{\partial t}+\left(\frac{\emptyset S_g}{B_w}C_g+\frac{\emptyset S_g}{B_w}C_r\right)\frac{\partial p}{\partial t}=\frac{\partial}{\partial x}\left(\frac{kk_{rg}}{\mu_g B_g}\frac{\partial P}{\partial x}\right)\frac{B_g}{B_w}+\frac{B_g}{B_w}q_g \qquad (17)$$

Аналогично, из уравнения (11) можно получить уравнение для водной фазы следующим образом:

$$\frac{\emptyset}{B_w}\frac{\partial S_w}{\partial t}+\left(\frac{\emptyset S_w}{B_w}C_w+\frac{\emptyset S_w}{B_w}C_r\right)\frac{\partial p}{\partial t}=\frac{\partial}{\partial x}\left(\frac{kk_{rw}}{\mu_w B_w}\frac{\partial P}{\partial x}\right)+q_w \qquad (18)$$

где, $C_w=B_w\frac{\partial}{\partial p}\left(\frac{1}{B_w}\right)$ и $C_r=\frac{1}{\emptyset}\frac{\partial\emptyset}{\partial p}$

Наконец, из уравнения (14) можно получить уравнение

для нефтяной фазы следующим образом:

$$\frac{\emptyset}{B_w}\frac{\partial S_o}{\partial t} + \left(\frac{\emptyset S_o}{B_w}C_o + \frac{\emptyset S_o}{B_w}C_r\right)\frac{\partial p}{\partial t} = \frac{\partial}{\partial x}\left(\frac{k k_{ro}}{\mu_o B_o}\frac{\partial P}{\partial x}\right)\frac{B_o}{B_w} + q_o\frac{B_o}{B_w} \quad (19)$$

где, $C_o = B_w\frac{\partial}{\partial p}\left(\frac{1}{B_w}\right)$ и $C_r = \frac{1}{\emptyset}\frac{\partial \emptyset}{\partial p}$

Уравнения (17-19) взаимосвязаны посредством применения ограничений, следующим образом:

$$S_o + S_w + S_g = 1, \quad (20)$$

где $\emptyset$ — это пористость, $S_o, S_g$ и $S_w$ — насыщенность нефти, газа и воды соответственно, $\vec{v}_w$ и $\vec{v}_o$ — скорости фаз воды и нефти, k — абсолютная проницаемость, $\mu_w, \mu_o$ и $\mu_g$ — вязкость воды, нефти и газа, соответственно. $B_w$, $B_o$ и $B_g$ — объемные коэффициенты образования воды, нефти и газа соответственно, $k_{rw}, k_{ro}$ и $k_{rg}$ — относительные проницаемости воды, нефти и газа соответственно. $C_w, C_o, C_g$ и $C_r$ — сжимаемости воды, нефти, газа и породы соответственно.

В начальный момент времени значения насыщенности и давления принимают следующие значения:

$S_w|_{t=0} = S_{w0}\,; S_o|_{t=0} = S_{o0}\,;\; S_g|_{t=0} = S_{g0}\,;\;\; P|_{t=0} = P_0$ (21)

Граничные условия в следующем виде:

$$S_w|_{x=0} = S_{w_inj}\,,\; \frac{\partial S_w}{\partial x}|_{x=1} = 0;\;\; \frac{\partial S_o}{\partial x}|_{x=1} = 0; \frac{\partial S_g}{\partial x}|_{x=1} = 0;$$
$$P|_{x=0} = P_{inj}\,, P|_{x=1} = P_{prod}\,; \quad (22)$$

где $S_{w_inj}$ обозначает насыщенность закачиваемой воды, а $P_{inj}$ и $P_{prod}$ – обозначают нагнетающее и добывающее давление.

***Численное решение модели Black Oil.***

Уравнение давления получается с использованием уравнений (17-20). Суммируя эти уравнения, в итоге было получено следующее уравнение:

$$\frac{\emptyset C_t}{B_w}\frac{\partial p}{\partial t} = \frac{\partial}{\partial x}\left(\lambda_g \frac{\partial P}{\partial x}\right)\frac{B_g}{B_w} + \frac{\partial}{\partial x}\left(\lambda_o \frac{\partial P}{\partial x}\right)\frac{B_o}{B_w} + \frac{\partial}{\partial x}\left(\lambda_w \frac{\partial P}{\partial x}\right) + \frac{B_g}{B_w} q_g + q_o \frac{B_o}{B_w} + q_w \quad (23)$$

где, $C_t = S_w C_w + S_o C_o + S_g C_g + C_r$; $\lambda_i = \frac{k k_{ri}}{\mu_i B_i}$ $(i = w, o, g)$ является отношением подвижностей.

В данной работе уравнение (23) решается методом конечных разностей с использованием известной итерационной схемы Якоби. Конечная разностная форма уравнения (23) имеет следующий вид:

$$\frac{\emptyset C_t}{B_w}\left(\frac{P_{i+1}^n - P_i^{n+1}}{\Delta t}\right) = \frac{1}{\Delta x_i}\frac{B_o}{B_w}\left(\lambda_{0,i-\frac{1}{2}} \frac{P_{i-1}^{n+1} - P_i^{n+1}}{\Delta x_{i-\frac{1}{2}}} + \lambda_{0,i+\frac{1}{2}} \frac{P_{i+1}^{n+1} - P_i^{n+1}}{\Delta x_{i+\frac{1}{2}}}\right) + \frac{1}{\Delta x_i}\frac{B_g}{B_w}\left(\lambda_{g,i-\frac{1}{2}} \frac{P_{i-1}^{n+1} - P_i^{n+1}}{\Delta x_{i-\frac{1}{2}}} + \lambda_{g,i+\frac{1}{2}} \frac{P_{i+1}^{n+1} - P_i^{n+1}}{\Delta x_{i+\frac{1}{2}}}\right) + \frac{1}{\Delta x_i}\left(\lambda_{w,i-\frac{1}{2}} \frac{P_{i-1}^{n+1} - P_i^{n+1}}{\Delta x_{i-\frac{1}{2}}} + \lambda_{w,i+\frac{1}{2}} \frac{P_{i+1}^{n+1} - P_i^{n+1}}{\Delta x_{i+\frac{1}{2}}}\right) + \frac{B_g}{B_w} q_g + q_o \frac{B_o}{B_w} + q_w; \quad (24)$$

Обе стороны уравнения (24) умножаются на объем потока жидкости $V_i = \Delta x_i A$, где A — это площадь поперечного сечения.

$$\frac{V_i\,\emptyset C_t}{B_w}\left(\frac{P_{i+1}^{n}-P_i^{n+1}}{\Delta t}\right)$$
$$=\left(\frac{AB_o}{B_w}\frac{\lambda_{0,i-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2}}}+\frac{AB_g}{B_w}\frac{\lambda_{g,i-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2}}}+A\frac{\lambda_{w,i-\frac{1}{2}}}{\Delta x_{i-\frac{1}{2}}}\right)(P_{i-1}^{n+1}$$
$$-P_i^{n+1})+$$
$$\left(\frac{AB_o}{B_w}\frac{\lambda_{0,i+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}+\frac{AB_g}{B_w}\frac{\lambda_{g,i+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}+A\frac{\lambda_{w,i+\frac{1}{2}}}{\Delta x_{i+\frac{1}{2}}}\right)(P_{i+1}^{n+1}-P_i^{n+1})+V_i\frac{B_g}{B_w}q_g+$$
$$V_iq_o\frac{B_o}{B_w}+V_iq_w; \qquad (25)$$

где, $\lambda_{p,i\pm\frac{1}{2}}=\frac{k_{i\pm\frac{1}{2}}k_{rp,i\pm\frac{1}{2}}}{\mu_pB_p}$, $k_{i+\frac{1}{2}}=\frac{k_i+k_{i+1}}{2}$, $k_{i-\frac{1}{2}}=\frac{k_i+k_{i-1}}{2}$, $k_{rp,i+\frac{1}{2}}=\frac{k_{rp,i}+k_{rp,i+1}}{2}$,
$k_{rp,i-\frac{1}{2}}=\frac{k_{rp,i}+k_{rp,i-1}}{2}$; $p=(w,o,g)$

Был заменен $\frac{\lambda_{0,i\pm\frac{1}{2}}}{\Delta x_{i\pm\frac{1}{2}}}$, $V_iq_g$, $V_iq_o$ , $V_iq_w$ на $T_{w,i\pm\frac{1}{2}}$, $Q_{g,i}$, $Q_{o,i}$, $Q_{w,i}$, в результате была получена

$$\frac{V_i\,\emptyset C_t}{B_w}\left(\frac{P_{i+1}^{n}-P_i^{n+1}}{\Delta t}\right)=\left(\frac{AB_o}{B_w}T_{o,i-\frac{1}{2}}+\frac{AB_g}{B_w}T_{g,i-\frac{1}{2}}+\right.$$
$$\left.AT_{w,i-\frac{1}{2}}\right)(P_{i-1}^{n+1}-P_i^{n+1})+\left(\frac{AB_o}{B_w}T_{o,i+\frac{1}{2}}+\frac{AB_g}{B_w}T_{g,i+\frac{1}{2}}+\right.$$
$$\left.AT_{w,i-\frac{1}{2}}\right)(P_{i+1}^{n+1}-P_i^{n+1})+\frac{B_g}{B_w}Q_{g,i}+\frac{B_o}{B_w}Q_{o,i}+Q_{w,i}; \qquad (26)$$

Обе стороны уравнения (26) умножаются на $\frac{B_w\Delta t}{V_i\emptyset C_t}$, и тогда окончательный вариант численного уравнения для неявного давления получается следующим образом:

$$P_i^{n+1} = P_i^n + \frac{B_w \Delta t}{V_i \emptyset C_t}\left[\left(\frac{B_o}{B_w} T_{o,i-\frac{1}{2}} + \frac{B_g}{B_w} T_{g,i-\frac{1}{2}} + T_{w,i-\frac{1}{2}}\right)(P_{i-1}^{n+1} - P_i^{n+1}) + \left(\frac{B_o}{B_w} T_{o,i+\frac{1}{2}} + \frac{B_g}{B_w} T_{g,i+\frac{1}{2}} + T_{w,i-\frac{1}{2}}\right)(P_{i+1}^{n+1} - P_i^{n+1}) + \frac{B_g}{B_w} Q_{g,i} + \frac{B_o}{B_w} Q_{o,i} + Q_{w,i}\right] \quad (27)$$

Аналогично, численное решение для насыщенности воды, нефти и газа решается методом конечных разностей с использованием итерационной схемы Якоби.

Из уравнения (18) численное решение для водонасыщенности было получено следующим образом:

$$S_{w,i}^{n+1} = S_{w,i}^n + \frac{B_{w,i} \Delta t}{V_i \emptyset_w}\left[T_{w,i-\frac{1}{2}}(P_{i-1}^{n+1} - P_i^{n+1}) + T_{w,i+\frac{1}{2}}(P_{i+1}^{n+1} - P_i^{n+1}) + Q_{w,i}\right]$$

$$-S_{w,i}^n (C_w - C_r)(P_{i1}^{n+1} - P_i^n) \quad (28)$$

Из уравнения (19) численное решение для нефтенасыщенности получается следующим образом:

$$S_{o,i}^{n+1} = S_{o,i}^n + \frac{B_{o,i} \Delta t}{V_i \emptyset_o}\left[T_{o,i-\frac{1}{2}}(P_{i-1}^{n+1} - P_i^{n+1}) + T_{o,i+\frac{1}{2}}(P_{i+1}^{n+1} - P_i^{n+1}) + Q_{o,i}\right]$$

$$-S_{o,i}^n (C_w - C_r)(P_{i1}^{n+1} - P_i^n) \quad (29)$$

Наконец, будет использовано уравнение (20), чтобы получить численное решение для насыщенности газа следующим образом:

$$S_{g,i}^{n+1} = 1 - \left(S_{o,i}^{n+1} + S_{w,i}^{n+1}\right) \quad (30)$$

Таблица 1 показывает входные параметры модели Black Oil, использованные в экспериментах.

**Таблица 1.** Параметры модели

| Параметр | Описание | Значение |
|---|---|---|
| $k$ | Абсолютная проницаемость | 0.001 |
| $\mu_w$ | Вязкость воды | 0.09 |
| $\mu_o$ | Вязкость нефти | 0.3 |
| $\mu_g$ | Вязкость газа | 0.01 |
| $\emptyset$ | Пористость | 0.25 |
| $P_{inj}$ | Нагнетающее давление | 0.5 |
| $P_{init}$ | Начальное давление | 0.3 |
| $P_{prod}$ | Добывающее давление | 0.1 |
| $S_{w_inj}$ | Насыщенность нагнетающей воды | 1.0 |
| $S_{w_init}$ | Начальная насыщенность воды | 0.2 |
| $S_{o_init}$ | Начальная насыщенность нефти | 0.7 |
| $S_{g_init}$ | Начальная насыщенность газа | 0.1 |
| $B_w$ | Объемный коэффициент воды | 1 |
| $B_o$ | Объемный коэффициент нефти | 1.2 |
| $B_g$ | Объемный коэффициент газа | 0.5 |

Используя различные вариации значений этих параметров, был собран синтетический набор данных. Следующий подраздел объясняет подробности этого набора

данных.

### 1.3.Разработка математических моделей для решения задачи неравновесной фильтрации (композиционная модель).

Моделирование неравновесной фильтрации представляет собой важный этап в анализе многих инженерных и геолого-геофизических задач. Композиционная модель, в свою очередь, представляет собой совокупность уравнений, учитывающих влияние различных компонентов смеси (нефти, газа, воды) и их взаимодействие с пластом на протяжении времени. Каждая из этих моделей имеет свои уникальные применения и предполагает различные уровни сложности при решении задач неравновесной фильтрации в различных геологических условиях.

Определим композиционную модель в общем виде. Для этого запишем закон сохранения массы:

$$\begin{gathered}\frac{\partial(\phi\xi_w S_w)}{\partial t} + div(\xi_w u_w) = q_w \\ \frac{\partial(\phi[x_{mo}\xi_o S_o + x_{mg}\xi_g S_g])}{\partial t} + div\left(x_{mo}\xi_o u_o + x_{mg}\xi_g u_g\right) = \\ q_m,\ m = 1,2,\dots,N_c \end{gathered} \quad (31)$$

где $\phi$ – пористость, $\xi_w$, $\xi_o$, $\xi_g$ – молярные плотности фаз воды, нефти и газа соответственно, $x_{mo}$, $x_{mg}$ – мольные доли компонента $m$ в фазах нефти и газа, $S_w$, $S_o$, $S_g$ – насыщенности фаз воды, нефти и газа, $q_w$ – молярная скорость, $q_m$ – молярная плотность компонента $m$, $u_w$, $u_o$, $u_g$ – молярные скорости фаз воды, нефти и газа, описываются законом Дарси:

$$u_\alpha = -\frac{k_{r\alpha}}{\mu_\alpha} k \nabla p_\alpha, \qquad \alpha = o,\ g,\ w \tag{32}$$

где $k_{r\alpha}$ – относительные фазовые проницаемости фаз, $\mu_\alpha$ – вязкость фаз, $k$ – абсолятная проницаемость, $p_\alpha$ – давление фаз. Молярные плотности и мольные доли можно рассчитать по следующей формуле:

$$\begin{gathered} \xi_\alpha = \sum_{m=1}^{N_c} \xi_{m\alpha}, \alpha = o,\ g \\ x_{m\alpha} = n_{m\alpha}/n_\alpha, m = 1,2,\dots,N_c, \alpha = o,\ g \\ \sum_{m=1}^{N_c} x_{mo} = 1, \sum_{m=1}^{N_c} x_{mg} = 1, S_o + S_w + S_g = 1, \end{gathered} \tag{33}$$

Здесь $n_{m\alpha}$ – количество молей компонента $m$, $n_\alpha$ – общее количество молей. Уравнения для капиллярных давлений выглядят следующим образом:

$$p_{cow} = p_o - p_w, \quad p_{cgo} = p_g - p_o$$

Ниже задаются условия равновесия, которое описывается законом Гиббса для композиционной системы:

$$\begin{gathered} f_{mo}(p_o,\ x_{1o},\ x_{2o},\dots,x_{N_co}) = \\ f_{mg}(p_g,\ x_{1g},\ x_{2g},\dots,x_{N_cg}), \quad m = 1,2,\dots,N_c \\ f_{m\alpha} = p_\alpha x_{m\alpha} \varphi_{m\alpha}, m = 1,2,\dots,N_c, \alpha = o,g \end{gathered} \tag{34}$$

где $f_{mo}$ и $f_{mg}$ функции летучести.

Распределение химических компонентов в углеводородной фазе описывается методом K-значения. Термодинамическое поведение жидкостей в пластовых условиях описывается уравнениями состояния Пенга-Робинсона.

Переписав эти уравнения для системы из трех фаз и двух компонентов получим следующие уравнения:

$$\phi\frac{\partial(\xi_w S_w)}{\partial t}+div(\xi_w u_w)=q_w$$
$$\phi\frac{\partial(x_{1o}\xi_o S_o+x_{1g}\xi_g S_g)}{\partial t}+div\left(x_{1o}\xi_o u_o+x_{1g}\xi_g u_g\right)=q_1$$
$$\phi\frac{\partial(x_{1o}\xi_o S_o+x_{2g}\xi_g S_g)}{\partial t}+div(x_{2o}\xi_o u_o+x_{2g}\xi_g u_g)=q_2$$
$$u_\alpha=-\frac{k_{r\alpha}}{\mu_\alpha}k\nabla p_\alpha,\qquad \alpha=o,\ g,\ w$$
$$x_{1o}+x_{2o}=1,\ x_{1g}+x_{2g}=1,\ S_o+S_w+S_g=1,$$
$$p_{cow}=p_0-p_w,\ \ p_{cgo}=p_g-p_0.$$
$$f_{1o}(p_o,\ x_{1o},\ x_{2o})=f_{1g}\left(p_g,\ x_{1g},\ x_{2g}\right)$$
$$f_{2o}(p_o,\ x_{1o},\ x_{2o})=f_{2g}(p_g,\ x_{1g},\ x_{2g}) \qquad (35)$$

Данная система нелинейных уравнений была линеаризована методом Ньютона-Рафсона.

В ($l$+1)-м слое итерационного процесса Ньютона значение неизвестных обновляется по следующему закону:

$$x_{mo}^{n+1,l+1}=x_{mo}^{n+1,l}+\Delta\, x_{mo}^{n+1,l+1},\quad m=1,2,\dots,N_c-1,$$
$$\zeta_{mo}^{n+1,l+1}=\zeta_{mo}^{n+1,l}+\Delta\,\zeta_{mo}^{n+1,l+1},\qquad m=1,2,\dots,N_c-1,$$
$$L^{n+1,l+1}=L^{n+1,l}+\Delta\, L^{n+1,l+1},$$
$$F^{n+1,l+1}=F^{n+1,l}+\Delta\, F^{n+1,l+1},$$
$$S^{n+1,l+1}=S^{n+1,l}+\Delta\, S^{n+1,l+1},$$
$$p^{n+1,l+1}=p^{n+1,l}+\Delta\, p^{n+1,l+1}. \qquad (36)$$

Переписав данные уравнения для каждой точки разностной сетки можно получить систему линейных уравнений вида $Ax=b$, где $x$ будет представлять собой вектор поправок метода Ньютона. Решение СЛАУ такого вида весьма распространенная задача, для которой существуют множество методов, из которых был выбран

метод GMRES (Generilized minimum residuals).

### 1.4. Стохастическая модель Баклея-Леверетта.

Как правило, метод стохастического моделирования используется для определения неопределенности нефтяного резервуара. Существуют различные методы, используемые для метода стохастического моделирования. Например, схема Upwind обычно используется для BL для обработки нелинейного образования ударной волны, а неявный/явный временной шаг используется для выбора на основе устойчивости нашего численного метода; в то время как обработка стохастичности используется для обработки стохастической природы проблемы, можно применить моделирование Монте-Карло (MCS), которое случайным образом выбирает стохастические переменные (например, поле проницаемости) и решает детерминированное уравнение Бакли-Леверетта для каждой реализации. Это даст нам статистическое распределение решения. Другой — стохастический метод Галеркина, который аппроксимирует стохастические компоненты, расширяя их в терминах ортогональных полиномов (например, разложение Винера-Эрмита). Это приводит к системе детерминированных уравнений, связанных через стохастические переменные.

В данной работе метод MCS был применен к стохастическому уравнению BL, которое широко используется для оценки неопределенностей при управлении нефтяными месторождениями. Он назначит распределение вероятностей для каждого неопределенного параметра, такого как проницаемость, пористость, вязкость и относительная проницаемость. Предположим, что проницаемость kx, ω — это случайное поле, описываемое распределением вероятностей, обычно логнормальным распределением:

$$k(x,\omega) = \underline{k}(x) + \sigma_k \delta(\omega); \quad (37)$$

где, $\underline{k}(x)$— средняя проницаемость, $\sigma_k$ — стандартное отклонение, а $\delta(\omega)$ представляет собой случайную величину, лежащую ниже стандартного нормального распределения, $N(\underline{k}(x),\ \sigma^2)$.

Учитывая случайность проницаемости, скорость Дарси v и фракционный поток $f_1(S_1)$ становятся стохастическими процессами:

$$\frac{\partial S_1(x,t)}{\partial t} + v(x)\left[\frac{\partial f_1(s_1(x,t))}{\partial x}\right] = 0; \quad (38)$$

где $v(x)$ — случайная величина, на которую влияет поле случайной проницаемости:

$$v(x) = \frac{-k(x,\omega)}{\mu_1(x,\omega)}\ \frac{\partial p(x)}{\partial x}; \quad (39)$$

А $f_1(S_1(x,t))$ — случайная величина, на которую влияет случайная вязкость воды и нефтяного месторождения:

$$f_1(S_1(x,t)) = \frac{1}{1+\frac{\mu_1(x,\omega)k_{r2}(S_1)}{\mu_2(x,\omega)k_{r1}(S_1)}} \quad (40)$$

**Результаты.**

В таблице 2 показаны параметры, использованные в нашем эксперименте.

**Таблица 2.** Параметры стохастической модели Баклея-Леверетта

| | |
|---|---|
| Начальная водонасыщенность | 0.2 |
| Впрыск воды | 1.0 |
| Начальное давление | 0.3 |
| Впрыск под давлением | 0.5 |
| Продукт давленият | 0.1 |

В этой работе четыре различных параметра неопределенности (пористость, вязкость нефти, вязкость воды и проницаемость) отбираются с использованием метода MCS. Таблица 3 показывает среднее и стандартное значение каждого параметра неопределенности для использования стандартного нормального распределения.

**Таблица 3.** Среднее и стандартное значение каждого параметра неопределенности

| | |
|---|---|
| Среднее значение проницаемости | 0.004 |
| Стандартное значение проницаемости | 0.002 |
| Среднее значение вязкости воды | 0.09 |
| Стандартное значение вязкости воды | 0.001 |
| Среднее значение вязкости нефти | 0.5 |
| Нормативное значение вязкости нефти | 0.1 |
| Среднее значение пористости | 0.5 |
| Стандартное значение пористости | 0.2 |

Эксперимент показывает, что результат отклонения водонасыщенности и коэффициента извлечения оказывает существенное влияние на эффективность извлечения нефти, причем более высокие отклонения приводят к снижению эффективности извлечения из-за неравномерного вытеснения и ухода нефти. В ходе работы над математическими моделями по каждой модели был собран синтетический набор данных, позволяющий изучить влияние различных параметров на результаты моделирования. Используя различные вариации значений этих параметров, можно наблюдать, как изменения влияют на динамику системы и ее поведение в различных условиях. Данный подход не только углубляет наше понимание процессов, происходящих в многокомпонентных системах, но также открывает новые горизонты для применения этих моделей в реальных сценариях. В следующем подразделе рассматривается использование собранного синтетического набора данных для предсказания коэффициента извлечения нефти (oil recovery factor) с применением методов машинного обучения.

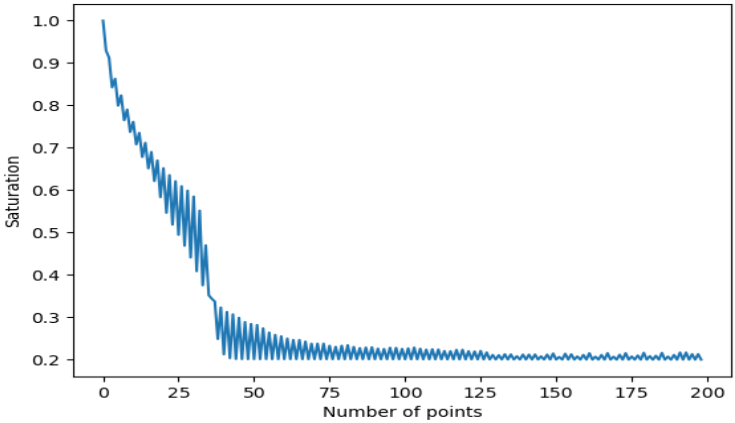



Рисунок 1. Водонасыщенность, отклонение водонасыщенности

## 2. Разработка интеллектуальных вычислительных моделей для решения задачи неравновесной фильтрации и стохастического моделирования.

В данном разделе сосредоточено внимание на разработке интеллектуальных вычислительных моделей, применяемых для решения задачи неравновесной фильтрации и стохастического моделирования. Эти модели опираются на современные методы машинного обучения и нейронные сети, что позволяет значительно повысить эффективность и точность вычислений.

### 2.1. Разработка интеллектуальных вычислительных моделей для решения задачи неравновесной фильтрации (композиционная модель).

Целью данного исследования является применение методов машинного обучения для решения задачи неравновесной фильтрации. Набор данных использованные в данном исследовании были получены из математической модели Бакли-Леверетта, в котором было смоделировано неравновесной фильтрации. Синтетический датасет был сгенерирован при различных значениях входных параметров добычи нефти. Каждый пример имеет 4 входных параметра: пористость, вязкость нефтеной фазы, абсолютная проницаемость породы и временная итерация. Кроме того, для применения алгоритмов машинного обучения на выходе был выбран коэффициент нефтеотдачи. Диапазоны и количество сгенерированных данных приведены в таблице 4.

Таблица 4. Описание и статистический анализ реализованных выборок данных

| Параметры | Количество вариации | Миним. Значение | Максим. значение | Среднее значение |
|---|---|---|---|---|
| Пористость | 41 | 0.11 | 0.289 | 0.167 |
| Вязкость | 41 | 0.105 | 0.485 | 0.243 |
| Абсолютная проницаемость | 6 | 0.01 | 0.095 | 0.045 |

В каждом запуске рассчитывается значение коэффициента нефтеотдачи пласта. Для исследования динамики коэффициента нефтеотдачи пласта были взяты 40 значений при различных промежуточных временных итерации математической модели для каждого примера выборки. Следовательно, общее количество набора данных составила более 403440 примера. На рисунке 2 вы можете увидеть сгенерированный набор данных из модели Бакли-Леверетта.

Out[4]:

| | Class | etta | Kviews | PORO | TIME | VISC_OIL |
|---|---|---|---|---|---|---|
| 0 | 1.0 | 0.004099 | 2.0 | 0.13089 | 0.0 | 0.414938 |
| 1 | 1.0 | 0.028301 | 2.0 | 0.13089 | 200.0 | 0.414938 |
| 2 | 1.0 | 0.049026 | 2.0 | 0.13089 | 400.0 | 0.414938 |
| 3 | 1.0 | 0.068843 | 2.0 | 0.13089 | 600.0 | 0.414938 |
| 4 | 1.0 | 0.087731 | 2.0 | 0.13089 | 800.0 | 0.414938 |

Рисунок 2. Набор данных из модели Бакли-Леверетта

Датасет был разделен на обучающий и тестовый сет. Для обучения использовано 80% от данных, а для теста остальные 20% данных. Рисунок 3 представляет собой описательную статистику для набора данных, содержащего

403440 записей, с шестью признаками: **Kviews**, **poro**, **visc**, **time**, **etta** и **class**. Признак **Kviews** имеет среднее значение 3.5 с разбросом в 1.71 (стандартное отклонение), минимальное значение 1 и максимальное 6.

| | Kviews | poro | visc | time | etta | class |
|---|---|---|---|---|---|---|
| count | 403440.000000 | 403440.000000 | 403440.000000 | 403440.000000 | 403440.000000 | 403440.000000 |
| mean | 3.500000 | 0.164208 | 0.217960 | 3900.000000 | 0.389585 | 4.407508 |
| std | 1.707827 | 0.050105 | 0.111983 | 2308.682137 | 0.203020 | 2.029491 |
| min | 1.000000 | 0.100200 | 0.105708 | 0.000000 | 0.003969 | 1.000000 |
| 25% | 2.000000 | 0.128535 | 0.125000 | 1950.000000 | 0.237126 | 3.000000 |
| 50% | 3.500000 | 0.155521 | 0.181488 | 3900.000000 | 0.399243 | 4.000000 |
| 75% | 5.000000 | 0.191939 | 0.283286 | 5850.000000 | 0.541174 | 6.000000 |
| max | 6.000000 | 0.279330 | 0.485437 | 7800.000000 | 0.986278 | 10.000000 |

Рисунок 3. Описательная статистика для набора данных

Признак poro (возможно, пористость) имеет среднее значение 0.164 и стандартное отклонение 0.05, при этом значения варьируются от 0.1 до 0.279. Вязкость (visc) имеет среднее значение 0.218 с минимальным значением 0.106 и максимальным 0.485. Time (время) варьируется от 0 до 7800, со средним значением 3900 и стандартным отклонением 2308. Признак etta имеет среднее значение 0.39 с разбросом 0.203, при этом значения находятся в диапазоне от 0.00397 до 0.986. Наконец, признак class представляет категориальные значения с медианой 4 и максимальным значением 10. Эти показатели позволяют оценить распределение данных и их вариативность для дальнейшего анализа и моделирования.

Наша задача относится к классу регрессионных задач с точки зрения методов машинного обучения. Реализованы и протестированы следующие регрессионные алгоритмы машинного обучения для прогнозирования коэффициента нефтеотдачи: множественная линейная регрессия (MLR),

метод опорных векторов (SVR), случайный лес (RF) и градиентный бустинг (GB).

Множественная линейная регрессия (MLR) — это метод статистического анализа, который используется для моделирования отношения между одной зависимой переменной и несколькими независимыми переменными. Он позволяет предсказать значения зависимой переменной на основе значений независимых переменных.

Модель множественной линейной регрессии описывается следующей формулой:

$$Y = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n + \varepsilon \quad (41)$$

где:

- $Y$ — зависимая переменная (например, цена дома).
- $\beta_0$ — свободный член (константа).
- $\beta_1, \beta_2 \ldots \beta_n$ — коэффициенты регрессии, показывающие, насколько изменяется Y при изменении соответствующей независимой переменной Xi.
- $x_1, x_2 \ldots x_n$ — независимые переменные (например, площадь, количество комнат и т.д.).
- $\varepsilon$ — ошибка модели (разница между фактическими и предсказанными значениями).

Коэффициенты регрессии β обычно оцениваются с помощью метода наименьших квадратов, который минимизирует сумму квадратов отклонений предсказанных значений от фактических:

$$min \sum_{i=1}^{N} (Y_i - \hat{Y}_i)^2 \quad (42)$$

где $\hat{Y}_i$ — предсказанные значения, полученные из модели.

Множественная линейная регрессия основывается на следующих допущениях:

**1. Линейная связь между зависимой и независимыми переменными.**

Первое допущение множественной линейной регрессии заключается в том, что существует линейная связь между зависимой переменной и каждой из независимых переменных. Лучший способ проверить линейные зависимости — создать диаграммы рассеяния и визуально проанализировать их на наличие линейности. Если связь, отображаемая на диаграмме рассеяния, не является линейной, то аналитик должен выполнить нелинейную регрессию или преобразовать данные с помощью статистического программного обеспечения, такого как SPSS.

**2. Независимые переменные не должны быть сильно коррелированы между собой.**

Данные не должны показывать мультиколлинеарность, которая возникает, когда независимые переменные (объясняющие переменные) сильно коррелированы между собой. Когда независимые переменные демонстрируют мультиколлинеарность, возникают проблемы с определением конкретной переменной, которая вносит вклад в дисперсию зависимой переменной. Лучший метод для проверки этого допущения — метод коэффициента инфляции дисперсии (Variance Inflation Factor).

**3. Дисперсия остатков должна быть постоянной.**

Множественная линейная регрессия предполагает, что величина ошибки остатков одинакова в каждой точке линейной модели. Это явление называется гомоскедастичностью. При анализе данных аналитик должен построить график стандартизированных остатков относительно прогнозируемых значений, чтобы определить, распределены ли точки равномерно по всем

значениям независимых переменных. Для проверки этого допущения данные можно изобразить на диаграмме рассеяния или с помощью статистического программного обеспечения, которое построит диаграмму рассеяния для всей модели.

**4. Независимость наблюдений.**

Модель предполагает, что наблюдения должны быть независимы друг от друга. Проще говоря, модель предполагает, что значения остатков являются независимыми. Для проверки этого допущения используется статистика Дарбина-Уотсона.

Тест покажет значения от 0 до 4, где значения от 0 до 2 указывают на положительную автокорреляцию, а значения от 2 до 4 — на отрицательную автокорреляцию. Среднее значение, то есть 2, указывает на отсутствие автокорреляции.

**5. Многомерная нормированность.**

Многомерная нормированность имеет место, когда остатки распределены нормально. Чтобы проверить это предположение, посмотрите, как распределены значения остатков. Это можно проверить двумя основными методами: гистограммой с наложенной нормальной кривой или методом нормального графика вероятности.

