

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

ДОПУСТИТЬ К ЗАЩИТЕ  
Заведующий кафедрой ЭВМ  
\_\_\_\_\_ Д.В. Топольский  
«\_\_» \_\_\_\_\_ 2024 г.

Моделирование переноса тепла в петлевой системе  
водяного охлаждения широких и узких стенок  
кристаллизатора машины непрерывного литья заготовок

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА

Руководитель работы,  
к.т.н., доцент каф. ЭВМ  
\_\_\_\_\_ Д.В. Топольский  
«\_\_» \_\_\_\_\_ 2024 г.

Автор работы,  
студент группы КЭ-405  
\_\_\_\_\_ Д.О. Чабанюк  
«\_\_» \_\_\_\_\_ 2024 г.

Нормоконтролёр,  
ст. преп. каф. ЭВМ  
\_\_\_\_\_ С.В. Сяськов  
«\_\_» \_\_\_\_\_ 2024 г.

Челябинск 2024 г.

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ  
Федеральное государственное автономное  
образовательное учреждение высшего образования  
«Южно-Уральский государственный университет  
(национальный исследовательский университет)»  
Высшая школа электроники и компьютерных наук  
Кафедра «Электронные вычислительные машины»

УТВЕРЖДАЮ

Заведующий кафедрой ЭВМ

\_\_\_\_\_ Д.В. Топольский

«\_\_» \_\_\_\_\_ 2024 г.

### ЗАДАНИЕ

на выпускную квалификационную работу бакалавра  
студенту группы КЭ-405  
Чабанюку Дмитрию Олеговичу  
обучающемуся по направлению  
09.03.01 «Информатика и вычислительная техника»

Тема работы: «Моделирование переноса тепла в петлевой системе водяного охлаждения широких и узких стенок кристаллизатора машины непрерывного литья заготовок»

утверждена приказом по университету от «\_\_» \_\_\_\_\_ 2024 г. №

Срок сдачи студентом законченной работы: 01.06.2024 г.

Исходные данные к работе

Кристаллизатор машины непрерывного литья заготовок (МНЛЗ-2)

Расход воды на кристаллизатор:  $Q_в=400-500$  м<sup>3</sup>/час;

Расход воды через широкую стенку: 225 м<sup>3</sup>/час;

Расход воды через узкую стенку: 30 м<sup>3</sup>/час;

Давление воды:  $P_в=6-8$  кг/см<sup>2</sup>;

Температура воды на входе:  $t_{вх}=20-23$  °С;

Температура воды на выходе:  $t_{вых}=31-35$  °С;

Температура металла в кристаллизатор:  $t_{кр}=1546$  °С;

Количество слитков в кристаллизаторе: 2;

## Перечень подлежащих разработке вопросов:

1. Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы;
2. Создание цифрового паспорта кристаллизатора МНЛЗ-2 как информационной модели системы первичного охлаждения;
3. Создать модель машинного обучения для прогнозирования тепловых параметров кристаллизатора МНЛЗ;
4. Реализовать приложение для визуализации модельных данных переноса тепла в петлевой системе водяного охлаждения широких и узких стенок кристаллизатора машины непрерывного литья заготовок; осуществить тестирование приложения.4.

Дата выдачи задания: 06 декабря 2023 г.

Руководитель работы \_\_\_\_\_ / Д.В. Топольский /

Студент \_\_\_\_\_ / Д.О. Чабанюк /

## КАЛЕНДАРНЫЙ ПЛАН

Этап	Срок сдачи	Подпись руководителя
Аналитический обзор научно-технической, нормативной и методической литературы по тематике работы	01.03.2024	
Создание цифрового паспорта кристаллизатора МНЛЗ-2 как информационной модели системы первичного охлаждения	01.04.2024	
Создание модели машинного обучения для анализа промышленных данных кристаллизатора МНЛЗ-2	15.04.2024	
Создание приложения для визуализации модельных данных тепловых характеристик кристаллизатора МНЛЗ	01.05.2024	
Тестирование приложения	15.05.2024	
Компоновка текста работы и сдача на нормоконтроль	24.05.2024	
Подготовка презентации и доклада	30.05.2024	

Руководитель работы \_\_\_\_\_ / Д.В. Топольский /

Студент \_\_\_\_\_ / Д.О. Чабанюк /

## **Аннотация**

Д.О. Чабанюк. Моделирование переноса тепла в петлевой системе водяного охлаждения широких и узких стенок кристаллизатора машины непрерывного литья заготовок. – Челябинск: ФГАОУ ВО «ЮУрГУ (НИУ)», ВШЭКН; 2024, 60 с., 22 ил., библиогр. список – 22 наим.

В рамках выпускной квалификационной работы исследована область обработки промышленных данных цифрового двойника кристаллизатора машины непрерывного литья заготовок с использованием нейросетевых технологий и машинного обучения, а также было разработано веб-приложение для визуализации модельных данных переноса тепла в петлевой системе водяного охлаждения широких и узких стенок кристаллизатора машины непрерывного литья заготовок.

В ходе работы были проведен обзор научной и технической литературы, собраны и предварительно обработаны промышленные данные для обучения нейронной сети, выбрана адекватная модель машинного обучения, проведено ее обучение и тестирование, разработано веб-приложение для визуализации модельных данных переноса тепла в широких и узких стенках кристаллизатора машины непрерывного литья заготовок, произведено тестирование приложения.

## ОГЛАВЛЕНИЕ

ВВЕДЕНИЕ.....	7
1. Анализ предметной области.....	9
1.1. Описание предметной области.....	9
1.2. Обзор литературы.....	10
1.3. Непрерывная разливка стали.....	15
2. Электронный паспорт кристаллизатора МНЛЗ как основа его цифрового двойника.....	19
2.1. Определение архитектуры цифрового двойника кристаллизатора МНЛЗ.....	19
2.2. Определение структуры электронного паспорта кристаллизатора МНЛЗ.....	20
3. Создание модели машинного обучения.....	23
3.1. Сбор и предобработка данных.....	23
3.2. Обучение и выбор модели.....	26
3.3. Полносвязная нейронная сеть.....	29
3.4. Тестирование модели.....	30
4. Разработка веб-приложения.....	34
4.1. Определение требований.....	34
4.2. Архитектура системы.....	36
4.3. Проектирование и реализация интерфейса веб-приложения.....	37
4.4. Проектирование REST API веб-приложения.....	39
4.5. Программные средства реализации.....	40
4.6. Тестирование веб-приложения.....	41
ЗАКЛЮЧЕНИЕ.....	43
БИБЛИОГРАФИЧЕСКИЙ СПИСОК.....	44
ПРИЛОЖЕНИЯ.....	47
Приложение А. Реализация модели машинного обучения.....	47
Приложение Б. Реализация веб-приложения.....	51
Приложение В. Макеты страниц веб-приложения.....	58
Приложение Г. Интерфейс веб-приложения.....	60

## **ВВЕДЕНИЕ**

Современная тяжелая индустрия в XXI веке претерпевает IV промышленную революцию и переходит в эпоху Индустрии 4.0, обусловленную цифровизацией и роботизацией производства. Концепция Индустрии 4.0 диктует необходимость применения инновационных технологий в производстве. Одним из наиболее перспективных направлений является внедрение в производственные процессы технологии цифровых двойников процессов, механизмов, устройств, а также крупномасштабных систем, предприятий или городов.

Основой цифрового двойника является электронный паспорт кристаллизатора МНЛЗ, агрегирующий данные о тепловых характеристиках процесса охлаждения и исходных параметрах кристаллизатора.

Однако вместе с ростом объема данных вычислительные мощности современных компьютеров и их сетей растут экспоненциально, создавая потенциал развития интеллектуальных методов анализа информации и машинного обучения на больших данных.

### **Постановка задачи**

**Целью** выпускной квалификационной работы является разработка веб-приложения для визуализации модельных данных переноса тепла в петлевой системе водяного охлаждения широких и узких стенок кристаллизатора машины непрерывного литья заготовок. Для достижения поставленной цели необходимо решить следующие **задачи**:

- 1) провести обзор научно-технической, нормативной и методической литературы;
- 2) разработать структуру цифрового паспорта кристаллизатора МНЛЗ-2 как информационной модели системы первичного охлаждения;
- 3) применить методы машинного обучения для анализа модельных данных и прогнозирования;
- 4) реализовать приложение для визуализации модельных данных переноса тепла

в петлевой системе водяного охлаждения широких и узких стенок кристаллизатора машины непрерывного литья заготовок;

5) осуществить тестирование приложения.

### **Структура и содержание работы**

Работа состоит из введения, четырех глав, заключения и списка литературы. Объем работы составляет 60 страниц, объем списка литературы – 22 источника.

В первой главе описываются теоретические сведения о предметной области, и осуществляется обзор существующих аналогов, которые решают проблему моделирования и визуализации модельных данных системы первичного охлаждения МНЛЗ, исследуется актуальное состояние проблемы разработки цифровых паспортов изделий, также просматриваются разные подходы к этим задачам и осуществляется сравнительный анализ для выявления существующих проблем и способов их решения.

Во второй главе описывается архитектура цифрового двойника кристаллизатора МНЛЗ. Основой цифрового двойника является электронный паспорт кристаллизатора МНЛЗ

В третьей главе описываются топологии выбранной нейронной сети, процесс предобработки данных выборки, реализация и обучения моделей нейронной сети. результаты валидации работы нейронной сети на тестированной выборке датасета.

В четвертой главе представлены функциональные и нефункциональные требования к разработанной системе, варианты ее использования, описания архитектуры и логики веб-приложения, и проектирование интерфейсов. Также представлено функциональное тестирование веб-интерфейса и REST API.

В заключении описываются результаты выполненной работы, возможные направления дальнейших исследований и перспективы веб-приложения.

В приложениях содержатся листинги с реализациями модели машинного обучения для обработки промышленных данных.



## **1. Анализ предметной области**

### **1.1. Описание предметной области**

Южный Урал является регионом с развитой тяжелой промышленностью, в частности – черной металлургии. Одной из важных и актуальных проблем для существующих в регионе металлургических комбинатов является снижение стоимости слябов при повышении качества готового проката. Снижение себестоимости слябов и последующей продукции прокатного производства возможно за счет сокращения затрат на ремонт машины непрерывного литья заготовок (МНЛЗ), связанных с восстановлением сменного оборудования и кристаллизаторов [1].

Одним из наиболее дорогостоящих узлов МНЛЗ является кристаллизатор, имеющий крайне низкую стойкость. Это обусловлено тяжелыми условиями его эксплуатации – высокими механическими нагрузками, непосредственным контактом с жидким металлом и шлаком. Основная функция кристаллизатора заключается в отводе тепла от жидкой стали к воде и формировании на поверхности образующейся заготовки твердой корочки определенной толщины, способной выдержать ферростатическое давление жидкого металла после выхода сляба (с жидкой фазой внутри него) из кристаллизатора [2, 3].

Эпоха Индустрии 4.0 диктует необходимость внедрения инновационных технологий, способных снизить производственные издержки и риски поломки дорогостоящего оборудования, такого как кристаллизатор, а также направленные на улучшение качества производимой продукции.

Внедрение технологии цифровых двойников позволяет решить эти проблемы. Цифровой двойник кристаллизатора слябовой МНЛЗ представляет собой совокупность моделей процессов, происходящих в широких и узких стенках кристаллизатора: гидравлической и тепловой работы первичной системы охлаждения слитка, а также приложение для визуализации и аналитики модельных и прогнозных данных [4].

## 1.2. Обзор литературы

В настоящее время существует ряд исследовательских работ, посвященных цифровым двойникам кристаллизатора МНЛЗ, в основе которых лежат расчеты тепловой и гидравлической работ в первичной системе охлаждения.

Исследователи ЮУрГУ под руководством И.М. Ячикова [1] произвели тепловой расчет широких стенок кристаллизатора с щелевыми каналами охлаждения. Для автоматизации и визуализации результатов расчёта был разработан модуль моделирования этих процессов в СКМ ЛП «ПолигонСофт». Это позволило прогнозировать тепловое поле на рабочей поверхности широкой стенки кристаллизатора для обеспечения заданного теплового режима его работы, а также предотвращения разупрочнения медного сплава и сохранения заданных прочностных свойств. Проведен тепловой расчет медных стенок с щелевыми каналами, а также полученные данные адаптировали к системе компьютерного моделирования «ПолигонСофт». Представленные результаты математического и компьютерного моделирования носят практический характер и могут быть применены при разработке новых унифицированных конструкций слябовых кристаллизаторов машины непрерывного литья заготовок.

В статье [5] представлены математическая модель и результаты моделирования теплообмена сляба с кристаллизатором машины непрерывного литья заготовок при динамических режимах разливки, когда изменяются скорость разливки и уровень жидкого металла в кристаллизаторе. Научной новизной данной работы является динамический характер созданной модели. В научной литературе представлено математическое описание теплопередачи от слитка к кристаллизатору при стационарных режимах разливки. Теплопередача от сляба к кристаллизатору при нестационарных (динамических) режимах разливки исследована недостаточно, в частности, отсутствуют математические модели, адекватно описывающие этот процесс.