Метод опорных векторов (Support Vector Regression, SVR) — это расширение метода опорных векторов (SVM) для задач регрессии. SVR позволяет находить функцию, которая предсказывает значения зависимой переменной на основе значений независимых переменных, при этом контролируя ошибки предсказания. Основные концепции:

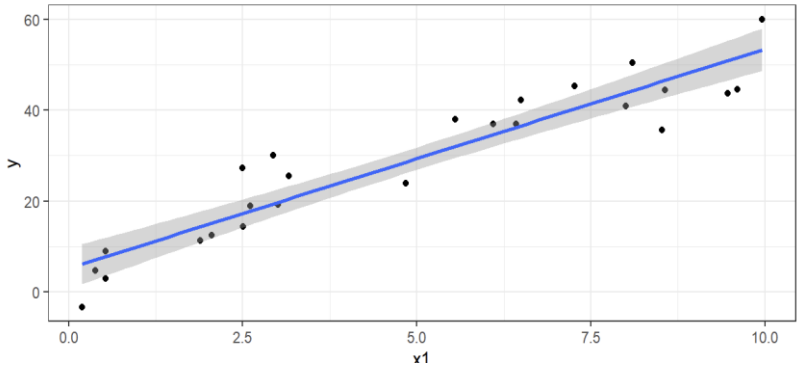



Рисунок 4. Интерпретация множественной линейной модели

**Цель SVR**: Найти функцию, которая минимизирует ошибки в пределах заданной допустимой зоны, называемой «epsilon-областью». Это значит, что предсказанные значения могут отклоняться от фактических, но в пределах этой области. За пределами этой области модель пытается минимизировать штраф.

**Форма функции**: В SVR функция может быть выражена как:

$$F(x) = w * x + b \quad (43)$$

где:

- $w$ — вектор весов,
- $x$ — вектор входных признаков,
- $b$ — свободный член.

SVR использует функцию потерь, которая включает как ошибку предсказания, так и регуляризацию:

$$Loss(y, f(x)) = \begin{cases} 0 & \text{если } |y - f(x)| \leq \varepsilon \\ |y - f(x)| - \varepsilon & \text{иначе} \end{cases} \quad (44)$$

Здесь y — истинное значение, f(x) — предсказанное значение. Регуляризация контролирует сложность модели, позволяя избежать переобучения.

SVR решает следующую задачу оптимизации:

$$min_{w,b}(\frac{1}{2}||w||^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*)) \quad (45)$$

при ограничениях:

$$y_i - (w * x_i + b) \leq \varepsilon + \xi_i$$
$$(w * x_i + b) - y_i \leq \varepsilon + \xi_i^*$$

$\xi_i, \xi_i^* \geq 0$

где $C$ — параметр регуляризации, который контролирует trade-off между точностью предсказания и сложностью модели.

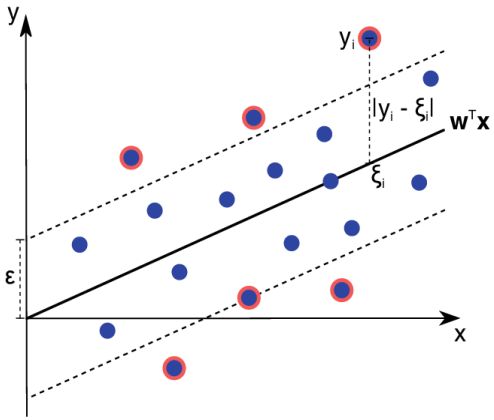



Рисунок 5. Support vector regression (SVR)

Случайный лес (Random Forest, RF) — это мощная техника машинного обучения, которая применяется как для задач классификации, так и регрессии. В основе этой модели лежит ансамблевый подход, состоящий из нескольких решающих деревьев, каждое из которых обучается на случайных подвыборках данных и принимает

самостоятельные решения. Модель улучшает точность прогнозов и устойчивость модели за счет агрегирования прогнозов каждого дерева в ансамбле, что делает её эффективной при работе с большими и сложными наборами данных.

**Ключевые концепции.**

**Бэггинг (Bootstrap Aggregating):** Основная интуиция бэггинга заключается в том, чтобы каждое дерево обучалось на случайной подвыборке данных с возвратом. Таким образом, можно построить несколько деревьев, каждое из которых частично основывается на разных наборах данных. Некоторые примеры могут несколько раз появляться для одного дерева, а для другого — не быть выбраны вообще. Этот метод уменьшает дисперсию модели, поэтому вероятность переобучения минимальна, что обычно случается в традиционных решающих деревьях.

**Случайный выбор признаков:** Помимо случайных подвыборок данных, на каждом этапе разбиения узлов в дереве случайный лес выбирает случайный набор признаков. Этот процесс уменьшает корреляцию между деревьями в лесу, увеличивая разнообразие предсказаний. Это важное отличие от простого случайного леса, который представляет собой бэггинг обычных решающих деревьев. Например, на каждом этапе вместо выбора лучшего признака из всех доступных признаков, случайный лес учитывает только подмножество признаков, что опять же делает модель более устойчивой к изменениям во входных данных.

**Структура алгоритма.**

**Обучение дерева:** Каждое дерево в случайном лесу строится рекурсивно, начиная с выбора случайных подвыборок данных, за которыми следует выбор случайных

подвыборок признаков. Лучший признак для разделения на каждом узле выбирается на основе таких критериев, как информационный выигрыш, энтропия или критерий Джини. Этот процесс продолжается до тех пор, пока не будут достигнуты некоторые остановочные условия, такие как максимальная глубина дерева или минимальное количество образцов в узле.

**Прогнозирование:** После того, как все деревья построены, предсказание осуществляется путем агрегирования предсказаний всех деревьев. В случае классификации используется метод большинства голосов, чтобы выбрать класс, за который проголосовало большинство деревьев. В случае регрессии для предсказания берется среднее значение предсказаний всех деревьев, что снижает вероятность крупных ошибок. Пример алгоритма показан на рисунке 6.

**Преимущества случайного леса.**

**Устойчивость к переобучению:** Большое количество деревьев и случайный выбор признаков делает случайный лес крайне устойчивым к переобучению. В то время как решающие деревья могут легко переобучаться из-за высокой сложности модели, случайный лес существенно снижает риск переобучения за счет объединения прогнозов большого числа деревьев.

**Высокая точность и производительность:** Случайный лес часто демонстрирует высокую производительность во многих задачах машинного обучения, где могут быть задействованы большие объемы данных и множество признаков. Модель легко справляется с отсутствующими данными, сохраняя точность при неполных данных.

**Работа с высокоразмерными данными:** Поскольку на каждом этапе алгоритм случайным образом выбирает подмножества признаков, он эффективно работает с

высокоразмерными данными, когда количество признаков велико. Случайный лес особенно полезен, когда заранее невозможно сказать, какой признак более важен.

**Оценка важности признаков:** Одним из самых полезных аспектов случайного леса является возможность оценки важности каждого признака для предсказания. Эта мера может быть использована для интерпретации модели и для выбора наиболее значимых переменных для дальнейшего анализа. Важность признака оценивается на основе того, как часто признак используется для разбиения узлов дерева и насколько он улучшает критерий разбиения.

**Устойчивость к выбросам:** Случайный лес достаточно устойчив к выбросам в данных. Поскольку предсказание основывается на усреднении результатов всех деревьев, выбросы не оказывают значительного влияния на окончательное предсказание, что делает алгоритм особенно полезным при наличии шумных данных.

**Гибкость:** Случайный лес можно адаптировать к любому типу данных — структурированным или неструктурированным. Алгоритм может работать как с категориальными, так и с числовыми данными, а также справляться с задачами многоклассовой классификации.

**Обнаружение аномалий:** Помимо задач классификации и регрессии, случайный лес может быть применен для обнаружения аномалий в данных. Деревья могут указать на необычные образцы, которые значительно отличаются от основной массы данных.

**Ограничения случайного леса.**

Несмотря на многочисленные преимущества, существуют и некоторые возможные ограничения использования случайного леса.

**Требования к вычислительным ресурсам:** Поскольку методология случайного леса предполагает построение

большого числа деревьев, анализ подвыборок данных и признаков может быть вычислительно затратным, особенно при работе с очень большими наборами данных.

**Сложность интерпретации:** В то время как одно решающее дерево легко интерпретировать, случайный лес состоит из множества деревьев, что делает модель менее интерпретируемой. Для интерпретации необходимо полагаться на такие методы, как важность признаков или визуализация усредненных прогнозов.

**Чувствительность к количеству деревьев:** Хотя случайный лес хорошо работает при большом количестве деревьев, слишком большое число деревьев может снизить производительность за счет переобучения модели.

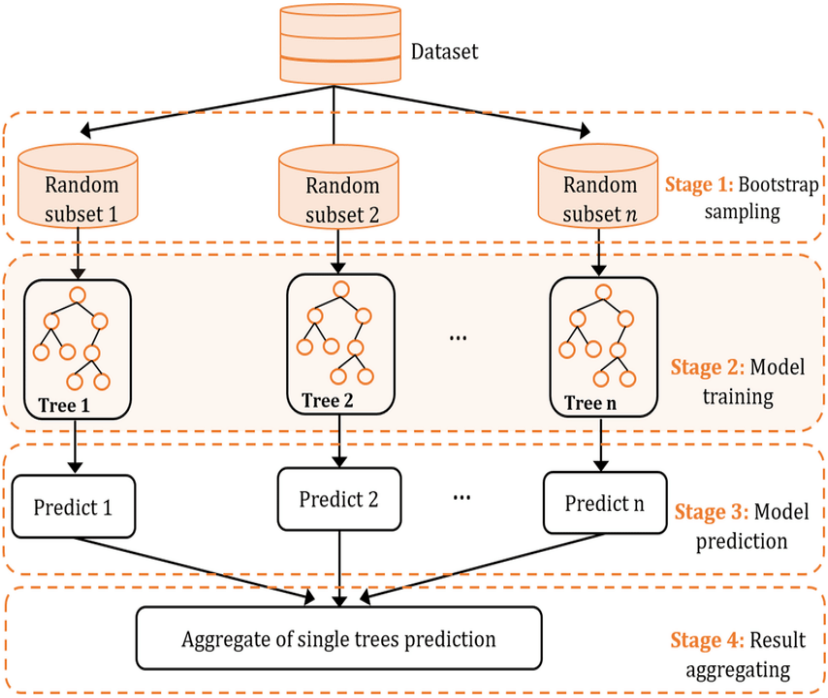



Рисунок 6. Пример алгоритма случайного леса (RF)

Градиентный бустинг (GB) — это метод ансамблевого обучения, который использует последовательное добавление слабых моделей для улучшения точности предсказаний. Основная идея заключается в том, чтобы обучать каждую новую модель на остатках предыдущей модели. Основные шаги градиентного бустинга:

- **Инициализация модели**: Начинаем с простой модели, часто это среднее значение целевой переменной для задач регрессии:

$$F_0(x) = argmin_\gamma \sum_{i=1}^{N} L(y_i, \gamma) \quad (46)$$

где L — функция потерь, yi — истинные значения, N — количество образцов.

- **Обучение слабых моделей**: На каждом шаге mmm добавляем новое дерево hm(x) для коррекции ошибок предыдущей модели. Остатки рассчитываются как:

$$r_i^{(m)} = y_i - F_{m-1}(x_i) \quad (47)$$

- **Градиентный шаг**: Обучаем дерево на остатках:

$$h_m(x) = argmin_h \sum_{i=1}^{N} L(y_i, F_{m-1}(x_i) + h(x_i)) \quad (48)$$

Здесь h(x) — новое дерево, которое минимизирует функцию потерь.

- **Обновление модели**: Обновляем текущую модель, добавляя новое дерево, с учетом коэффициента обучения η:

$$F_m(x) = F_{m-1}(x) + \eta h_m(x) \quad (49)$$

Обычно η выбирается малым, чтобы модель обновлялась постепенно.

Чтобы избежать переобучения, в градиентный бустинг могут добавляться методы регуляризации, такие как уменьшение глубины деревьев или добавление штрафов на весы деревьев.

На рисунке 7 показана архитектура, разработанная для прогнозирования на основе данных нефтяной промышленности.

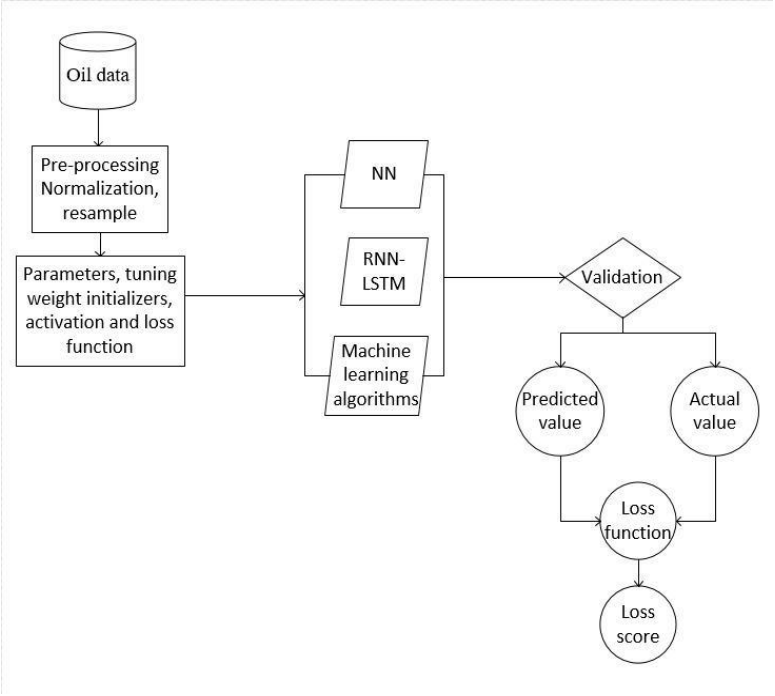



Рисунок 7. Архитектура машинного обучения для добычи нефти

На первом этапе используются сырые данные, которые проходят предварительную обработку. Это включает нормализацию данных, что помогает привести их в единый диапазон для улучшения качества обучения модели, а также ресемплирование, особенно актуальное для временных

рядов, чтобы устранить пропуски и привести данные к единообразным временным интервалам.

На рисунке 8 представлены графики плотности распределения для шести признаков из набора данных

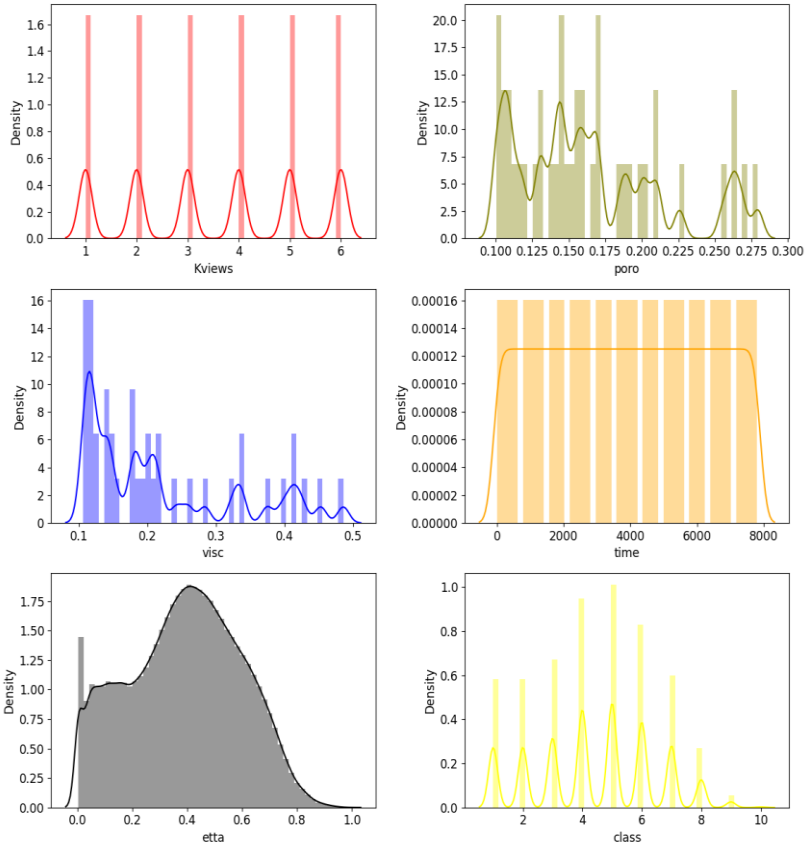



Рисунок 8. График плотности распределения для набора данных

Парный график показывает несколько заметных связей между переменными в наборе данных.

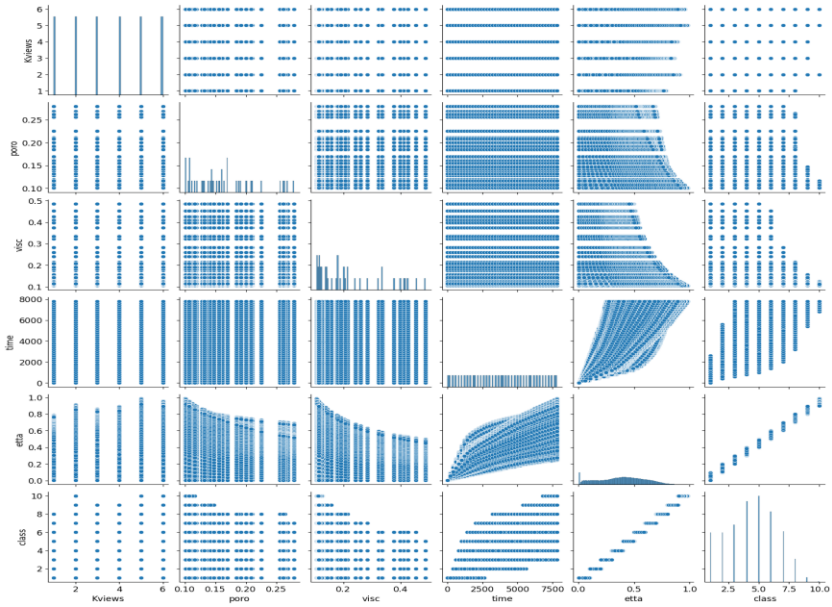



Рисунок 9. Парный график связей между переменными в наборе данных

Наиболее заметной является сильная положительная линейная связь между «etta», «class» и «time», которая совпадает с высокими корреляциями, наблюдавшимися ранее, что указывает на то, что эти переменные тесно связаны. Напротив, переменные, такие как «visc», показывают отрицательную связь как с «etta», так и с «class», на что указывают нисходящие тенденции на их диаграммах рассеяния. «Kviews» и «class» демонстрируют отдельные кластеры, отражающие их дискретную природу, в то время как переменные, такие как «poro» и «time», не демонстрируют четких закономерностей, что предполагает более слабое взаимодействие с другими переменными.

Матрица корреляции в рисунке 10 показывает, что большинство переменных в наборе данных слабо или совсем не коррелируют друг с другом, за исключением

«etta», «class» и «time», которые демонстрируют сильные положительные связи. «Etta» и «class» почти идеально коррелируют (0,99), что говорит о том, что они измеряют очень похожие аспекты данных, в то время как обе также показывают сильную корреляцию с «time» (0,79 и 0,78 соответственно), что указывает на то, что с течением времени эти переменные увеличиваются вместе. С другой стороны, такие переменные, как «visc» и «poro», показывают слабые отрицательные корреляции с «etta» и «class», что подразумевает обратную связь между этими свойствами. «Kviews» кажется в значительной степени независимым от всех других переменных, поскольку он демонстрирует корреляции, близкие к нулю на протяжении всего периода.

Рисунок 10. Матрица корреляции

После предобработки данных настраиваются ключевые параметры модели, такие как инициализация весов, выбор функции активации и функции потерь. Инициализация весов играет важную роль в скорости сходимости модели и её эффективности, а функции активации (например, ReLU или Sigmoid) используются для решения, активировать ли нейрон. Функция потерь, например, среднеквадратическая ошибка (MSE), измеряет, насколько предсказания модели отличаются от фактических значений, и направляет процесс обучения на минимизацию этой ошибки.

На следующем этапе данные поступают на вход различных моделей машинного обучения, таких как полносвязные нейронные сети (NN), рекуррентные сети с длинной краткосрочной памятью (RNN-LSTM), которые эффективны для анализа временных рядов, и другие алгоритмы машинного обучения. Эти модели обучаются на основе данных и пытаются предсказать значения будущих показателей.

На рисунке 11 показана нейронная сеть — это базовая модель, состоящая из входного, скрытого и выходного слоя. Входной слой принимает на вход пять характеристик: Etta, Kviews, Poro, Time и Visc_oil. Эти данные обрабатываются в скрытом слое, состоящем из 128 нейронов, и затем передаются на выходной нейрон, который выдает окончательное предсказание.

После получения предсказаний проводится этап валидации, где предсказанные значения сравниваются с фактическими данными. Ошибка между ними вычисляется с помощью выбранной функции потерь, и на основе этой оценки вычисляется итоговая величина ошибки модели. Цель — минимизировать этот показатель на каждом этапе обучения, что улучшит точность предсказаний модели в будущем.

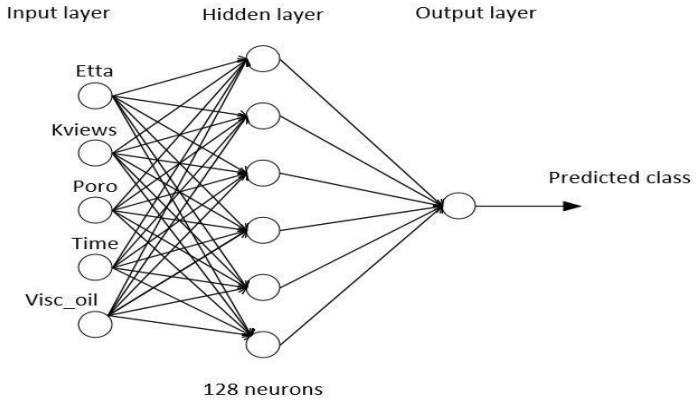



Рисунок 11. Архитектура нейронной сети для решение задачи вытеснения нефти

.

На рисунке 12 представлен результат прогнозирования коэффициента нефтеотдачи для случайно выбранных 1000 примеров:

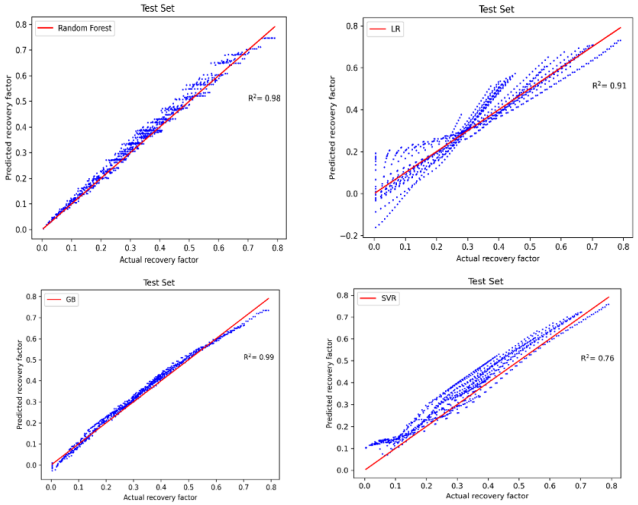



Рисунок 12. Результаты прогнозирования коэффициента нефтеотдачи для регрессионных алгоритмов

Для оценки регрессионных моделей были выбраны метрики коэффициент корреляции R2, средняя квадратическая ошибка MSE. В таблице 5 представлено сравнение между использованными регрессионными моделями с точки зрения индексов статистических ошибок.

Таблица 5. Результат оценки регрессионных алгоритмов

| Метрика | MLR | RF | SVR | GB |
|---|---|---|---|---|
| Коэффициент корреляции обучений ($R^2$) | 0.87 | 0.98 | 0.92 | 0.99 |
| Коэффициент корреляции теста ($R^2$) | 0.91 | 0.98 | 0.76 | 0.99 |
| MSE | 0.003 | 0.0006 | 0.009 | 0.0004 |

Прогнозы, сделанные одним деревом решений, могут быть неточными, но в сочетании прогноз будет ближе к среднему. Причина, по которой случайный лес обычно более точен, чем одно дерево решений, заключается в том, что из многих прогнозов включается гораздо больше знаний. Для задач регрессии случайный лес использует среднее значение деревьев решений для окончательного прогноза. Результаты модели опорных векторов (SVR) указывают на то, что она хорошо справляется с обучающими данными, но имеет более низкую способность обобщения на новые данные из-за переобучения. Результаты сравнения показали, что регрессионные модели случайного леса (RF) и градиентного бустинга (GB) дают более высокую точность и обладают отличной способностью к обобщению на тестовые данные для нашей выборки.

Кроме регрессионных алгоритмов, были использованы нейронные сети. На этапе создания и обучения нейронной сети была разработана модель с использованием Tensorflow

и Pytorch. Модель представляла собой последовательную нейронную сеть с двумя полносвязными слоями, активацией ReLU, и одним выходным слоем без активации. Модель была скомпилирована с функцией потерь в виде среднеквадратичной ошибки и оптимизатором 'adam'. Обучение модели проводилось на обучающей выборке с 100 эпохами и размером батча 32. На рисунке 13 представлен результат прогнозирования коэффициента нефтеотдачи для случайно выбранных 1000 примеров:

| | Тестовые данные | Прогноз |
|---|---|---|
| 0 | 0.486674 | 0.485748 |
| 1 | 0.082572 | 0.080607 |
| 2 | 0.360197 | 0.360908 |
| 3 | 0.175699 | 0.174224 |
| 4 | 0.590396 | 0.589551 |
| ... | ... | ... |
| 13443 | 0.024453 | 0.021270 |
| 13444 | 0.306247 | 0.304377 |
| 13445 | 0.564379 | 0.564172 |
| 13446 | 0.374263 | 0.373374 |
| 13447 | 0.481870 | 0.480828 |

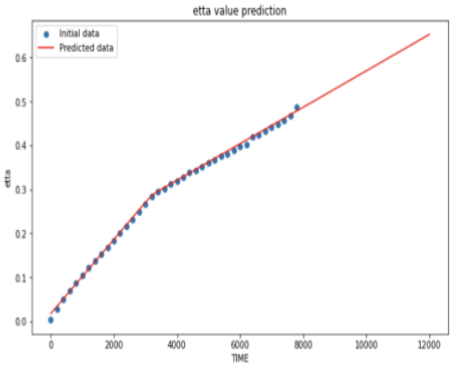



Рисунок 13. Результаты прогнозирования коэффициента нефтеотдачи для нейронных сетей алгоритмов

Кроме этого, была реализована вторая нейронная сеть LSTM (Long Short-Term Memory). Для создания модели была использована архитектура нейронной сети LSTM и функция активации ReLU, которая принимает входные данные в форме временных последовательностей с одним признаком, предназначенными для анализа и извлечения временных зависимостей, а затем слой Dense с одним выходом для решения задачи регрессии.

Для оценки моделей были выбраны метрики коэффициент корреляции R2, средняя квадратическая

ошибка MSE. В таблице 6 представлено сравнение метрик двух нейронных сетей.

Таблица 6. Результат оценки нейронных сетей

| **Метрики** | **NN** | **LSTM** |
|---|---|---|
| MSE | 6.57374404966383e-11 | 2.156884934119221e-07 |
| MAE | 6.597564930893270 6e-06 | 0.00045655597873409565 |
| R^2 | 0.95699474605946 | 0.9999999983987063 |
| Время обучения | 86.344532542 | 108.54556783 |

Модель нейронной сети (NN) продемонстрировала впечатляющую точность в предсказании значений, имея низкую среднеквадратичную ошибку (MSE) и высокий коэффициент детерминации (R^2), что составляет 95.7% объясненной дисперсии в данных. Модель LSTM также показала высокую точность, но с немного выше значением MSE и ниже временем обучения по сравнению с моделью NN. Это может указывать на баланс между точностью и вычислительным временем.

## 2.2. Разработка интеллектуальных вычислительных моделей для модели Black Oil.

В данном разделе были использованы методы машинного обучения и нейронных сетей для задачи Black Oil. Процесс прогнозирования коэффициента нефтеотдачи включал четыре ключевых этапа:

- создание синтетических данных на основе математической модели Black Oil;
- предварительная обработка и анализ данных;
- разработка моделей машинного обучения для прогнозирования нефтеотдачи;

- оценка производительности моделей с целью определения их эффективности.

Обширный синтетический набор данных был сгенерирован с помощью численного решения модели Black Oil, что послужило основой для последующего анализа с применением методов машинного обучения. В процессе генерации использовались основные параметры, описанные в таблице 1, такие как абсолютная проницаемость, вязкость жидкостей (нефти, газа и воды), пористость, давление закачки, начальное давление и другие параметры насыщенности. Для каждого из этих параметров были заданы диапазоны значений, которые моделировали различные сценарии внутри резервуара.

Основные этапы генерации синтетического набора данных включали:

- Выбор параметров. Были определены ключевые параметры, такие как вязкость флюидов (нефти и газа), проницаемость, давление закачки и уровни насыщенности каждого компонента.

- Создание сценариев. Эти параметры были варьированы для создания различных сценариев. Каждый сценарий представлял собой уникальный набор условий, которые моделировали различные состояния резервуара. Это позволило создать объемный набор данных, состоящий из тысяч различных ситуаций.

Данный набор данных включает 369 600 точек, собранных с помощью симуляций математической модели Black Oil. Исходный набор данных, представленный в таблице 4, содержит следующие параметры модели такие как абсолютная проницаемость (k), вязкость нефти (Visc_oil), вязкость газа (Visc_gas), пористость (Porosity), давление (Pressure), средняя насыщенность водой (Sw_avg), средняя насыщенность нефтью (So_avg), средняя насыщенность газа (Sg_avg), коэффициент нефтеотдачи

(Oil_recovery_factor). Эти данные были использованы для обучения модели, которая затем применяется для прогнозирования коэффициента нефтеотдачи. Следует подчеркнуть, что таблица 6 представляет собой лишь выборочный фрагмент данных, предоставленный для иллюстрации общей структуры набора данных.

Таблица 6. Набор данных, полученный с использованием модели Black Oil.

| k | Visc_oil | Visc_gas | Porosity | Oil_recover_factor | Gas_recovery_factor | Sw_avg | So_avg | Sg_avg | Pressure |
|---|---|---|---|---|---|---|---|---|---|
| 0.0001 | 0.3 | 0.01 | 0.2 | 0.305602 | 0.03125 | 0.03125 | 0.2251 | 0.6781 | 0.09687 |
| 0.0001 | 0.3 | 0.01 | 0.2 | 0.30562 | 0.031664 | 0.03081 | 0.2252 | 0.6778 | 0.09691 |
| 0.0001 | 0.3 | 0.01 | 0.2 | 0.305651 | 0.032026 | 0.03048 | 0.2254 | 0.6775 | 0.09695 |
| 0.0001 | 0.3 | 0.01 | 0.2 | 0.305689 | 0.032388 | 0.03014 | 0.2256 | 0.6773 | 0.09698 |

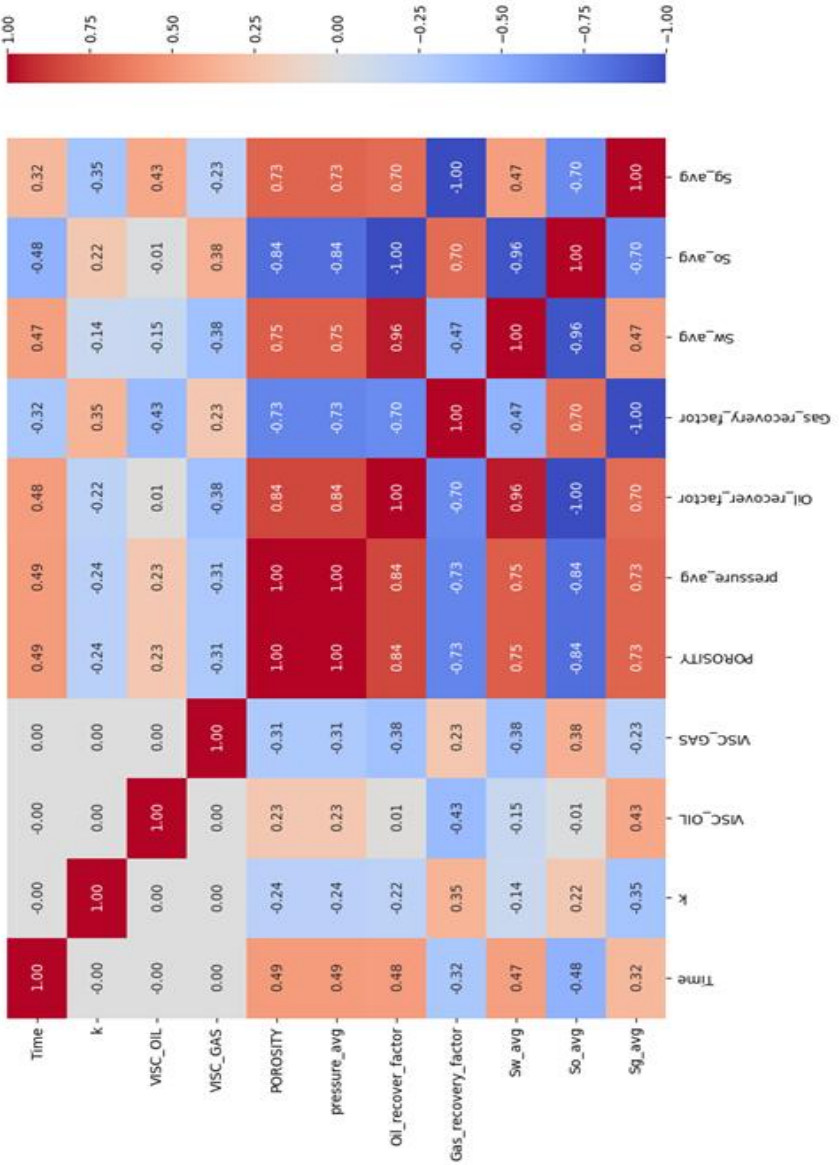
В наборе данных наблюдаются значимые корреляции между переменными, что представлено в корреляционной матрице на рисунке 14. Сильная положительная корреляция выявлена между давлением, коэффициентом нефтеотдачи и пористостью (коэффициенты Пирсона 0.84 и 1.00 соответственно), указывая на то, что более высокие давления связаны с увеличением нефтеотдачи и большей пористостью. В то же время коэффициент газоотдачи имеет отрицательную корреляцию с давлением и пористостью (-0.73), а насыщенность водой и нефтью демонстрируют почти идеальную отрицательную взаимосвязь (-0.96).