В то же время при разливке стали на МНЛЗ довольно часто происходит изменение скорости вытягивания слитка, расхода и параметров жидкого металла,

подаваемого в кристаллизатор, обусловленное следующими причинами:

- нарушением стабильной работы огнеупоров;
- прожиганием разливочных стаканов;
- уменьшением скорости разливки при смене промежуточного ковша,

сталеразливочного стаканов, при запуске и остановке МНЛЗ.

В работе [6] представлена математическая модель процесса охлаждения слитка в кристаллизаторе машины непрерывного литья заготовок. Математическая модель учитывает особенности охлаждения формирующегося слитка через боковые поверхности: информацию о температурах ребер слитка как зон наибольшего градиента температур. Для адекватного описания процесса образования слитка в кристаллизаторе автор решает пространственную задачу, учитывая теплообмен в кристаллизаторе.

Автором работы [7] предлагается аналитическая математическая модель тепловых процессов, протекающих при непрерывной отливке в заготовке при ее движении с постоянной скоростью в кристаллизаторе, учитывающая выделение скрытой теплоты плавления и охлаждающее воздействие стенок кристаллизатора, а также других факторов.

Предполагается, что затвердевание заготовки происходит в прямом цилиндрическом кристаллизаторе с металлической вставкой, охлаждение постоянно по всей длине кристаллизатора, а основная доля скрытой теплоты кристаллизации выделяется в пределах жидкого ядра отливки.

В этих условиях процесс охлаждения расплава описывается дифференциальным уравнением конвективной теплопроводности, после решения которого было получено инженерное выражение для расчета температурных полей в заготовке.

Расчетные эксперименты по построению температурных полей проводились для непрерывно-литого чугуна слитка. В отличие от известных работ в построенной модели было изучено влияние скорости движения слитка в

процессе затвердевания расплава.

Таким образом, автором предложена аналитическая теплофизическая модель охлаждения металла при непрерывном процессе литья в кристаллизаторе. Модель учитывает основные производственные и теплофизические условия формирования отливок: скорость извлечения заготовки из кристаллизатора, теплообмен между кристаллизирующимся металлом и внутренней поверхностью кристаллизатора, геометрические параметры заготовки и физические особенности металла, включая выделение в заготовке скрытой теплоты плавления.

Также существует ряд работ, посвященных поиску и расчету оптимального терморежима кристаллизатора. Например, группой ученых из Беларуси [8] разработана методика определения теплового режима кристаллизатора, где полученные данные позволяют перейти к расчету процесса затвердевания отливки и определению термонапряженного состояния рабочей втулки кристаллизатора, а также к анализу термомеханического взаимодействия кристаллизатора и отливки в процессе ее формирования.

Однако, данная выпускная квалификационная работа посвящена в большей степени информационной модели кристаллизатора МНЛЗ и тепловой работе в нем. Таким образом, необходимо также рассмотреть исследования, направленные на обработку промышленных данных и цифровых двойников МНЛЗ.

Группа ученых под руководством М.П. Гусева выпустила ряд работ, посвященных созданию цифрового двойника МНЛЗ и кристаллизатора – в частности [9-10].

Ученые утверждают, что несмотря на значительное количество работ, посвященных повышению качества слитка, в настоящее время не разработана комплексная математическая модель, охватывающая процесс разлива стали от зоны кристаллизатора до затвердевания слитка, функционирующая в режиме реального времени, отражающая качество слитка в режиме реального времени,

позволяющая в автоматическом режиме рассчитывать и изменять параметры разливки для достижения оптимального качества слитка и связанная с платформой сбора данных или промышленным интернетом. Систему, обладающую подобным функционалом, следует назвать Цифровым Двойником процесса непрерывного литья заготовок и в своей работе данный коллектив предлагает концепт такой системы [9].

Далее та же группа ученых занимается поиском оптимального средства обмена промышленными данными для работы цифрового двойника. В статье [10] представлены результаты тестирования оптоволоконных датчиков измерения температуры в составе разработанного опытного комплекса мониторинга тепловых процессов в кристаллизаторе машины непрерывного литья заготовок (МНЛЗ). Проведено сравнение качества измерения температур оптоволоконными датчиками и классическими хромель-алюмелевыми термопарами. Разработан прототип цифрового двойника кристаллизатора для оценки полного теплового состояния кристаллизатора.

Авторский коллектив [11] рассматривает методику проведения исследований тепловых параметров работы кристаллизатора с помощью измерительного комплекса двухзонных термопарных датчиков. В статье описывается опытно-экспериментальная работа по установлению взаимосвязи показателей качества поверхности слябовых заготовок и технологических параметров, теплотехнических показателей работы кристаллизатора и химического состава разливаемой стали. В результате было разработано программное обеспечение для накопления и обработки диагностической информации на ЭВМ. Применение этой системы в комплексе со средствами статистической обработки данных позволяет разработать систему оптимального управления режимами охлаждения слитка и установить различные технологические параметры разливки для разных марок стали. Данное исследование также может быть полезно в рамках работы над цифровым двойником кристаллизатора МНЛЗ.

Как уже отмечалось выше, цифровой двойник представляет собой совокупность физико-математической модели и элементов физического уровня. С помощью системы моделей можно смоделировать критические неисправности, которые не повторялись ранее, которые мы не наблюдали в исторических данных, и когда имеется мало информации об объекте.

Однако, имея исторические данные Интернета вещей (промышленные данные), следует использовать машинное обучение: на этих данных можно построить статистические модели, прогнозировать события, которые в исторических данных происходили.

Так, например, группа австрийских ученых [12] представила научный обзор методов машинного обучения для анализа процесса непрерывной разливки стали: обзор машинного обучения и обзор процесса непрерывной разливки. Авторы проанализировали существующую работу, касающуюся машинного обучения при непрерывной разливке стали, а затем привели общие концепции по категориям, помогая определить общие варианты использования и подходы. Анализ завершается проработкой проблем, потенциальных решений и перспективами дальнейших исследований.

Однако, стоит отметить, что данные, полученные от сенсоров и приборов не всегда однородны и структурированы. Таким образом, необходимо найти способ обработки неструктурированных или частично структурированных данных промышленного Интернета вещей.

Исследовательский коллектив [13] привел практическое применение инновационной технологии обработки и анализа неоднородного набора данных с использованием промышленного Интернета вещей для сбора информации и технологии искусственного интеллекта для ее консолидации и централизованной потоковой обработки в реальном времени. В работе показан принцип интеграции разработки компании с облачными платформами на примере Microsoft Azure. Приведены примеры результатов внедрения и повышения производительности.

Система поддержки принятия решений является неотъемлемой частью

цифрового двойника кристаллизатора МНЛЗ. В статье [14] был представлен анализ использования промышленных больших данных для интеллектуального принятия решений, а также рассмотрены проблемы в этой области. В реальной производственной деятельности утечка информации, помехи и другие факторы будут оказывать косвенное влияние на принятие решений. Эта концептуальная основа новой парадигмы вводит промышленный анализ на основе больших данных в производственную систему. Однако гипотетическая модель анализа больших данных построена в идеальной среде, надежность и практичность этой концептуальной основы нуждается в дальнейшей проверке.

Интеллектуальное производство тесно связано с оцифровкой всех производственных операций. Это увеличивает объем доступных данных для повышения производительности и получения прибыли с помощью программ принятия решений, основанных на данных. Коллектив исследователей [15] составил работу, предназначенную помочь инженерам по обработке данных в разработке конвейеров анализа больших данных для данных производственного процесса. Таким образом, в этом документе охарактеризованы требования к конвейерам анализа технологических данных и проведен обзор существующих платформ из академической литературы. В работе представлены этапы хранения и анализа конвейеров, получения, передачи и визуализации, инструменты для обработки разнородных данных, пакетной обработки и потоковой обработки в реальном времени. Также в статье предлагаются рекомендации для каждой фазы конвейера.

### **1.3. Непрерывная разливка стали**

Цифровизация металлургического производства, особенно технологической непрерывной разливки с использованием численных моделей тепло- и массопереноса и последующего затвердевания, была разработана для достижения высокой эффективности производства с минимальными дефектами и, следовательно, низким уровнем брака. По-прежнему сложно выполнять адаптивную настройку процесса с замкнутым циклом с использованием высокоточных вычислений в режиме реального времени. Для решения этой

проблемы суррогатные модели являются хорошим вариантом замены высокоточной модели с приемлемой точностью и меньшими затратами на вычисления время и затраты. Этот инновационный и надежный метод обладает большим потенциалом в интеллектуальном управлении процессом металлургического производства.

Непрерывная разливка стали — это сложный металлургический процесс, используемый для производства большинства технологических стальных изделий (включая заготовки, блюмы и слябы) по всему миру. В заготовках всегда имеется несколько типов дефектов литья, включая поверхностные и угловые трещины, макросегрегацию по центру, усадку по центру и пористость, которые снижают эксплуатационные характеристики стальных изделий. Эффективным решением для устранения этих дефектов является использование технологии мягкого восстановления, при которой ключевыми техническими параметрами являются положение и степень уменьшения, а также соответствующий контроль подачи вторичной охлаждающей воды. Для эффективного управления вторичным охлаждением и плавным обжатием необходимо точно рассчитать толщину затвердевшей оболочки и металлургическую длину (расстояние, на котором затвердевание в центре сляба является достаточно полным) в процессе литья. Поэтому очень важно в целом проанализировать теплопередачу, затвердевание, многофазное турбулентное течение и другие взаимодействующие явления в процессе литья. Однако это довольно сложно выполнить в достаточной степени эксперименты с процессом литья для получения данных для оцифровки процесса, а также из-за ограниченности методов измерений и суровых производственных условий большая часть важной информации о процессах литья не может быть получена прямым экспериментом. Численное моделирование широко применимо для лучшего понимания этих фундаментальных характеристик и последующего прогнозирования трехмерного распределения температуры и затвердевающей оболочки, которые являются ключевой информацией для оптимизации процесса и стратегии устранения дефектов.



Модель прямого численного моделирования, основанная на вычислительной гидродинамике для непрерывного литья, значительно улучшилась с 1980-х годов и использовалась для оптимизации условий эксплуатации, таких как расположение охлаждающей воды в зоне вторичного охлаждения, что приводит к уменьшению дефектов отливки в заготовке. Модели прямого численного моделирования были разработаны для моделирования сложных явлений в процессе литья, и исследования объединили данные модели с алгоритмами оптимизации-регулирования, чтобы ускорить процесс оптимизации. Однако общей проблемой этих моделей является чрезмерное вычислительное время и потребление ресурсов, особенно при параметрических исследованиях для оптимизации процессов, где необходимо выполнить большое количество вычислений. Кроме того, всегда возникают ситуации, когда реальное поведение процесса ведет себя неожиданным образом. В этом случае крайне необходимы быстрое принятие решения и автозамена, чтобы предотвратить дальнейшее ухудшение процесса.

Глубокое обучение превратилось в мощную технологию, демонстрирующую самую современную производительность при решении самых разных задач. Обладая исключительной способностью извлекать уроки из огромных объемов данных, данная технология широко используется в процессе литья для обеспечения непрерывного мониторинга, контроля (оптимизация температурного режима и предварительная настройка температуры расплавленной стали), и оценка реализации (прогнозирование внутренних трещин и прорывов). Несмотря на преимущества, которые дают методы глубокого обучения для повышения интеллектуальности и эффективности процесса литья, по-прежнему существует настоятельная необходимость в разработке методов, обеспечивающих быстрое принятие решений и быструю реакцию на автокоррекцию в этой области.

## **Вывод по Главе 1**

В данной главе описаны предметная область и теоретическая часть

моделирования тепломассопереноса в кристаллизаторе машины непрерывного литья заготовок. Также был произведен обзор литературных источников, которые легли в основу данной работы.

## **2. Электронный паспорт кристаллизатора МНЛЗ как основа его цифрового двойника**

### **2.1. Определение архитектуры цифрового двойника кристаллизатора МНЛЗ**

С помощью цифрового двойника могут быть обнаружены аномалии в производственных процессах и достигнуты различные функциональные цели, такие как управление в режиме реального времени, автономная аналитика, проверка работоспособности, профилактическое обслуживание, оптимизация управления производственным процессом, адаптация процессов, анализ больших данных, машинное обучение и т.д.

Существует три основных типа моделей, основанных на данных, в зависимости от их применения:

- деградирующие модели (отражают прогрессирующее ухудшение состояния системы с течением времени);
- локальное моделирование (когда виртуальная модель заменяет физическую, что требует высоких вычислительных затрат, но сокращает время моделирования и требования к ресурсам);
- модели, управляемые данными, для динамической идентификации систем с использованием измерений от датчиков в физической системе [11].