Корреляции между вязкостью нефти и другими параметрами слабые, с максимальной отрицательной корреляцией -0.43 с газоотдачей. Сильная корреляция целевой переменной коэффициента нефтеотдачи с другими

характеристиками набора данных создаёт прочную основу для разработки точной модели прогнозирования.

Анализ представленных распределений данных демонстрирует распределения различных параметров, вероятно, связанных с нефтегазовой отраслью. Каждый график отображает частоту встречаемости определенных значений параметров в исследуемом наборе данных. В целом, большинство графиков демонстрируют многомодальные распределения, что говорит о наличии нескольких пиков для каждого параметра. Это может указывать на существование различных групп или классов в данных. Например, параметр "poro" (пористость) имеет несколько пиков, что может отражать различия в типах пород в исследуемой области. Асимметрия в некоторых распределениях, например, параметра "visc" (вязкость), указывает на неравномерное распределение данных, с преобладанием значений в одной части графика и длинным хвостом в другой. Это может свидетельствовать о наличии выбросов или технических ограничений процесса измерения.

Рисунок 14. Матрица корреляции



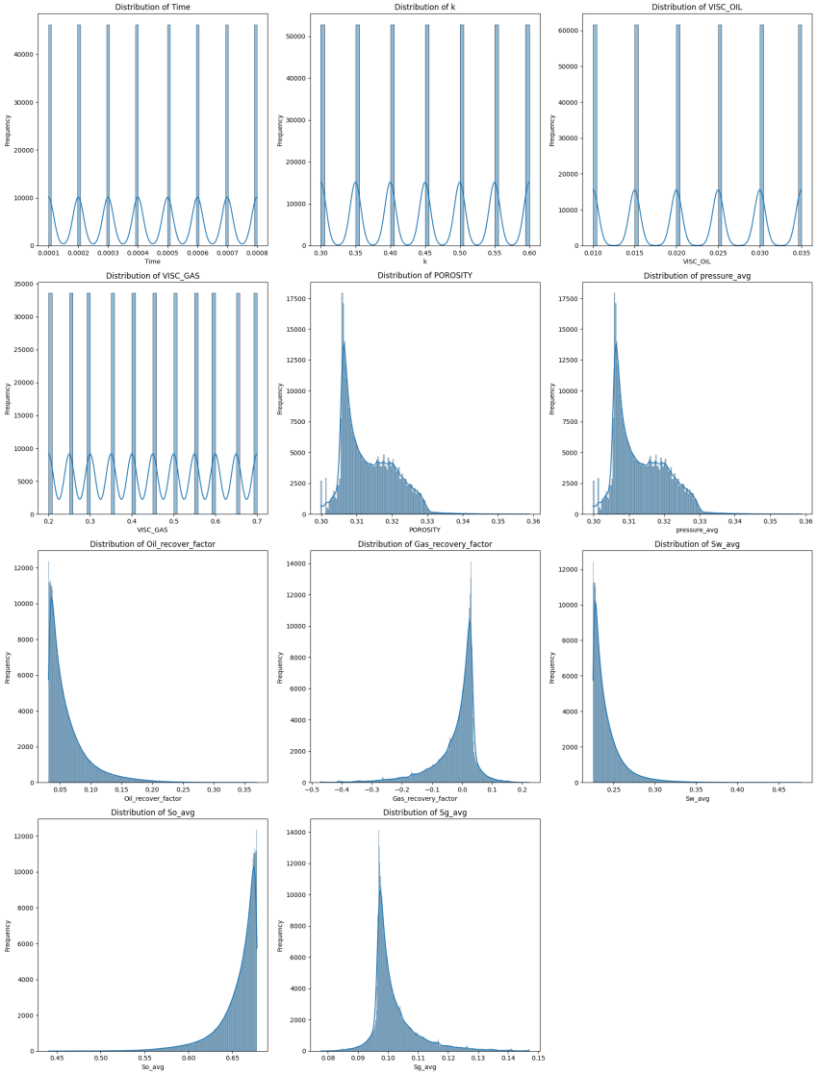



Рисунок 15. Анализ представленных распределений данных

На графиках также наблюдаются шумы и выбросы, что может быть связано с погрешностями измерений, редкими событиями или особыми условиями сбора данных. Эти особенности требуют дальнейшего анализа для более точной интерпретации результатов. Например, многомодальные распределения могут свидетельствовать о геологической неоднородности, наличии различных пород и тектонических структур. Асимметричные распределения могут быть результатом специфики технологических процессов, таких как добыча нефти или применение методов обработки данных.

Анализ диаграмм распределений параметров показывает ключевые статистические характеристики различных параметров, вероятно, связанных с нефтегазовой отраслью.

Каждая диаграмма наглядно демонстрирует медиану (черная линия внутри), которая делит данные пополам, а также границы первого и третьего квартилей, которые определяют основные 50% данных. Усы диаграммы отображают диапазон значений, исключая выбросы, которые представлены отдельными точками за пределами усов. Выбросы наблюдаются на многих диаграммах, что может свидетельствовать об аномалиях или высокой изменчивости некоторых параметров. Размах значений варьируется между параметрами: например, параметр "POROSITY" имеет более узкий размах, чем параметр "Oil recover factor". Также на некоторых диаграммах видна асимметрия в распределении, где медиана смещена относительно центра ящика, как это видно на примере параметра "Oil recover factor", что свидетельствует о том, что большая часть значений сосредоточена в левой части, а хвост распределения удлиняется вправо.

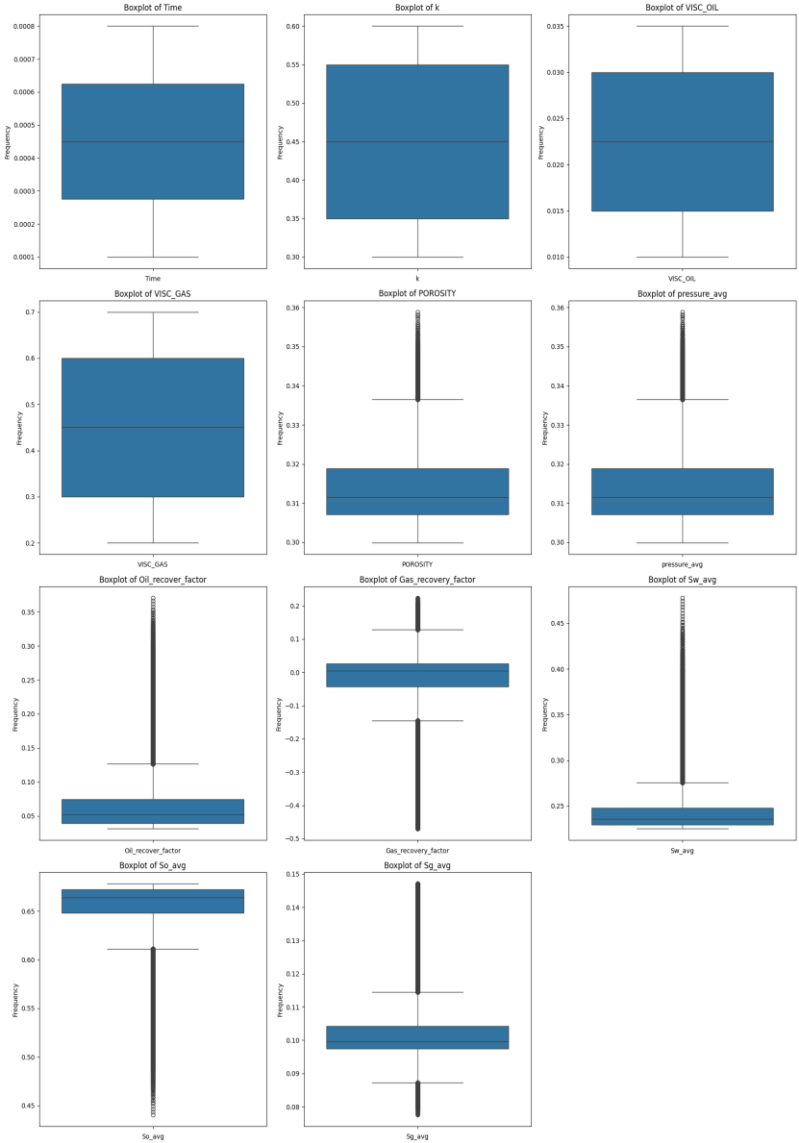



Рисунок 16. Диаграмма распределений параметров

В данном исследовании в симулированный набор данных было добавлено шумовое воздействие в виде шума Лапласа для моделирования стохастической изменчивости, отражающей случайные и неопределенные свойства природных резервуаров. Шум Лапласа, также известный как двойной экспоненциальный шум, выбран из-за его способности описывать данные с "тяжёлыми хвостами", что увеличивает вероятность появления редких, но значимых отклонений, характерных для геологических симуляций. Это позволяет более точно моделировать влияние небольших колебаний параметров резервуара. Использованная формула функции плотности вероятности для распределения Лапласа выглядит следующим образом:

$$f(x|u,b) = \frac{1}{2b} ex\left(-\frac{|x-\mu|}{b}\right), \quad (50)$$

где $x$ - переменная, $\mu$ — параметр положения, а $b$ — параметр масштаба, определяющий ширину распределения. Параметр $b$ был установлен на уровне 10% от стандартного отклонения каждой переменной, что обеспечивает масштабирование шума в соответствии с изменчивостью переменных, сохраняя относительную пропорцию флуктуаций. В рисунке 17 показана диаграмма рассеяния данных после добавления шума.

Для выбора наиболее точной модели прогнозирования на данном наборе данных была проведена сравнительная оценка производительности различных алгоритмов. После анализа существующей литературы по прогнозированию добычи нефти и методам машинного обучения были выбраны следующие алгоритмы, продемонстрировавшие высокую эффективность в решении задач регрессии: деревья решений, случайные леса, градиентный бустинг и искусственные нейронные сети. Помимо этих методов,

были также обучены алгоритмы экстремального градиентного бустинга (XGBoost) и стекинг-регрессор.

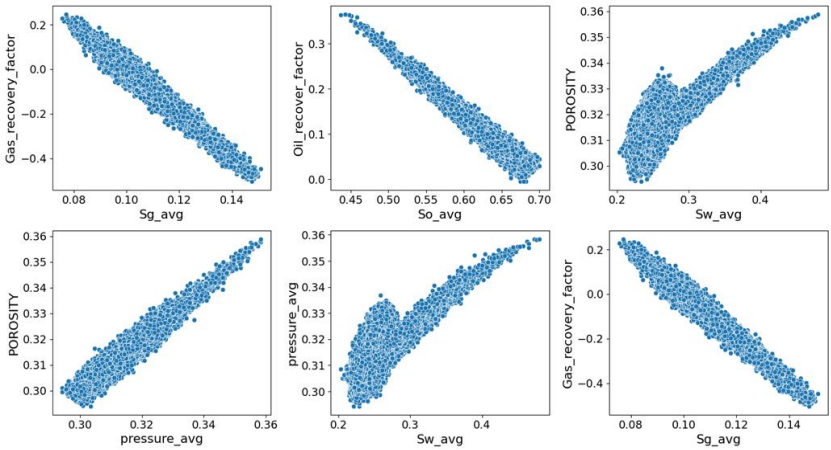



Рисунок 17. Набор данных после добавления шума

Дерево решений - это алгоритм машинного обучения под наблюдением, используемый для предсказания значения целевой переменной путем обучения на основе правил принятия решений, полученных из признаков в данных. Оно состоит из узлов, которые описывают вопросы о данных, известные как тесты признаков; ветвей, представляющих ответы или результаты на эти вопросы, которые ведут к другим узлам; и листовых узлов, которые представляют классификации или окончательные решения. Основная цель дерева решений - разделить набор данных на более мелкие и однородные группы, которые потенциально могут быть более удобными для обработки и интерпретации.

**Основные элементы дерева решений:**

**Корневой узел:** Это корень любого дерева. В нем находится весь набор данных, а значит, и начальная точка принятия решения. Это узел, представляющий собой

наилучший признак, по которому можно разделить данные на основе какого-либо критерия, например, прироста информации или примеси Джини.

**Внутренние узлы:** Это узлы, которые проверяют различные атрибуты входных данных. Каждый из этих тестов задает вопрос о данных, например «Превышает ли значение признака X пороговое значение?». В зависимости от ответа на этот вопрос он разбивает данные на подмножества, которые идут по ветвям.

**Ветви:** Это возможные результаты тестов, проведенных на каждом внутреннем узле. В зависимости от результата данные разделяются и поступают в следующие узлы.

**Окончательные прогнозы или классификация:** Как только данные достигают узла листа, больше не происходит расщепления данных, и, следовательно, принимается решение. В задачах классификации каждый лист представляет собой метку класса, а в задачах регрессии - предсказанное значение.

**Как работают деревья решений. Дерево решений строится с помощью рекурсии на нескольких важных этапах:**

**Выбор лучшего атрибута:** Алгоритм выбирает наиболее полезный атрибут для разбиения данных. Он основывается на критерии разбиения; это может быть примесь Джини, энтропия или прирост информации. По сути, выбирается тот атрибут, который лучше всего разбивает данные на подмножества, где вариации целевой переменной наименьшие.

**Разделение набора данных:** На основе значения выбранного атрибута набор данных делится на подмножества. В каждом подмножестве есть набор данных, которые имеют общие характеристики на основе выбранного атрибута. Этот шаг деления будет продолжен для каждого подмножества рекурсивно.

**Критерий остановки:** Процесс создания узлов и разбиения данных продолжается до тех пор, пока не будет достигнут критерий остановки. Общими критериями остановки являются максимальная глубина дерева, минимальное количество образцов в узле или когда все экземпляры в узле принадлежат к одному классу.

**Рекурсия:** Тот же процесс повторяется для каждого вновь сформированного подмножества рекурсивным образом. Каждая рекурсия создает внутренний узел, пока не приведет к формированию листовых узлов. Таким образом, создается дерево решений, на основе которого делается прогноз.

**Метрики разделения.**

**Gini impurity** - это мера вероятности того, что случайно выбранный экземпляр из данного набора данных будет неправильно помечен, если метка класса определяется случайным образом в зависимости от относительных частот классов в данном наборе данных. Gini Imputiry, равная 0, соответствует полной чистоте, когда все экземпляры принадлежат одному классу.

$$Gini \ = \ 1 \ - \sum_{i=1}^{n} (pi)^2 \qquad (51)$$

где $p_i$ - вероятность того, что экземпляр будет отнесен к определенному классу.

**Entropy:** Подобно Gini Impurity, Entropy измеряет неопределенность или примесь в наборе данных. Высокая Entropy означает высокий уровень беспорядка; при низкой Entropy набор данных считается более упорядоченным, в том смысле, что экземпляры преимущественно принадлежат к одному классу.

$$Entropy \ = \ - \sum_{i=1}^{n} p_i log_2(p_i)$$

(52)

где $p_i$ - вероятность того, что экземпляр будет отнесен к определенному классу.

**Information Gain:** Измеряет уменьшение Entropy или Gini Impurity после разделения набора данных на основе атрибута. Для разбиения выбирается атрибут с наибольшим Information Gain.

$$Information\ Gain\ =\ Entropy_{parent}\ -\ \sum_{i=1}^{n}(\frac{|D_i|}{|D|}*\ Entropy(D_i)) \quad (53)$$

где $D_i$ - подмножество $D$ после разбиения по атрибуту.

**Преимущества деревьев принятия решений.**

Простота и интерпретируемость: Деревья решений просты для понимания и интерпретации; эти системы, по сути, отражают человеческие процессы рассуждения, визуально отображая различные решения и их последствия. Таким образом, они применимы в тех областях, где необходимо объяснять причины, лежащие в основе решения.

**Универсальность:** Деревья решений одинаково хорошо работают как с классификационными, так и с регрессионными задачами. Они гибкие и могут работать как с категориальными, так и с числовыми данными.

**Отсутствие необходимости в масштабировании признаков:** В отличие от многих моделей машинного обучения, деревья решений не нуждаются в нормализации или масштабировании данных, поскольку модель основана на пороговом разбиении, а не на расстоянии между точками данных.

**Улавливает нелинейные отношения:** Дерево решений отражает сложные нелинейные отношения между переменными.

**Устойчивость к нерелевантным признакам:** Деревья решений в первую очередь выбирают наиболее важные

признаки для выполнения разбиения и могут полностью отбрасывать признаки, которые кажутся нерелевантными, без особого влияния на их производительность.

**Недостатки деревьев принятия решений.**

**Переоптимизация:** У деревьев принятия решений есть один большой недостаток - их склонность к перестройке на обучающих данных по мере увеличения глубины дерева. Переподгонка - это состояние, при котором модель изучает шум или небольшие колебания в данных, которые не очень хорошо обобщаются на новые данные.

**Нестабильность:** Деревья решений чувствительны даже к незначительным изменениям входных данных. Даже незначительные изменения в наборе обучающих данных могут полностью изменить структуру созданного дерева. Таким образом, решения могут повлиять на стабильность модели.

**Предвзятое отношение к признакам с большим количеством уровней:** Деревья решений могут быть предвзяты к признакам, имеющим большое количество уровней, поскольку такие признаки доминируют в структуре дерева решений.

**Обрезка: Уменьшение перебора**

Деревья решений могут быть обрезаны, чтобы противостоять чрезмерной подгонке. Обрезка включает в себя процесс уменьшения размера дерева путем удаления узлов, которые обеспечивают ограниченную предсказательную способность, что позволяет получить более простую и обобщенную модель.

**Предварительная обрезка или ранняя остановка:** Это процесс остановки роста дерева до того, как оно начнет перестраиваться. Обычно он выполняется путем задания условий для дерева, таких как максимальная глубина, минимальное количество образцов, необходимых для

разделения узла, или минимальное количество образцов, необходимых для узла листа. Если условие выполнено, дальнейшее разбиение прекращается.

**После обрезки:** Полное дерево строится до максимальной глубины с последующей обрезкой ветвей, которые имеют низкую точность классификации. Пост-обрезка обычно дает гораздо лучшее дерево по сравнению с пред-обрезкой.

Случайные леса — это метод ансамблевого обучения, который строит несколько деревьев решений и объединяет их для получения более точного и стабильного прогноза. Каждое дерево в лесу обучается на случайной подвыборке данных, а итоговый прогноз получается путем усреднения прогнозов всех деревьев в ансамбле:

$$\hat{y} = \frac{1}{T} \sum_{t=1}^{T} \hat{y}^t \,, \tag{54}$$

где $T$ — количество деревьев, а $\hat{y}^t$ — прогноз от $t$-го дерева.

Градиентный бустинг — это ансамблевый метод, который строит модели последовательно, каждая новая модель пытается исправить ошибки предыдущих. Модели добавляются до тех пор, пока не перестанут происходить значимые улучшения. Он минимизирует дифференцируемую функцию потерь $L(y, \hat{y})$ с использованием градиентного спуска.

XGBoost — это эффективная и гибкая реализация алгоритма градиентного бустинга, который является одним из самых мощных алгоритмов машинного обучения для задач классификации, регрессии и ранжирования. Его успех можно объяснить превосходной производительностью и высокой скоростью, что сделало его фаворитом в большинстве конкурсов по машинному обучению, таких как те, что проводятся на платформе Kaggle. XGBoost

особенно хорош в создании предсказательных моделей с ансамблем деревьев решений. Эти деревья создаются итеративным образом с целью улучшения производительности. Система разработана с особым акцентом на вычислительную эффективность, масштабируемость для работы с большими наборами данных и борьбу с переобучением благодаря механизмам, которые улучшают обобщение.

Преимущества XGBoost достигаются благодаря ряду оптимизаций и улучшений, включая добавление регуляризации, обрезку деревьев, параллельную обработку и работу с пропущенными значениями. Эти возможности, наряду с поддержкой разреженных данных и гибкостью в настройках, делают XGBoost одним из самых универсальных инструментов для предсказательных задач в таких областях, как финансы, здравоохранение, электронная коммерция и многих других, где применяется предсказательная аналитика.

**Преимущества использования XGBoost.**

**Высокая производительность:** XGBoost показывает высокую точность и часто превосходит другие алгоритмы в конкурентных условиях. Он выиграл множество конкурсов по машинному обучению благодаря своей эффективности в создании более точных моделей.

**Гибкость:** XGBoost подходит для решения широкого круга задач, от классической классификации и регрессии до задач ранжирования. Он также поддерживает пользовательские функции потерь, что позволяет адаптировать его к различным типам предсказательных задач.

**Эффективность:** XGBoost очень эффективен как по использованию памяти, так и по скорости вычислений. Благодаря таким оптимизациям, как параллельная обработка и обрезка деревьев, обучение в XGBoost

проходит быстрее, чем в других алгоритмах, без потери точности.

**Обработка пропущенных данных:** Автоматическая обработка пропущенных данных снижает необходимость ручной иммутации и, таким образом, упрощает рабочий процесс машинного обучения.

**Масштабируемость:** XGBoost очень масштабируем и поэтому подходит для приложений, связанных с большими данными, таких как финансы, здравоохранение и электронная коммерция.

**Регуляризация:** Встроенные техники регуляризации позволяют XGBoost контролировать переобучение, что является частой проблемой моделей, основанных на деревьях решений.

**Недостатки XGBoost.**

**Сложность настройки гиперпараметров:** У XGBoost много гиперпараметров, что дает большую гибкость и производительность, но этот процесс настройки гиперпараметров может быть довольно трудоемким и сложным. Тем не менее, можно использовать такие техники, как поиск по сетке (grid search) или случайный поиск (random search) для выбора наилучших настроек.

**Потребление памяти:** Несмотря на свою эффективность, XGBoost может быть весьма требователен к памяти, особенно при работе с очень большими наборами данных. Сохранение большого числа деревьев решений в памяти может создать проблемы для систем с ограниченными ресурсами.

**Интерпретируемость:** Хотя XGBoost более точен, чем большинство других моделей, он не так легко интерпретируем по сравнению с простыми моделями, такими как линейная регрессия или одно дерево решений. С помощью оценки важности признаков можно получить

некоторую информацию, но детальная интерпретация ансамбля деревьев затруднена.

**Время обучения на больших наборах данных:** Хотя XGBoost значительно быстрее традиционных алгоритмов бустинга, его обучение на очень больших наборах данных все равно может занимать много времени, особенно при применении глубоких деревьев.

Для этих моделей настройка гиперпараметров играла ключевую роль. Использовались методы, такие как GridSearchCV с перекрестной проверкой, для определения оптимальных гиперпараметров, включая максимальную глубину дерева, минимальное количество образцов для разделения узла и минимальное количество образцов в листовом узле.

**Стекинг моделей машинного обучения.**

Стекинг регрессоров — это мощная техника, которая улучшает результаты предсказаний, объединяя множество регрессионных моделей. Он работает, беря предсказания от нескольких базовых моделей и подавая их в качестве входных данных для модели более высокого уровня, называемой мета-обучателем. Мета-модель обучается с целью научиться комбинировать эти предсказания, чтобы достичь более высокой точности, чем это возможно с любой одной моделью по отдельности.

В дальнейшем была использована нейронная сеть, которая по своей природе отличается от вышеперечисленных алгоритмов.

Нейронные сети извлекают из данных идентифицирующие признаки, не требующие запрограммированного понимания. Компоненты сети включают нейроны, связи, веса, смещения, функции распространения и правила обучения. Нейроны получают входные сигналы, управляемые порогами и функциями активации. Соединения включают веса и смещения,

регулирующие передачу информации. Обучение, корректировка весов и смещений, происходит в три этапа: вычисление входных данных, генерация выходных данных и итеративное совершенствование, повышающее квалификацию сети в решении различных задач.

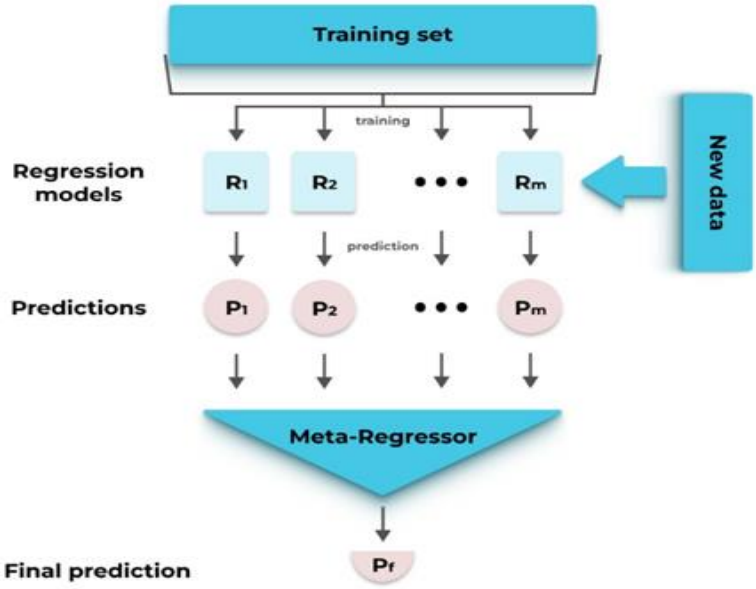



Рисунок 18. Архитектура стекового регрессора

**Важность нейронных сетей.**

Способность нейронных сетей выявлять закономерности, решать сложные головоломки и приспосабливаться к меняющейся обстановке имеет огромное значение. Их способность обучаться на основе данных имеет далеко идущие последствия: от революции в таких технологиях, как обработка естественного языка и самоуправляемые автомобили, до автоматизации процессов принятия решений и повышения эффективности во многих

отраслях. Развитие искусственного интеллекта во многом зависит от нейронных сетей, которые также являются движущей силой инноваций и влияют на направление развития технологий.

**Работа нейронной сети.**

Нейронные сети - это сложные системы, которые имитируют некоторые особенности работы человеческого мозга. Она состоит из входного слоя, одного или нескольких скрытых слоев и выходного слоя, состоящего из слоев искусственных нейронов, которые соединены между собой. Два этапа основного процесса называются обратным и прямым распространением.

**Прямое распространение.**

**Входной слой:** Каждый признак во входном слое представлен узлом сети, который получает входные данные.

**Веса и связи:** Вес каждого соединения нейронов показывает, насколько сильной является связь. В процессе обучения эти веса меняются.

**Скрытые слои:** Каждый нейрон скрытого слоя обрабатывает входные данные, умножая их на веса, складывая их, а затем пропуская через функцию активации. Таким образом вводится нелинейность, позволяющая сети распознавать сложные закономерности.

**Выход:** Конечный результат получается путем повторения процесса до тех пор, пока не будет достигнут выходной слой.

**Обратное распространение.**

**Вычисление потерь:** Выход сети оценивается в сравнении с реальными значениями цели, и для вычисления разницы используется функция потерь. Для задачи регрессии в качестве функции затрат обычно используется средняя квадратичная ошибка (MSE).

**Функция потерь:**

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2 \quad (55)$$

**Градиентный спуск:** Градиентный спуск используется сетью для уменьшения потерь. Чтобы уменьшить погрешность, веса изменяются на основе производной потерь по отношению к каждому весу.

**Регулировка весов:** Веса корректируются в каждом соединении путем применения этого итерационного процесса, или обратного распространения, в обратном направлении по всей сети.

**Обучение:** Во время обучения на различных образцах данных весь процесс прямого распространения, вычисления потерь и обратного распространения выполняется итеративно, что позволяет сети адаптироваться и изучать закономерности на основе данных.

**Функции активации:** Нелинейность модели вносится функциями активации, такими как выпрямленная линейная единица (ReLU) или сигмоид. Их решение о том, следует ли «запустить» нейрон, основывается на взвешенном входном сигнале в целом.

Наша модель была построена с пятью слоями, включая четыре полностью связанных (плотных) слоя и один выходной слой. Была использована функция активации ReLU. Нейронная сеть обучается на входных данных, которые включают различные параметры, связанные с нефтяными скважинами, такие как глубина скважины, ее геологические характеристики, характеристики нефтяного пласта и т. д. После обучения нейронная сеть может предсказать коэффициент извлечения нефти на основе новых входных данных.

Благодаря процессу обучения, который минимизирует среднеквадратичную ошибку между прогнозируемыми и фактическими значениями, сеть учится точно

прогнозировать извлечение нефти. В этом обучении используется оптимизатор Adam — эффективный алгоритм настройки внутренних параметров сети для наилучшего соответствия данным.

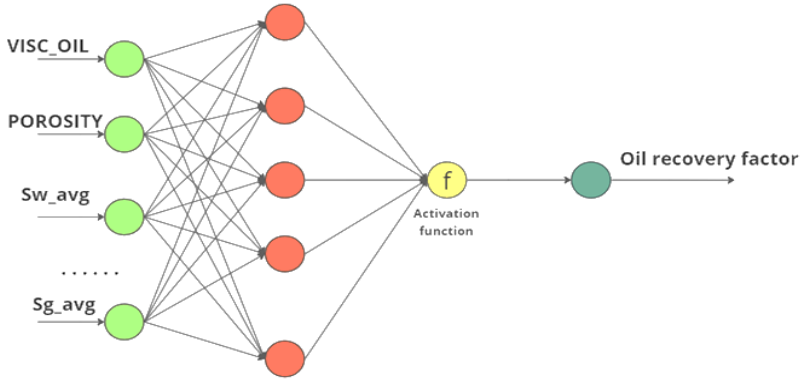



Рисунок 19. Архитектура нейронной сети для модели Black Oil

Текущая реализация нашей нейронной сети является статической, фокусируясь на прогнозировании коэффициентов нефтеотдачи на основе моментальных данных характеристик резервуара. Динамическая природа нефтяных пластов и предполагается, что будущие исследования могут изучить динамические модели нейронных сетей, которые могут включать временные изменения и данные о процессе добычи, чтобы лучше отражать изменяющиеся условия нефтяных пластов..

В этом исследовании набор данных разделен на обучающий и тестовый в соотношении 80% к 20%. Для нейронной сети проверочная выборка составила 20% обучающего набора. Модели оценивались с использованием ключевых метрик: MSE, MAE, $R^2$. Масштабирование обучающего и тестового наборов представлено в Таблице 7.

Таблица 7. Размер обучающего и тестового набора

| **Data Type** | **Data Points** |
|---|---|
| Training set | 295 680 |
| Test set | 73 920 |
| Total | 369 600 |

На рисунке 6 показана диаграмма рассеяния, сравнивающая данные обучения и данные тестирования для обученных моделей. Каждый график отображает ряд точек данных, где фактические значения зависимой переменной сравниваются с прогнозируемыми значениями, полученными из модели линейной регрессии.

Чтобы найти оптимальные параметры, был применен поиск по сетке для оптимизации гиперпараметров моделей на основе дерева и искусственных нейронных сетей (ИНС). Оптимальные комбинации гиперпараметров были определены на основе точности тестового набора, используемого в качестве метрики производительности. На рисунке 20 показана архитектура нейронной сети.

Диапазон гиперпараметров для этих моделей был определен на основе размера набора данных и их характеристик. В пределах этих заданных диапазонов различные комбинации параметров систематически исследовались для поиска оптимального набора. Таблица 6 показывает оценку $R^2$ с использованием оптимизированных параметров.