Важно отметить, что методы, основанные на данных, требуют значительного объема данных для обучающих моделей. Поэтому решающее значение имеет обеспечение репрезентативности и точности собираемых данных.

Рассматривая подходы к построению архитектуры цифрового двойника, необходимо учитывать требования национального стандарта, где приведены типовые архитектуры цифровых двойников: сущностная, доменная и комбинированная [12].

В интересах данной работы использована комбинированная архитектурная модель (рисунок 1).

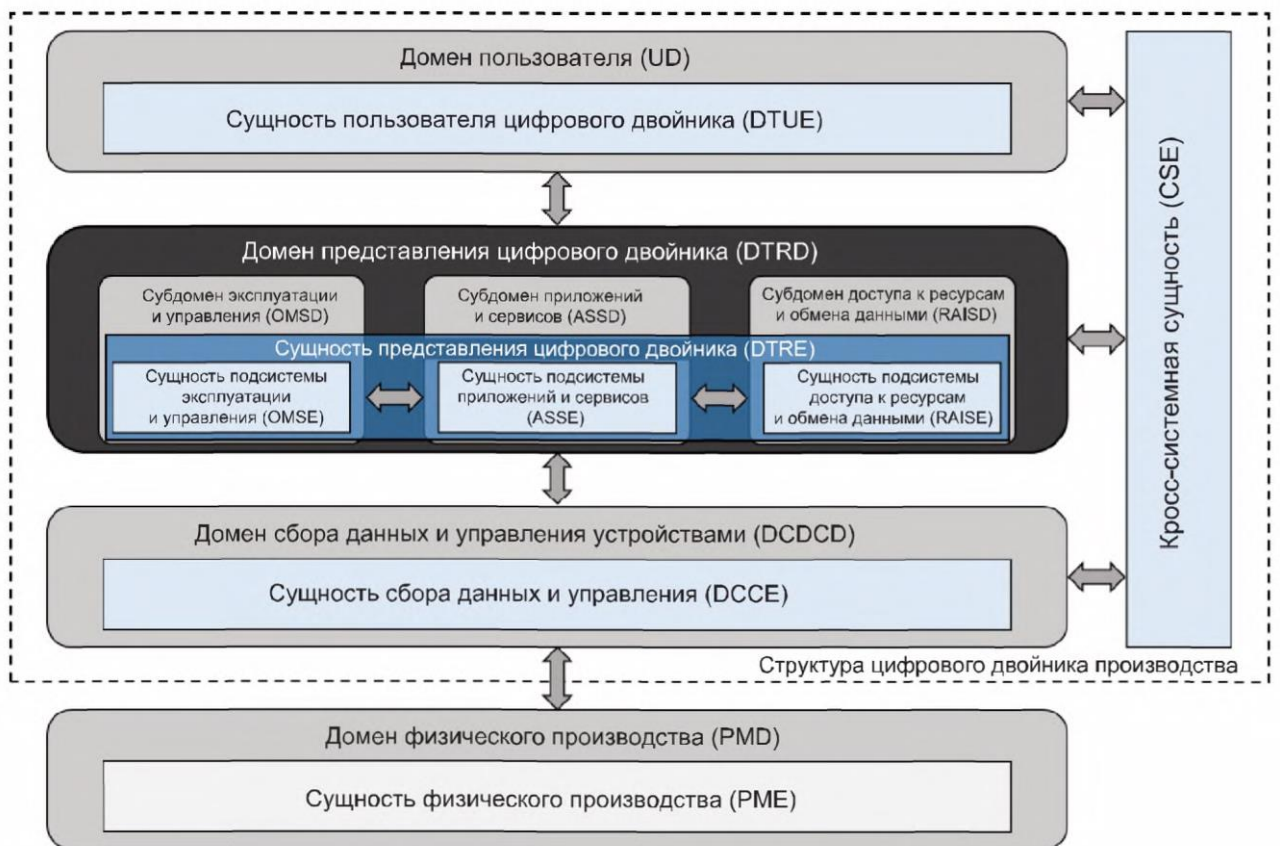


Рисунок 1 - Комбинированная универсальная модель, основанная на доменах и сущностях

Сущности, лежащие в основе данной архитектуры, представляют собой электронный паспорт кристаллизатора МНЛЗ, который включает в себя базу данных, агрегирующую текущие и расчетные параметры системы охлаждения. В роли доменов выступает система датчиков, передающих текущие показания приборов, и Интернет вещей.

## 2.2. Определение структуры электронного паспорта кристаллизатора МНЛЗ

Динамические и паспортные данные используются в дальнейшем для модели машинного обучения, и визуализируются в приложении оператора (рисунок 2) [11, 12].

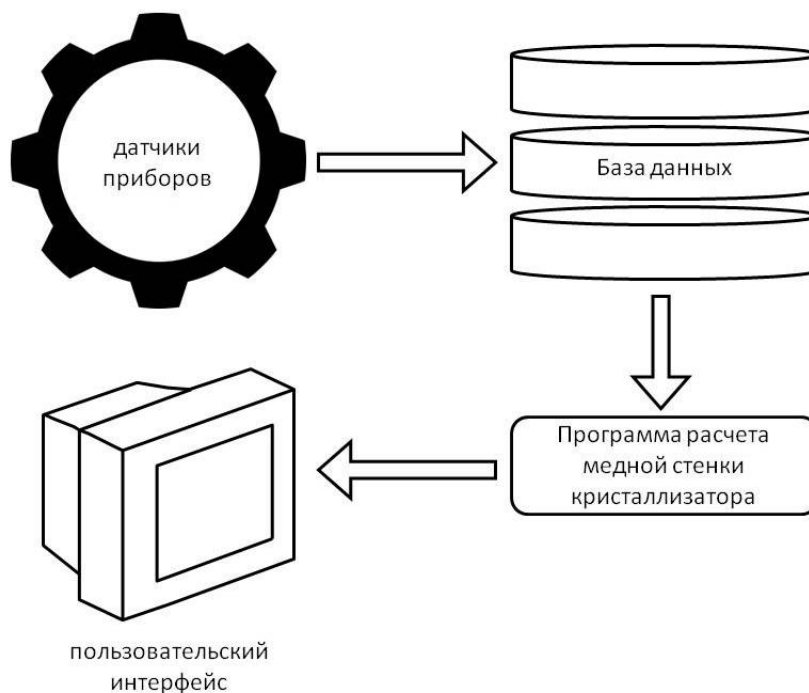


Рисунок 2 – Схема решения задачи имитационного моделирования

Текущие показания приборов посредством СОМ-портов поступают в базу данных и заполняют поля электронного паспорта объекта моделирования (рисунок 5) – динамические параметры. На основе динамических данных заполняются расчетные значения и уточняются модельные параметры.

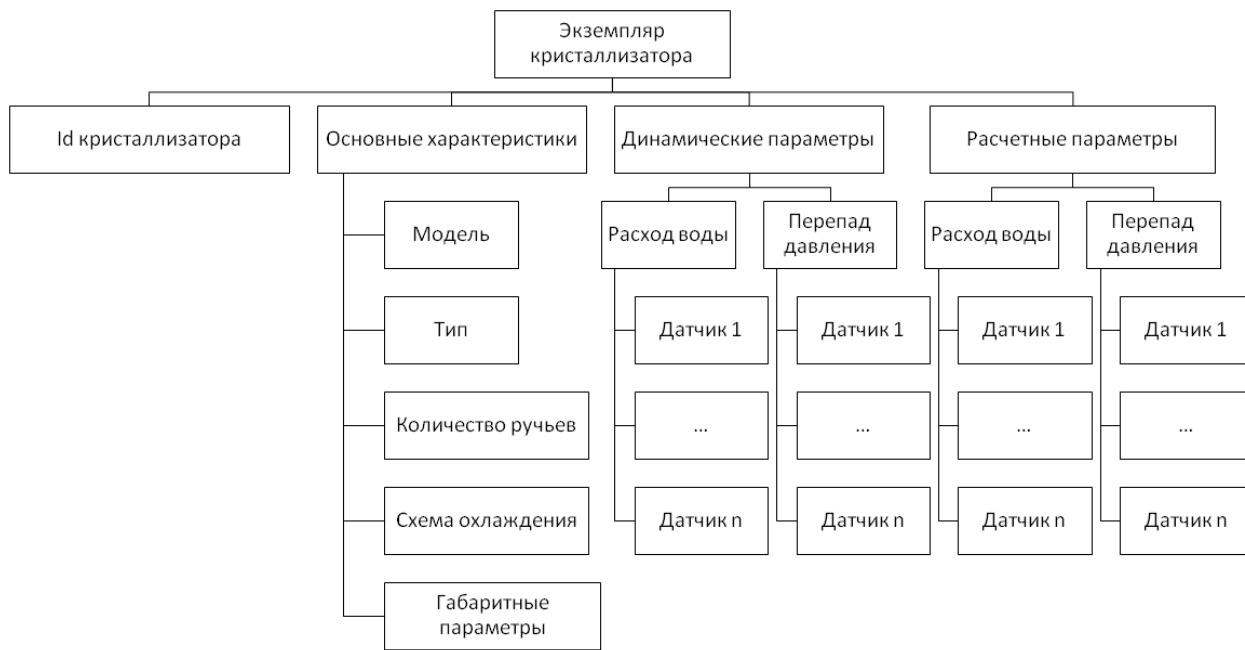


Рисунок 3 – Структура электронного паспорта кристаллизатора МНЛЗ

Электронный паспорт хранит такие поля как номер кристаллизатора, количество ручьев, основные характеристики о конкретной модели кристаллизатора, динамические параметры (расход воды, перепад давления).

Данная структура электронного паспорта легла в основу модели машинного обучения.

## **Вывод по Главе 2**

Задачей данной работы было определение структуры электронного паспорта цифрового двойника кристаллизатора МНЛЗ.

Для понимания специфики данной области были изучены требования Предварительного национального стандарта РФ ПНСТ 430-2020 «Умное производство. Цифровые двойники производства. Часть 2. Типовая архитектура» [16].

Таким образом за основу цифрового двойника кристаллизатора машины непрерывного литья заготовок принята комбинированная типовая архитектура, где в роли сущности сбора данных выступает электронный паспорт кристаллизатора, сущностью управления является модель машинного обучения, а в роли домена пользователя – веб-приложение для взаимодействия с обученной моделью машинного обучения для обработки промышленных данных.

### 3. Создание модели машинного обучения

#### 3.1. Сбор и предобработка данных

Проведем исследовательский анализ данных: проверим наличие пропусков и аномалий, изучим распределение признаков. Для этого напишем функцию для получения информации о датасете, выводящую первые пять строк, описание информации в столбцах, наличие явных дубликатов и количество пропусков (рисунок 4).

```
file_path = '/content/ccm_dataset.csv'

data = pd.read_csv(file_path)
data.info()
data.describe()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17460 entries, 0 to 17459
Data columns (total 9 columns):
#   Column                                                                 Non-Null Count  Dtype
---  -
0   steel_type                                                             17460 non-null  object
1   steel_temperature_grab1_Celsius_deg  17460 non-null  float64
2   water_consumption_liter_per_minute  17460 non-null  int64
3   water_temperature_delta_Celsius_deg  17460 non-null  int64
4   water_consumption_secondary_cooling_zone_num1_liter_per_minute  17460 non-null  int64
5   water_consumption_secondary_cooling_zone_num2_liter_per_minute  17460 non-null  int64
6   water_consumption_secondary_cooling_zone_num3_liter_per_minute  17460 non-null  float64
7   temperature_measurement1_Celsius_deg  17460 non-null  float64
8   temperature_measurement2_Celsius_deg  17460 non-null  float64
dtypes: float64(4), int64(4), object(1)
memory usage: 1.2+ MB
```

	steel_temperature_grab1_Celsius_deg	water_consumption_liter_per_minute	water_temperature_delta_Celsius_deg
count	17460.000000	17460.000000	17460.0
mean	1566.883047	2127.163230	8.9
std	23.762489	71.769604	0.2
min	567.000000	1255.000000	6.0
25%	1563.000000	2155.000000	9.0
50%	1567.000000	2155.000000	9.0
75%	1570.000000	2155.000000	9.0
max	1667.000000	2155.000000	9.0

Рисунок 4 – Предварительный анализ датасета

Из полученных данных видим, что пропусков в данных нет, полных дубликатов в данных также нет. Рассмотрим более детально каждый столбец.

Поле `steel_type` хранит информацию о сорте разливаемой стали. На рисунке 5 показан количественный анализ: присутствуют записи об 11 сортах стали.

```
data.groupby('steel_type')['steel_type'].count().value_counts(normalize=True)

24      0.166667
30      0.083333
117     0.083333
518     0.083333
220     0.083333
414     0.083333
96      0.083333
13816   0.083333
804     0.083333
1352    0.083333
45      0.083333
Name: steel_type, dtype: float64
```

Рисунок 5 – Сорта стали

Проведем исследовательский анализ данных о температуре на наличие пропусков и аномалий, а также на распределение признаков, где целевым будет выходное значение температуры. Начальную температуру будем использовать как входной признак.

Рассчитаем перепад температур охлаждающей жидкости внутри кристаллизатора, предварительно написав функцию для его расчета. Функция на входе принимает строку датасета и в зависимости от условия рассчитывает время нагрева, которое сохраняется в новом столбце `time_heat`. На выходе получаем новую строку с добавленным столбцом. Затем применим данную функцию ко всему датасету (рисунок б)

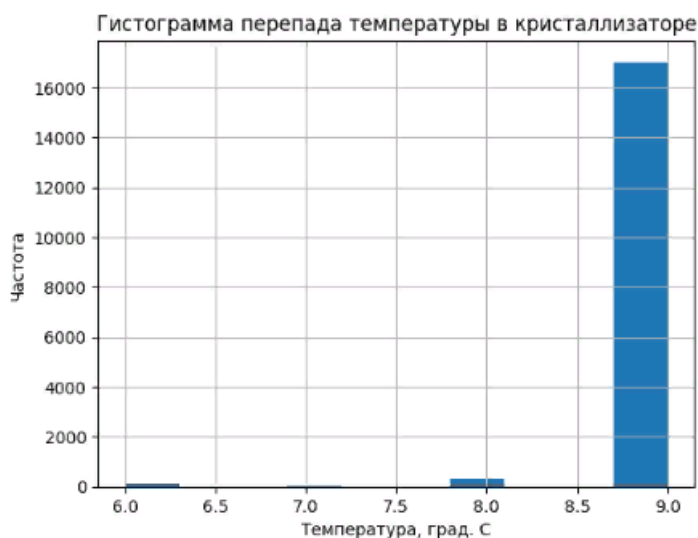


Рисунок б – Перепад температур охлаждающей жидкости

Далее проверим распределение величин температуры с слитках, а также

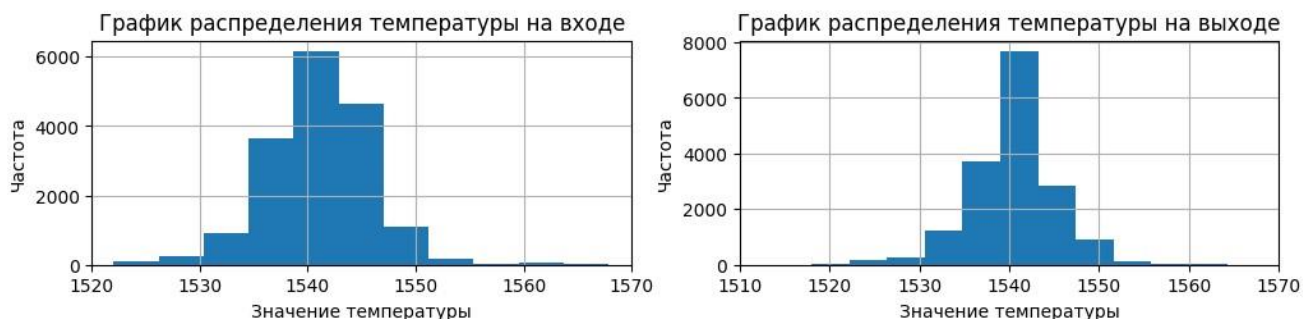


ВХОДНЫЕ И ВЫХОДНЫЕ ЗАМЕРЫ ТЕМПЕРАТУРЫ ЖИДКОСТИ КРИСТАЛЛИЗАТОРА (рисунок 7).

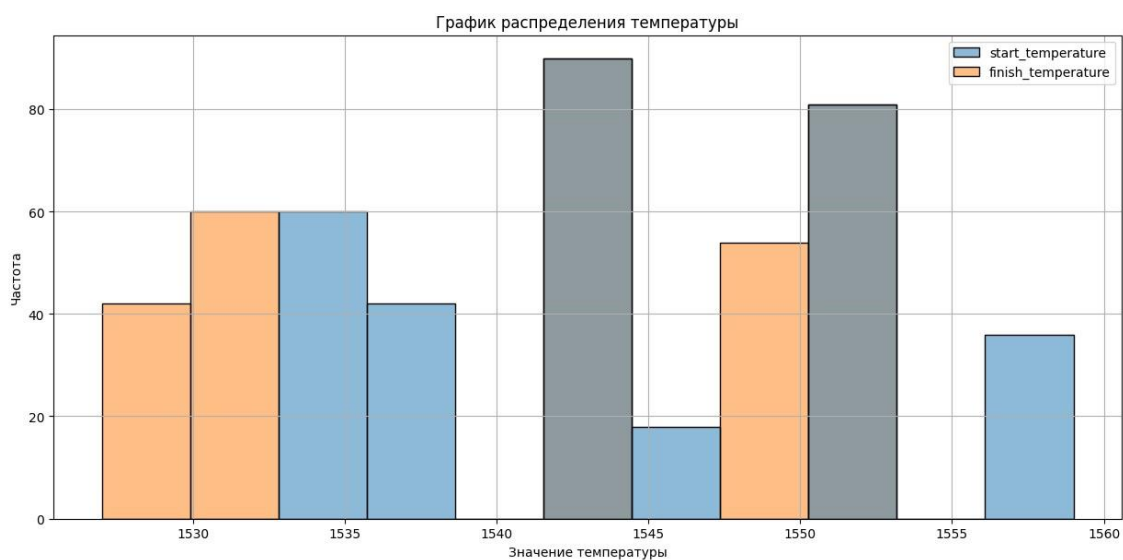


## Рисунок 7 – Анализ распределения данных

Распределение нормальное, аномалий нет. Построим график распределения температуры (рисунок 8, рисунок 9).



## Рисунок 8 – Распределение температур слитка



## Рисунок 9 – Распределение температуры слитка

**Вывод:** на данном этапе датасет разделили на признаки, где целевой признак – конечная температура (`finish_temperature`), которую мы пытаемся предсказать в данном проекте. Также разделили выборки на обучающую и тестовую в соотношении 3:1.

### 3.2. Обучение и выбор модели

Рассмотрим линейную модель `Ridge()`, которую можно использовать в случае мультиколлинеарности данных (рисунок 10).

```

%%time
columns = features_train.columns
column_transformer_rd = make_column_transformer((StandardScaler(),columns),
                                                remainder = 'passthrough')

model_rd = Ridge(random_state=RANDOM_STATE)
pipeline_rd = make_pipeline(column_transformer_rd, model_rd)
pipeline_rd
param_rd = {
    'ridge__alpha': range(1,15,1)
}
grid_rd = GridSearchCV(pipeline_rd, param_rd, cv=3,
                       scoring='neg_mean_absolute_error')
grid_rd.fit(features_train, target_train)
ridge_best_mae = round(abs(grid_rd.best_score_), 3)
ridge_best_model = grid_rd.best_estimator_
ridge_best_params = grid_rd.best_params_
print(f'MAE лучшей модели Ridge на обучающей выборке: {ridge_best_mae}')
print(f'Гиперпараметры лучшей модели: {ridge_best_params}')

```

MAE лучшей модели Ridge на обучающей выборке: 1.75  
Гиперпараметры лучшей модели: {'ridge\_\_alpha': 14}  
CPU times: user 1.09 s, sys: 816 ms, total: 1.9 s  
Wall time: 1.01 s

Рисунок 10 – MAE лучшей модели Ridge на обучающей выборке

Рассмотрим модель RandomForestRegressor. Поступаем аналогично, но так как гиперпараметров больше, будем использовать RandomizedSearchCV. Также произведем масштабирование данных (рисунок 11).

```

%%time
column_transformer_rf = make_column_transformer((StandardScaler(),columns),
                                                remainder = 'passthrough')
model_rf = RandomForestRegressor(random_state=RANDOM_STATE)
pipeline_rf = make_pipeline(column_transformer_rf, model_rf)

param_rf = {
    'randomforestregressor__n_estimators': range(10, 80, 10),
    'randomforestregressor__max_depth': range(1, 10),
    'randomforestregressor__min_samples_split': (2, 3, 4),
    'randomforestregressor__min_samples_leaf': (1, 2, 3, 4)
}
grid_rf = RandomizedSearchCV(pipeline_rf, param_rf, cv=3, n_iter=30,
                             scoring='neg_mean_absolute_error', n_jobs=-1,
                             random_state=RANDOM_STATE)
grid_rf.fit(features_train, target_train)
rf_best_mae = round(abs(grid_rf.best_score_), 3)
rf_best_model = grid_rf.best_estimator_
rf_best_params = grid_rf.best_params_
print(f'MAE лучшей модели RandomForestRegressor на обучающей выборке: {rf_best_mae}')
print(f'Гиперпараметры лучшей модели:')
for k, v in rf_best_params.items():
    print('\t', k, ': ', v)

```

MAE лучшей модели RandomForestRegressor на обучающей выборке: 1.48  
Гиперпараметры лучшей модели:  
randomforestregressor\_\_n\_estimators : 30  
randomforestregressor\_\_min\_samples\_split : 3  
randomforestregressor\_\_min\_samples\_leaf : 3  
randomforestregressor\_\_max\_depth : 9  
CPU times: user 1.86 s, sys: 184 ms, total: 2.04 s  
Wall time: 41.7 s

Рисунок 11 – MAE лучшей модели RandomForestRegressor на обучающей выборке

Далее рассмотрим модель LGBMRegressor, где аналогично произведем

масштабирование данных (рисунок 12).

```
%%time
model_lg = LGBMRegressor(random_state=RANDOM_STATE)

param_lg = {
    'n_estimators': range(10, 100, 10),
    'num_leaves': range(5, 15),
    'learning_rate': [0.01, 0.1]
}

grid_lg = RandomizedSearchCV(model_lg, param_lg, cv=3, n_iter=30,
                             scoring='neg_mean_absolute_error', n_jobs=-1,
                             random_state=RANDOM_STATE)
grid_lg.fit(features_train, target_train)

lg_best_mae = round(abs(grid_lg.best_score_), 3)
lg_best_model = grid_lg.best_estimator_
lg_best_params = grid_lg.best_params_

print(f'MAE лучшей модели LGBMRegressor на обучающей выборке: {lg_best_mae}')
print(f'Гиперпараметры лучшей модели: {lg_best_params}')
```

[LightGBM] [Info] Auto-choosing row-wise multi-threading, the overhead of testing was 0.000769 seconds. You can set `force\_row\_wise=true` to remove the overhead. And if memory is not enough, you can set `force\_col\_wise=true`.

[LightGBM] [Info] Total Bins 1073

[LightGBM] [Info] Number of data points in the train set: 12783, number of used features: 12

[LightGBM] [Info] Start training from score 1540.555347

MAE лучшей модели LGBMRegressor на обучающей выборке: 1.678

Гиперпараметры лучшей модели: {'num\_leaves': 14, 'n\_estimators': 70, 'learning\_rate': 0.1}

CPU times: user 527 ms, sys: 37.6 ms, total: 565 ms

Wall time: 8.57 s

Рисунок 12 – MAE лучшей модели LGBMRegressor на обучающей выборке

Рассмотрим модель CatBoostRegressor. В данном случае не будем заниматься подбором гиперпараметров, посмотрим значение MAE на кроссвалидации, а также попробуем подобрать гиперпараметры без масштабирования (рисунок 13).

```

%%time
model_cb_1 = CatBoostRegressor(random_seed=RANDOM_STATE, silent=True)

param_cb_1 = {
    'learning_rate':[0.1, 0.3],
    'iterations': [40, 50, 60, 80],
    'depth': range(1, 10, 1),
}

grid_cb = RandomizedSearchCV(model_cb_1, param_cb_1, cv=3, n_iter=20,
                             scoring='neg_mean_absolute_error', n_jobs=-1,
                             random_state=RANDOM_STATE)
grid_cb.fit(features_train, target_train)

cb_best_mae = abs(grid_cb.best_score_)
cb_best_model = grid_cb.best_estimator_
cb_best_params = grid_cb.best_params_

print(f'MAE лучшей модели CatBoostRegressor на обучающей выборке: {cb_best_mae}')
print(f'Гиперпараметры лучшей модели: {cb_best_params}')

```

```

MAE лучшей модели CatBoostRegressor на обучающей выборке: 1.4269717351971882
Гиперпараметры лучшей модели: {'learning_rate': 0.3, 'iterations': 60, 'depth': 7}
CPU times: user 1.01 s, sys: 156 ms, total: 1.17 s
Wall time: 16 s

```

Рисунок 13 – MAE лучшей модели CatBoostRegressor на обучающей выборке

Значение MAE без подбора гиперпараметров со стандартными настройками получилось лучше, будем использовать данную модель при сравнении.