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 dense (Dense)               (None, 256)               2816

 dense_1 (Dense)             (None, 128)               32896

 dense_2 (Dense)             (None, 64)                8256

 dense_3 (Dense)             (None, 32)                2080

 dense_4 (Dense)             (None, 1)                 33

=================================================================
Total params: 46,081
Trainable params: 46,081
Non-trainable params: 0
```

Рисунок 20. Архитектура нейронной сети

Таблица 8. Оценка $R^2$ различных алгоритмов на обучающих и тестовых наборах после оптимизации

| | DT | RF | GB | XGBoost | NN | Stacking Regressor |
|---|---|---|---|---|---|---|
| Training set | 0.98 | 0.99 | 0.99 | 0.98 | 0.97 | 0.97 |
| Test set | 0.97 | 0.98 | 0.98 | 0.98 | 0.97 | 0.97 |
| Average | 0.975 | 0.985 | 0.985 | 0.98 | 0.97 | 0.97 |

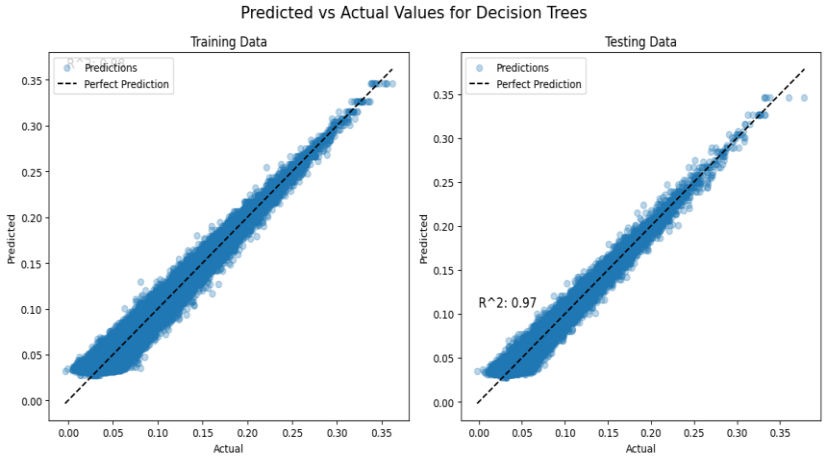



Рисунок 21. Сравнение прогнозируемых и фактических значений на обучающем наборе.

График сравнения предсказанных и фактических значений, полученных с помощью модели дерева решений, демонстрирует высокую точность модели. Точки данных плотно группируются вокруг линии идеального предсказания как на тренировочном, так и на тестовом наборах, что указывает на сильную корреляцию между предсказанными и фактическими значениями. Высокое значение коэффициента детерминации $R^2$ подтверждает эту высокую точность, свидетельствуя о том, что модель успешно объясняет значительную часть вариации целевой переменной. Отсутствие систематических отклонений точек от линии регрессии говорит о том, что модель не склонна к систематическим ошибкам в своих предсказаниях. Такое поведение модели можно объяснить способностью деревьев решений эффективно улавливать нелинейные зависимости между признаками и целевой переменной. Кроме того, деревья решений обладают высокой интерпретируемостью, что позволяет анализировать, какие именно признаки вносят наибольший

вклад в предсказание. Несмотря на впечатляющие результаты, следует помнить о потенциальной проблеме переобучения. Для ее оценки рекомендуется построить графики обучения и валидации. В целом, представленный график свидетельствует о том, что модель деревьев решений является эффективным инструментом для решения данной задачи.

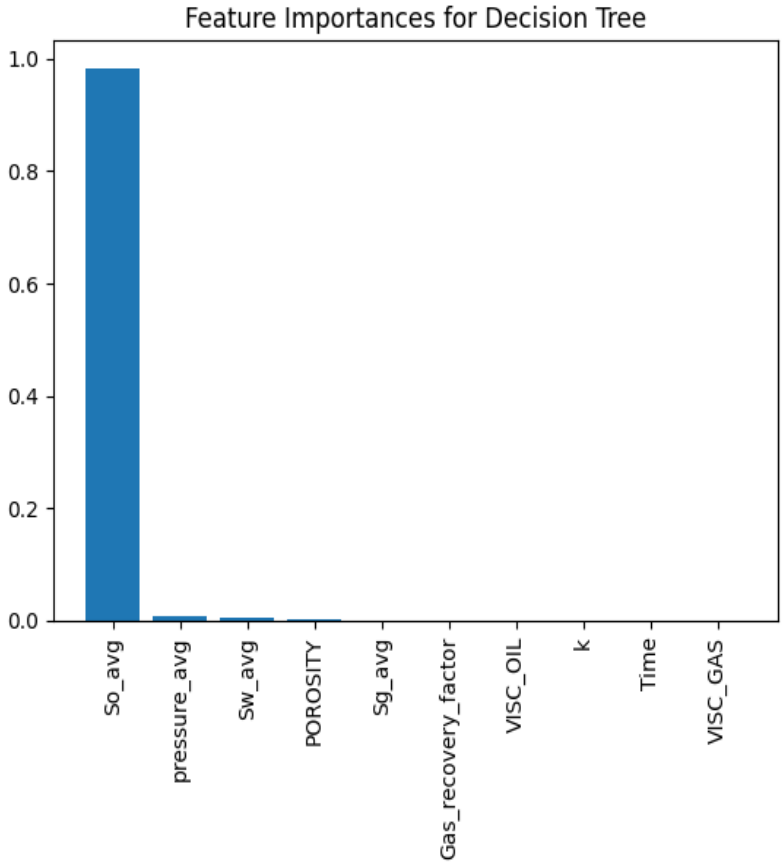



Рисунок 22. График важности признаков для дерева решени

График важности признаков для дерева решений наглядно демонстрирует доминирующее влияние признака So_avg на предсказания модели. Этот признак значительно превосходит по важности остальные, что свидетельствует о его ключевой роли в процессе принятия решений деревом. Такая значимость может быть обусловлена сильной корреляцией с целевой переменной, нелинейной зависимостью или взаимодействием с другими признаками. Менее важные признаки, в свою очередь, оказывают значительно меньшее влияние на результат. Полученные результаты указывают на возможность упрощения модели путем исключения наименее значимых признаков, что может привести к повышению ее интерпретируемости и снижению риска переобучения.

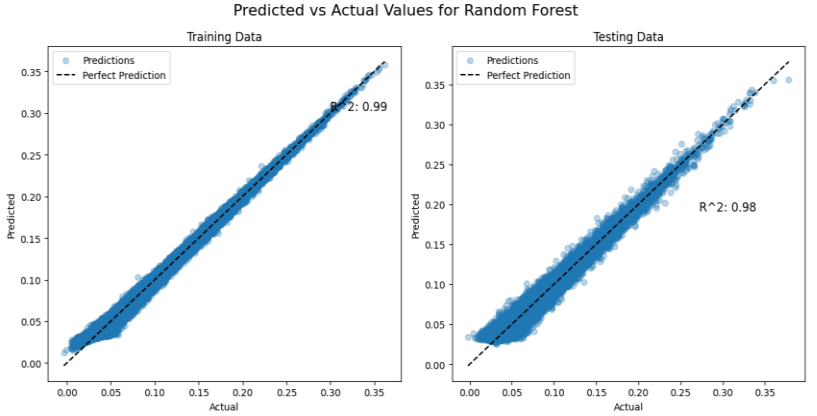



Рисунок 23. Графики сравнения предсказанных и фактических значений для RF

Графики сравнения предсказанных и фактических значений для случайного леса демонстрируют высокую точность модели как на тренировочных, так и на тестовых данных. Точки данных плотно группируются вокруг линии идеального предсказания, что говорит о сильной

корреляции между предсказанными и фактическими значениями. Высокое значение коэффициента детерминации $R^2$ подтверждает эту высокую точность, свидетельствуя о том, что модель успешно объясняет значительную часть вариации целевой переменной. Отсутствие систематических отклонений точек от линии регрессии говорит о том, что модель не склонна к систематическим ошибкам в своих предсказаниях. Такое поведение модели можно объяснить способностью случайного леса эффективно учитывать нелинейные зависимости между признаками и целевой переменной за счет ансамбля из множества деревьев решений.

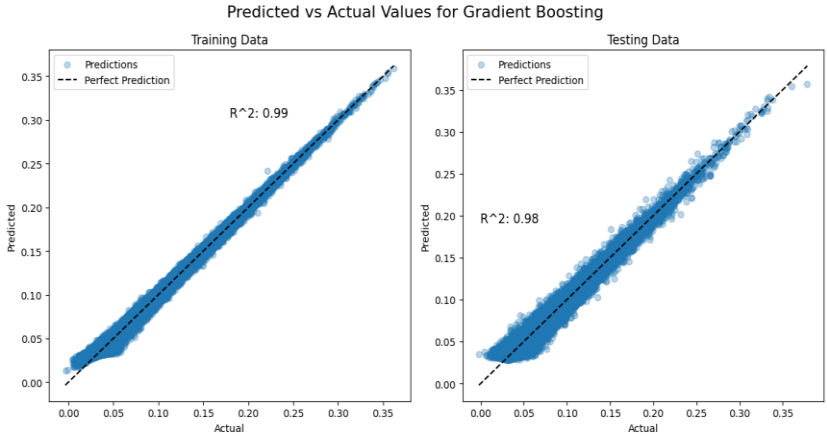



Рисунок 24. Графики сравнения предсказанных и фактических значений для GB

Анализ графиков сравнения предсказанных и фактических значений модели градиентного бустинга свидетельствует о ее высокой эффективности. Модель демонстрирует отличную согласованность между прогнозами и реальными значениями как на обучающей, так и на тестовой выборках. Высокое значение

коэффициента детерминации $R^2$ указывает на то, что модель объясняет значительную часть вариации целевой переменной. Отсутствие систематических ошибок в прогнозах подтверждает надежность модели. Такой результат достигается благодаря способности градиентного бустинга последовательно строить ансамбль слабых моделей, что позволяет эффективно моделировать сложные зависимости в данных.

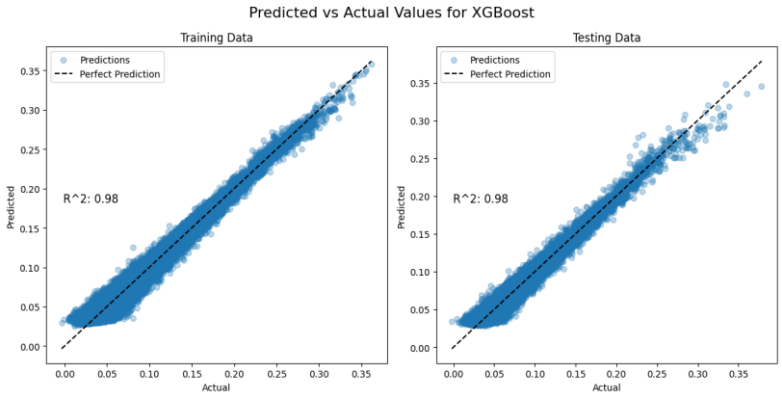



Рисунок 25. Графики сравнения предсказанных и фактических значений для XGBoost

XGBoost продемонстрировал превосходные результаты на данной задаче. Модель показала высокую точность предсказаний и способность адаптироваться к сложным взаимосвязям в данных. Это достигается благодаря ее уникальной архитектуре, которая позволяет последовательно улучшать модель путем добавления новых слабых моделей. Высокое значение коэффициента детерминации $R^2$ подтверждает, что модель является отличным инструментом для объяснения и прогнозирования целевой переменной.

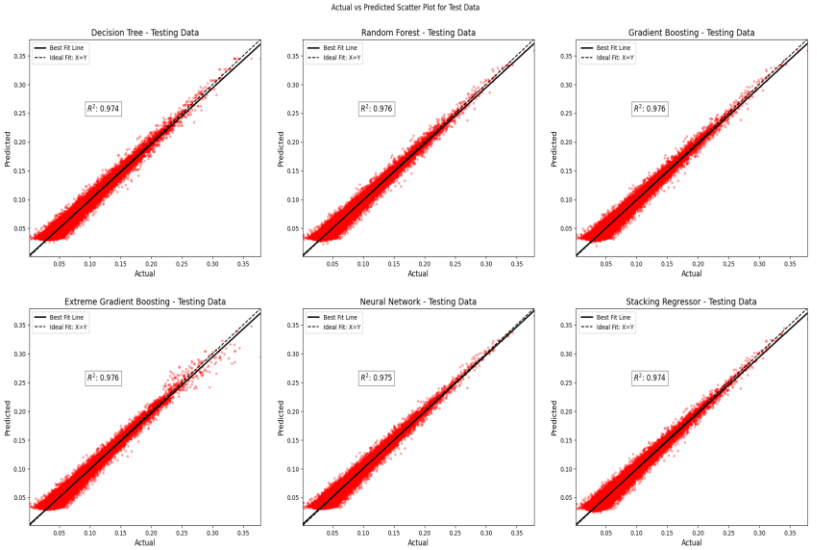



Рисунок 26. Сравнение прогнозируемых и фактических значений на тестовом наборе.

Модели показывают сопоставимые коэффициенты детерминации $R^2$, что указывает на их эффективность в выявлении базовых закономерностей в данных, даже при воздействии новых, невиданных данных. Примечательно, что модель PR достигла высокого значения $R^2$ 0,977, эффективно фиксируя нелинейные связи между переменными. Остальные модели показали результаты около 0,974, которые также очень высоки. Эти результаты подчеркивают способность как традиционных алгоритмов, так и передовых методов машинного обучения, включая нейронные сети, достигать высокой точности прогнозирования и поддерживать согласованность между различными фрагментами данных.

Второе наблюдение заключается в том, что Stacking Regressor показывает сопоставимую производительность с другими моделями на тестовом наборе данных, но немного

отстает на обучающем наборе. Это может указывать на то, что Stacking Regressor более устойчив к переобучению и способен лучше обобщать новые данные. Сравнение оценок $R^2$ до и после оптимизации модели, показанное на рисунке 8, важно для оценки эффективности улучшений, внесенных в алгоритмы.

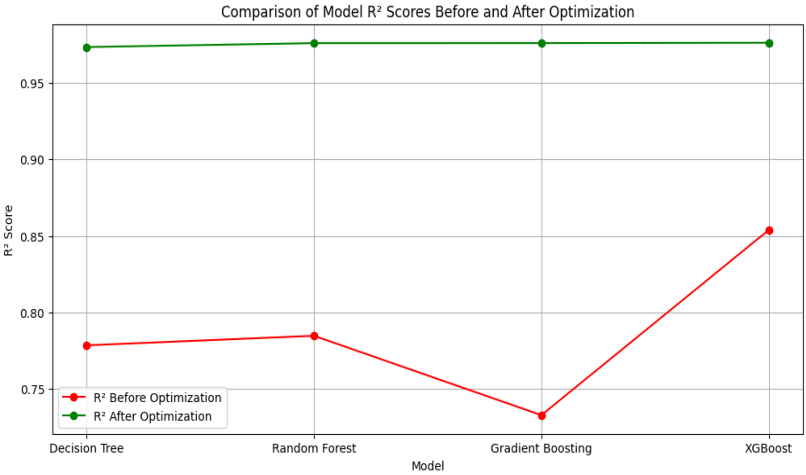



Рисунок 27. Оценка $R^2$ различных алгоритмов на обучающем и тестовом наборах.

Видно, что деревья решений, случайный лес, градиентный бустинг и экстремальный градиентный бустинг показали значительное улучшение после оптимизации, увеличив свой показатель $R^2$ более чем на 10%. Это означает, что после оптимизации эти модели лучше приспосабливаются к изменениям в данных и могут делать более точные прогнозы, чем исходные значения. Эти результаты подтверждают, что оптимизация моделей может значительно улучшить их производительность и сделать их более надежными для использования в предиктивных моделях. Сравнение результатов среднеквадратической

ошибки (MSE) до и после оптимизации моделей показано на рисунке 9.

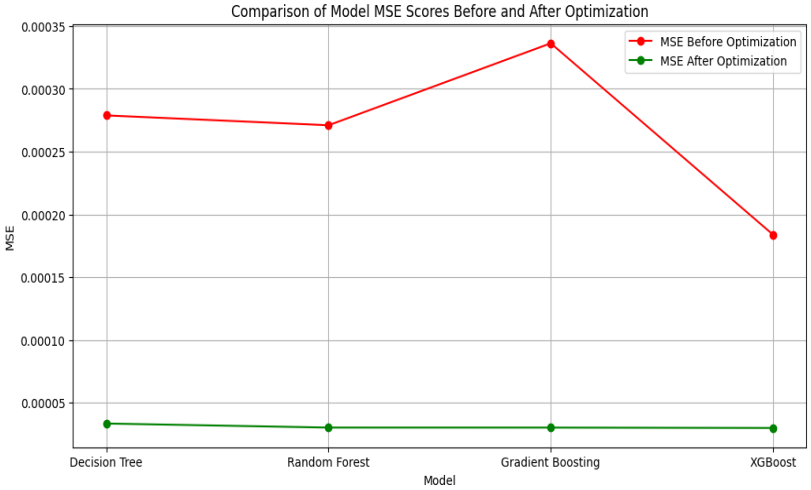



Рисунок 28. Среднеквадратическая ошибка (MSE) различных алгоритмов на обучающем и тестовом наборах.

До оптимизации MSE для дерева решений, случайного леса, градиентного бустинга и экстремального градиентного бустинга были значительно выше. После оптимизации MSE значительно снизился значительно снизился.

Представленный график сравнивает различные модели машинного обучения по метрике R-квадрат. Как видно из графика, все представленные модели демонстрируют высокую точность, с показателями R-квадрат, близкими к 1. Это означает, что модели очень хорошо объясняют вариацию целевой переменной и могут делать точные прогнозы. Лидирующие позиции занимают полиномиальная регрессия, XGBoost, градиентный бустинг и случайный лес, все с показателем R-квадрат 0.98. Это говорит о том, что эти модели практически идеально

подходят для данной задачи. Чуть ниже по рейтингу оказались дерево решений и линейная регрессия с показателем 0.97. Хотя эти модели также показали хорошие результаты, они все же немного уступают лидерам. Важно отметить, что высокая точность на тренировочных данных еще не гарантирует хорошей обобщающей способности модели на новых данных. Для более полной оценки моделей необходимо провести дополнительную проверку на тестовых данных или применить кросс-валидацию. В целом, график показывает, что для данной задачи существует несколько моделей, которые могут дать очень хорошие результаты.

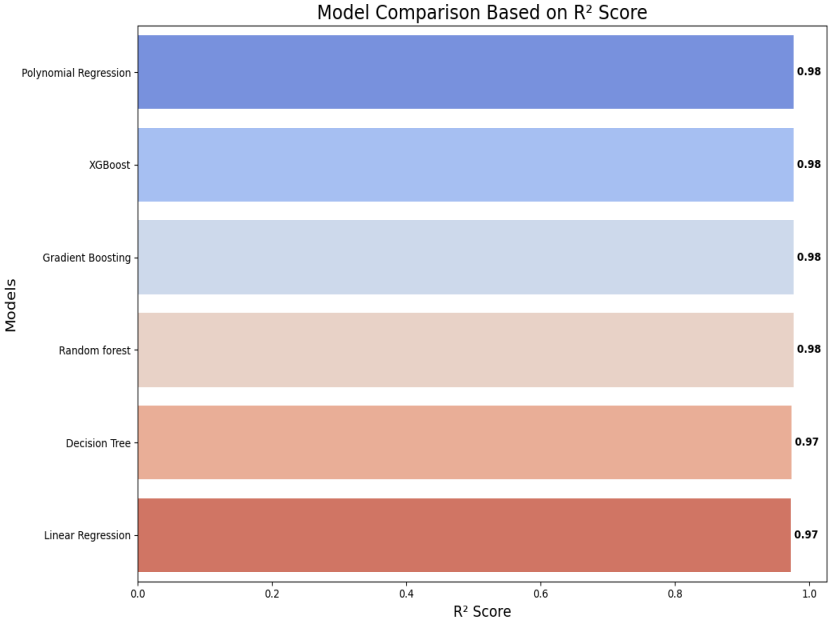



Рисунок 29. график сравнивает различные модели машинного обучения по метрике R-квадрат

Анализ графика остатков для RF играет важную роль в оценке качества модели.

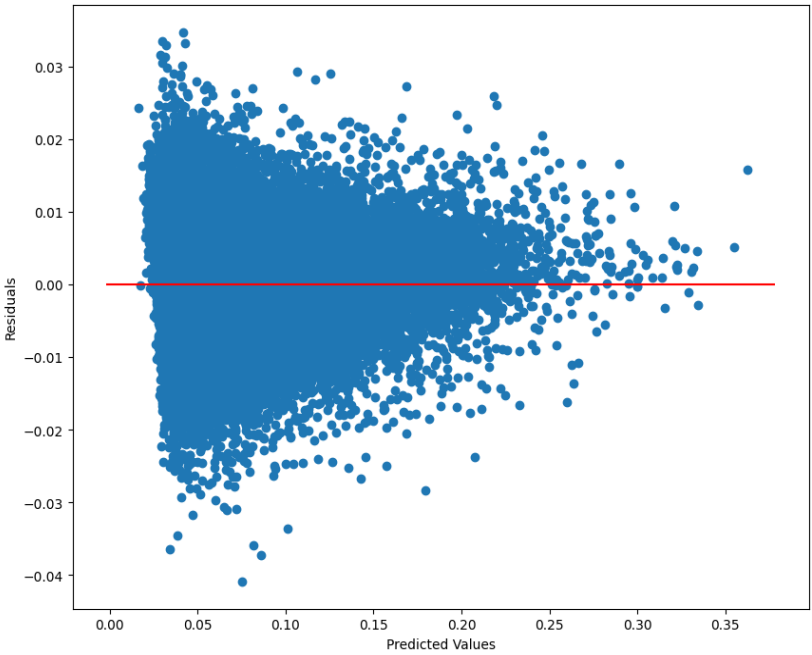



Рисунок 30. График остатков для RF

График остатков (residual plot) представляет собой диаграмму рассеяния, где по оси X откладываются предсказанные моделью значения, а по оси Y – остатки, то есть разности между фактическими и предсказанными значениями. Такой график используется для оценки того, насколько хорошо линейная модель подходит к данным. В идеальном случае точки на графике должны быть случайно распределены вокруг горизонтальной линии, проходящей через ноль, что говорит об отсутствии систематических отклонений. Это указывает на то, что модель адекватно описывает зависимость между переменными и остатки распределены равномерно. Важно также, чтобы на графике

не было видимых паттернов, таких как кластеры, кривые или сужающиеся/расширяющиеся воронки, которые могли бы указывать на проблемы с гетероскедастичностью или пропущенные нелинейные зависимости. Таким образом, анализ графика остатков позволяет выявить потенциальные проблемы модели и принять меры для их устранения.

График потерь модели демонстрирует характерный процесс обучения нейронной сети. На начальных этапах обучения наблюдается быстрое снижение ошибки как на тренировочных, так и на валидационных данных, что свидетельствует об эффективном обучении модели. Затем кривые потерь стабилизируются, указывая на то, что модель достигла насыщения. Незначительное различие между ошибками на тренировочном и валидационном наборах говорит о том, что модель не переобучена и способна хорошо обобщать на новые данные. Однако, небольшое переобучение или возможность дальнейшего улучшения модели не исключены. Для более глубокого анализа рекомендуется рассмотреть другие метрики, визуализировать предсказания и экспериментировать с различными гиперпараметрами и архитектурами сети.

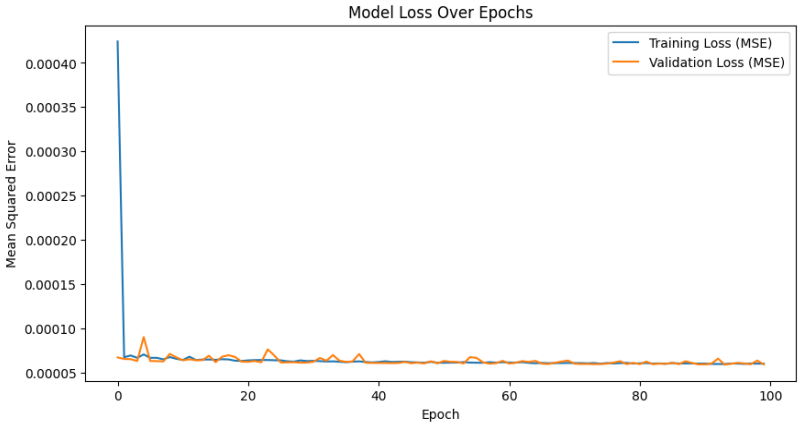



Рисунок 31. График потерь модели

Чтобы увидеть влияние признаков на выходные данные модели, использовались графики значений SHAP (объяснения по добавкам Шепли), как показано на рисунке 10. Значения SHAP объясняют выходные данные модели путем количественной оценки вклада каждой характеристики. Характеристики ранжируются в порядке важности влияния на выходные данные модели сверху вниз. Наибольшее влияние оказывает водонасыщенность, за ней следуют нефтенасыщенность и коэффициент извлечения газа для модели линейной регрессии. В то время как для модели дерева решений только среднее значение нефти показывает значительное влияние.

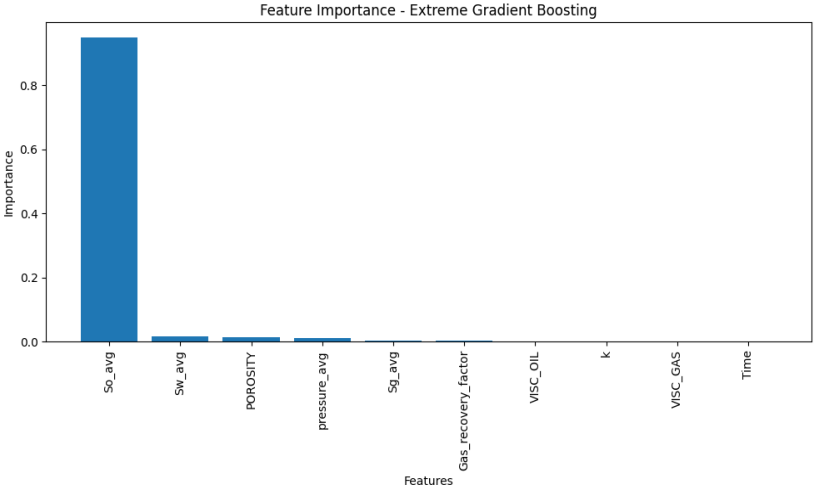



Рисунок 32. XGBoost feature importance

Подводя итог, можно сказать, что оптимизация гиперпараметров привела к значительному улучшению производительности всех моделей, снижению среднеквадратической ошибки и получению более точных прогнозов.

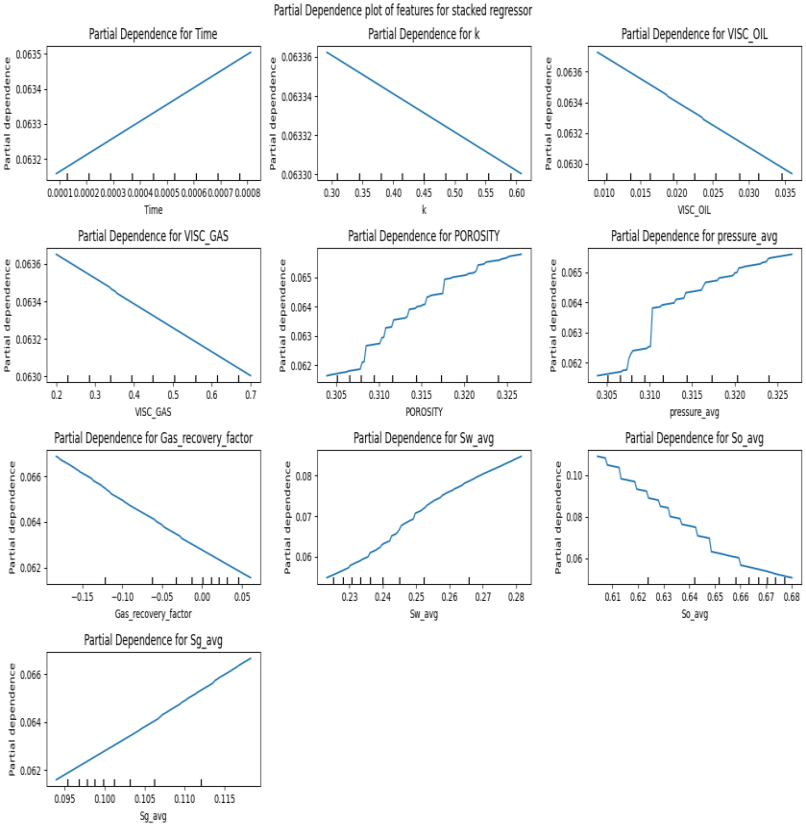



Рисунок 33. Графики частичной зависимости

Графики частичной зависимости предоставляют визуальное представление того, как изменение каждого признака влияет на целевую переменную в модели стекированного регрессора. Анализируя эти графики, можно сделать вывод о том, что многие признаки имеют монотонную зависимость от целевой переменной, то есть с увеличением значения признака целевая переменная также либо увеличивается, либо уменьшается. Однако, встречаются и нелинейные зависимости, а также зависимости со ступенчатым характером, что может

указывать на наличие пороговых значений или сложных взаимодействий между признаками. В контексте нефтегазовой отрасли, такие графики позволяют понять, как физические свойства пород, технологические параметры и геологические факторы влияют на добычу углеводородов. Несмотря на свою информативность, графики частичной зависимости имеют свои ограничения, связанные с упрощенным представлением взаимосвязей между признаками.

### 2.3. Использование методов машинного обучения для стохастического моделирование.

Целью данной работы является разработка интеллектуальных вычислительных моделей для прогнозирования коэффициента извлечения нефти на основе стохастического моделирования параметров нефтегазового месторождения, таких как пористость, вязкость воды и нефти, проницаемость. С помощью методов машинного обучения планируется создать модель, которая позволит оценить коэффициент извлечения нефти в условиях неопределенности, вызванной стохастическими факторами.

Исходные данные для моделирования были получены в результате стохастического моделирования системы двухфазного потока с использованием модели Бакли-Леверетта. Входные параметры, включенные в датасет, представляют собой:

- **porosity** (пористость): доля пористого пространства в пласте;
- **vis_water** (вязкость воды): вязкость воды в пласте;
- **vis_oil** (вязкость нефти): вязкость нефти в пласте;

- **k_permiability** (проницаемость): способность пористой среды пропускать жидкость;
- **oil_recovery_factor** (коэффициент извлечения нефти): целевая переменная, которую необходимо предсказать.

Для каждого параметра было сгенерировано 100 образцов и рассчитаны водонасыщенность, отклонение водонасыщенности и коэффициент извлечения нефти. На рисунке 34 показано сочетание различных образцов и коэффициент извлечения нефтяной фазы, результат водонасыщенности и отклонение водонасыщенности.

| | porosity | vis_water | vis_oil | k_permiability | oil_recovery_factor |
|---|---|---|---|---|---|
| 0 | 0.475607 | 0.089878 | 0.487803 | 0.003756 | 0.484094 |
| 1 | 0.475607 | 0.089878 | 0.487803 | 0.001826 | 0.686820 |
| 2 | 0.475607 | 0.089878 | 0.487803 | 0.005369 | 0.382396 |
| 3 | 0.475607 | 0.089878 | 0.487803 | 0.001850 | 0.682822 |
| 4 | 0.475607 | 0.089878 | 0.487803 | 0.004067 | 0.463761 |
| 5 | 0.475607 | 0.089878 | 0.487803 | 0.005490 | 0.372825 |
| 6 | 0.475607 | 0.089878 | 0.487803 | 0.004067 | 0.463718 |
| 7 | 0.475607 | 0.089878 | 0.487803 | 0.002947 | 0.547889 |
| 8 | 0.475607 | 0.089878 | 0.487803 | 0.004925 | 0.415150 |
| 9 | 0.475607 | 0.089878 | 0.487803 | 0.004401 | 0.443625 |
| 10 | 0.475607 | 0.089878 | 0.487803 | 0.008249 | 0.237190 |

Рисунок 34. Синтетический набор данных, созданный с использованием стохастического моделирования Баклея-Леверетта

Для исследования динамики коэффициента нефтеотдачи пласта были взяты 100 значений при различных промежуточных временных итерации математической модели для каждого примера выборки. Следовательно,

общее количество набора данных составила более 1 000 000 примера. Датасет был разделен на обучающий и тестовый сет. Для обучения использовано 80% от данных, а для теста остальные 20% данных. Наша задача относится к классу регрессионных задач с точки зрения методов машинного обучения. В данном монографии были реализованы и протестированы те же регрессионные алгоритмы машинного обучения, которые использовались для моделей **Black Oil** и **Buckley-Leverett**, с целью прогнозирования коэффициента нефтеотдачи. Оценка эффективности и точности этих алгоритмов применялась в условиях неопределенности, вызванной стохастическим моделированием.

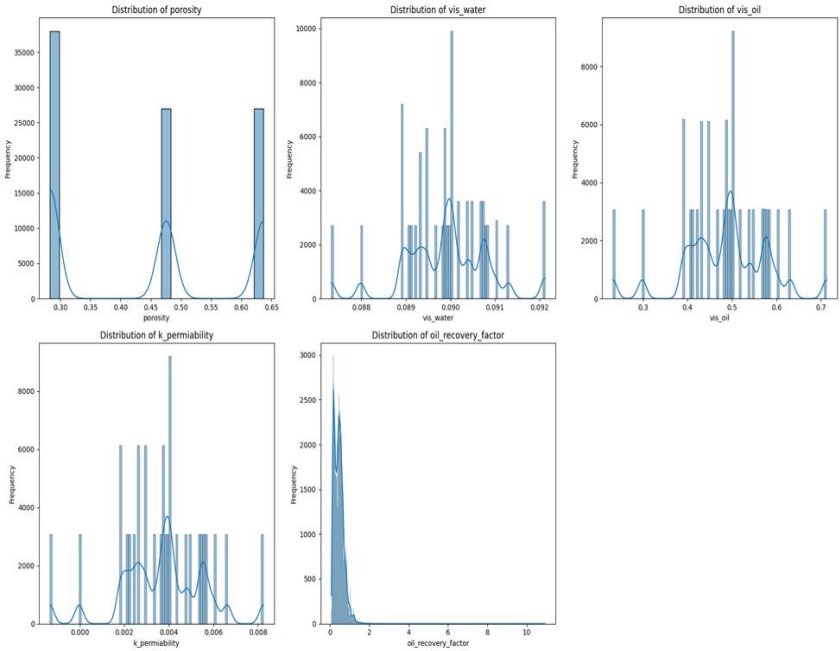



Рисунок 35. Гистограмма пористости

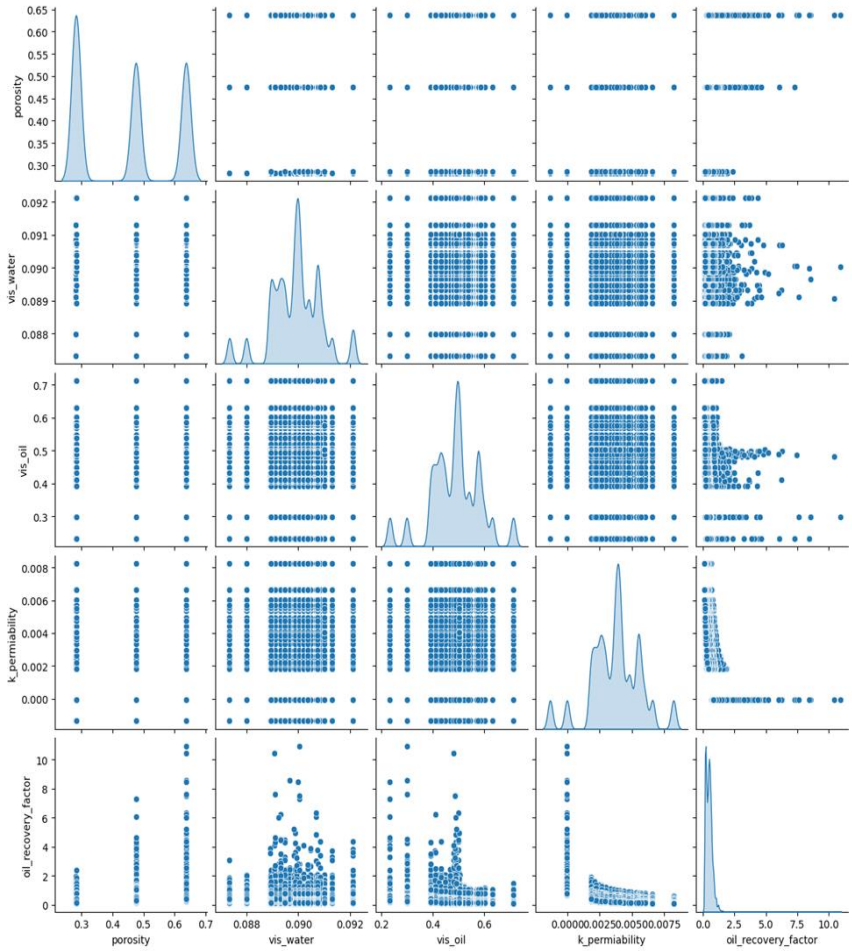



Рисунок 36. Матрица рассеяния для взаимосвязи между переменными

Матрица рассеяния дополнительно исследует взаимосвязи между этими переменными. Пористость и k_permeability показывают некоторую степень корреляции, что и ожидалось, поскольку эти свойства часто связаны в подземных материалах. Однако эта взаимосвязь не является

линейной, что указывает на другие факторы, влияющие на проницаемость. Связь между коэффициентом извлечения нефти и k_permeability примечательна, причем более высокие коэффициенты извлечения часто связаны с более высокой проницаемостью, что предполагает, что проницаемость играет критическую роль в определении эффективности извлечения. Диаграммы рассеяния, включающие вязкость нефти, показывают менее очевидную связь с другими переменными, хотя могут быть нелинейные зависимости, особенно в ее отношении к коэффициенту извлечения и проницаемости. Эти результаты указывают на важность проницаемости в управлении извлечением нефти, причем вязкость нефти усложняет поведение системы.

Рисунок 37. Тепловая карта

Пористость имеет умеренную положительную корреляцию с коэффициентом извлечения нефти (0,613), что говорит о том, что более высокая пористость связана с большей нефтеотдачей. Вязкость нефти показывает отрицательную корреляцию с коэффициентом извлечения

нефти (-0,295), что указывает на то, что более высокая вязкость нефти имеет тенденцию к снижению эффективности извлечения нефти. Проницаемость (k_permiability) имеет сильную отрицательную корреляцию с коэффициентом извлечения нефти (-0,542), что говорит о том, что более низкая проницаемость связана с уменьшением извлечения, что согласуется с ожидаемым поведением в потоке пористой среды.

Другие переменные, такие как вязкость воды, демонстрируют слабую или незначительную корреляцию как с коэффициентом извлечения нефти, так и с другими ключевыми параметрами, что позволяет предположить, что они оказывают минимальное влияние на коэффициент извлечения в этом наборе данных.

**Результаты машинного обучения.**

Для оценки регрессионных моделей, направленных на прогнозирование коэффициента нефтеотдачи, были выбраны следующие метрики: коэффициент детерминации $R^2$ и средняя квадратическая ошибка (MSE). Метрика $R^2$ используется для оценки качества модели, показывая, насколько хорошо предсказанные значения соответствуют фактическим данным, с максимальным значением 1, что указывает на идеальное соответствие. Средняя квадратическая ошибка (MSE) измеряет среднее значение квадратов ошибок, что позволяет оценить, насколько сильно предсказанные значения отклоняются от реальных.

Модель дерева решений показала оценку $R^2$ 0,96 на обучающих данных и 0,94 на тестовых данных, что указывает на сильную предсказательную силу с минимальной переподгонкой. Диаграммы рассеяния предсказанных и фактических значений показывают довольно близкое соответствие идеальной линии предсказания, хотя есть некоторые точки с большими

ошибками, особенно при более высоких значениях фактора восстановления.

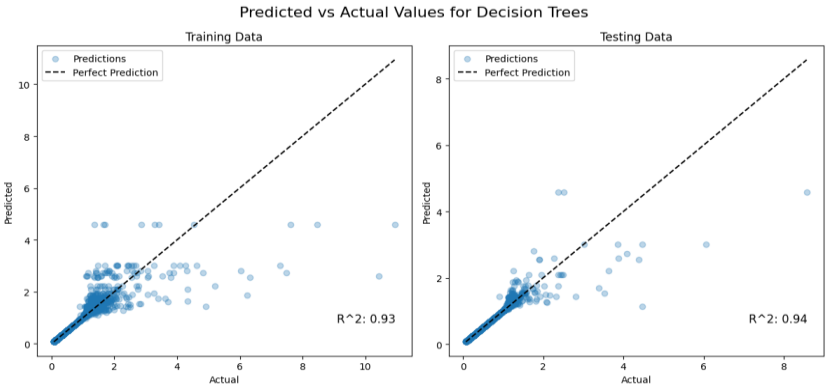



Рисунок 38. Результаты прогнозирования коэффициента нефтеотдачи с использованием алгоритма дерева решений

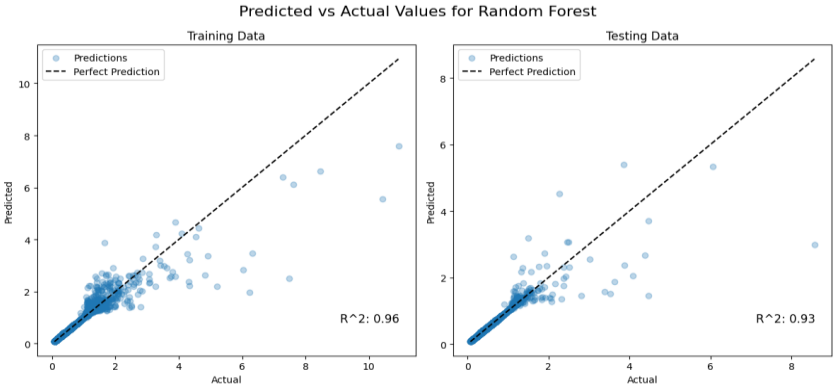



Рисунок 39. Результаты прогнозирования коэффициента нефтеотдачи с использованием алгоритма случайного леса

Модель случайного леса достигла оценки $R^2$ 0,96 на обучающем наборе и 0,93 на тестовом наборе. Подобно деревьям решений, модель работает хорошо, с немного улучшенным обобщением из-за ансамблевой природы случайного леса. Однако есть некоторая небольшая недооценка более высоких значений фактора восстановления, как видно на диаграммах рассеяния.

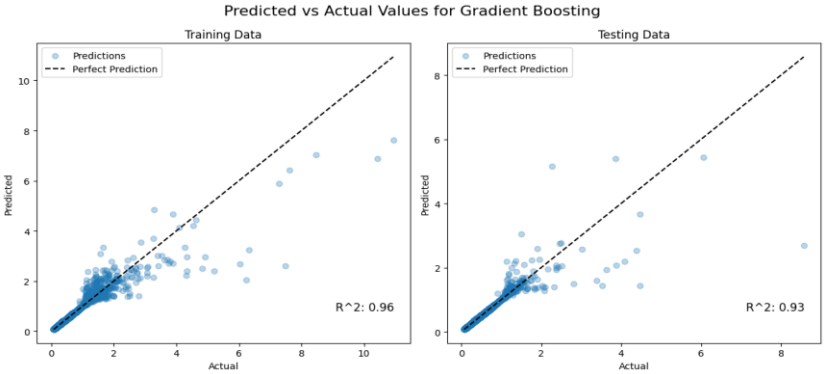



Рисунок 40. Результаты прогнозирования коэффициента нефтеотдачи с использованием алгоритма градиентного бустинга

Модель градиентного спуска показала оценку $R^2$ 0,96 на обучающем наборе и 0,93 на тестовом наборе. Модель хорошо улавливает взаимосвязь между признаками и фактором восстановления, хотя производительность на тестовых данных немного снижается для более высоких значений фактора восстановления, как и в случае со случайным лесом.

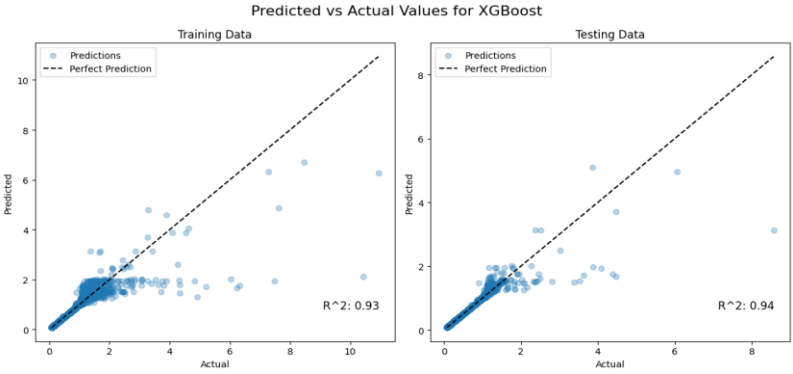



Рисунок 41. Результаты прогнозирования коэффициента нефтеотдачи с использованием алгоритма XGBoost

XGBoost продемонстрировал оценку $R^2$ 0,93 для обучающих данных и 0,94 для тестовых данных. Хотя его точность обучения немного ниже по сравнению со случайным лесом и градиентным усилением, он хорошо обобщает тестовые данные. Диаграмма рассеяния показывает хорошее соответствие между прогнозируемыми и фактическими значениями с относительно небольшим количеством выбросов.

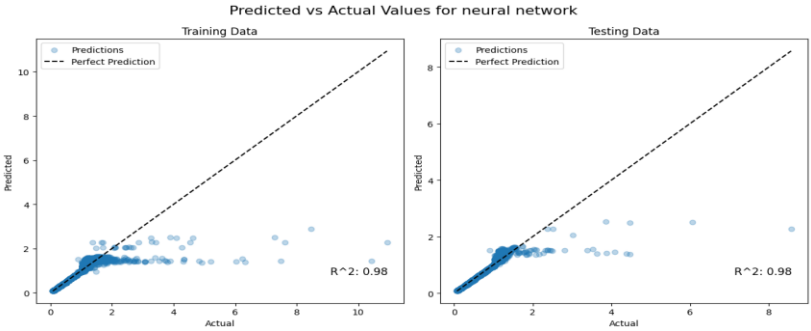



Рисунок 42. Результаты прогнозирования коэффициента нефтеотдачи с использованием нейронных сетей

Модель нейронной сети достигла оценки $R^2$ 0,98 на обучающем наборе и 0,98 на тестовом наборе, что предполагает очень высокую производительность. Близкое соответствие прогнозируемых и фактических значений на графике рассеяния подтверждает способность модели улавливать базовые закономерности в данных. Несмотря на свою сильную предсказательную силу, нейронные сети, как известно, подвержены переобучению, и необходима тщательная проверка.

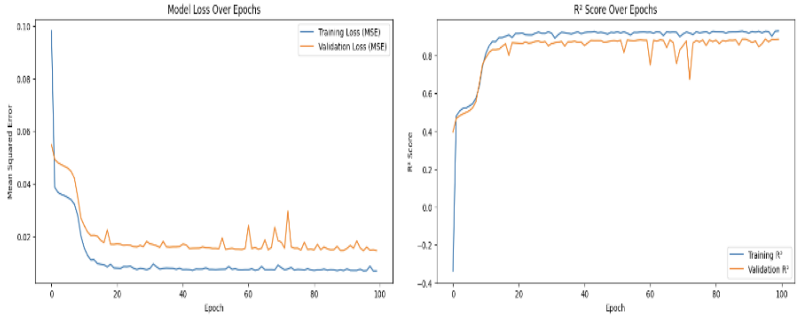



Рисунок 43. Сравнение обучающих и тестовых данных для $R^2$ и MSE нейронной сети

Сравнение обучающих и тестовых данных для нейронной сети, основанное на метриках $R^2$ и среднеквадратичной ошибки (MSE), показывает высокую эффективность модели. Значение $R^2$ для обучающей выборки составило 91, что указывает на отличное качество подгонки модели к данным, в то время как значение для валидационной выборки немного ниже и равно 90.5. Это небольшое снижение может свидетельствовать о том, что модель хорошо обобщает, хотя и может быть подвержена небольшому переобучению. Среднеквадратичная ошибка (MSE) также подтверждает успешное обучение: для обучающей выборки она составляет 0.01, а для

валидационной выборки — 0.02. Эти результаты подчеркивают стабильность модели и ее способность точно предсказывать значения, что делает ее подходящей для применения в задачах неравновесной фильтрации и стохастического моделирования.

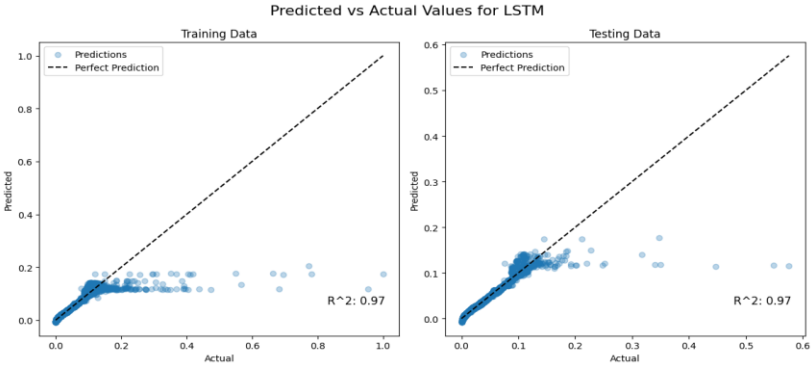



Рисунок 44. Результаты прогнозирования коэффициента нефтеотдачи с использованием LSTM

LSTM показала лучшую производительность среди всех моделей с оценкой $R^2$ 0,99 как для обучающего, так и для тестового наборов. Почти идеальное предсказание модели на обоих наборах указывает на то, что она способна исключительно хорошо обобщать, фиксируя долгосрочные зависимости в данных. Это особенно заметно на диаграммах рассеяния, где предсказанные значения почти идеально соответствуют фактическим значениям, отражая силу LSTM в обработке временных рядов или последовательных данных.

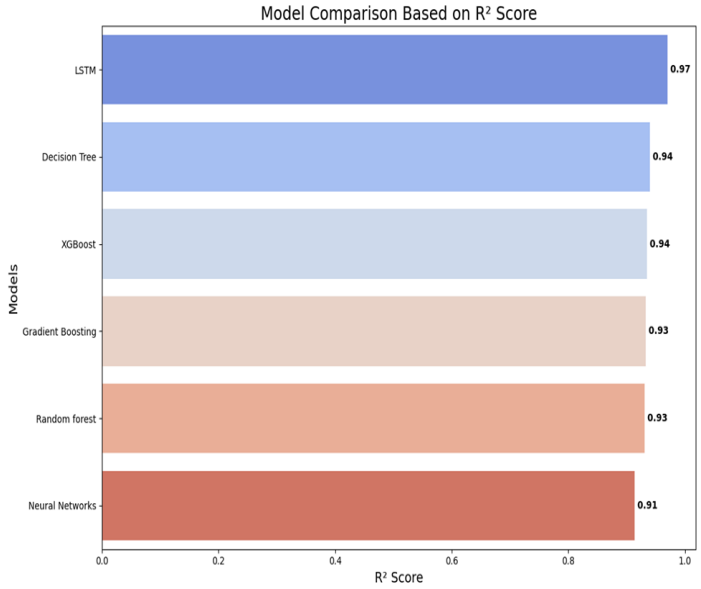



Рисунок 45. Результаты прогнозирования коэффициента нефтеотдачи для регрессионных алгоритмов

При сравнении моделей на основе их оценок $R^2$ LSTM оказывается самой эффективной моделью с почти идеальным $R^2$ 0,99, за ней следуют нейронные сети с 0,98 и модели на основе деревьев (деревья решений, случайный лес, градиентный бустинг, XGBoost) в диапазоне от 0,93 до 0,96. Сравнительная диаграмма наглядно демонстрирует превосходную производительность LSTM, за которыми следуют нейронные сети и методы на основе ансамблей.

На основе оценок $R^2$ и анализа прогнозируемых и фактических значений модель LSTM является наиболее эффективной для прогнозирования коэффициента извлечения нефти, за ней следует нейронная сеть. Древовидные модели, хотя и очень точны, показали немного более низкую производительность, особенно при более высоких значениях коэффициента извлечения, что

говорит о том, что эти модели могут быть менее чувствительны к выбросам или экстремальным значениям. Учитывая высокую производительность LSTM, она может быть наиболее подходящей моделью для этой задачи прогнозирования, особенно если данные имеют временные или последовательные зависимости. Однако относительно высокая точность других моделей также подчеркивает надежность древовидных и нейронных сетевых подходов в этой области.

В следующем разделе будет рассмотрена разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи неравновесной фильтрации и стохастического моделирования, включая использование методов распараллеливания и технологий CUDA.

## 3. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи неравновесной фильтрации и стохастичкеского моделирование.

В рамках работ прошлого года были разработаны и реализованы параллельные алгоритмы методов линейной регрессии, RF, GB, SVM с помощью стандарта MPI. Линейная регрессия была распараллелена разделением матриц и векторов по параллельным процессам. RF был параллелизован разделением деревьев решений по процессам. GB и SVM были распараллелены разделением данных по процессам. То есть, на каждом процессе были обучены копии модели поддатасета. Ответ был получен средним решением со всех процессов.

Также были разработаны и реализованы алгоритмы машинного обучения с использованием графического процессора (GPU) и различных методов параллелизации. Были реализованы линейная регрессия, метод опорных векторов (SVM), случайный лес (Random Forest) и градиентный бустинг. Для оптимизации вычислений использовались технологии CUDA и библиотеки из набора RAPIDS (cuDF и cuML), что позволило значительно ускорить обучение моделей и обработку больших объемов данных. Также были проведены сравнительные исследования производительности различных методов, которые показали значительное улучшение времени выполнения алгоритмов при использовании GPU.

### 3.1. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи неравновесной фильтрации (композиционная модель).

В рамках программы были разработаны параллельные алгоритмы методов машинного обучения. Существуют различные параллельные архитектуры. Современные кластерные и суперкомпьютерные системы строятся на гибридной архитектуре, т. е. система состоит из вычислительных узлов с разделенной памятью, а каждый узел может иметь многоядерные CPU и GPU как ускоритель.

Работа методов, последовательная реализация была описана в предыдущей подглаве. Существуют два основных подхода реализации параллельных алгоритмов: параллелизм модели и параллелизм данных. В первом подходе параллельно выполняются расчеты внутри модели. Во втором подходе данные разделяются по вычислительным узлам, обучение проходит параллельно, а при прогнозировании каждый узел выдает собственный прогноз, и окончательное решение можно принять на основе этих прогнозов выбором большинства или расчетом среднего значения.

Разработан параллельный алгоритм линейной регрессии для задачи нефтедобычи. В данном методе веса можно найти с помощью градиентного спуска, а в некоторых случаях с помощью нормального уравнения аналитическим методом. Нормальное уравнение будет выглядеть следующим образом:

$$\theta = (X^T X)^{-1} X^T y$$

где $X$ - матрица признаков, где каждая строка представляет собой одно наблюдение, а каждый столбец - признак (включая единичный столбец для свободного члена); $y$ - вектор целевых переменных; $\theta$ - вектор параметров модели. Уравнение состоит из матричного умножения, нахождения обратной матрицы и умножения матрицы на вектор. Данные операции были распараллелены с помощью стандарта MPI для систем с разделенной памятью. Результаты параллелизации можно увидеть в таблице 9.

Алгоритм SVM для регрессии был параллелизован с помощью параллелизма данных. Выборка была разделена по параллельным процессам, каждый процесс проводил обучение на свой подвыборке, затем подмодели на каждом процессе собирались на главном процессе, который в свою очерель проводил прогнозирование. Результаты параллелизации можно увидеть в таблице 10.

Самым удобным методом для параллелизации на кластерных системах является метод Random Forest. В методе строятся N деревья решений на рандомных подвыборках. При прогнозировании находится усредненное значение с каждого дерева. При параллельной реализации N деревья делятся на NP параллельных процессов и каждый процесс строит свои деревья решений независимо друг от друга. Результаты параллелизации можно увидеть в таблице 11.

Алгоритм gradient boosting также был параллелизован с помощью параллелизма данных. Метод является сложным для параллелизации, так как в алгоритме деревья решений строятся последовательно и зависят друг от друга. Параллелизация проводилась аналогично с методом SVM. Результаты параллелизации можно увидеть в таблице 13.

Таблица 9. Время выполнения параллельного алгоритма LR, с

| NP\m | 600000 | 1200000 | 2400000 |
|---|---|---|---|
| 1 | 4,22 | 7,52 | 14,93 |
| 2 | 1,61 | 4,05 | 8,44 |
| 4 | 1,29 | 3,18 | 6,18 |
| 8 | 1,24 | 2,83 | 5,74 |

Таблица 10. Время выполнения параллельного алгоритма SVM, с

| NP\m | 100000 | 200000 | 400000 |
|---|---|---|---|
| 1 | 2,25 | 4,77 | 11,3 |
| 2 | 1,2 | 2,5 | 5,46 |
| 4 | 0,68 | 1,36 | 2,96 |
| 8 | 0,59 | 1,02 | 1,91 |

Таблица 11. Время выполнения параллельного алгоритма Random Forest, с

| P\m | 10000 | 20000 | 40000 | 80000 |
|---|---|---|---|---|
| 1 | 0,37 | 1,05 | 2,69 | 7,45 |
| 2 | 0,15 | 0,46 | 1,15 | 3,16 |
| 4 | 0,06 | 0,17 | 0,49 | 1,4 |
| 8 | 0,03 | 0,1 | 0,34 | 0,87 |

Таблица 12. Время выполнения параллельного алгоритма Gradient Boosting, с

| NP\m | 100000 | 200000 | 400000 |
|---|---|---|---|
| 1 | 3,71 | 8,41 | 16,14 |
| 2 | 1,74 | 4,64 | 9,11 |
| 4 | 1,12 | 2,29 | 5,33 |
| 8 | 0,75 | 1,61 | 3,57 |

Во всех таблицах первый столбец означает количество параллельных процессов, первая строка означает размер выборки(количество строк матрицы $X$), а каждая следующая строка время выполнения параллельной программы. По данным на таблицах можно увидеть что все алгоритмы дали ускорения относительно последовательной программы. В методе SVM меньше всего размер выборки, так как метод сам по себе выполняется для данных размером до 50000 наблюдений. Но именно данный метод дает больше всего ускорения, данное явление зависит от сложности алгоритма, при разделении данных обучение проходит быстрее на каждом процессе, в сумме с ускореним от параллелизации получаем суперускорение. Например на 8 процессах при количестве наблюдений равным 10000 получили ускорение в 12 раз вместо теоритического восьми. Полученные результаты дают предположить что имеет смысл проводить параллельное обучение приведенных алгоритмов в системах с разделенной памятью с целью ускорения процесса на больших данных.

Распараллеливание искусственных нейронных сетей с использованием MPI позволяет эффективно распределять вычислительную нагрузку между несколькими процессами или узлами. Это может значительно сократить время обучения больших и сложных нейронных сетей. Описанный выше алгоритм представляет собой пошаговое руководство по распараллеливанию искусственной

нейронной сети с использованием MPI. Разделяя входные данные, инициализируя отдельные модели нейронной сети и координируя обновления веса и смещения, алгоритм гарантирует, что распараллеленная нейронная сеть работает эффективно и дает точные результаты. Дальнейшие исследования и эксперименты могут быть проведены для оптимизации алгоритма и изучения других методов распараллеливания для искусственных нейронных сетей. Время выполнения параллельной программы MPI ANN VS номера процессов показано в рисунке А.3.

Также были реализованы алгоритмы с использованием графического процессора. Для каждого алгоритма используются разные подходы использования GPU: параллелизация матричных операции с использованием CUDA, использование библиотеки numba для вычисления на графическом процессоре и использование платформы RAPIDS для анализа данных и машинного обучения, разработанная NVIDIA.

Был создан параллельный алгоритм линейной регрессии (LR), который выполняет обучение модели и выполнение предсказаний на графических процессорах (GPU) с использованием технологии CUDA.

Отдельно была реализована линейная регрессия с использованием библиотек cuML и cuDF из набора инструментов RAPIDS. cuDF – это библиотека для манипуляции и анализа данных, которая предоставляет функциональность, аналогичную библиотеке Pandas, но спроектирована для работы на GPU. Это позволяет значительно увеличить скорость обработки данных, особенно для больших объемов данных. cuML – это библиотека которая предоставляет реализации различных алгоритмов машинного обучения, таких как классификация, регрессия и кластеризация, работающие на GPU. Это ускоряет обучение моделей и позволяет

обрабатывать большие объемы данных. Эти библиотеки предоставляют оптимизированные и высокопроизводительные функции для манипуляции данными и машинного обучения на GPU.

Оба подхода обеспечивают параллельное выполнение линейной регрессии на графических процессорах, что улучшает производительность и ускоряет решение задач машинного обучения, особенно на больших объемах данных.

Таблица 13. Время выполнения алгоритма LR с разными подходами

| Размер матрицы | Послед C++ | CUDA C++ | scikit-learn | Nvidia Rapids |
|---|---|---|---|---|
| 600 000 | 1.86 | 1.41 | 0.22 | 0.007 |
| 1 200 000 | 3.786 | 2.63 | 0.57 | 0.12 |
| 2 097 120 | 7.775 | 4.798 | 0.933 | 0.12 |

В продолжение работы были разработаны алгоритмы машинного обучения, включая SVM (Метод опорных векторов), Random Forest (Случайный лес) с использованием инструментов RAPIDS (cuDF и cuML), а также Gradient Boosting с использованием библиотеки XGBoosting. Процесс обучения проводился на наборе данных, содержащем более 400,000 примеров.

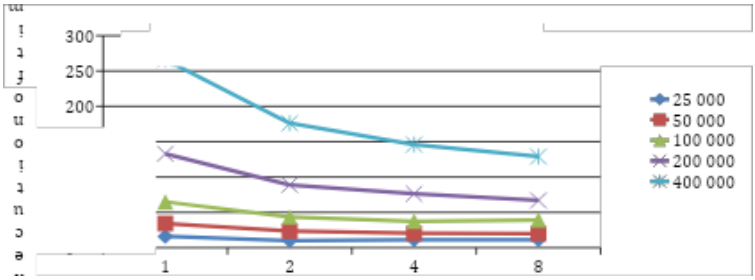



Рисунок 46. Время выполнения параллельной программы MPI ANN VS номера процессов

Во время обучения было выделено 80% данных для обучения моделей, а оставшиеся 20% данных использовались для тестирования. Полученные результаты обучения были сопоставимы с ранее реализованными алгоритмами машинного обучения, однако важно отметить, что использование инструментов RAPIDS привело к значительному увеличению производительности обучения. Результаты сравнительных исследований производительности представлены в таблице 14.

Таблица 14. Сравнение производительности методов машинного обучения

| Алгоритм | Подход с GPU, сек. | Ранее реализованный подход, сек. |
|---|---|---|
| Линейная регрессия (LR) | 0.01 | 0.022 |
| SVM | 45.84 | 4809.84 |
| Random Forest | 0.55 | 18.95 |
| Gradient Boosting | 0.32 | 11.39 |

Эффективность обучения на GPU при использовании инструментов RAPIDS (cuDF и cuML) и XGBoosting позволила значительно увеличить скорость обработки данных и обучения моделей машинного обучения.

А также было использовано методы параллельных искусственных нейронных сетей (ИНС) с использованием технологии CUDA. Архитектура распараллеливания данных с использованием искусственной нейронной сети показано на рисунке 47.

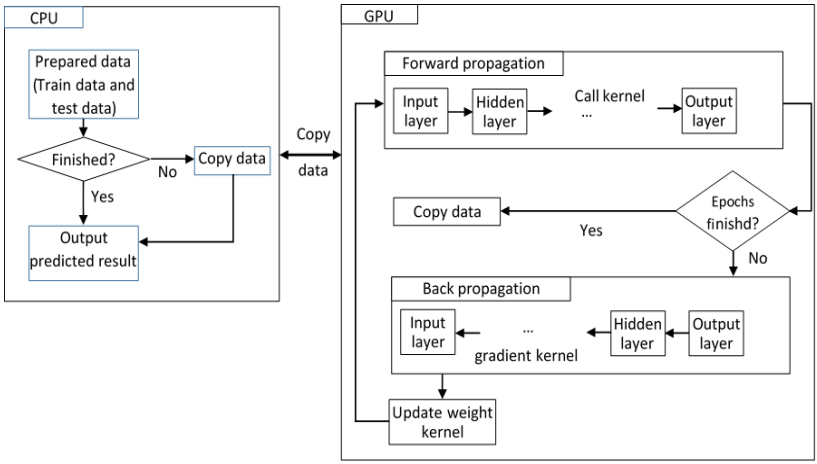



Рисунок 47. Распараллеливание искусственной нейронной сети (ИНС) с помощью CUDA

С использованием технологии CUDA результаты параллельного выполнения данных, полученных по модели Бакли-Леверетта вытеснения жидкости в пористых средах с помощью искусственной нейронной сети (ИНС), представлены в таблице 10:

Таблица 15. Время работы DNN_CPU и DNN_GPU

| Batch size | CPU_DNN | GPU_DNN |
|---|---|---|
| 32 | 457.53 | 551.56 |
| 64 | 233.34 | 277.20 |
| 128 | 136.6 | 139.02 |
| 256 | 67.95 | 70.99 |
| 512 | 37.30 | 38.40 |
| 1024 | 32.54 | 19.71 |
| 2048 | 61.82 | 10.52 |
| 3000 | 129.15 | 8.0 |
| 4096 | 127.62 | 8.15 |
| 5000 | 128.82 | 8.85 |

DNN-CPU также имеет значительное уменьшение времени выполнения между размером пакета 32 и 1024, но, когда размер пакета превышает 1024, время выполнения значительно уменьшается. А DNN-GPU значительно сокращает время выполнения между размером пакета 32 и 3000, но, когда размер пакета превышает 4096, время выполнения начинает уменьшаться. Однако при обработке аналогичного объема данных время выполнения GPU в 4 раза быстрее, чем у CPU. На рисунке 48 показано время обработки процессора и графического процессора.

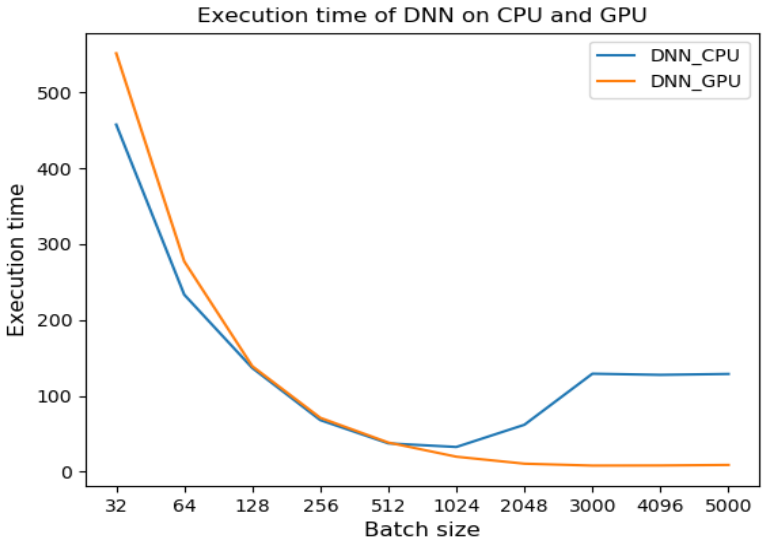



Рисунок 48. Относительное время выполнения

Обработка данных искусственной нейронной сети (ИНС) лучше всего работала на ЦП с размером пакета 1024, а графический процессор — с размером пакета 3000.

## 3.2. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи моделирования трехфазного течения Black Oil.

Для ускорения процесса обучения были реализованы запуски на нескольких параллельных процессах. Параллельное обучение моделей машинного обучения и нейронных сетей можно разделить на два основных подхода: параллелизм данных и параллелизм модели.

Параллелизм данных предполагает разделение набора данных на узлы параллельной вычислительной системы и обучение независимых моделей на каждом узле на своих поднаборов данных. Каждая модель независимо выполняет прогнозирование, а итоговое решение является средним значением всех моделей.

Параллелизм модели включает распределение различных вычислений между несколькими узлами, где каждый узел обрабатывает часть общей задачи. Несмотря на распределение задач, все узлы работают над решением единой модели. Конечная цель заключается в том, чтобы эффективно решить одну согласованную модель, используя вычислительную мощность нескольких узлов одновременно.

Для небольших наборов данных существует риск потери точности при параллелизме данных. Однако, при поддержании оптимальной точности и достаточного ускорения, использование параллелизма данных может быть полезным. Визуальная диаграмма параллелизма данных представлена на рисунке 49.

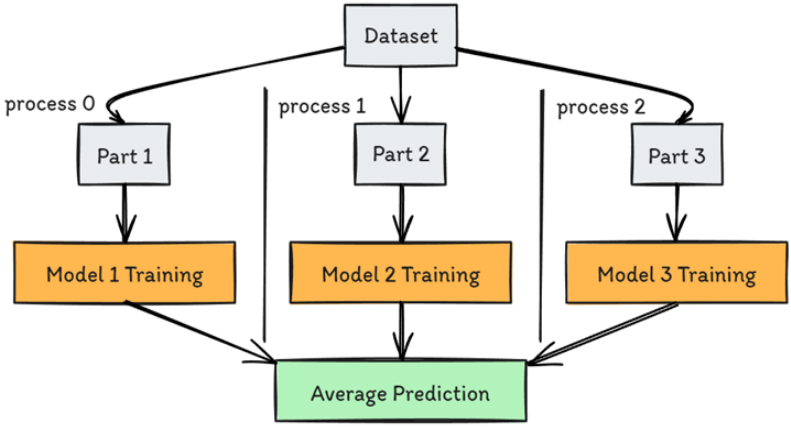


Рисунок 49. Схема параллелизма данных

В нашем подходе к параллелизму данных та же самая модель машинного обучения, включая нейронные сети, применяется к каждому поднабору данных. Это обеспечивает согласованность в обучении и прогнозировании модели на всех поднаборах. В частности, данные разделяются на несколько поднаборов, и одна и та же модель (например,RF или нейронная сеть) обучается на каждом поднаборе независимо.

Модель нейронной сети, используемая для каждого поднабора, имеет идентичные параметры и архитектуру, включая количество слоев, скорость обучения и другие связанные параметры. После обучения прогнозы от каждой модели агрегируются для получения окончательного результата. Агрегация может осуществляться путем усреднения прогнозов или с использованием другого метода для их эффективного объединения. Этот подход не только повышает вычислительную эффективность, но и поддерживает точность и надежность предсказательной модели.

Использование графических процессоров (GPU) для задач машинного обучения значительно ускоряет

вычисления, особенно при обработке больших данных, благодаря параллельной обработке и высокой пропускной способности. Это делает GPU идеальным решением для ресурсоемких задач по сравнению с обычными процессорами. В рамках обучения на GPU применялась библиотека cuML из пакета RAPIDS, предоставляющая эффективные реализации методов регрессии, таких как линейная регрессия, случайный лес, деревья решений и градиентный бустинг. Библиотека использует архитектуру GPU и платформу NVIDIA CUDA для ускорения вычислительных процессов благодаря параллелизму и оптимальному использованию ресурсов GPU. Алгоритмы в cuML специально адаптированы для выполнения на графических процессорах, включая распараллеливание операций и эффективное управление памятью. Для ускорения обучения нейронных сетей использовался TensorFlow, поддерживающий работу на GPU через библиотеку CUDA. Это позволяет эффективно распределять вычислительные задачи между ядрами GPU, что существенно сокращает время обучения моделей. TensorFlow автоматически распознаёт доступные графические процессоры и оптимизирует распределение нагрузки, обеспечивая максимальную производительность и минимальные задержки, что способствует быстрому и точному результату. На рисунке 49 представлен классический метод параллельных вычислений на CPU, тогда как наша реализация на GPU с использованием CUDA использует дополнительные механизмы как на рисуноке 24. RAPIDS был выбран для реализации машинного обучения благодаря его совместимости с scikit-learn. Это позволяет использовать привычный API scikit-learn с дополнительными возможностями ускорения на GPU через CUDA. Методы в RAPIDS разработаны с учётом

совместимости со scikit-learn, что упрощает проверку и сравнение производительности моделей.

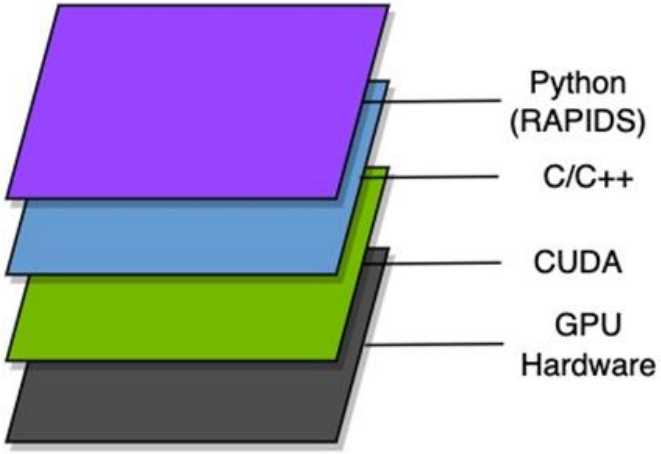



**Рисунок 50.** Параллелизм на GPU

Параллельное обучение с использованием технологий RAPIDS и TensorFlow позволило значительно ускорить процесс обучения моделей, что критически важно при работе с большими объёмами данных. RAPIDS cuML и cuDF обеспечили высокую скорость обработки данных на GPU, а TensorFlow применялся для обучения нейронных сетей, что также увеличивало производительность вычислений.

RAPIDS включает в себя несколько ключевых компонентов, работающих вместе для обеспечения высокопроизводительных вычислений:

1. cuDF — библиотека для работы с табличными данными, аналогичная Pandas, но оптимизированная для работы на GPU.
2. cuML — библиотека машинного обучения, обеспечивающая ускорение таких алгоритмов, как

регрессия, классификация и кластеризация, с помощью GPU.

3. cuGraph — библиотека для работы с графами, обеспечивающая высокоскоростную обработку графовых данных.

Каждый из этих компонентов использует технологию CUDA для распараллеливания задач и переноса вычислений на графический процессор.

CUDA — это платформа и модель программирования, разработанная NVIDIA для работы с графическими процессорами. GPU отличается от CPU тем, что имеет тысячи небольших ядер, которые могут выполнять множество задач одновременно. Это делает GPU эффективным для задач, которые можно распараллелить, например, обработка данных, обучение моделей машинного обучения, и работа с большими массивами данных.

В RAPIDS каждое ядро GPU может обрабатывать часть данных или выполнять отдельные операции параллельно с другими ядрами. CUDA предоставляет API, позволяющий программистам напрямую управлять вычислениями на GPU, используя язык программирования C или Python через высокоуровневые библиотеки (такие как cuDF и cuML).

RAPIDS использует парадигму разделения задач на потоки, которые могут одновременно исполняться на разных ядрах GPU. Например, при обработке табличных данных (операции фильтрации, агрегации и т. д.), данные могут быть разбиты на блоки, которые распределяются между ядрами GPU для параллельного выполнения. Это позволяет обрабатывать большие объёмы данных с высокой производительностью.

cuDF — это основная библиотека RAPIDS для работы с табличными данными на GPU. Она предоставляет функциональность, аналогичную библиотеке Pandas, но использует мощь GPU для ускорения операций с данными.

cuDF работает с типами данных, аналогичными типам Pandas, и поддерживает основные операции, такие как:

- Фильтрация данных;
- Агрегирование;
- Группировка;
- Слияние и соединение таблиц.

Основная задача cuDF — обеспечить высокую производительность при работе с большими наборами данных, которые могут занимать много времени на обработку с помощью традиционных инструментов на CPU. Благодаря использованию параллельных вычислений на GPU, cuDF может обрабатывать миллионы строк данных за секунды. Он оптимизирован для работы с операциями, которые можно распараллелить, такими как сортировка и фильтрация, где каждое ядро GPU может обрабатывать часть данных независимо от других.

Схема работы RAPIDS на GPU может быть представлена следующим образом:

1. Загрузка данных на GPU — данные загружаются в память GPU для дальнейшей обработки.
2. Разделение данных — данные разбиваются на блоки, и каждая часть данных отправляется на отдельные ядра GPU.
3. Параллельное выполнение операций — каждое ядро GPU одновременно выполняет операцию над своей частью данных.
4. Агрегация результатов — результаты вычислений собираются в итоговый массив или таблицу.
5. Возвращение данных — обработанные данные могут быть возвращены обратно на CPU (если требуется).

cuDF использовался для предварительной обработки данных, таких как параметры, связанные с физическими характеристиками системы (давление, проницаемость, объемы фаз и др.). Этот инструмент позволяет

обрабатывать большие таблицы данных и выполнять такие операции, как фильтрация, агрегация и слияние данных, значительно быстрее по сравнению с традиционными методами на основе CPU. В контексте задачи, где объём данных может быть очень большим, скорость обработки на GPU с использованием cuDF обеспечила важное ускорение всех этапов подготовки данных перед обучением моделей машинного обучения.

cuML был задействован для обучения машинных моделей, таких как градиентный бустинг (GB), деревья решений (DT) и случайные леса (RF). Основное преимущество cuML — это возможность быстро обучать модели на GPU, что позволило значительно сократить время выполнения сложных вычислительных задач.

Использование фреймворков RAPIDS продемонстрировало значительное ускорение обучения моделей. Результаты времени обучения моделей машинного обучения показывают, что задачи, которые могли бы занимать минуты или часы на CPU, на GPU выполняются за доли секунд. Это особенно важно в задачах моделирования трёхфазного течения Black Oil, где необходимо быстро обрабатывать и анализировать большие объемы данных. RAPIDS cuML позволил уменьшить время обучения моделей до менее одной секунды для GB, DT и RF, что делает эту технологию идеальной для задач, где производительность играет ключевую роль.

Наряду с классическими моделями машинного обучения, для задачи моделирования трёхфазного течения Black Oil использовались нейронные сети, обучение которых проводилось на GPU с использованием фреймворка TensorFlow. В этой задаче нейронные сети применялись для решения сложных задач предсказания, где требуется учесть нелинейные зависимости между входными параметрами системы.

Использование TensorFlow позволило значительно ускорить обучение нейронных сетей, благодаря эффективной параллелизации и поддержке вычислений на GPU. Хотя обучение нейронной сети требует больше времени по сравнению с другими моделями, она обеспечивает возможность точного предсказания в условиях сложных физических процессов.

TensorFlow также предоставляет инструменты для гибкого управления процессом обучения, включая возможность настройки архитектуры сети и параллельного выполнения задач на GPU. Это позволило достичь необходимой точности предсказаний при минимальных затратах времени, что особенно важно для моделей, которые требуют учета большого количества факторов и взаимодействий.

В результате использования машинного обучения, задача моделирования трёхфазного течения стала решаться с большей эффективностью, что позволяет применять такие подходы для более крупных и сложных систем в реальном времени.

Алгоритмы для параллельного обучения регрессионных моделей были разработаны и реализованы с использованием библиотеки mpi4py. «mpi4py» — это пакет для Python, который предоставляет привязки к библиотеке MPI (Message Passing Interface). MPI является стандартным интерфейсом, используемым для обмена сообщениями между процессами, выполняемыми на разных узлах в вычислительном кластере или параллельной вычислительной системе.

С помощью рангов процессов набор данных был разделен и распределен по разным узлам. Каждый узел затем осуществлял обучение модели параллельно, независимо от других. После обучения производительность и точность модели анализировались коллективно для