### 3.3. Полносвязная нейронная сеть

Создаем класс для задания архитектуры нейронной сети. Так как у нас задача регрессии, то на выходном слое у нас будет 1 нейрон. На входе количество нейронов равно количеству признаков. Для задач регрессии на выходе в качестве функции активации предпочтительнее использовать линейную функцию, в скрытых слоях попробуем использовать функцию ReLU(). Переведем выборки в тензоры.

Для сравнения моделей сведем данные в таблицу. Будем сравнивать по величине метрики MAE, полученной при обучении моделей на обучающей

выборке (рисунок 14).

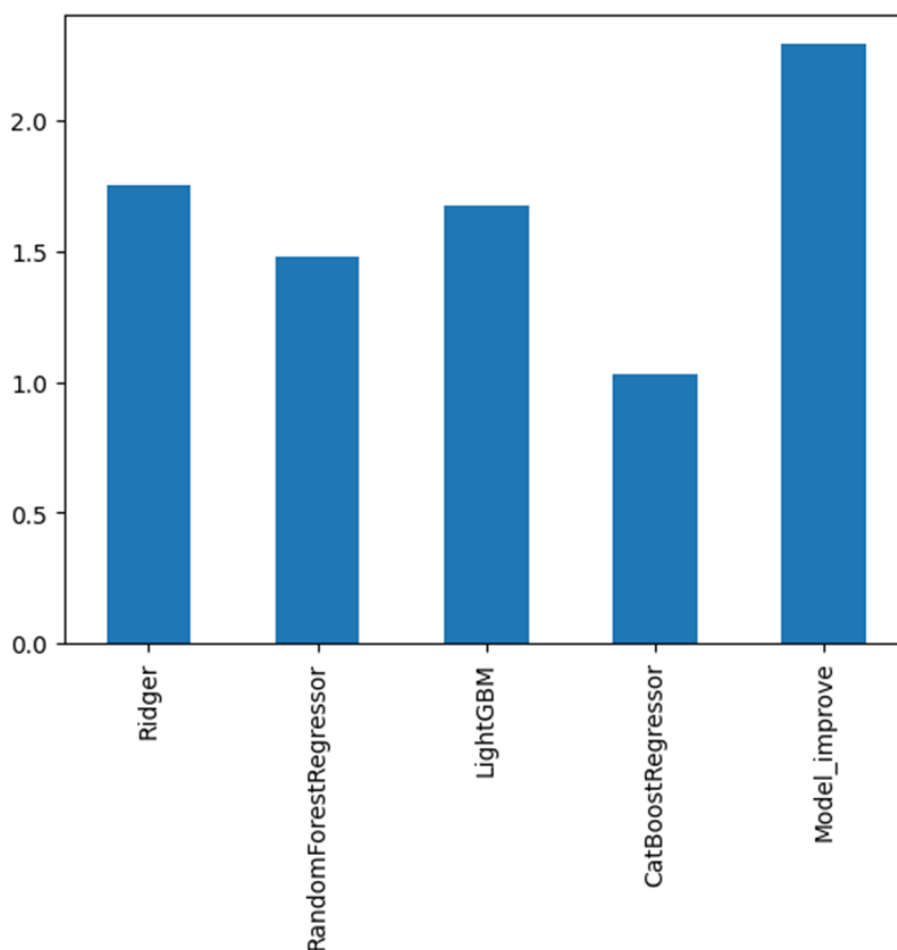


Рисунок 14 – Сравнение MAE различных моделей на обучающей выборке

**Вывод:** На данном этапе было рассмотрено и обучено 5 моделей: Ridger (линейная модель), RandomForestRegressor (случайный лес), LightGBM (градиентный бустинг), CatBoostRegressor (градиентный бустинг), Model\_improve (трехслойная полносвязная нейронная сеть). CatBoostRegressor рассматривали с дефолтными значениями и с подбором гиперпараметров. Модель с дефолтными значениями показала результат лучше, поэтому в итоговый рейтинг включили ее. Из рассмотренных моделей лучше всего себя показала модель градиентного бустинга CatBoostRegressor, на дефолтных параметрах. Полученный значение метрики MAE соответствует условию поставленной задачи, она менее 1,2.

### 3.4. Тестирование модели

Обучаем модель CatBoostRegressor, затем построим график важности

признаков для данной модели (рисунок 15).

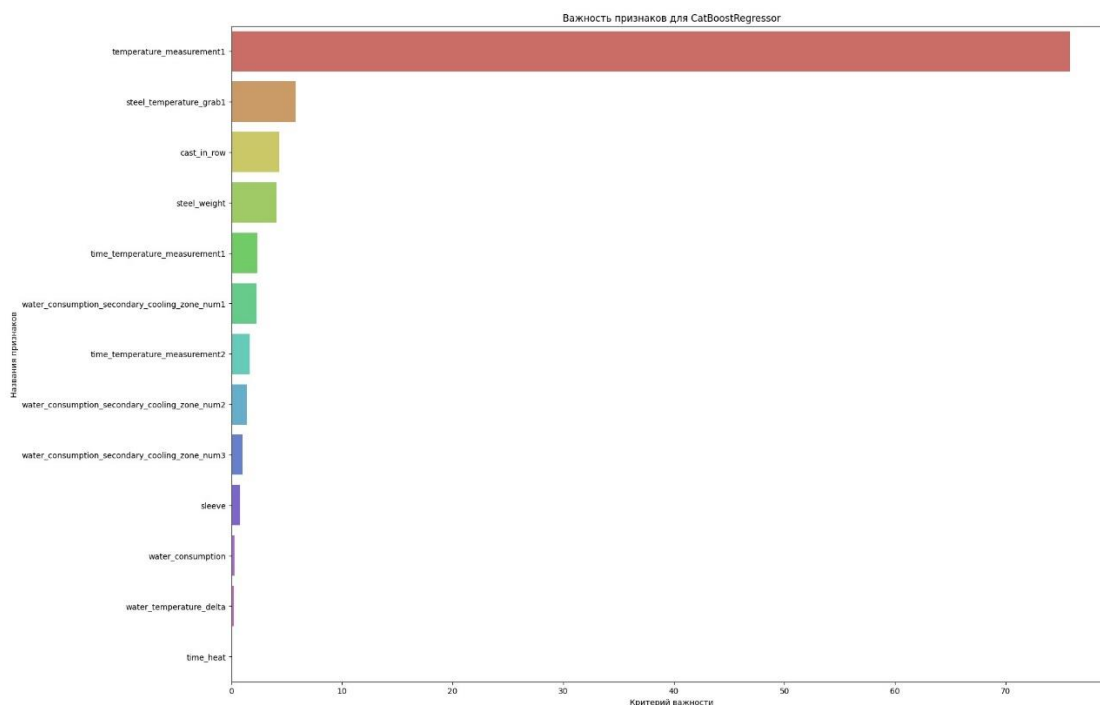
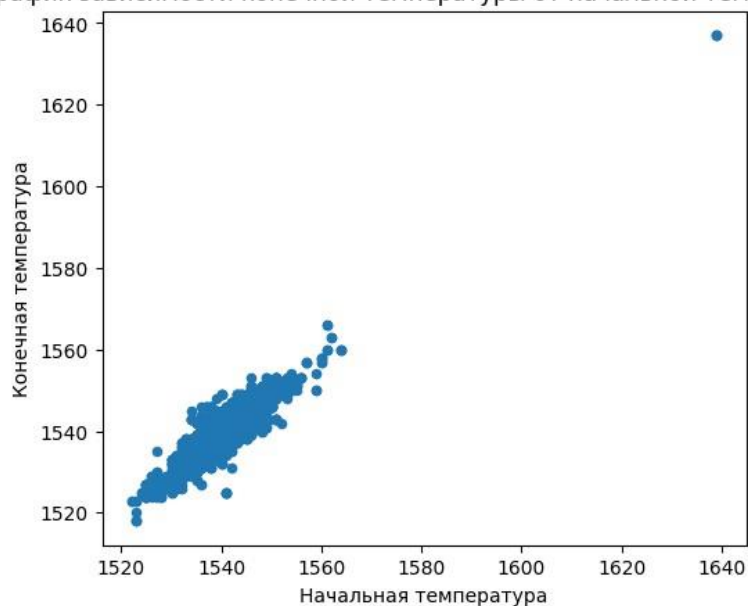


Рисунок 15 – График важности признаков для модели CatBoostRegressor

Из полученных данных видим, что наиболее важным признаком для модели CatBoostRegressor является начальная температура.

Построим график зависимости выходной температуры от начальной температуры (рисунок 16).

График зависимости конечной температуры от начальной температуры



## Рисунок 16 – График зависимости конечной температуры от начальной

Наблюдается следующий тренд: при увеличении начальной температуры увеличивается конечная температура. Также можно видеть, что есть скученность значений в центре графика (при значении начальной температуры в пределах 1530 -1550) и достаточно большой разброс значений вокруг.

### **Вывод:**

- протестирована модель CatBoostRegressor, показавшая лучший результат на обучающей выборке, MAE на тестовой выборке показало результат, удовлетворяющий поставленной задаче:  $MAE < 1,2$ ;
- определены признаки, которые были наиболее значимы для модели CatBoostRegressor, где наиболее важный признак – начальная температура;
- построен график зависимости конечной температуры от начальной.

### **Вывод по Главе 3**

Задачей данной работы было построение модели, которая будет предсказывать температуру сплава. Это необходимо, чтобы оптимизировать производственные расходы металлургического комбината, за счет уменьшения потребления электроэнергии на этапе обработки стали.

Для решения поставленной задачи было выполнено следующее:

1. Проведена предварительная обработка данных и исследовательский анализ датасета, сгенерирован ряд новых признаков (например время нагрева).
2. Из таблицы с замерах температуры был выделен целевой признак – конечная температура (`finish_temperature`).
3. Разделен итоговый датасет на признаки и целевой признак – конечная температура (`finish_temperature`), которую мы пытаемся предсказать в данном проекте. А также выборки разделены на обучающую и тестовую в соотношении 3:1.
4. Рассмотрены, обучены и подобраны гиперпараметры для 5 моделей: Ridger



(линейная модель), RandomForestRegressor (случайный лес), LightGBM (градиентный бустинг), CatBoostRegressor (градиентный бустинг), Model\_improve (трехслойная полносвязная нейронная сеть). CatBoostRegressor рассматривали с дефолтными значениями и с подбором гиперпараметров. Модель с дефолтными значениями показала результат лучше, поэтому в итоговый рейтинг включили ее. Из рассмотренных моделей лучше всего себя показала модель градиентного бустинга CatBoostRegressor на дефолтных параметрах. Полученный значение метрики MAE соответствует условию поставленной задачи, она менее 1,2.

5. На заключительном этапе:

- протестирована модель CatBoostRegressor, показавшая лучший результат на обучающей выборке, MAE на тестовой выборке показало результат удовлетворяющий поставленной задаче:  $MAE < 1,2$ ;
- определены признаки, которые были наиболее значимы для модели CatBoostRegressor. Наиболее важный признак – начальная температура;
- построен график зависимости конечной температуры от начальной.

Таким образом задача разработки модели тепловой работы кристаллизатора машины непрерывного литья заготовок выполнена. Данная модель позволяет прогнозировать конечную температуру сплава с метрикой качества  $MAE < 1,2$ .

## **4. Разработка веб-приложения**

### **4.1. Определение требований**

Сервис для работы с промышленными данными будет представлять собой веб-приложение. В ходе проектирования приложения были определены функциональные и нефункциональные требования к системе.

#### **Функциональные требования**

1. Система должна предоставлять возможность авторизации пользователя.
2. Система должна предоставлять возможность пользователю дать обратную связь по результатам анализа промышленных данных.

#### **Нефункциональные требования**

1. Система должна быть веб-приложением с REST API.
2. Система должна иметь возможность обработать файлы, размером до 5 мегабайт.
3. Система должна иметь возможность обработать файлы с расширением .csv.

На рисунке 17 изображена диаграмма вариантов использования веб-приложения для работы с промышленными данными о тепловых процессах кристаллизатора МНЛЗ. Главный актер – пользователь веб-приложения, также к приложению может подключиться другое программное обеспечение по REST API, исходя из этого можно представить еще одного актера – клиентская программа.