оценки эффективности подхода распределенного обучения. В таблицах ниже показаны результаты параллельного обучения моделей, описанных в главе 2.1.

**Таблица 16.** Время параллельного обучения регрессионных моделей, секунд:

| Количество процессов | GB | DT | RF | NN |
|---|---|---|---|---|
| 1 | 12841.98 | 705.69 | 1054.32 | 124.97 |
| 2 | 6377.08 | 323.54 | 502.66 | 62.47 |
| 4 | 3104.33 | 146.97 | 235.31 | 24.23 |
| 8 | 1508.85 | 66.95 | 106.15 | 16.76 |

Градиентный бустинг (GB) показывает самое длительное время обучения при всех количествах процессов. Это ожидаемо, так как алгоритм GB требует больше вычислительных ресурсов из-за итеративного характера. Деревья решений (DT) и Случайные леса (RF) демонстрируют относительно низкое время обучения по сравнению с GB, при этом DT является самой быстрой моделью. Нейронные сети (NN) имеют самое короткое время обучения. Это может быть связано с тем, что архитектура сети либо достаточно лёгкая, либо хорошо оптимизирована для параллельной обработки. С увеличением числа процессов время обучения значительно сокращается для всех моделей. Это свидетельствует о высокой эффективности параллелизации, особенно заметно для GB, время обучения которого сократилось с 12,841.98 секунд (1 процесс) до 1,508.85 секунд (8 процессов), а также для RF — с 1,054.32 секунд до 106.15 секунд при аналогичном увеличении числа процессов.

**Таблица 17.** Ускорение параллельного обучения регрессионных моделей:

| Количество процессов | GB | DT | RF | NN |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 2 | 2.013771 | 2.18115 | 2.097481 | 2.0004 |
| 4 | 4.136796 | 4.80159 | 4.480557 | 5.1576 |
| 8 | 8.511104 | 10.5405 | 9.932359 | 7.4564 |

Градиентный бустинг (GB) демонстрирует линейное и даже суперлинейное ускорение (8.51x при 8 процессах). Это указывает на значительную выгоду от параллелизации, возможно, благодаря сокращению накладных расходов на память или лучшей работе кэша при увеличении числа процессов. Деревья решений (DT) показывают наибольшее ускорение, достигая 10.54x при 8 процессах. Это говорит о том, что алгоритм деревьев решений очень хорошо параллелизуется, так как каждое дерево в ансамбле может строиться независимо. Случайные леса (RF) также показывают сильные преимущества параллелизации с ускорением в 9.93x при 8 процессах. Нейронные сети (NN) демонстрируют ускорение в 7.46x при 8 процессах, что немного меньше, чем у других моделей, но всё же значительное. Это может быть связано с более сложной параллелизацией нейронных сетей, где коммуникация между процессами может добавить накладные расходы по мере увеличения их количества.

**Таблица 18.** Значение коэффициента детерминации ($R^2$) для различных алгоритмов при разном количестве процессов:

| Количество процессов | GB | DT | RF | NN |
|---|---|---|---|---|
| 1 | 0.9757 | 0.9735 | 0.975 | 0.9756 |
| 2 | 0.9758 | 0.9742 | 0.976 | 0.9758 |
| 4 | 0.9759 | 0.9753 | 0.976 | 0.9752 |
| 8 | 0.9758 | 0.975 | 0.975 | 0.9751 |

Градиентный бустинг (GB) показывает стабильно высокие значения $R^2$ при любом количестве процессов (от 0.9757 до 0.9759). Это свидетельствует о том, что параллелизация не влияет на точность модели. Деревья решений (DT) показывают незначительное увеличение $R^2$ с увеличением числа процессов, но значения остаются стабильными около 0.975. Случайные леса (RF) сохраняют высокие и стабильные значения $R^2$ (от 0.975 до 0.976), что свидетельствует о надёжной работе модели независимо от числа процессов. Нейронные сети (NN) демонстрируют небольшие колебания, но значения $R^2$ остаются стабильно высокими (около 0.975), что говорит о сохранении прогностической мощности модели даже при увеличении числа процессов.

Параллелизация значительно сокращает время обучения для всех регрессионных моделей, причём деревья решений и случайные леса показывают наилучшее ускорение. Градиентный бустинг также получает значительные преимущества, особенно учитывая его вычислительную сложность. Анализ ускорения показывает, что параллельная обработка может привести к почти линейным или даже суперлинейным улучшениям

производительности. Деревья решений (DT) и случайные леса (RF) демонстрируют наибольшее ускорение, за ними следует градиентный бустинг (GB), в то время как нейронные сети (NN) демонстрируют более умеренное ускорение из-за сложностей параллелизации. Точность модели ($R^2$) остаётся стабильной для всех моделей при любом количестве процессов, что указывает на то, что параллелизация не снижает точности предсказаний моделей. Это важно, так как улучшение времени обучения не сопровождается снижением точности модели. Для высоко параллелизуемых задач, таких как случайные леса и деревья решений, увеличение числа процессов обеспечивает значительное сокращение времени обучения при сохранении высокой точности модели. Для более сложных моделей, таких как градиентный бустинг и нейронные сети, параллелизация также помогает, но масштабирование оказывается менее драматичным для нейронных сетей. Оптимальное количество процессов следует выбирать на основе доступных ресурсов и компромисса между ускорением и потреблением ресурсов.

Как видно из приведенных выше результатов, для почти всех моделей было получено хорошее ускорение. При этом ускорение достигнуто практически без потери точности. Эти результаты позволяют рассматривать параллелизацию с использованием MPI как отличную возможность для ускорения обучения моделей.

Для оценки производительности параллельного выполнения на графическом процессоре были проведены тесты на видеокарте Nvidia RTX 3070. Результаты этих тестов представлены в таблице 19.

**Таблица 19.** Время обучения регрессионных моделей на GPU,секунд

| GB | DT | RF | NN |
|---|---|---|---|
| 0.28 | 0.68 | 0.66 | 15.8 |

Как показали результаты, применение GPU существенно ускорило процесс обучения. Однако данные по полиномиальной регрессии отсутствуют в таблице 11, поскольку этот метод не поддерживается напрямую в библиотеке cuML.

### 3.3. Разработка интеллектуальных высокопроизводительных вычислительных моделей для решения задачи стохастичкеского моделирования.

С помощью параллелизма данных было проведено параллельное обучение моделей описанных в разделе 2.2 для данных полученных от стохастического моделирования. Параллельное обучение проводилось также как в главе 3.1. Результаты параллельного обучения с использованием распараллеливания данных для четырёх регрессионных моделей (GB - градиентный бустинг, DT - деревья решений, RF - случайные леса, NN - нейронные сети) при различном количестве процессов (1, 2, 4, 8) приведены в таблицах 20-22.

**Таблица 20.** Время параллельного обучения регрессионных моделей, секунд:

| Количество процессов | GB | DT | RF | NN |
|---|---|---|---|---|
| 1 | 310.29 | 16.78 | 26.57 | 26.26 |
| 2 | 157.65 | 8.39 | 15.09 | 13.33 |
| 4 | 81.21 | 4.35 | 7.42 | 11.06 |
| 8 | 42.83 | 2.38 | 3.79 | 8.73 |

Градиентный бустинг (GB) имеет самое долгое время обучения при всех количествах процессов, однако наблюдается стабильное снижение времени с 310,29 секунд (1 процесс) до 42,83 секунд (8 процессов). Деревья решений (DT) и Случайные леса (RF) демонстрируют гораздо меньшее время обучения, причём DT является самой быстрой моделью. Время обучения DT уменьшается с 16,78 секунд до всего 2,38 секунд, а RF — с 26,57 секунд до 3,79 секунд при 8 процессах. Нейронные сети (NN) демонстрируют более стабильное время по сравнению с RF и DT, снижая время обучения с 26,26 секунд (1 процесс) до 8,73 секунд (8 процессов). С увеличением количества процессов время обучения значительно сокращается для всех моделей, что указывает на эффективную параллелизацию. Однако нейронные сети получают меньшее преимущество по сравнению с деревьями решений и случайными лесами.

**Таблица 21.** Ускорение параллельного обучения регрессионных моделей:

| Количество процессов | GB | DT | RF | NN |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 2 | 1,9682 | 1,7607 | 1,7607 | 1,9699 |
| 4 | 3,8208 | 3,5808 | 3,5808 | 2,3743 |
| 8 | 7,2446 | 7,0105 | 7,0105 | 3,008 |

Градиентный бустинг (GB) демонстрирует ускорение 7,24x при 8 процессах, что указывает на почти линейное масштабирование. Деревья решений (DT) и Случайные леса (RF) показывают почти одинаковое ускорение с максимальным значением 7,01x при 8 процессах. Нейронные сети (NN) демонстрируют более медленное ускорение (3,01x при 8 процессах), что отражает сложность параллелизации обучения нейронных сетей. Несмотря на улучшение скорости, оно не столь драматично, как для моделей на основе деревьев. Все модели, кроме NN, демонстрируют постоянное улучшение ускорения с увеличением количества процессов, что свидетельствует о высокой эффективности параллелизации данных. В случае нейронных сетей может возникать проблема с накладными расходами на коммуникацию между процессами или ограничениями архитектуры при масштабировании.

**Таблица 22.** Значение коэффициента детерминации ($R^2$) для различных алгоритмов при разном количестве процессов:

| Количество процессов | GB | DT | RF | NN |
|---|---|---|---|---|
| 1 | 0.923 | 0.931 | 0.959 | 0.965 |
| 2 | 0.921 | 0.928 | 0.955 | 0.961 |
| 4 | 0.917 | 0.92 | 0.94 | 0.958 |
| 8 | 0.916 | 0.918 | 0.929 | 0.94 |

Градиентный бустинг (GB) показывает небольшое снижение значений $R^2$ по мере увеличения количества процессов, с 0,923 (1 процесс) до 0,916 (8 процессов). Хотя это уменьшение невелико, оно может указывать на незначительное влияние параллелизации на производительность модели. Деревья решений (DT) также демонстрируют небольшое снижение $R^2$ с 0,931 до 0,918 при увеличении числа процессов. Случайные леса (RF) показывают более заметное снижение точности, с 0,959 (1 процесс) до 0,929 (8 процессов), что может свидетельствовать о том, что параллелизация влияет на способность модели к обобщению. Нейронные сети (NN) демонстрируют незначительное снижение $R^2$, с 0,965 до 0,94. Несмотря на это снижение, нейронные сети сохраняют высокую предсказательную способность даже при увеличении числа процессов. Параллелизация приводит к небольшому снижению точности для всех моделей, причём RF страдает больше всего. Следует учитывать компромисс между ускорением и точностью, особенно для более сложных моделей, таких как RF.

Как показали результаты экспериментов, увеличение числа процессов приводит к значительным ускорениям в обучении регрессионных моделей. Однако с ростом

количества процессов наблюдается снижение точности моделей. Это явление можно объяснить уменьшением объема данных, обрабатываемого каждым процессом. При увеличении числа процессов поддатасет, распределяемый на каждый из них, становится меньше, что приводит к недостаточному объему данных для эффективного обучения модели. В результате, уменьшение информации, доступной для каждого процесса, негативно сказывается на качестве обучения и, соответственно, на точности предсказаний модели.

Для анализа данных на основе стахастической модели были разработаны и реализованы алгоритмы машинного обучения с использованием графических процессоров **(GPU).** В рамках исследования применялись следующие методы: случайный лес (Random Forest), градиентный бустинг (Gradient Boosting), и нейронные сети (NN). Дополнительно был рассмотрен алгоритм k-ближайших соседей (K-Nearest Neighbors, KNN) для оценки его производительности на данном датасете. Использование GPU позволило значительно ускорить процесс обучения моделей и обработки данных, что особенно важно при работе с большими наборами данных.

**Таблица 23.** Время обучения регрессионных моделей на GPU, секунд.

| GB | RF | KNN | NN |
|---|---|---|---|
| 1.42 | 0.227 | 0.01 | 6.69 |

Градиентный бустинг (GB) показал время обучения 1.42 секунды, что является относительно быстрым результатом для модели, требующей большого количества вычислений. Случайный лес (RF) оказался ещё быстрее, с временем обучения 0.227 секунды. Это связано с тем, что каждая

модель дерева может быть обучена независимо, что хорошо распараллеливается на GPU. K-ближайших соседей (KNN) продемонстрировал самое короткое время обучения — всего 0.01 секунды. KNN является неконструктивным методом, поэтому время обучения минимально, однако время предсказания на больших датасетах может быть более продолжительным. Нейронные сети (NN) потребовали 6.69 секунд для обучения, что ожидаемо дольше по сравнению с другими методами, так как тренировка нейронных сетей часто требует большего количества итераций и вычислительной мощности.

**Таблица 24.** Значение коэффициента детерминации ($R^2$) для различных алгоритмов на GPU:

| GB | RF | KNN | NN |
|---|---|---|---|
| 0.93 | 0.94 | 0.931 | 0.965 |

Градиентный бустинг (GB) достиг значения $R^2$ 0.93, что свидетельствует о хорошей способности модели предсказывать результаты, хотя это немного ниже по сравнению с другими моделями. Случайный лес (RF) показал $R^2$ 0.94, что чуть выше по сравнению с GB, демонстрируя высокую точность модели. K-ближайших соседей (KNN) продемонстрировал $R^2$ 0.931, что весьма близко к результатам случайного леса и градиентного бустинга, несмотря на существенно меньшее время обучения. Нейронные сети (NN) показали наивысшее значение $R^2$ — 0.965. Это подтверждает высокую способность нейронных сетей к моделированию сложных зависимостей в данных.

Таким образом, можно сделать вот такой вывод:

1. Время обучения: Использование GPU значительно сократило время обучения для всех моделей. Особенно

заметна эффективность для таких моделей, как K-ближайших соседей (KNN) и случайный лес (RF), которые требуют минимального времени на обучение. Нейронные сети занимают больше времени, но всё равно демонстрируют ускорение по сравнению с обучением на CPU.

2. Точность модели ($R^2$): Несмотря на существенную разницу во времени обучения, все модели показывают высокие значения коэффициента детерминации ($R^2$), что указывает на их высокую точность при предсказании. Наиболее точные результаты были достигнуты нейронными сетями (NN), в то время как остальные модели — GB, RF, и KNN — также демонстрируют конкурентоспособные показатели.

3. Рекомендация: Использование GPU для обучения моделей машинного обучения эффективно как с точки зрения времени, так и точности. Модели, такие как нейронные сети, несмотря на более долгое время обучения, обеспечивают максимальную точность. Случайный лес и градиентный бустинг также являются отличными кандидатами для ускоренного обучения на GPU, предоставляя хорошее соотношение между скоростью и точностью.

*Сводка результатов тестирования на CPU (с использованием MPI) и GPU (с использованием CUDA)*

Были проведены два набора экспериментов для оценки производительности обучения моделей машинного обучения на CPU (с использованием MPI для параллелизма) и на GPU (с использованием CUDA). Тестировались следующие модели: градиентный бустинг (GB), случайный лес (RF), нейронные сети (NN), и дополнительно на GPU тестировался алгоритм k-ближайших соседей (KNN). Ниже представлено сравнительное резюме результатов. Время

обучения моделей значительно сократилось при увеличении числа процессов. Например, время обучения градиентного бустинга (GB) уменьшилось с 310,29 секунд (при 1 процессе) до 42,83 секунд (при 8 процессах). Случайный лес (RF) и деревья решений (DT) показали наилучшие ускорения, практически линейно увеличивая производительность с ростом числа процессов. Нейронные сети (NN) также выиграли от параллелизма, однако их ускорение оказалось менее заметным из-за архитектурных ограничений. Использование GPU значительно сократило время обучения. Например, GB обучался всего за 1,42 секунды на GPU по сравнению с 42,83 секундами на 8 процессах CPU. RF и KNN продемонстрировали еще более быстрое обучение — 0,227 секунд и 0,01 секунд соответственно, благодаря эффективной параллелизации на GPU. Нейронные сети (NN) обучались за 6,69 секунд, что хоть и дольше, чем у других моделей на GPU, но всё же существенно быстрее по сравнению с CPU.

Ускорение для обучения на CPU показало эффективное масштабирование. GB достиг ускорения в 7,24 раза при 8 процессах, тогда как RF и DT показали еще более высокие результаты, достигая ускорений до 7,01 раз. Нейронные сети (NN) продемонстрировали более скромное ускорение — всего 3,01 раза при 8 процессах, что, вероятно, связано с накладными расходами на коммуникацию и сложностью архитектуры модели. Использование GPU обеспечило мгновенное ускорение, особенно для моделей RF, KNN и GB. Например, KNN на GPU обучался всего за 0,01 секунд, что является значительным улучшением. Нейронные сети (NN), хоть и показали меньшее ускорение, все же значительно выиграли от параллелизации на GPU по сравнению с их выполнением на CPU.

Все модели сохраняли высокие значения коэффициента детерминации ($R^2$), хотя наблюдалось небольшое снижение

точности с увеличением числа процессов. Например, $R^2$ для GB снизилось с 0,923 до 0,916 при увеличении числа процессов с 1 до 8. Модели RF и NN продемонстрировали стабильные результаты с небольшими вариациями, поддерживая высокую точность. На GPU точность ($R^2$) осталась на высоком уровне для всех моделей. Например, GB показал значение $R^2$ на уровне 0,93, RF — 0,94, а нейронные сети (NN) достигли наивысшего значения $R^2$ — 0,965. Это подтверждает, что обучение на GPU не оказало значительного влияния на точность моделей.

В итоге, несмотря на то, что параллелизация на CPU с MPI даёт хорошие результаты по масштабированию и эффективности, обучение на GPU с CUDA значительно сокращает время обучения, делая его предпочтительным выбором, особенно при работе с большими наборами данных и вычислительно сложными моделями.

В следующем разделе будет рассмотрено проектирование и разработка виртуального прототипа нефтяной скважины, затем его тестирование с использованием современных методов моделирования и анализа.

## 4. Разработка виртуального прототипа нефтяной скважины.

### 4.1. Проектирование архитектуры виртуального прототипа нефтяной скважины.

Проект включает интеграцию технологий цифровых двойников, интернета вещей, промышленной автоматизации, контроллеров, датчиков и киберфизических систем. Был представлен всесторонний обзор предпосылок и контекста проекта, подчеркнувший его соответствие решению проблем в нефтегазовой отрасли. Интеграция технологий цифровых двойников выполнена с высокой степенью точности, обеспечивая отображение физической нефтяной скважины в режиме реального времени. Систематически осуществлялось внедрение устройств интернета вещей, датчиков и систем управления, что позволило эффективно осуществлять сбор данных и управление ими. Создание эффективной киберфизической системы (CPS) обеспечило беспрепятственное взаимодействие между виртуальными и физическими системами, способствуя расширению возможностей по мониторингу и контролю. В рисунке 51 показана архитектура виртуального прототипа нефтяной скважины.

Виртуальный прототип был тщательно разработан и представляет собой цифровой двойник физической нефтяной скважины. Интеграция IoT-устройств, датчиков и систем управления была выполнена с особой тщательностью. Сбор данных включал различные типы данных, чему способствовали датчики и устройства IoT. Обеспечена передача и безопасное хранение данных. Последовательно применялись методы анализа данных, включая алгоритмы предварительной обработки данных и анализа. Была проведена интеграция компонентов

промышленной автоматизации с использованием ПЛК (программируемых логических контроллеров) или других систем управления. Была проведена разработка процедур автоматизации для управления такими аспектами, как работа насосов, управление клапанами и мониторинг давления. Была достигнута реализация киберфизической системы (КФС). Взаимодействие между виртуальным прототипом и физическим маслом было хорошо обеспечено благодаря интеграции датчиков и системы управления.

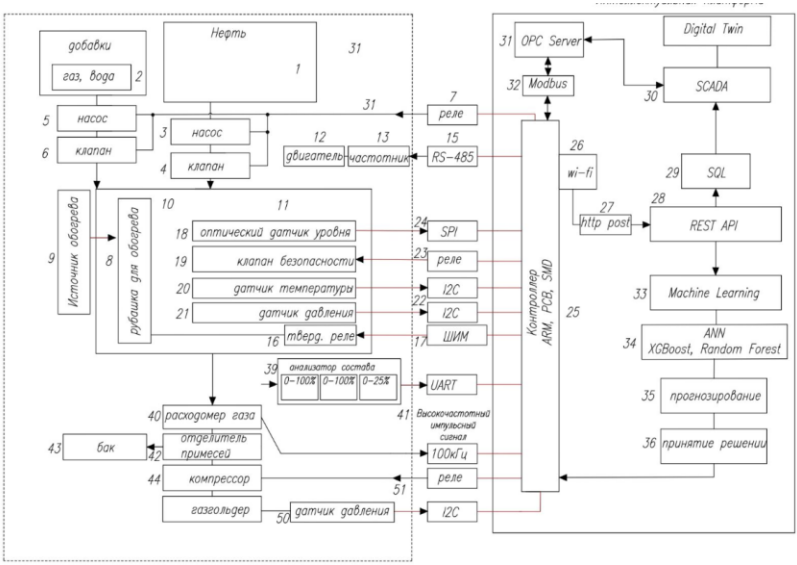



Рисунок 51. Архитектура виртуального прототипа нефтяной скважины

Архитектура цифрового двойника была тщательно разработана. Синхронизация между цифровым двойником и физическим маслом была тщательно отлажена. Поток данных и связь между цифровым двойником и физической системой управлялись тщательно. Интеграция устройств и датчиков Интернета вещей включала использование

определенных аппаратных и программных компонентов. Систематически настраивалась связь между устройствами Интернета вещей, датчиками и цифровым двойником. Постоянно обеспечивалось соответствие протоколам связи и стандартам IoT-коммуникаций. Разработана и реализована архитектура систем промышленной автоматизации и управления. Методически проводилась разработка контуров управления, логики и алгоритмов управления работой виртуальной нефтяной скважины. Тщательно реализовано использование обратной связи с датчиков для принятия управляющих решений в режиме реального времени. Архитектура киберфизической системы (КФС) была тщательно разработана. Систематически были установлены протоколы взаимодействия и связи между виртуальными и физическими системами. Меры безопасности для защиты работы CPS были тщательно продуманы. В рисунке 52 показана система сбора данных и управления виртуального прототипа нефтяной скважины.

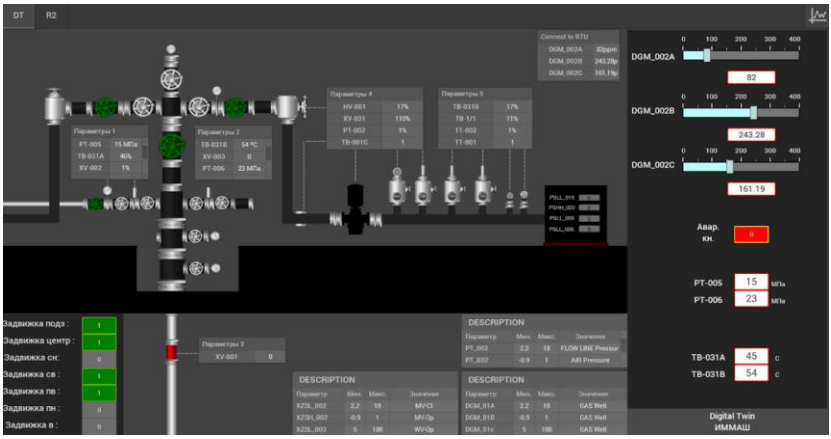



Рисунок 52. Система сбора данных и управления виртуального прототипа нефтяной скважины

Для проекта были специально выбраны и развернуты аппаратные компоненты, включая серверы, датчики и контроллеры. Стек программного обеспечения был тщательно настроен, включая операционные системы, языки программирования и среды разработки. Систематически выполнялась настройка IoT-устройств. Любые проблемы, возникшие при настройке устройства, были эффективно решены. Датчики были точно интегрированы в виртуальный прототип. Систематически выполнялись калибровка, сбор данных и обработка сигналов датчиков. Разработка логики управления и процедур автоматизации проводилась тщательно. Алгоритмы управления систематически кодировались и документировались. Внедрение мер кибербезопасности для защиты виртуального прототипа от потенциальных угроз было тщательно выполнено. Обеспечению конфиденциальности данных и целостности системы уделялось первостепенное внимание.

В рисунке 53 показан блок управления виртуального прототипа нефтяной скважины.

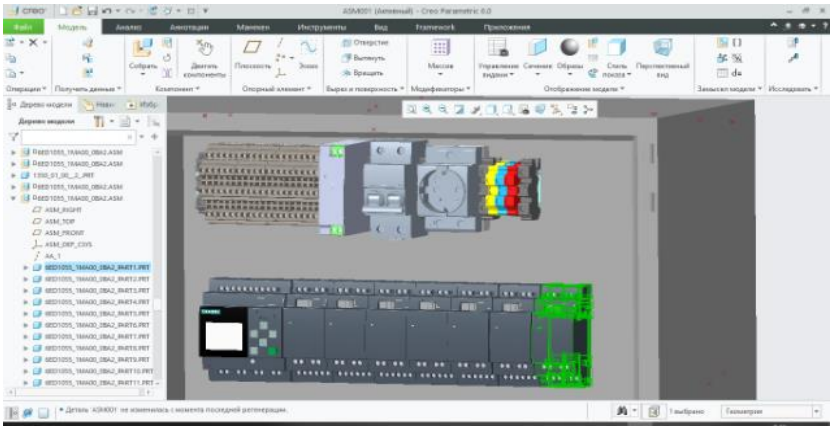


Рисунок 53. Блок управления виртуального прототипа нефтяной скважины

Количественные данные о производительности виртуального прототипа, такие как время отклика, энергоэффективность или надежность системы, были тщательно собраны и проанализированы. Для иллюстрации результатов использовалась визуализация данных с помощью графиков, диаграмм и таблиц. Результаты анализа данных датчиков, подчеркивающие тенденции или аномалии, были тщательно представлены. Данные датчиков сыграли решающую роль в эффективном мониторинге и управлении виртуальной нефтяной скважиной. Проведен сравнительный анализ, сопоставивший достигнутые результаты с первоначальными целями проекта. Было начато всестороннее обсуждение для оценки того, соответствует ли виртуальный прототип указанным целям и требованиям. Была проведена оценка эффективности и результативности виртуальной нефтяной скважины с точки зрения использования ресурсов и общей производительности. Особое внимание было уделено выявлению областей, в которых можно было бы добиться улучшений.

## 4.2. Разработка виртуального прототипа нефтяной скважины.

Разработка виртуального прототипа нефтяной скважины — это инновационный подход к совершенствованию управления нефтяными скважинами, обеспечивающий интеграцию данных в реальном времени и расширенную аналитику для более эффективного принятия решений. Данная задача направлена на революцию в нефтегазовой отрасли путем обеспечения эффективной и устойчивой эксплуатации нефтяных скважин.

Нефтяные скважины являются важнейшим активом в энергетическом секторе, и их оптимальная