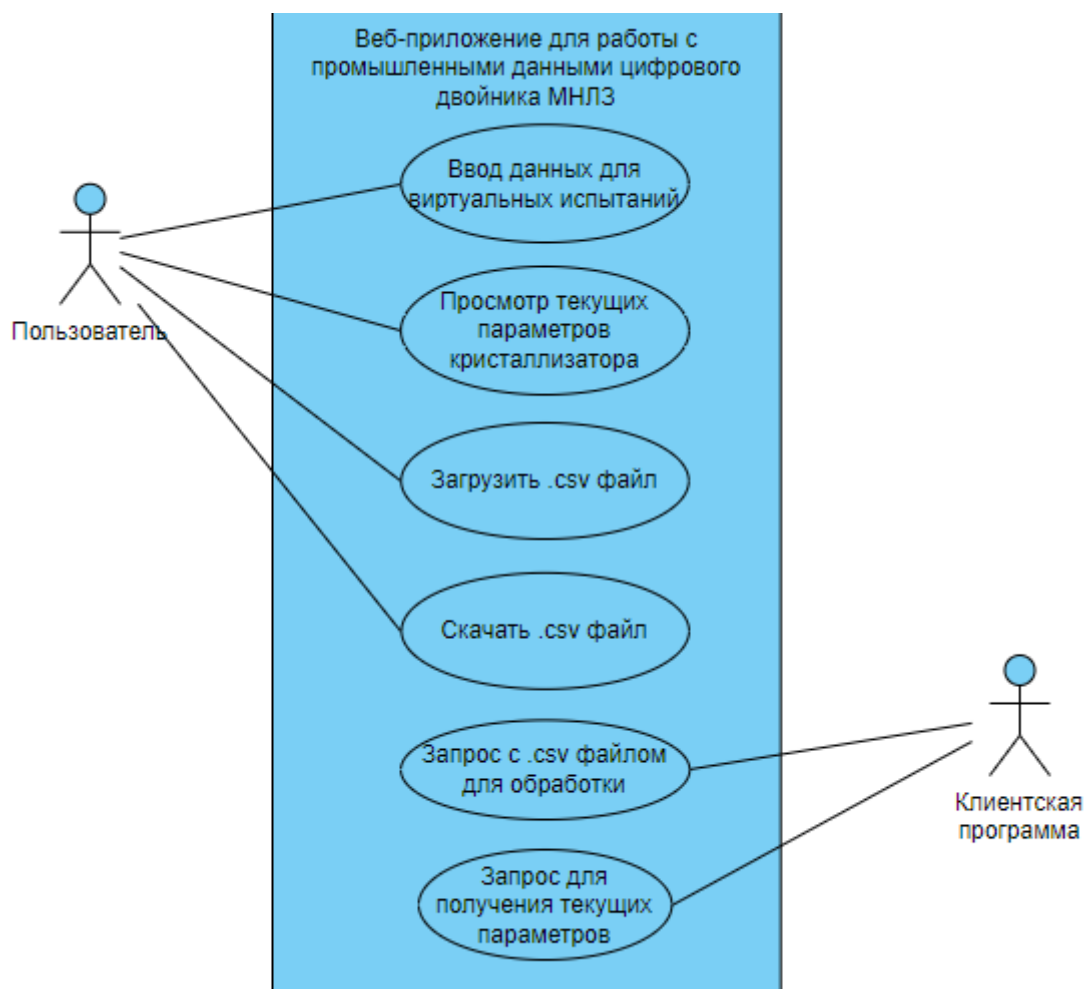


Рисунок 17 – UML-диаграмма вариантов использования

Актерам доступны следующие варианты использования системы.

1. Вариант использования «Ввод данных для виртуальных испытаний». Пользователь вводит параметры кристаллизатора, которые обрабатываются существующей моделью машинного обучения и прогнозируют состояние кристаллизатора МНЛЗ при введенных значениях.
2. Вариант использования «Загрузка файла .csv, содержащего промышленные данные и начало обработки». Пользователь может загрузить файл .csv, содержащий промышленные данные, для которого будет производиться аналитика.
3. Вариант использования «Просмотр текущих параметров кристаллизатора». Пользователь может посмотреть актуальное состояние кристаллизатора, извлекая данные полей цифрового

паспорта.

4. Вариант использования «Скачать файл .csv». Пользователь может скачать результирующие данные аналитики в формате .csv.

## 4.2. Архитектура системы

Архитектуру систему можно декомпозировать на несколько компонентов, которые взаимодействуют между собой для реализации требуемого функционала. На рисунке 18 предоставлена диаграмма компонентов системы.

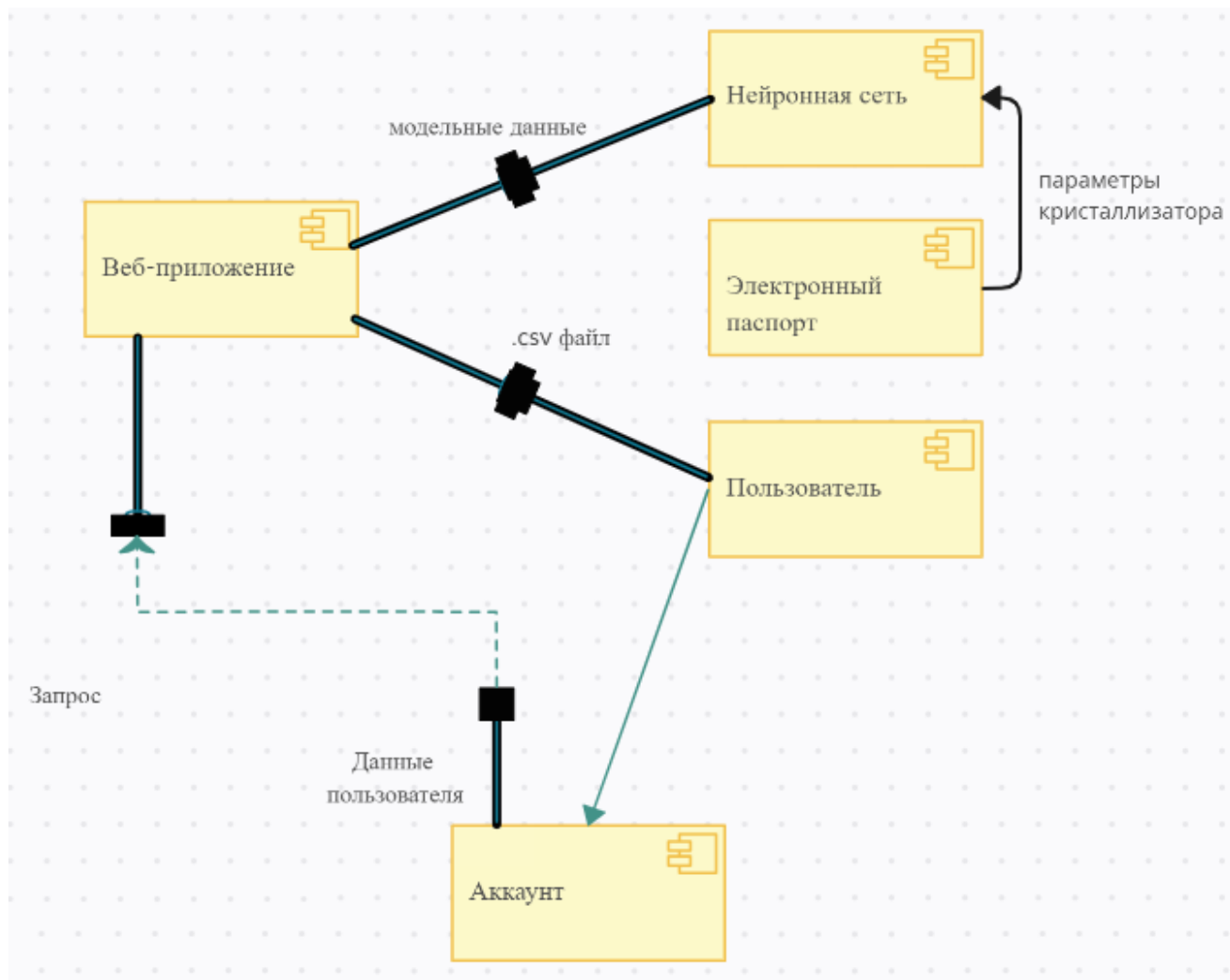


Рисунок 18 – Диаграмма компонентов системы

Рассмотрим каждый компонент.

1. Нейронная сеть – компонент обученной и описанной в предыдущей главе модели нейронной сети.

2. Электронный паспорт – агрегатор данных цифрового двойника, хранящий паспортные данные и архив расчетных, модельных данных

кристаллизатора МНЛЗ.

3. Пользователь – компонент, хранящий данные о пользователе, его роли и параметрах авторизации.

4. Аккаунт – компонент, являющийся частью базы данных и системы авторизации.

5. Веб-приложение – компонент, отвечающий за организацию передачи данных конечному пользователю, и выполнять операции для удовлетворения клиентских запросов.

Основная деятельность системы представляется диаграммой (рисунок 19), где показывается последовательность действий, происходящих при загрузке файла промышленных данных .csv или ввода параметров для производства виртуального испытания.

#### **4.3. Проектирование и реализация интерфейса веб-приложения**

Важной частью веб-приложения является графический интерфейс, которым пользователям предоставляется возможность удобного и доступного способа взаимодействия с системой, независимо от операционных систем, и параметров устройств.

Графический интерфейс состоит из нескольких веб страниц, которые играют роль представлений в архитектуре приложения:

1. Страница авторизация – страница, где отображается форма с полями логина и пароля, после успешного заполнения которой осуществляется вход в систему, и авторизованный пользователь сможет воспользоваться функциями приложения (рисунок 19).

## Кристаллизатор прогноз

Логин

Пароль

**Вход**

Рисунок 19 – Форма авторизации

2. Главная страница – основной функцией системы является регрессионный анализ состояния кристаллизатора МНЛЗ на основании загруженных промышленных данных. На главной странице, авторизованным пользователям представляется возможность загрузить файл и ожидать результат обработки, или выгрузки полученных данных (рисунок 20).

Кристаллизатор прогноз Главная Ввод параметров вручную Загрузка промышленных данных [Вход](#)

### Прогноз температурных значений кристаллизатора МНЛЗ

[Ввод параметров вручную](#)

[Загрузка промышленных данных](#)

Рисунок 20 – Главная страница

3. Форма ввода данных кристаллизатора вручную для получения прогнозных значений. На рисунке 21 представлен макет страницы.

## Прогноз температурных значений

Введите параметры кристаллизатора:

cast_in_row
sleeve
steel_weight
steel_temperature_grab1
water_consumption
water_temperature_delta
water_consumption_secon
water_consumption_secon
water_consumption_secon
time_temperature_measur
temperature_measurement1
time_temperature_measur
time_heat

Рассчитать

Рисунок 21 – Ввод параметров кристаллизатора вручную

### 4. Форма загрузки файлов с промышленными данными (рисунок 22)

## Загрузка промышленных данных

Выберите файл с промышленными данными

Выберите файл

Выбрать файлы | Файл не выбран

Выбран файл

Выберите файл | Файл не выбран

Рассчитать

Рисунок 22 – Форма загрузки файлов с промышленными данными

### 4.4. Проектирование REST API веб-приложения

Кроме графического интерфейса, веб-приложение имеет программный интерфейс, который реализуется на базе асинхронного REST API. Далее перечислим запросы, доступные клиентским программам.

1. POST api/login. С помощью данного запроса, можно осуществлять авторизацию уже зарегистрированного пользователя. В теле запроса должны быть поля login и password, в которых отправляются логин и пароль пользователя соответственно. В результате успешного запроса, клиенту отправляется ответ в виде JSON с токенами для авторизации по JWT [19]. HTTP код ответа – «202

(Accepted)».

2. PUT `api/data`. Данным запросом пользователь может отправить файл с промышленными данными для обработки в модели машинного обучения. В результате успешного запроса, сервер отправит файл с прогнозными данными. HTTP код ответа – «201 (Created)».

3. GET `api/manual/`. С помощью данного запроса, можно получить прогнозные данные по введенным вручную параметрам кристаллизатора. HTTP код ответа – «200 (OK)».

#### 4.5. Программные средства реализации

Для реализации веб-приложения для анализа промышленных данных и взаимодействия с электронным паспортом кристаллизатора с использованием нейросетевых технологий был использован язык программирования Python 3.9. Веб-интерфейс приложения разработан с помощью HTML, CSS. Для написания кода и его отладки была использована среда разработки PyCharm.

В процессе разработки веб-приложения были использованы следующие библиотеки и фреймворки.

1. Django (4.1.6) [17]. Веб-фреймворк на базе Python с открытым исходным кодом, который следует архитектурному шаблону `model-view-controller`.

2. Django REST framework (3.14.0) [18]. Фреймворк, для работы со стандартными моделями Django для создания гибкого и мощного API для проекта.

3. Simple JWT (5.2.2) [19]. Плагин фреймворка Django REST framework, реализующий аутентификации через JSON Web Token.

4. Celery (5.2.7) [20]. Библиотека, реализующая асинхронную очередь задач, основанная на распределенной передаче сообщений.

5. RabbitMQ (3.11.16) [21]. Система обмена сообщениями, которая обеспечивает связь и взаимодействие между различными компонентами программного обеспечения.

6. Tailwind (3.3.2) [22]. Современный CSS-фреймворк для разработки



пользовательского интерфейса путем комбинирования набора готовых классов.

#### 4.6. Тестирование веб-приложения

Для проверки работы веб-сервера было проведено функциональное тестирование для веб-интерфейса, и для REST API отдельно. Набор функциональных тестов представлена веб-интерфейса в таблице 1.

Таблица 1 – Результаты функционального тестирования веб-интерфейса

Название теста	Входные данные	Ожидаемый результат	Тест пройден?
Загрузка файла с промышленными данными	Пользователь заходит на страницу загрузки файла, загружает датасет и нажимает на кнопку «Рассчитать»	Система возвращает прогнозные значения и пользователь сможет увидеть результат или скачать выходной файл	Да
Загрузка файла некорректного формата	Пользователь заходит на главную страницу, загружает файл некорректного расширения, и нажимает на кнопку «Рассчитать»	Система возвращает ошибку, и отображается сообщение о правильном формате файла	Да
Ввод параметров кристаллизатора вручную	Пользователь вводит ручную параметры кристаллизатора	Система возвращает прогнозные значения и пользователь сможет увидеть результат или скачать выходной файл	Да
Авторизация пользователя	Пользователь вводит логин и пароль, и нажимает на кнопку «Вход»	Система осуществляет авторизацию пользователя	Да
Некорректный пароль при авторизации	Пользователь вводит некорректные логин или пароль, и нажимает на кнопку «Вход»	Система должна отказать запрос и отображать ошибку неверного пароля	Да

Для тестирования API была использована программа Postman, в таблице 2 представлен набор тестов REST API сервиса.

Таблица 2 – Результаты функционального тестирования REST API

Название теста	Входные данные	Ожидаемый результат	Тест пройден?
Авторизация и запрос JWT токена	Клиент отправляет запрос GET по адресу /api/login с параметрами логина и пароля	Ответ с данными аккаунта, и JWT токен, который используется для авторизации при дальнейших запросах, в формате JSON	Да
Запрос на отправку файла с промышленными данными для обработки в модели машинного обучения	Пользователь отправляет PUT api/data запрос с файлом с промышленными данными для обработки в модели машинного обучения	Сервер отправит файл с прогнозными данными в формате JSON	Да
Запрос на получение прогнозных данных по введенным вручную параметрам кристаллизатора	С помощью запроса GET api/manual/ пользователь получает прогнозные данные по введенным вручную параметрам кристаллизатора	Сервер отправит файл с прогнозными данными в формате JSON	Да

#### Вывод по Главе 4

В рамках данной работы были определены функциональные и нефункциональные требования к веб-интерфейсу, выполнено его проектирование и реализация, а также спроектировано REST API веб-приложение.

## ЗАКЛЮЧЕНИЕ

В Главе 1 описаны предметная область и теоретическая часть моделирования тепломассопереноса в кристаллизаторе машины непрерывного литья заготовок. Также был произведен обзор литературных источников, которые легли в основу данной работы.

В Главе 2 была определена структура электронного паспорта цифрового двойника кристаллизатора МНЛЗ. За основу цифрового двойника кристаллизатора машины непрерывного литья заготовок принята комбинированная типовая архитектура, где в роли сущности сбора данных выступает электронный паспорт кристаллизатора, сущностью управления является модель машинного обучения, а в роли домена пользователя – веб-приложение для взаимодействия с обученной моделью машинного обучения для обработки промышленных данных.

В Главе 3 были собраны и предварительно обработаны промышленные данные для обучения нейронной сети, выбрана адекватная модель машинного обучения, проведено ее обучение и тестирование, разработано веб-приложение для визуализации модельных данных переноса тепла в широких и узких стенках кристаллизатора машины непрерывного литья заготовок, а также произведено тестирование приложения.

В Главе 4 были определены функциональные и нефункциональные требования к веб-интерфейсу, выполнено его проектирование и реализация, а также спроектировано REST API веб-приложение

В будущем планируется улучшение результатов предсказаний нейронной сети путем дополнительного обучения на более широком датасете, и увеличения размера входных данных, для чего требуется больше вычислительной мощности. Также можно расширить функционал веб-приложения добавлением новых возможностей.

## БИБЛИОГРАФИЧЕСКИЙ СПИСОК

1. Ячиков, И.М. Моделирование тепловых полей в кристаллизаторе с щелевыми каналами охлаждения / И.М. Ячиков, Н.А. Феоктистов, А.С. Савинов, Т.И. Шафиков, И.В. Михалкина // ТиТМП. – 2022. – №1 (40). – С. 12–18.
2. Емельянов, В.А. Тепловая работа машин непрерывного литья заготовок: М.: Metallurgy, 1988. – 53 с.
3. Расчетно-аналитическое исследование тепловых процессов в кристаллизаторе / Буланов Л.В., Асанова Р.Ф., Волегова В.Е., Лобанов Е.П. // Сталь. – 1999. – №9. – С. 24 – 26.
4. Topolskaia, I. Use of Digital Twins for the Technological Equipment Condition Monitoring / I. Topolskaia, V. Pochinskaia and J. Plaksina // 2023 International Russian Automation Conference (RusAutoCon). – 2023. – P. 675 – 679.
5. Лукин, С.В. Математическое моделирование теплообмена сляба с кристаллизатором машины непрерывного литья заготовок при динамических режимах разлива / С.В. Лукин, Н.И. Шестаков, В.В. Мухин, В.И. Славов // Вестник Череповецкого государственного университета. – 2012. – №2 (39). – С. 22 – 30.
6. Повитухин, С.А. Математическое моделирование процесса охлаждения слитка в кристаллизаторе машины непрерывного литья заготовок // Вестник ТГТУ. – 2009. – Том 15. – № 3. – С. 672 – 681.
7. Прохоров, А.В. Математическая модель отливки металла в кристаллизаторе // Международный журнал прикладных и фундаментальных исследований. – 2015. – № 8. – С. 401– 401.
8. Бодяко, А.М. Тепловой режим кристаллизатора / А. М. Бодяко, В. Ф. Бевза // Литье и металлургия. – 2001. – № 2. – С. 68 – 71.
9. Гусев, М.П. Концепт цифрового двойника процесса непрерывного литья

- слитков / М.П. Гусев, К.Н. Анисимов, С.В. Зарубин, А.М. Лонгинов, И.К. Ужинский // ISCON. – 2018. – С.223–226.
10. Гусев, М.П. Результаты опытного тестирования оптоволоконных измерителей температуры в имитаторе кристаллизатора // Научно-технический прогресс в черной металлургии - 2019 : Материалы IV Международной научной конференции / отв. за выпуск Н. Л. Балабанова. – Череповец: ЧГУ, 2019. – С. 48–54.
  11. Виноградов, В.П., Разработка и испытание измерительного комплекса для контроля тепловых параметров работы кристаллизатора МНЛЗ // Вестник ЧГУ. – 2020. – № 4 (97). – С. 9–18.
  12. Cemernek, D. Machine learning in continuous casting of steel: a state-of-the-art survey. // Intelligent Manufacturing. – 2022. – vol. 33. – P. 1561–1579.
  13. Шиболденков, В.А. Построение технологического ландшафта инновационных решений на основе интеллектуального анализа данных / В.А. Шиболденков, Д.М. Кхан // Искусственный интеллект и принятие решений. – 2024. – № 1. – С. 79–91.
  14. Абрамова, А.А. Анализ использования больших данных для принятия решений в промышленной сфере // Экономика и качество систем связи. – 2023. – №3 (29). – С. 29-31.
  15. Ismail, A. Manufacturing process data analysis pipelines: a requirements analysis and survey // Big Data. – 2019. – vol. 6, №1. – P. 128 –130.
  16. Предварительный национальный стандарт Российской Федерации "ПНСТ 430-2020. Умное производство. Двойники цифровые производства. Часть 2. Типовая архитектура" от 01.01.2021 № ПНСТ 430-2020 // Официальный интернет-портал правовой информации. – 2021.
  17. Django documentation. [Электронный ресурс] URL: <https://docs.djangoproject.com/en/4.1/> (дата обращения: 20.04.2024 г.).

18. Django REST framework documentation. [Электронный ресурс] URL: <https://www.django-rest-framework.org/tutorial/quickstart/> (дата обращения: 02.04.2024 г.).
19. Simple JWT documentation. [Электронный ресурс] URL: [https://django-rest-framework-simplejwt.readthedocs.io/en/latest/getting\\_started.html](https://django-rest-framework-simplejwt.readthedocs.io/en/latest/getting_started.html) (дата обращения: 15.04.2024 г.).
20. Celery documentation. [Электронный ресурс] URL: <https://docs.celeryq.dev/en/stable/> (дата обращения: 22.04.2024 г.).
21. RabittMQ documentation. [Электронный ресурс] URL: <https://www.rabbitmq.com/documentation.html> (дата обращения: 22.04.2024 г.).
22. Tailwind documentation. [Электронный ресурс] URL: <https://tailwindcss.com/docs/> (дата обращения: 25.05.2023 г.).

## ПРИЛОЖЕНИЯ

### Приложение А. Реализация модели машинного обучения

Функция для получения информации о датасете (вывод первых пяти строк, общей информации методом, описания информации в столбцах, наличия явных дубликатов, количества пропусков) представлена в листинге А.1 приложения А.

#### Листинг А.1 – Модуль разведки данных

```
RANDOM_STATE = 40324
engine = create_engine(connection_string)
file_path = '/content/ccm_dataset.csv'
data = pd.read_csv(file_path, sep=';')
data.info()
data.describe()
data.info()

def get_info(data):
    display(f'Вывод первых 5-ти строк датасета', data.head())
    display(data.info())
    print('Количество явных дубликатов:', data.duplicated().sum())
    print('Количество пропусков:\n', data.isna().sum())
    display(data.describe())

data_total=data
features = data_total.drop('temperature_measurement2', axis=1)
target = data_total['temperature_measurement2']
data_total.info()
print(features.shape)

target.shape
features_train, features_test, target_train, target_test =
train_test_split(features, target, test_size=0.25, random_state=RANDOM_STATE)

print(features_train.shape)
print(target_train.shape)
print(features_test.shape)
print(target_test.shape)
```

Модель CatBoostRegressor представлена на листинге А.2 приложения А. Произведен анализ значений MAE на кроссвалидации, а также подбор гиперпараметров.

## Листинг А.2 – Модуль обучения и выбора модели

```
%%time
columns = features_train.columns
column_transformer_rd = make_column_transformer((StandardScaler(),columns),
                                                remainder = 'passthrough')

model_rd = Ridge(random_state=RANDOM_STATE)
pipeline_rd = make_pipeline(column_transformer_rd, model_rd)
pipeline_rd
param_rd = {
    'ridge__alpha': range(1,15,1)
}
grid_rd = GridSearchCV(pipeline_rd, param_rd, cv=3,
                      scoring='neg_mean_absolute_error')
grid_rd.fit(features_train, target_train)
ridge_best_mae = round(abs(grid_rd.best_score_), 3)
ridge_best_model = grid_rd.best_estimator_
ridge_best_params = grid_rd.best_params_
print(f'MAE лучшей модели Ridge на обучающей выборке: {ridge_best_mae}')
print(f'Гиперпараметры лучшей модели: {ridge_best_params}')

model_cb_1 = CatBoostRegressor(random_seed=RANDOM_STATE, silent=True)

param_cb_1 = {
    'learning_rate':[0.1, 0.3],
    'iterations': [40, 50, 60, 80],
    'depth': range(1, 10, 1),
}

grid_cb = RandomizedSearchCV(model_cb_1, param_cb_1, cv=3, n_iter=20,
                             scoring='neg_mean_absolute_error', n_jobs=-1,
                             random_state=RANDOM_STATE)
grid_cb.fit(features_train, target_train)

cb_best_mae = abs(grid_cb.best_score_)
cb_best_model = grid_cb.best_estimator_
cb_best_params = grid_cb.best_params_

print(f'MAE лучшей модели CatBoostRegressor на обучающей выборке: {cb_best_mae}')
print(f'Гиперпараметры лучшей модели: {cb_best_params}')
```

Класс для задания архитектуры нейронной сети представлен на листинге А.3 приложения А. На выходном слое расположен 1 нейрон. На входе количество нейронов равно количеству признаков.