производительность имеет решающее значение для успеха отрасли. Для решения задач мониторинга и оптимизации эксплуатации нефтяных скважин был приступлен к разработке испытательного стенда цифрового двойника. На данном испытательном стенде используются передовые технологии для создания виртуальной копии физической нефтяной скважины, что позволяет нам отслеживать, анализировать и оптимизировать ее производительность в режиме реального времени.

Процесс разработки включал несколько ключевых этапов:

- Сбор данных: Были собраны комплексные данные с датчиков и устройств мониторинга, установленных на площадке нефтяной скважины. Эти данные включают температуру, давление, скорость потока и другие соответствующие параметры.
- Создание цифрового двойника. Используя эти данные, была создана модель цифрового двойника, которая воспроизводит поведение физической нефтяной скважины. Цифровой двойник — это сложное моделирование, включающее в себя физические модели и алгоритмы машинного обучения для прогнозирования поведения скважины.
- Мониторинг в реальном времени: Была создана система мониторинга в реальном времени, которая постоянно обновляет цифровой двойник данными из физической скважины. Это гарантирует, что цифровой двойник точно отражает текущее состояние скважины.
- Аналитика и оптимизация. К цифровому двойнику применяются расширенные алгоритмы аналитики и оптимизации, что позволяет нам выявлять потенциальные проблемы, прогнозировать будущую производительность и рекомендовать оптимальные операционные стратегии.

Электрический шкаф управления оснащён автоматизационными компонентами для регулировки и контроля нефтяных процессов. В шкафу установлен контроллер ПР205, который обеспечивает функции программирования и управления для интегрированных систем. К шкафу подключены пневматические цилиндры и регуляторы давления, используемые для точной регулировки потоков воздуха в автоматизированных процессах. В шкафу также присутствуют манометры для отслеживания давления и множество быстроразъемных соединений, обеспечивающих надёжное и удобное подключение к воздушным линиям.

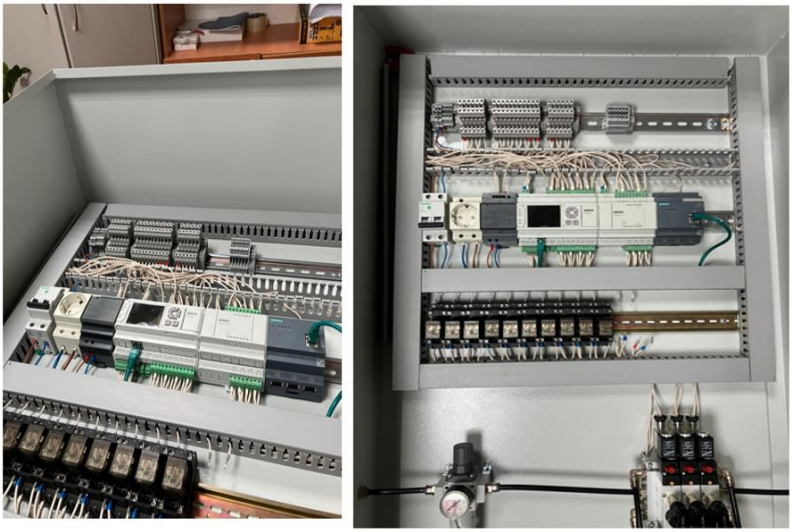


Рисунок 54. Панель электрического шкафа управления

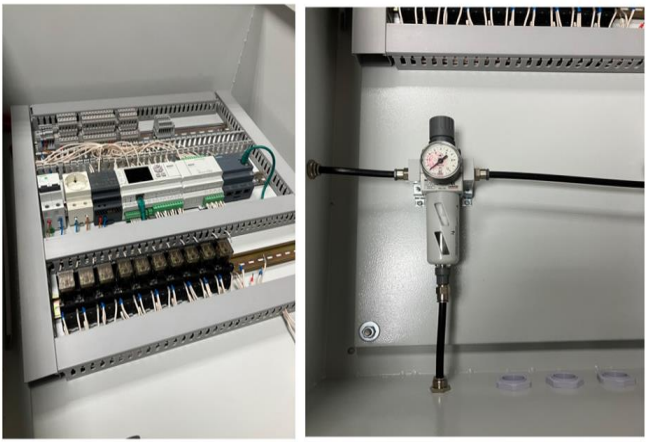


Рисунок 55. Электрический шкаф управления и фильтр-регулятор подачи воздуха

Регулятор давления воздуха с манометром регулирования давления воздуха, поступающего в пневматические цилиндры. Распределительный блок с электромагнитными клапанами, подключенными к электрическим цепям. Пневматический цилиндр подключённый к распределительному шкафу.

Рисунок 56. Соединения регулятора давления воздуха с пневматическим цилиндром

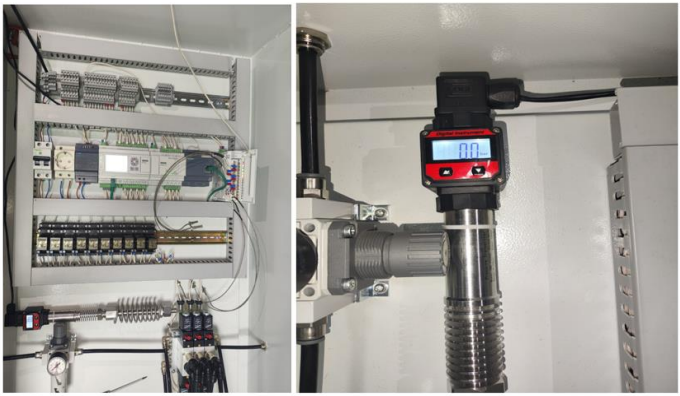


Рисунок 57. Дифференциальный датчик давления

Датчик давления с диапазоном 0-40 бар используется для измерения давления в системах, нефтяные скважины, трубопроводы, резервуары.Датчик постоянно измеряет давление в системе, передавая информацию на контроллер.

MB210-101 – это многофункциональный модуль ввода/вывода, предназначенный для измерения температуры с использованием термопар и передачи данных в систему автоматизации. Модуль поддерживает подключение различных типов термопар (например, типа K, J, T), что позволяет измерять широкий диапазон температур в различных процессах.

Термопары подключаются непосредственно к входам модуля MB210-101, который преобразует их сигналы в цифровые данные для дальнейшей обработки и передачи по сети. Этот модуль поддерживает интерфейс RS-485 и протокол Modbus RTU, что делает его совместимым с большинством систем автоматизации и позволяет интегрировать его с другим оборудованием, таким как контроллеры и панели оператора.

Использование ПО "ОВЕН Конфигуратор" позволяет легко настроить модуль MB210-101 для работы с термопарами. Через конфигуратор можно:

Рисунок 58. Интеллектуальный модуль ввода/вывода

1. Выбрать тип используемой термопары.
2. Настроить адрес модуля для работы по сети RS-485.
3. Определить параметры фильтрации и сглаживания показаний для точных измерений.
4. Проверить состояние подключённых термопар в реальном времени и отладить конфигурацию.

Система на базе MB210-101 и термопар обеспечивает надёжный и точный контроль температуры, что особенно важно в производственных процессах, где требуется постоянное и точное измерение температур для поддержания технологических параметров.

Рисунок 59. Расходомер контроллера расхода газа и жидкости

Alicat — расходомер  контроллера расхода газа и жидкости, которые широко используются в нефтяных скважинах. Расходомеры Alicat работают на основе технологии термальной масс-спектрометрии

На рисунке 60 показано стенд виртуального прототипа нефтяной скважины.

Рисунок 60. Стенд виртуального прототипа нефтяной скважины

Результаты: Испытательный стенд цифрового двойника уже продемонстрировал значительные перспективы в улучшении эксплуатации нефтяных скважин. Ключевые результаты включают в себя:

1. Повышение эффективности. Мониторинг и аналитика в режиме реального времени позволили повысить эффективность эксплуатации скважин, сократить время простоев и повысить производительность.

2. Прогнозируемое обслуживание: цифровой двойник позволяет нам прогнозировать отказы оборудования и заранее планировать техническое обслуживание, сводя к минимуму дорогостоящие простои.

3. Повышенная безопасность. Постоянно контролируя состояние скважин, можно выявлять потенциальные угрозы безопасности и принимать превентивные меры.

4. Воздействие на окружающую среду: алгоритмы оптимизации помогают снизить воздействие на окружающую среду за счет минимизации выбросов и потребления ресурсов.

Разработка испытательного стенда с виртуальным прототипом для мониторинга и оптимизации нефтяных скважин представляет собой значительный прогресс в нефтегазовой отрасли. Этот инновационный подход обещает повысить эффективность, безопасность и устойчивость при одновременном снижении эксплуатационных расходов.

В этом разделе будет рассмотрено тестирование разработанного виртуального прототипа нефтяной скважины, который был создан в рамках проекта по проектированию и разработке цифрового двойника. В процессе тестирования виртуального прототипа будут оценены его функциональные характеристики, производительность и способность моделировать различные сценарии эксплуатации. Симуляции будут проводиться в условиях, приближенных к реальным, с целью выявления возможных недостатков и подтверждения эффективности решений, реализованных на этапе разработки.

Для тестирования виртуального прототипа рассматривается интеграция виртуального датчика на основе машинного обучения (ML) с системами диспетчерского управления и сбора данных (SCADA) для

повышения эффективности предиктивного обслуживания и эксплуатации при мониторинге нефтяных скважин. Виртуальный датчик, построенный на основе моделей машинного обучения, таких как Long Short-Term Memory (LSTM), был обучен на наборе данных из 21 644 эксплуатационных записей. Эти модели предназначены для прогнозирования критического эксплуатационного параметра FlowRate, который необходим для поддержания эксплуатационной целостности и эффективности.

Виртуальный датчик отправляет данные в программируемый логический контроллер (ПЛК) через API. Данные включают прогнозируемые параметры работы скважины, которые используются для дальнейшего управления. ПЛК анализирует поступающие данные и управляет реле на основе предсказаний виртуального датчика, что позволяет контролировать операционные параметры в режиме реального времени. Для ускорения анализа данных в системе предусмотрено устройство с PCB и SMD, оснащенное микроконтроллером RP2040, расположенное в шкафу управления. Это устройство помогает ПЛК в обработке данных и управлении, повышая общую скорость и точность системы. Виртуальный датчик также способен обнаруживать отклонения от нормальных параметров работы скважины, что позволяет предотвратить потенциальные аварийные ситуации на ранней стадии. Эти предиктивные данные, поступающие в ПЛК, способствуют автоматической корректировке работы исполнительных механизмов, таких как клапаны и насосы, обеспечивая безопасную и эффективную эксплуатацию оборудования в любых условиях.

После анализа данных ПЛК передает информацию на OPC сервер через протокол ModBus TCP, что обеспечивает обмен данными между различными промышленными системами. SCADA система получает данные от OPC

сервера для визуализации и мониторинга в реальном времени. Операторы могут отслеживать результаты работы виртуального датчика и управляемые параметры скважины на удобных интерфейсах. Операционные данные из SCADA системы записываются в SQL базу данных для хранения, анализа и последующего использования, что позволяет проводить долгосрочный анализ работы системы и производить отчетность.

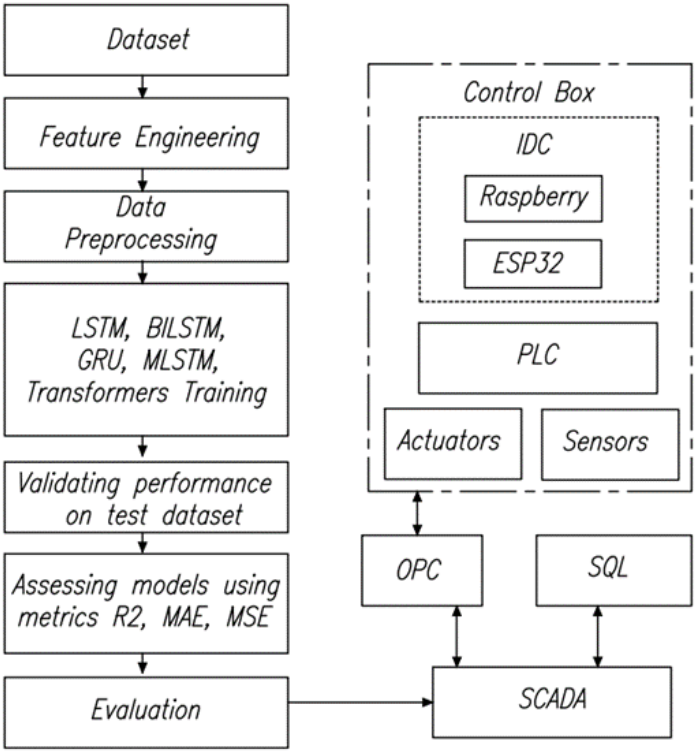



Рисунок 61. Рабочий процесс системы.

Комплексная структура набора данных обеспечивает целостное представление как гидравлических, так и электрических аспектов эксплуатации нефтяных скважин.

Данные использовались для создания и проверки структуры цифрового двойника, направленной на улучшение эксплуатационных прогнозов и принятия решений в реальном времени. В частности, этот набор данных позволяет точно прогнозировать FlowRate и обнаруживать аномалии в CasingPressure. Эти прогнозы имеют решающее значение для оптимизации производительности скважин, особенно в сложных условиях эксплуатации. Кроме того, путем мониторинга CasingPressure и TubingPressure можно реализовать стратегии предиктивного обслуживания, выявляя потенциальные проблемы до того, как они перерастут в простои в работе.

В рисунке 61 представлены ключевые параметры, которые играют важную роль при сборе данных. Цифровая модель системы поддерживается программируемым логическим контроллером (ПЛК) и устройствами, установленными в шкафу управления, такими как микроконтроллеры Raspberry Pi и ESP32, которые отвечают за анализ данных и обмен информацией с виртуальным датчиком. Эти устройства собирают данные от датчиков и передают их на ПЛК для дальнейшего анализа. Данные с ПЛК передаются на OPC сервер по протоколу ModBus TCP, который взаимодействует с системой SCADA для визуализации и мониторинга в реальном времени. Адрес регистрации указывает на место, где данные устройства регистрируются, что обеспечивает точность в отслеживании их местоположения и конфигурации. Значение отображает текущее состояние устройства или передаваемые данные, тогда как тип данных помогает определить формат и характер информации, будь то числовой, текстовый или иной вид данных. Количество указывает на объем передаваемых данных, что необходимо для анализа пропускной способности сети и эффективности работы системы.

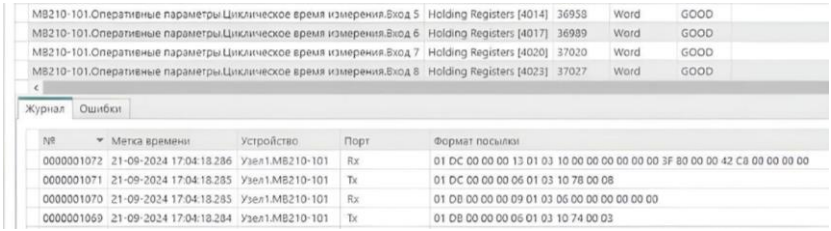



Рисунок 62. Процесс мониторинга параметров

А также метки по времени позволяют синхронизировать данные и следить за временными интервалами событий, что особенно важно для корректного анализа данных. Порт определяет физический или виртуальный канал, через который передаются данные, а формат посылки описывает структуру передаваемой информации, что критически важно для интерпретации и декодирования полученных данных. Эти параметры обеспечивают целостность и достоверность данных при их сборе и обработке.

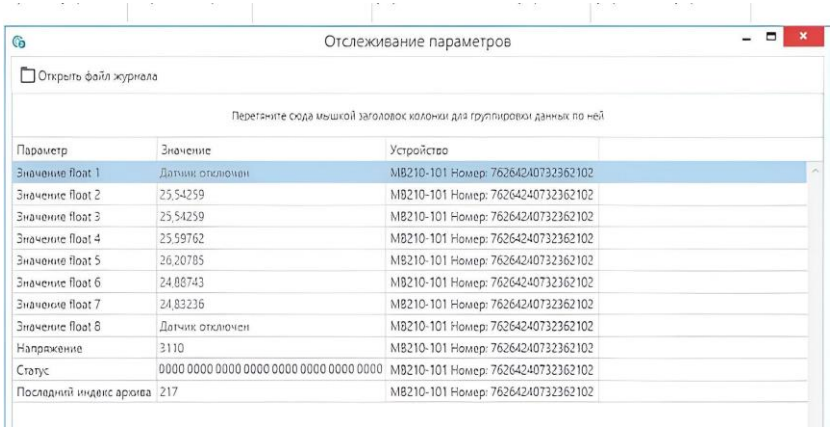



Рисунок 63. Отслеживание параметров во время сбора данных

В дополнение к ранее упомянутым параметрам, на рисунке 27 также показано, какие устройства и датчики были успешно подключены, а какие были отклонены. Это важная информация, поскольку позволяет оперативно оценить состояние системы и выявить возможные неисправности или проблемы с подключением. Устройства, которые были подключены успешно, указывают на их готовность к передаче данных, в то время как отклоненные устройства сигнализируют о сбоях, требующих внимания и устранения. Такая визуализация значительно упрощает процесс мониторинга и контроля работы всех компонентов системы.

В сфере мониторинга нефтяных скважин блок управления выступает в качестве основного компонента, организуя бесшовную интеграцию оборудования и программного обеспечения для эффективного управления и контроля полевых работ. В основе этого сложного аппарата лежит программируемый логический контроллер (ПЛК), который обрабатывает входящие данные с различных датчиков, включая датчики, контролирующие расход и давление, для выполнения логики управления в реальном времени.

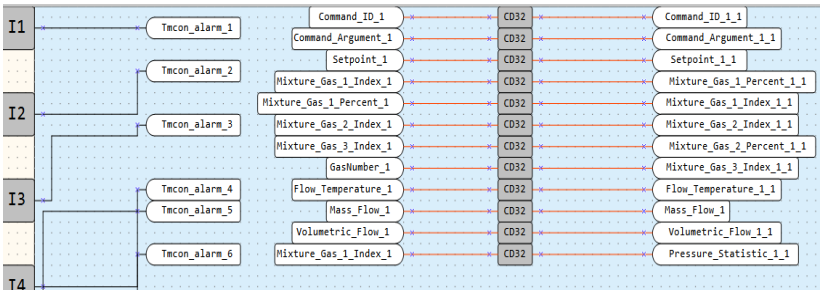



Рисунок 64. Преобразование значений с помощью макросов

Реле в блоке управления имеют решающее значение для операций переключения, позволяя управлять более мощными нагрузками, которые превышают возможности прямого вывода ПЛК. Они дополняются модулями расширения, которые увеличивают диапазон входов и выходов ПЛК, тем самым повышая масштабируемость системы и вмещая более широкий спектр датчиков и механизмов управления. Такая настройка облегчает обработку как аналоговых, так и цифровых сигналов — где аналоговые входы собирают непрерывные данные с датчиков, цифровые входы управляют двоичными сигналами с переключателей, а индикаторы состояния обеспечивают комплексную среду мониторинга.

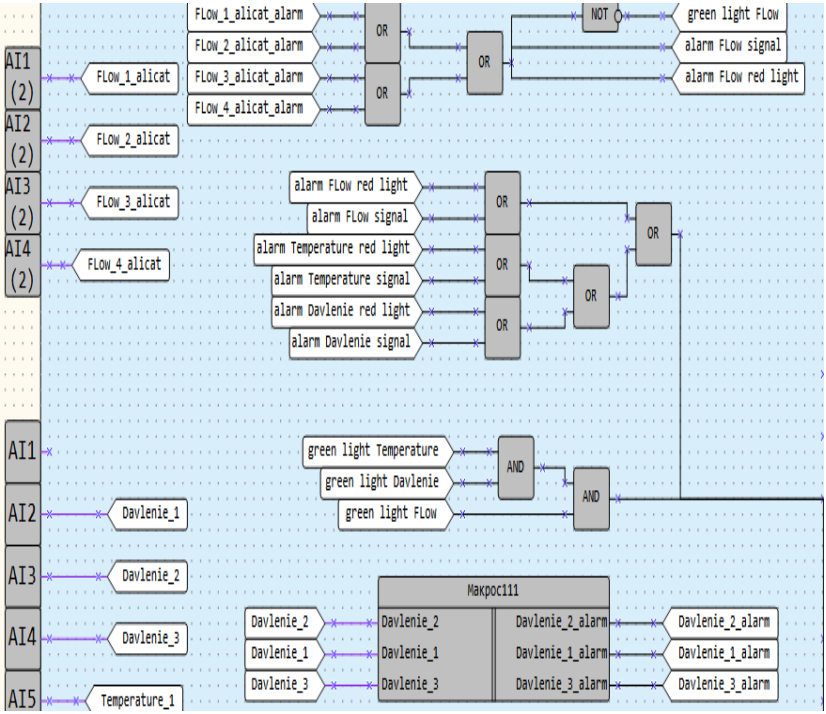



Рисунок 65. Макросы ПЛК

Алгоритм считывания данных процесс считывания данных с аналоговых входов **AI2** и **AI4** от датчиков, которые поступают на два блока **Scale. Scale 1** обрабатывает данные от датчика температуры. **Scale 2** обрабатывает данные от датчика давления.

Высокоскоростные счетные входы в блоке управления тщательно спроектированы для захвата быстрых импульсных сигналов от турбинных счетчиков, которые необходимы для точного измерения расхода. Напротив, выходы широтно-импульсной модуляции (ШИМ) играют важную роль в регулировании работы таких устройств, как насосы, которые управляют динамическими потоками жидкости в скважине, обеспечивая точный контроль над такими критическими рабочими параметрами.

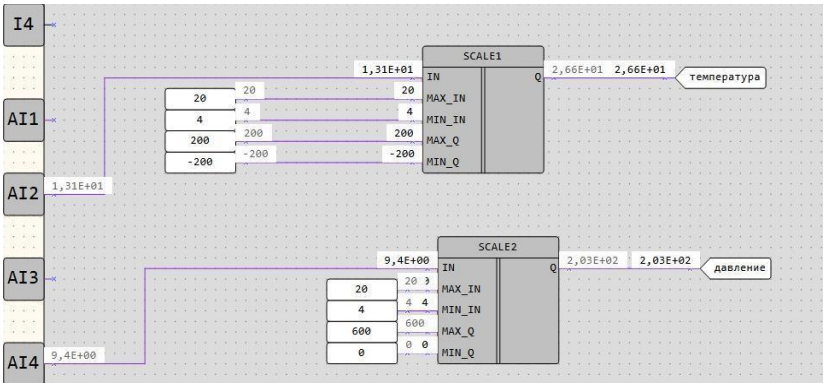



Рисунок 66. Преобразование сигналов в ПЛК

**LIM** (ограничение): Входное значение **IN** проходит через блок, который ограничивает его в пределах заданных минимальных (**MIN_IN**) и максимальных (**MAX_IN**) значений.

**Выражения для пересчета**: Значение сначала уменьшается на **MIN_IN** (отнимается минимальное

значение диапазона). Затем результат делится на разницу между максимальным и минимальным. После этого значение умножается на разницу между **MAX_Q** и **MIN_Q** (это перевод нормализованного значения в новый диапазон, температуры или давления). И, наконец, прибавляется **MIN_Q**, чтобы сдвинуть итоговое значение к нижней границе шкалы выходного диапазона.

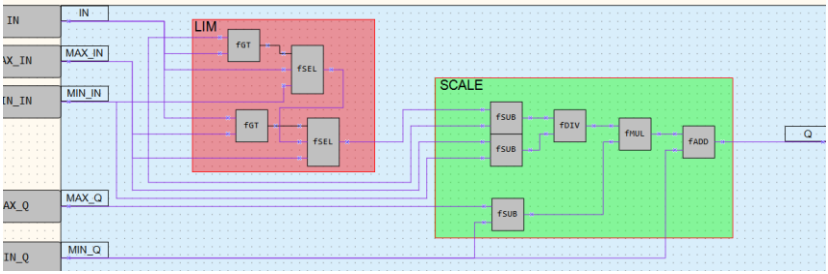



Рисунок 67. Логика алгоритма преобразования полученных данных с помощью макроса Scale

Одной из основных функций блока управления является управление пневматическими клапанами, которые регулируют поток масла и поддерживают давление в безопасных пределах. Эти клапаны регулируются на основе прогнозных данных, предоставляемых передовыми моделями машинного обучения, которые обрабатывают исторические и данные в реальном времени для прогнозирования рабочих параметров. Данная прогнозная возможность позволяет выполнять упреждающие корректировки, оптимизировать производительность и предотвращать потенциальные эксплуатационные опасности. Более того, блок управления усилен Ethernet-подключением и поддерживает протокол Modbus TCP, основной элемент промышленных коммуникаций, что обеспечивает надежную интеграцию с другими промышленными системами и позволяет осуществлять

эффективный удаленный мониторинг и управление. В блок управления встроена специализированная печатная плата, оснащенная SMD-устройством, работающим под управлением операционной системы Linux. Данная конфигурация не только обрабатывает сложные вычисления и обработку данных, но и выступает в качестве важнейшего звена между ПЛК и сетью, обеспечивая стабильную и универсальную платформу для разработки пользовательских приложений и интеграции алгоритмов машинного обучения.

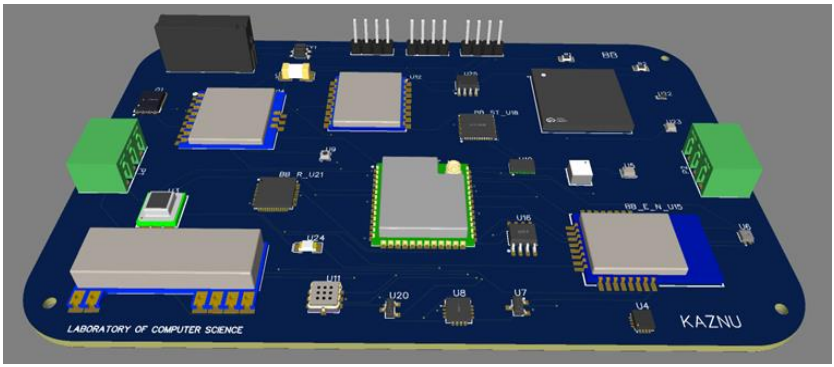



Рисунок 68. Интеллектуальное устройство для сбора данных

Устройство PCB SMD, тщательно спроектированное для блока управления, включает в себя передовые компоненты, такие как Raspberry Pi Pico 2040 и Espressif ESP32. Эти микроконтроллеры предлагают надежные возможности обработки и гибкость для различных задач мониторинга. Raspberry Pi Pico 2040, построенный на архитектуре ARM Cortex-M0+, эффективно обрабатывает сложные вычисления, в то время как ESP32 облегчает подключение по Wi-Fi, обеспечивая бесперебойную связь TCP/IP. Данная настройка обеспечивает надежный сбор данных от массива датчиков, стратегически расположенных по всей системе

нефтяных скважин. Интеграция подключения по Wi-Fi позволяет передавать данные в режиме реального времени, повышая отзывчивость системы и обеспечивая немедленную операционную корректировку на основе данных. Эта сложная электронная структура имеет решающее значение для оптимизации операционной эффективности и надежности мониторинга нефтяных скважин, отражая передовую технологическую интеграцию, характерную для современных промышленных систем.

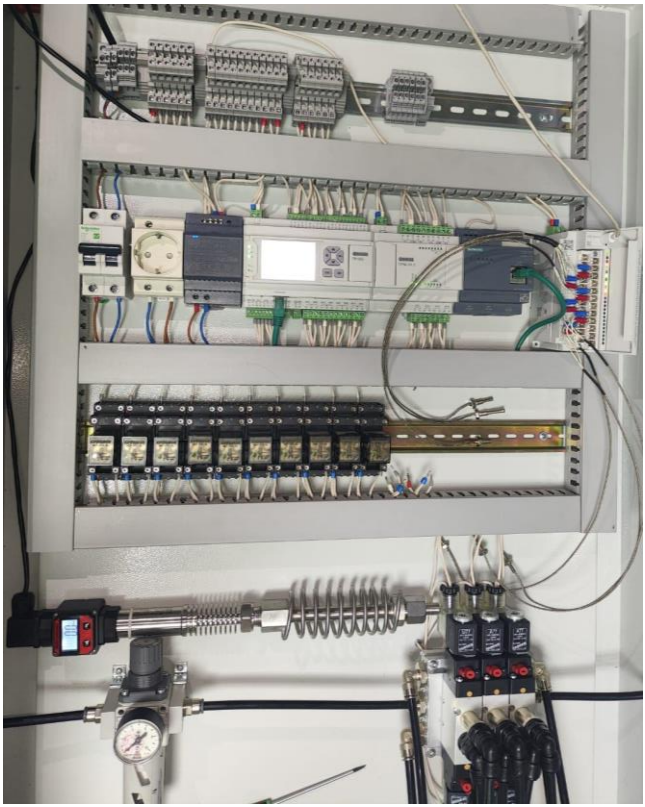


Рисунок 69. Шкаф управления

Шкаф управления включает в себя несколько ключевых компонентов, которые обеспечивают эффективное функционирование системы мониторинга и управления нефтяной скважиной. Основные элементы шкафа — это программируемый логический контроллер (ПЛК) ПР205, модуль расширения ПРМ24.2 и модуль ввода-вывода МВ210. Эти устройства работают в связке с различными датчиками, такими как датчики температуры и давления, что позволяет системе собирать и обрабатывать данные в режиме реального времени. Программное обеспечение, используемое для настройки системы, обеспечивает гибкость в управлении, позволяя настраивать алгоритмы и взаимодействие компонентов.

Для управления исполнительными механизмами, такими как насосы и клапаны, используется реле DPDT с током до 10А и катушкой на 24VDC, что обеспечивает надежное переключение цепей. ПЛК ПР205 принимает сигналы через транзисторные входы, что повышает быстроту и точность обработки данных. ModBus TCP по протоколу Ethernet используется для передачи данных между ПЛК и сервером, а также для обмена информацией с виртуальным датчиком. Этот протокол поддерживает взаимодействие через модель OSI, что обеспечивает надежную и быструю передачу пакетов данных, а также стабильное соединение с сервером. Для повышения надежности системы и предотвращения сбоев в работе используются специальные модули расширения с гальванической развязкой, которые добавляют отказоустойчивость в условиях электромагнитных помех. ПЛК ПР205 также спроектирован с учетом требований к работе в сложных промышленных условиях, что гарантирует его отказоустойчивость и устойчивость к электромагнитным полям, характерным для производственных площадок.

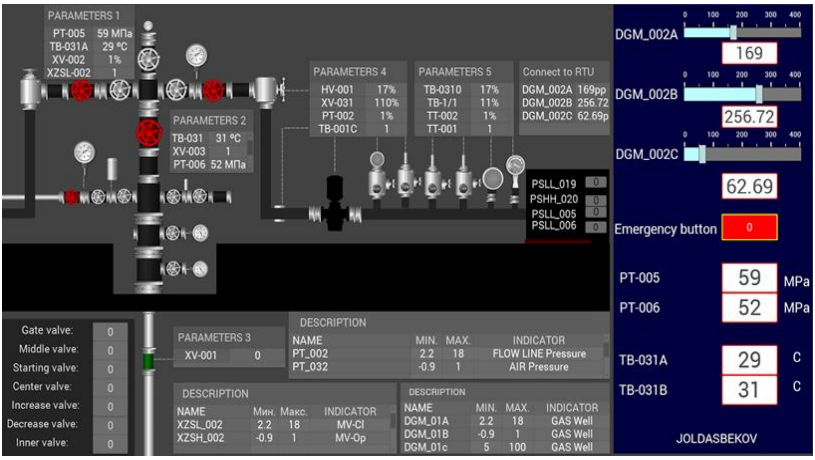



Рисунок 70. Мониторинг и управление в режиме реального времени

Виртуальный прототип системы включает в себя сбор данных с помощью датчиков, их обработку и передачу через контроллер в систему SCADA, и последующее управление через алгоритмы Python. Этот процесс обеспечивает эффективный мониторинг и управление нефтяными процессами, что критично для обеспечения безопасности и эффективности производства. Схема описывает критическую роль системы SCADA в мониторинге и управлении операциями на нефтяных скважинах. Интегрированная с базой данных SQL и сервером OPC, система эффективно управляет основными параметрами, такими как расход и давление, для обеспечения эксплуатационной эффективности и безопасности. Система SCADA использует прогнозируемые значения из внешних моделей машинного обучения для оптимизации ответов и действий, повышая ее способность поддерживать эксплуатационную целостность. Данная настройка подчеркивает стратегическую

функциональность системы и надежную связь при управлении операциями на нефтяных скважинах.

Для подготовки данных для обучения моделей машинного обучения данные были разделены на три основных набора. Для обучения моделей использовался обучающий набор данных, составлявший 80% от исходного набора данных. Для настройки гиперпараметров и предотвращения переобучения использовался проверочный набор данных, составлявший 15% от обучающего набора данных с использованием параметра разделения проверки. Оставшиеся 20% исходного набора данных были выделены в тестовый набор данных и использовались для оценки окончательной производительности модели.

Предварительная обработка данных включала обработку пропущенных значений, которые были заменены средними значениями числовых признаков, и нормализацию значений признаков с использованием метода Min-Max таким образом, чтобы значения признаков находились в диапазоне от 0 до 1. Каждая модель была обучена с использованием следующей конфигурации: 10 эпох, размер партии 64 и подробный параметр 1 для вывода прогресса обучения. Для проверки использовался набор данных, содержащий 15% набора данных обучения. Каждая модель была составлена с использованием следующей конфигурации: функция потерь среднеквадратической ошибки (mean_squared_error), оптимизатор Adam со скоростью обучения 1e-4 и метрика оценки $R^2$. Модели хорошо справились с обоими тестовыми наборами данных. Все модели достигли значений $R^2$ выше 0,96 на тестовом наборе данных, что указывает на то, что они смогли объяснить большую часть дисперсии целевой переменной. Momentum LSTM и Bidirectional LSTM справились лучше всего, со значениями $R^2$, очень близкими к 1. Кроме того,

модели GRU и standard LSTM показали уровень производительности не хуже, чем MLSTM и BILSTM.

Трансформеры показали коэффициент детерминации $R^2$ 0,9685, что хуже рекуррентных моделей, что объясняется последовательным характером данных, благоприятным для рекуррентных сетей. Однако результаты трансформеров остаются конкурентоспособными. Сравнение LSTM, BiLSTM, MLSTM и GRU показало их высокую производительность с $R^2$ в диапазоне 0,9720–0,9726, что указывает на точные прогнозы. Средняя абсолютная ошибка (MAE) для всех моделей составляет около 0,0089, а средняя квадратичная ошибка (MSE) около 0,0001, что подчеркивает их хорошую предсказательную способность. LSTM и её варианты предпочтительны для данной задачи прогнозирования по сравнению с трансформерами.

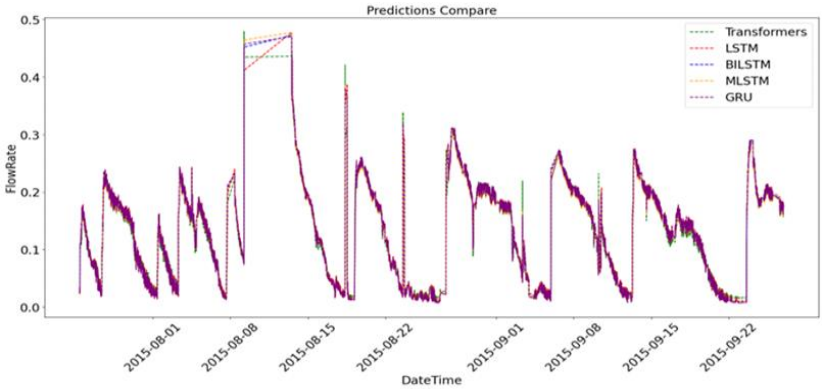



Рисунок 71. Сравнительный график результатов всех алгоритмов

Исследование показало, что LSTM и ее варианты, включая BiLSTM и GRU, продемонстрировали превосходную производительность в прогнозировании эксплуатационных параметров нефтяных скважин, при этом значения $R^2$ постоянно превышали 0,97. Это указывает

на высокий уровень точности моделирования и прогнозирования важнейших эксплуатационных показателей, что необходимо для эффективного управления скважинами. Модель Transformer, хотя и немного менее эффективная со значением $R^2$ 0,9685, все же продемонстрировала существенные прогностические возможности, что свидетельствует о ее потенциальной полезности в сценариях, требующих обработки больших наборов данных со сложными шаблонами.

Более того, внедрение этих моделей МО в системы SCADA способствовало улучшенному принятию решений в реальном времени. Интегрируя предиктивные модели непосредственно в операционные рабочие процессы, системы могли заранее подстраиваться под изменяющиеся условия, тем самым предотвращая потенциальные сбои и оптимизируя эффективность производства. Эта интеграция не только подчеркивает надежность моделей машинного обучения в операционных условиях, но и подчеркивает развивающийся ландшафт промышленных систем мониторинга, где основанные на данных идеи становятся центральными для операционных стратегий.

Интеграция моделей машинного обучения в системы SCADA облегчила принятие решений в режиме реального времени, что позволило системам проактивно адаптироваться к изменяющимся условиям. Например, использование прогностических моделей позволило проактивно настраивать пневматические, электромагнитные клапаны и другие механизмы управления, тем самым поддерживая давление в безопасных пределах и обеспечивая оптимальные скорости потока. Кроме того, исследование подчеркивает надежность и универсальность рекуррентных моделей нейронных сетей при обработке данных временных рядов. Немного более низкая производительность модели

Transformer указывает на то, что, хотя она является мощным инструментом для определенных приложений, последовательный характер данных о добыче нефтяных скважин благоприятствует использованию рекуррентных моделей.

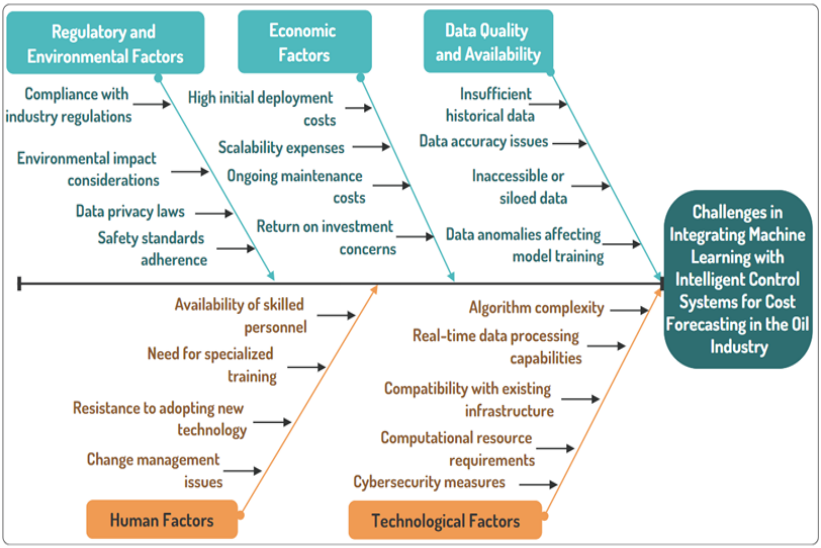



Рисунок 72. Анализ эффективности интеграции машинного обучения с интеллектуальными системами управления для прогнозирования затрат в нефтяной промышленности в соответствии с диаграммой Ишикавы

Мы провели аналитическое исследование с использованием диаграммы Ишикавы для систематической визуализации факторов, влияющих на интеграцию машинного обучения с интеллектуальными системами управления для прогнозирования затрат в нефтяной промышленности. Диаграмма облегчила выявление и организацию потенциальных причин проблем, связанных с этим процессом интеграции.

Хотя результаты многообещающие, это исследование имеет несколько ограничений, которые заслуживают

рассмотрения. Набор данных, хотя и полный, ограничен конкретными эксплуатационными параметрами из одного источника, что может не полностью отражать изменчивость и сложность операций на нефтяных скважинах по всему миру. Это может повлиять на обобщаемость результатов. Это подчеркивает необходимость более обширного тестирования невидимых данных и методов перекрестной проверки. Внедрение этих моделей в режиме реального времени в реальных полевых операциях представляет технические и логистические проблемы, поскольку для этого требуются надежные инфраструктуры для развертывания и интеграции в существующие системы SCADA. Более того, вычислительные ресурсы, необходимые для обучения и развертывания этих передовых моделей машинного обучения, значительны, что может ограничить их практичность в средах с ограниченными ресурсами.

Кроме того, важно признать ограничения использования набора данных, охватывающего 21 644 записи из конкретной нефтяной скважины. Этот ограниченный объем может не полностью отражать огромную неоднородность и динамичную природу операций на нефтяных скважинах в различных геологических формациях и в различных условиях.

Будущие исследования должны устранить эти ограничения, включив более разнообразные наборы данных, протестировав производительность моделей в различных условиях эксплуатации, разработав эффективные фреймворки развертывания и оптимизировав модели для повышения вычислительной эффективности. Кроме того, решение проблем смещения данных и дрейфа концепций с помощью таких методов, как непрерывное обучение и трансферное обучение, имеет решающее значение для обеспечения долгосрочной надежности и

обобщаемости прогностических моделей на основе МО в реальных приложениях нефтяных скважин.

Сбор и передача данных: Датчики давления и температуры (использовались датчики с диапазоном 0-40 бар и термопары типа K, J, T) собирают данные о критических параметрах скважины. Эти данные передаются на контроллер ПР205, который далее направляет их на OPC сервер.

Обработка данных: На контроллере с помощью ПО «OWEN Logic» осуществляется первичная обработка данных, включая преобразование сигналов с датчиков в цифровые данные. Это позволяет точно контролировать процессы, с помощью макроса Scale в ПО.

Интеграция с SCADA: Данные поступают в SCADA систему, где операторы в реальном времени могут наблюдать и анализировать все параметры работы скважины. Программное обеспечение Simple Scada 2 позволяет визуализировать эти данные на графическом интерфейсе.

Управление процессом: Python алгоритмы используются для дальнейшего управления системой, анализируя данные из SCADA и оптимизируя работы системы в соответствии с анализом данных.

Результаты тестирования: Тест виртуального прототипа системы мониторинга и управления нефтяной скважины показали значительный прогресс в возможностях улучшения эксплуатации нефтяных процессов. Ключевые достижения включают:

Эффективность сбора данных: Датчики и контроллер успешно справились с задачей точной передачи данных о давлении и температуре.

Точность обработки данных: Программирование на ПО «OWEN Logic» и использование макросов позволили точно откалибровать и преобразовать аналоговые сигналы в

цифровые, что обеспечило высокую точность данных для SCADA системы.

Надежность SCADA системы: SCADA система показала высокую надежность в отображении данных, помогая операторам в мониторинге и управлении процессами.

Управление процессом: Python алгоритмы демонстрировали эффективное управление, оптимизируя операции в зависимости от анализа входящих данных.

Тестирование виртуального прототипа нефтяной скважины успешно завершено. Все компоненты системы — от датчиков до алгоритмов управления — продемонстрировали высокую эффективность и надежность. Виртуальный прототип готов к применению в реальных условиях эксплуатации, обещая значительное улучшение управления нефтяными скважинами.

## Заключение

В ходе проведенных исследований были достигнуты значительные результаты в четырех ключевых направлениях: разработка математической модели, применение методов машинного обучения, использование высокопроизводительных вычислений (HPC) и создание цифрового двойника. В первой части была разработана и улучшена математическая модель, которая стала основой для дальнейших вычислительных экспериментов. Во второй части был успешно применен машинный интеллект для оптимизации сложных вычислительных процессов, что позволило повысить точность прогнозов и снизить вычислительные затраты. Третья часть касалась применения HPC для решения задач с высокой вычислительной сложностью, что значительно увеличило скорость расчетов и масштабируемость моделей. В заключительной части был создан цифровой двойник для интеграции и моделирования сложных систем в реальном времени, что открывает возможности для улучшения мониторинга и управления системами. В данном заключении подведено итоги каждой из этих областей и подчеркнем их важность и потенциальную пользу.

В рамках первой области проведенного исследования были рассмотрены и проанализированы четыре основные модели, применяемые для описания динамики фильтрационных процессов в пористых средах: модель Бакли-Леверетта, Black Oil, композиционная и стохастическая модели. Каждая из этих моделей играет важную роль в решении задач неравновесной фильтрации, предоставляя различные уровни детализации и учета фазовых переходов, многокомпонентных взаимодействий и стохастической неопределенности. Модель Бакли-Леверетта предложила аналитический подход для описания

двухфазного потока, тогда как модель Black Oil обеспечила основу для синтетических данных и симуляции нефтяных резервуаров. Композиционная модель позволила учесть влияние различных компонентов на поведение системы во времени, а стохастическая модель внесла элементы неопределенности, повышая точность при моделировании неоднородных сред. Таким образом, использование данных моделей позволяет комплексно и более точно моделировать процессы фильтрации и динамики насыщенности в геологических системах, что способствует улучшению методов прогнозирования и принятия решений в инженерной практике

В рамках второй области данного исследования станет последовательное применение методов машинного обучения для оптимизации процессов добычи нефти и газа. В первом году основное внимание уделяется модели Бакли-Леверетта, что позволит исследовать процесс вытеснения жидкости из резервуара и анализировать влияние таких параметров, как проницаемость и вязкость, на эффективность вытеснения. Разработан интегрированный инструмент для моделирования процессов и анализа результатов в реальном времени. Во второй год был достигнут значительный прогресс в разработке интеллектуальных вычислительных моделей для прогнозирования коэффициентов извлечения нефти. Использование машинного обучения, таких как деревья решений, случайные леса, градиентный бустинг и искусственные нейронные сети, покажет высокую точность в предсказании ключевых параметров модели. Основное внимание уделяется синтетическому набору данных, содержащему более 369 600 точек данных, что позволит моделировать различные сценарии поведения нефтяного пласта.. Применение алгоритмов машинного обучения позволит эффективно выявлять взаимосвязи между

переменными, такими как давление, пористость и коэффициенты нефте- и газоотдачи. В ходе сравнительного анализа будет установлено, что градиентный бустинг и XGBoost демонстрируют наивысшую точность, что подтверждается метриками $R^2$, а нейронная сеть, построенная с использованием пяти слоев, также покажет высокий уровень точности. На третьем году акцент будет сделан на стохастическом моделировании с применением машинного обучения, что позволит более точно прогнозировать поведение резервуаров и их реакцию на изменения условий эксплуатации. Используется композиционная модель для сочетания результатов предыдущих лет и создания комплексного инструмента, способного адаптироваться к разным условиям, предлагая оптимальные стратегии разработки месторождений на основе анализа данных. Несмотря на достигнутые результаты, статическая природа текущих моделей оставляет пространство для дальнейших исследований, в частности, изучение динамических моделей, которые могут учитывать временные изменения и динамику процесса добычи. Данный подход представляет собой мощный инструмент для повышения точности прогнозирования и оптимизации работы нефтяных резервуаров, что может быть полезно в практических применениях.

В третьей части работы были использованы алгоритмы высокопроизводительных вычислений (HPC) и машинного обучения для разработки и внедрения интеллектуальных высокопроизводительных вычислительных моделей для решения задач, связанных с моделированием модели Бакли-Леверетта, Black Oil, композиционными моделями и стохастическим моделированием. В рамках задачи были применены параллельные алгоритмы случайного леса (RF), градиентного бустинга (GB) и деревьев решений (DT) с использованием стандарта MPI, что позволило добиться

значительного ускорения обучения моделей с минимальными потерями точности. Алгоритмы были оптимизированы для работы на графических процессорах (GPU) с использованием технологий CUDA и библиотек RAPIDS, что привело к заметному уменьшению времени обработки больших объемов данных. Можно отметить, что разработка и реализация интеллектуальных высокопроизводительных вычислительных моделей на основе методов машинного обучения с применением как параллелизации с помощью технологии MPI, так и графических процессоров (GPU), позволили значительно повысить эффективность и производительность вычислений. Параллельное обучение моделей, реализованное с использованием библиотеки mpi4py, показало хорошие результаты как в плане ускорения процессов, так и сохранения точности при увеличении числа процессов. При этом было продемонстрировано, что методология параллелизма данных и моделей обеспечивает высокий уровень масштабируемости без существенных потерь в точности прогнозов.

Использование GPU, в свою очередь, показало себя как мощный инструмент для ускорения машинного обучения, особенно при работе с большими объемами данных. Применение методов машинного обучения и нейронных сетей, с использованием библиотек cuML и TensorFlow, позволило существенно сократить время обучения и повысить производительность моделей. GPU, благодаря своей архитектуре и параллельной обработке, доказали свою эффективность в задачах, требующих больших вычислительных ресурсов. Совокупное применение MPI и GPU открыло новые возможности для эффективной и быстрой обработки данных, предоставив гибкие и масштабируемые решения для задач машинного обучения. Полученные результаты показывают, что такие технологии

являются важными инструментами для решения современных сложных вычислительных задач, обеспечивая как высокую точность, так и значительное ускорение процессов.

Таким образом, внедрение HPC-алгоритмов в комбинации с методами машинного обучения не только значительно ускоряет вычислительные процессы, но и предоставляет возможность масштабирования решений для работы с большими объёмами данных. Эти технологии представляют собой важный шаг вперёд в области высокопроизводительных вычислений и машинного обучения, предлагая эффективные и точные решения для решения сложных задач в различных научных и инженерных областях.

В заключительной части работ по проектированию, разработке и тестированию виртуального прототипа нефтяной скважины была проделана значительная работа, направленная на создание инновационного решения для мониторинга и оптимизации работы нефтяных скважин. На этапе проектирования была разработана архитектура системы, включающая интеграцию технологий цифровых двойников, моделей машинного обучения, интернета вещей и киберфизических систем, что обеспечило синхронизацию между виртуальным прототипом и физической скважиной в режиме реального времени. В процессе разработки была создана модель цифрового двойника, способная воспроизводить поведение физической скважины, с использованием данных, полученных от датчиков и IoT-устройств. Разработанная система мониторинга в реальном времени обеспечила актуальность данных, что в свою очередь позволило применять аналитические алгоритмы для прогнозирования производительности и выявления потенциальных проблем.

На этапе тестирования системы были проведены всесторонние испытания, которые подтвердили высокую эффективность сбора и обработки данных, надежность работы системы SCADA и успешность применения алгоритмов управления на Python для оптимизации процессов. Результаты тестирования продемонстрировали улучшение эксплуатационных характеристик нефтяной скважины, включая повышение эффективности, снижение простоев, прогнозирование отказов и уменьшение воздействия на окружающую среду. Это исследование демонстрирует эффективность интеграции современных моделей машинного обучения, в частности архитектур на основе LSTM, с системами SCADA для предиктивного обслуживания и оптимизации эксплуатации в операциях на нефтяных скважинах. Модели LSTM, BiLSTM и GRU последовательно достигают значений $R^2$ выше 0,97, что подчеркивает их превосходную производительность в точном прогнозировании расхода и обнаружении аномалий.

Успешное тестирование этих моделей машинного обучения в системы SCADA способствовало более эффективному принятию решений в реальном времени, значительно сократив время простоя и оптимизировав эффективность тестируемого объекта. Результаты исследования подчеркивают необходимость интеграции таких прогностических моделей в нефтегазовую отрасль для повышения общей производительности работы. Нефтегазовым компаниям следует рассмотреть возможность внедрения современных моделей машинного обучения для улучшения своих систем SCADA. Используя возможности прогнозирования моделей LSTM, BiLSTM и GRU, компании могут добиться более точного прогнозирования потока и своевременного обнаружения аномалий, что приведет к улучшению стратегий обслуживания и оптимизации процессов. Рекомендуется

инвестировать в необходимую вычислительную инфраструктуру и экспертизу для эффективного внедрения и поддержки этих моделей.

Таким образом, проект виртуального прототипа нефтяной скважины стал важным шагом вперед в нефтегазовой отрасли, обеспечивая основу для более устойчивого и эффективного управления нефтяными ресурсами.

Все проведенные исследования демонстрируют важность инноваций и применения современных технологий в различных областях.

**Список использованных литератур**

[1] Larry C. Young. Compositional Reservoir Simulation: A Review, SPE J. 27 (05): 2746-2792, 2022. http://doi.org/10.2118/208610-PA

[2] John Smith, A new Compositional Simulation Approach to Model Recovery, Journal of Natural Gas Science and Engineering, 2012.

[3] Gang Wang et al., Detailed Assessment of Compositional and Interfacial Tension Effect on the Fluid Behaviour During Immiscible and Near-Miscible CO2 Continuous and WAG Displacements, SPE Journal, 2018.

[4] Najeeb S.Alharthy, Compositional Modeling of Multiphase Flow and Enhanced Oil Recovery Prospects in Liguid-Rich Unconventional Reservoirs, Colorado School of Mines, 2015.

[5] Kumar, A., & Giddey, S., Multiphase Flow Modeling with Compositional Methods in Tight Oil Reservoir, SPE Reservoir Simulation Conference, Society of Petroleum Engineers, 2017.

[6] Sylvain Pasquier, Michel Quintard, and Yohan Davit, Modeling Two-Phase Flow of Immiscible Fluids in Porous Media: Buckley-Leverett Theory with Explicit Terms, Physical Review Fluids, 2017.

[7] Dominique Guerillot, Mostafa Kadiri, and Saber Trabelsi, Buckley-Leverett Theory for Two-Phase Immiscible Fluids Flow Model with Explicit Phase-Coupling Terms, Water, 2020.

[8] S. Geiger, S. Matthai, J. Niessner, and R. Helmig, Black-Oil Simulations for Three-Component, Three-Phase Flow in Fractured Porous Media, SPE Journal, 2009.

[9] Zhangxin Chen, Formulations and Numerical Methods of the Black Oil Model in Porous Media, SIAM Journal on Numerical Analysis, 2001.

[10] D. Zhang, Stochastic Methods for Flow in Porous Media: Coping with Uncertainties. Academic Press, 2001.

[11] C. E. Auger, G. C. Buscaglia, P. A. Briozzo. Stochastic Modeling of Multiphase Flow in Reservoir Simulation:

Addressing Uncertainty in Reservoir Properties. Computational Geosciences, 2018.

[12] True Tamplin, Stochastic modeling: Meaning, Application, Benefits, Drawbacks, Finance Strategists, 2023.

[13] Liu, Y., &Reynolds, A. C. Ensemble Kalman filter for automatic history matching of geologic models with faults. SPE Journal, 13(4), 458-470. Society of Petroleum Engineers. 2008

[14] Shirazi, A.; Hezarkhani, A.; Beiranvand Pour, A.; Shirazy, A.; Hashim, M. Neuro-Fuzzy-AHP (NFAHP) Technique for Copper Exploration Using Advanced Spaceborne Thermal Emission and Reflection Radiometer (ASTER) and Geological Datasets in the Sahlabad Mining Area, East Iran. *Remote Sens.* **2022**, *14*, 5562.

[15] Shirazy, A.; Hezarkhani, A.; Timkin, T. Investigation of Magneto-/Radio-Metric Behavior in Order to Identify an Estimator Model Using K-Means Clustering and Artificial Neural Network (ANN) (Iron Ore Deposit, Yazd, IRAN). *Minerals* **2021**, *11*, 1304.

[16] Johnson, E.; Obot, O.; Attai, K.; Akpabio, J.; Inyang, U. The Use of Machine Learning in Oil Well Petrophysics and Original Oil in Place Estimation: A Systematic Literature Review Approach. *J. Eng. Res. Rep.* **2023**, *25*, 40–54.

[17] Koray, A.-M.; Bui, D.; Ampomah, W.; Appiah Kubi, E.; Klumpenhower, J. *Application of Machine Learning Optimization Workflow to Improve Oil Recovery*; OnePetro: Richardson, TX, USA, 2023.

[18] Helland, J.O.; Friis, H.A.; Assadi, M.; Nagy, S. Machine Learning for Underground Gas Storage with Cushion CO2 Using Data from Reservoir Simulation. *IOP Conf. Ser. Mater. Sci. Eng.* **2023**, *1294*, 012058.

[19] Mousavi, S.M.; Bakhtiarimanesh, P.; Enzmann, F.; Kersten, M.; Sadeghnejad, S. Machine-Learned Surrogate

Models for Efficient Oil Well Placement Under Operational Reservoir Constraints. *SPE J.* **2024**, *29*, 518–537.

[20] Samnioti, A.; Gaganis, V. Applications of Machine Learning in Subsurface Reservoir Simulation—A Review—Part II. *Energies* **2023**, *16*, 6727.

[21] Chen, X.; Zhang, K.; Ji, Z.; Shen, X.; Liu, P.; Zhang, L.; Wang, J.; Yao, J. Progress and Challenges of Integrated Machine Learning and Traditional Numerical Algorithms: Taking Reservoir Numerical Simulation as an Example. *Mathematics* **2023**, *11*, 4418.

[22] Kazemi, M.; Takbiri-Borujeni, A.; Nouroizeh, H.; Kazemi, A.; Takbiri, S.; Wallrich, C. A Novel Surrogate Model for Reservoir Simulations Using Fourier Neural Operators. In Proceedings of the SPE Annual Technical Conference and Exhibition? San Antonio TX, USA, 9 October 2023; OnePetro: Richardson, TX, USA, 2023.

[23] Faraji MA, Shooshtari A, El-Hag A. Stacked Ensemble Regression Model for Prediction of Furan. Energies. 2023; 16(22):7656. https://doi.org/10.3390/en16227656

[24] Yang Y, Tan C, Cheng Y, Luo X, Qiu X. Using a Deep Neural Network with Small Datasets to Predict the Initial Production of Tight Oil Horizontal Wells. Electronics. 2023; 12(22):4570. https://doi.org/10.3390/electronics12224570

[25] S. R. Upadhyaya, “Parallel approaches to machine learning—A comprehensive survey,” *J. Parallel Distrib. Comput.*, vol. 73, no. 3, pp. 284–292, Mar. 2013, doi: 10.1016/j.jpdc.2012.11.001.

[26] J. Verbraeken, M. Wolting, J. Katzy, J. Kloppenburg, T. Verbelen, and J. S. Rellermeyer, “A Survey on Distributed Machine Learning,” *ACM Comput. Surv.*, vol. 53, no. 2, p. 30:1-30:33, Mar. 2020, doi: 10.1145/3377454.

[27] M. P. Skënduli, M. Biba, and M. Ceci, “Implementing Scalable Machine Learning Algorithms for

Mining Big Data: A State-of-the-Art Survey," in *Big Data in Engineering Applications*, S. S. Roy, P. Samui, R. Deo, and S. Ntalampiras, Eds., Singapore: Springer, 2018, pp. 65–81. doi: 10.1007/978-981-10-8476-8_4.

[28] M.-C. Lee, W.-L. Chiang, and C.-J. Lin, "Fast Matrix-Vector Multiplications for Large-Scale Logistic Regression on Shared-Memory Systems," in *2015 IEEE International Conference on Data Mining*, Nov. 2015, pp. 835–840. doi: 10.1109/ICDM.2015.75.

[29] J. Liu, X. Liang, W. Ruan, and B. Zhang, "High-performance medical data processing technology based on distributed parallel machine learning algorithm," *J. Supercomput.*, vol. 78, Mar. 2022, doi: 10.1007/s11227-021-04060-4.

[30] G. Mutlu and Ç. İ. Acı, "SVM-SMO-SGD: A hybrid-parallel support vector machine algorithm using sequential minimal optimization with stochastic gradient descent," *Parallel Comput.*, vol. 113, p. 102955, Oct. 2022, doi: 10.1016/j.parco.2022.102955.

[31] B. Yan, Y. Ren, and Z. Yang, "A GPU Based SVM Method with Accelerated Kernel Matrix Calculation," in *2015 IEEE International Congress on Big Data*, Jun. 2015, pp. 41–46. doi: 10.1109/BigDataCongress.2015.16.

[32] G. Sun, X. Li, X. Hou, and and F. Lang, "GPU-Accelerated Support Vector Machines for Traffic Classification," *Int. J. Perform. Eng.*, vol. 14, no. 5, p. 1088, May 2018, doi: 10.23940/ijpe.18.05.p28.10881098.

[33] World Geospatial Industry Council (WGIC). Spatial Digital Twins: Global Status, Opportunities, and the Way Forward. 2022. Available online: **https://wgicouncil.org/world-geospatial-industry-council-releases-a-report-on-spatial-digital-twins-trends-and-opportunities/**.

[34] Wanasinghe, T.R.; Wroblewski, L.; Petersen, B.K.; Gosine, R.G.; James, L.A.; Silva, O.D.; Mann, G.K.I.; Warrian, P.J. Digital Twin for the Oil and Gas Industry: Overview, Research Trends, Opportunities, and Challenges. *IEEE Access* **2020**, *8*, 104175–104197.

[35] Zhang, T.; Sun, S. An Exploratory Multi-scale Framework to Reservoir Digital Twin. *Adv. Geo-Energy Res.* **2021**, *5*, 239–251.

[36] Shen, F.; Ren, S.S.; Zhang, X.Y.; Luo, H.W.; Feng, C.M. A Digital Twin-Based Approach for Optimisation and Prediction of Oil and Gas Production. *Math. Probl. Eng.* **2021**, *2021*, 3062841.

[37] Xue, X.; Li, B.; Gai, J. Asset Management of Oil and Gas Pipeline System Based on Digital Twin. *IFAC Pap.* **2020**, *53*, 715–719.

[38] Lv, Z.; Lv, H.; Fridenfalk, M. Digital Twins in the Marine Industry. *Electronics* **2023**, *12*, 2025.

**Мухамбетжанов Салтанбек**

# ИНТЕЛЛЕКТУАЛЬНЫЕ ВЫЧИСЛИТЕЛЬНЫЕ МОДЕЛИ И ЦИФРОВЫЕ ДВОЙНИКИ ДЛЯ АНАЛИЗА И ПРОГНОЗИРОВАНИЯ ХАРАКТЕРИСТИК НЕФТЯНЫХ МЕСТОРОЖДЕНИЙ

**Монография**

Басылуға 27.10.2024 қол қойылды.
Пішімі 60x84/16. Көлемі 9,7 б.т.
Сандық басылым. Офсеттік қағаз.
Таралымы 500 дана

«Дарын баспасы» ЖШС
Алматы, көш. Жамбыл 114/85,
BC Business City, литр D, 510-кеңсе.
info@darynbaspa.kz, +7 747 150 11 77

«Дарын» баспасы» ЖШС баспаханасында басылған