### Листинг А.3 – Класс архитектуры нейронной сети

```
random.seed(RANDOM_STATE)
np.random.seed(RANDOM_STATE)
torch.manual_seed(RANDOM_STATE)
torch.use_deterministic_algorithms(True)
scaler = StandardScaler()
num_columns = features_train.columns
features_train_st = features_train.copy()
features_test_st = features_test.copy()
features_train_st[num_columns] =
scaler.fit_transform(features_train[num_columns])
features_test_st[num_columns] = scaler.transform(features_test[num_columns])
display(features_train_st.head())
features_test_st.head()
x_train = torch.FloatTensor(features_train_st.values)
x_test = torch.FloatTensor(features_test_st.values)
y_train = torch.FloatTensor(target_train.values)
y_test = torch.FloatTensor(target_test.values)
y_test.size()
class Model_improve(nn.Module):
    def __init__(self, n_in_neurons, n_hidden_neurons_1, n_hidden_neurons_2,
n_out_neurons, dropout_1, dropout_2, activation_1, activation_2):
        super(Model_improve, self).__init__()
        self.fc1 = nn.Linear(n_in_neurons, n_hidden_neurons_1)
        self.a1 = activation_1
        self.drop1 = nn.Dropout(dropout_1)
        self.fc2 = nn.Linear(n_hidden_neurons_1, n_hidden_neurons_2)
        self.a2 = activation_2
        self.drop2 = nn.Dropout(dropout_2)
        self.fc3 = nn.Linear(n_hidden_neurons_2, n_out_neurons)
    def forward(self, x):
        x = self.drop1(self.a1(self.fc1(x)))
        x = self.drop2(self.a2(self.fc2(x)))
        x = self.fc3(x)
        return x
params = {
    'module__dropout_1': [0, 0.3],
    'module__dropout_2': [0, 0.1],
    'module__n_hidden_neurons_1': [20],
    'module__n_hidden_neurons_2': [15],
    'module__activation_1': [nn.ReLU()],
    'module__activation_2': [nn.ReLU()],
    'module__n_in_neurons': [x_train.shape[1]],
    'module__n_out_neurons': [1],
    'batch_size': [50]
}
def mae(y_true, y_pred):
    return mean_absolute_error(y_true, y_pred)
```

На листинге А.4 приложения А представлен код сравнения выбранной модели с константной, которая предсказывает среднее значение, построен график важности признаков для модели CatBoostRegressor, а также график зависимости конечной температуры кристаллизатора от начальной.

#### Листинг А.4 –Тестирование модели

```
%%time
start_time = time.time()
predictions_cb = model_cb.predict(features_test)
end_time = time.time()
pred_time = end_time - start_time
print('MAE', mean_absolute_error(target_test, predictions_cb).round(3))
print('R2', r2_score(target_test, predictions_cb).round(3))
print('Время предсказания', pred_time)
model_dr = DummyRegressor(strategy="mean")
model_dr.fit(features_train, target_train)
predictions = model_dr.predict(features_test)
print('MAE', mean_absolute_error(target_test, predictions).round(3))
plot_data = {
    'features': features_test.columns,
    'feature_importance': model_cb.feature_importances_,
}
plot_data = pd.DataFrame(plot_data)
plot_data.sort_values(by='feature_importance', ascending=False, inplace=True)
plt.figure(figsize=(20, 15))
sns.barplot(x=plot_data['feature_importance'],
y=plot_data['features'], palette='hls')
plt.title('Важность признаков для CatBoostRegressor')
plt.xlabel('Критерий важности')
plt.ylabel('Названия признаков')
plt.show()
data_temp = {
    'start_temperature': features_test['temperature_measurement1'],
    'finish_temperature': target_test,
}
data_temp = pd.DataFrame(data_temp)
plt.figure(figsize=(20, 15))
data_temp.plot(x='start_temperature', y='finish_temperature', kind='scatter',
figsize=(6,5))
plt.title('График зависимости конечной температуры от начальной температуры')
plt.xlabel('Начальная температура')
plt.ylabel('Конечная температура')
plt.show()
```

## Приложение Б. Реализация веб-приложения

На листинге Б.1 приложения Б представлен html-код главной страницы.

### Листинг Б.1 – Главная страница

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Кристаллизатор прогноз</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scrip
t>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href=" ../main.html">Кристаллизатор прогноз</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Главная</a></li>
      <li><a href=" ../manual.html">Ввод параметров вручную</a></li>
      <li><a href=" ../upload.html">Загрузка промышленных данных</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
      <li><a href=" ../auth.html"><span class="glyphicon glyphicon-log-in"></span>
Вход </a></li>
    </ul>
  </div>
</nav>
<div class="login" align="center">
  <h2>Прогноз температурных значений кристаллизатора МНЛЗ</h2>
  <a href=" ../manual.html"><button width="300">Ввод параметров
вручную</button></a><br><br>
  <a href=" ../upload.html"><button width="300">Загрузка промышленных
данных</button></a>
  </form>
  <br>
  <br>
  <b> {{ prediction_text }} </b>
</div>
</body>
</html>
```

На листинге Б.2 приложения Б представлен html-код страницы авторизации.

### Листинг Б.2 –Страница авторизации

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8" />
  <meta name="viewport"
    content="width=device-width,
      initial-scale=1,
      shrink-to-fit=no" />
  <link rel="stylesheet" href=
"https://maxcdn.bootstrapcdn.com/bootstrap/4.0.0/css/bootstrap.min.css" />
  <title>Авторизация</title>
</head>
<body>
  <h1 class="text-success text-center"> Кристаллизатор прогноз </h1>
  <div class="container mt-5">
    <div class="row justify-content-center">
      <div class="col-md-6">
        <div class="card">
          <div class="card-body">
            <form id="registrationForm" action="">
              <div class="form-group">
                <label for="login"> Логин </label>
                <input type="login"
                  class="form-control"
                  id="login"
                  placeholder="Логин" required />
              </div>
              <div class="form-group">
                <label for="password">
                  Пароль
                </label>
                <input type="password"
                  class="form-control"
                  id="password"
                  placeholder="Пароль"
                  required />
              </div>
              <button class="btn btn-danger">
                Вход
              </button>
            </form>
          </div>
        </div>
      </div>
    </div>
  </div>
</div>
</div>
```

```
</body>
</html>
```

На листинге Б.3 приложения Б представлен html-код страницы загрузки csv файлов промышленных данных.

### Листинг Б.3 –Страница загрузки промышленных данных

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Кристаллизатор прогноз</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script
src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js"></script>
  <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js"></scrip
t>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href=" ../main.html">Кристаллизатор прогноз</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Главная</a></li>
      <li><a href=" ../manual.html">Ввод параметров вручную</a></li>
      <li><a href=" ../upload.html">Загрузка промышленных данных</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
      <li><a href=" ../auth.html"><span class="glyphicon glyphicon-log-in">Вход
</a></li>
    </ul>
  </div>
</nav>
<h1 class="text-success"> Загрузка промышленных данных </h1>
<h3>Выберите файл с промышленными данными</h3>
<div class="mb-3">
  <label for="SmallFile" class="form-label"> Выберите файл </label>
  <input class="form-control form-control-sm" id="SmallFile" type="file"
multiple>
</div><br>
<div>
  <label for="largeFile" class="form-label"> Выбран файл </label>
  <input class="form-control form-control-lg" id="largeFile" type="file"
disabled> </div><br>
<button type="submit" class="btn btn-primary btn-block btn-large"> Рассчитать
</button>
```

```
</body>
</html>
```

На листинге Б.4 приложения Б представлен html-код страницы ручного ввода данных дальнейшего расчета с помощью модели машинного обучения.

#### Листинг Б.4 – Страница ввода данных

```
<!DOCTYPE html>
<html lang="en">
<head>
  <title>Bootstrap Example</title>
  <meta charset="utf-8">
  <meta name="viewport" content="width=device-width, initial-scale=1">
  <link rel="stylesheet"
href="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/css/bootstrap.min.css">
  <script src="https://ajax.googleapis.com/ajax/libs/jquery/3.7.1/jquery.min.js">
</script> <script
src="https://maxcdn.bootstrapcdn.com/bootstrap/3.4.1/js/bootstrap.min.js">
</script>
</head>
<body>
<nav class="navbar navbar-inverse">
  <div class="container-fluid">
    <div class="navbar-header">
      <a class="navbar-brand" href="#">Кристаллизатор прогноз</a>
    </div>
    <ul class="nav navbar-nav">
      <li class="active"><a href="#">Главная</a></li>
      <li><a href="manual.html">Ввод параметров вручную</a></li>
      <li><a href="upload.html">Загрузка промышленных данных</a></li>
    </ul>
    <ul class="nav navbar-nav navbar-right">
      <li><a href="auth.html"><span class="glyphicon glyphicon-log-in"></span>
Вход </a></li>
    </ul>
  </div>
</nav>
<div class="login" align="center">
  <h2>Прогноз температурных значений</h2>
  <p>Введите параметры кристаллизатора:</p>
  <!-- Inputs for our ML model -->
  <form action="{ { url_for('predict') }}"method="post">

    <input type="text" name="cast_in_row" placeholder="cast_in_row"
required="required" />
    <br><input type="text" name="sleeve" placeholder="sleeve"
required="required"/>
    <br><input type="text" name="steel_weight" placeholder="steel_weight"
required="required"/>
    <br><input type="text" name="steel_temperature_grab1"
placeholder="steel_temperature_grab1" required="required"/>
```

```
<br><input type="text" name="water_consumption"
placeholder="water_consumption" required="required"/>
```

#### Окончание листинга Б.4

```
<br><input type="text" name="water_temperature_delta"
placeholder="water_temperature_delta" required="required" />
<br><input type="text"
name="water_consumption_secondary_cooling_zone_num1"
placeholder="water_consumption_secondary_cooling_zone_num1" required="required"/>
<br><input type="text"
name="water_consumption_secondary_cooling_zone_num2"
placeholder="water_consumption_secondary_cooling_zone_num2" required="required"/>
<br><input type="text"
name="water_consumption_secondary_cooling_zone_num3"
placeholder="water_consumption_secondary_cooling_zone_num3" required="required"/>
<br><input type="text" name="time_temperature_measurement1"
placeholder="time_temperature_measurement1" required="required"/>
<br><input type="text" name="temperature_measurement1"
placeholder="temperature_measurement1" required="required"/>
<br><input type="text" name="time_temperature_measurement2"
placeholder="time_temperature_measurement2" required="required"/>
<br><input type="text" name="time_heat" placeholder="time_heat"
required="required"/>
<button type="submit" class="btn btn-primary btn-block btn-
large">Рассчитать</button>
</form>
<br>
<br>
<b> {{ prediction_text }} </b>
</div>
</body>
</html>
```

На листинге Б.5 приложения Б представлен код основного приложения.

#### Листинг Б.5 – Код приложения

```
from flask import Flask, request, render_template
import pickle
from catboost import CatBoostRegressor, Pool

app = Flask( name )

import pickle
model = CatBoostRegressor()
pickle.dump(model, open('model.pkl', 'wb'))

@app.route('/')
def home():
    return render_template('main.html')
```

```
@app.route('/manual.html')
```

## Продолжение листинга Б.5

```
def manual():
    return render_template('manual.html')
@app.route('/auth.html')
def auth():
    return render_template('auth.html')
@app.route('/upload.html')
def upload():
    return render_template('upload.html')

@app.route('/predict',methods=['POST'])
def predict():

    steel_type = int(request.form["steel_type"])
    cast_in_row = int(request.form["cast_in_row"])
    sleeve = int(request.form["sleeve"])
    steel_weight = int(request.form["steel_weight"])
    steel_temperature_grab1 = int(request.form["steel_temperature_grab1"])
    water_consumption = int(request.form["water_consumption"])
    water_temperature_delta = int(request.form["water_temperature_delta"])
    water_consumption_secondary_cooling_zone_num1 =
int(request.form["water_consumption_secondary_cooling_zone_num1"])
    water_consumption_secondary_cooling_zone_num2 =
int(request.form["water_consumption_secondary_cooling_zone_num2"])
    water_consumption_secondary_cooling_zone_num3 =
int(request.form["water_consumption_secondary_cooling_zone_num3"])
    time_temperature_measurement1 =
int(request.form["time_temperature_measurement1"])
    temperature_measurement1 = int(request.form["temperature_measurement1"])
    time_temperature_measurement2 =
int(request.form["time_temperature_measurement2"])
    temperature_measurement2 = int(request.form["temperature_measurement2"])
    time_heat = int(request.form["time_heat"])
    prediction =
model.predict(['steel_type','cast_in_row','sleeve','steel_weight','steel_tempera
ture_grab1',
'water_consumption','water_temperature_delta','water_consumption_secondary_coolin
g_zone_num1',
'water_consumption_secondary_cooling_zone_num2','water_consumption_secondary_cool
ing_zone_num3',
'time_temperature_measurement1','temperature_measurement1','time_temperature_meas
urement2',
'temperature_measurement2',
'time_heat'],[steel_type,cast_in_row,sleeve,steel_weight,steel_temperature_grab1,
```



```
water_consumption,water_temperature_delta,water_consumption_secondary_cooling_zone_num1,
```

## Окончание листинга Б.5

```
water_consumption_secondary_cooling_zone_num2,water_consumption_secondary_cooling_zone_num3,
```

```
time_temperature_measurement1,temperature_measurement1,time_temperature_measurement2,
```

```
temperature_measurement2, time_heat]]) # this returns a list e.g. [127.20488798], so pick first element [0]
```

```
output = round(prediction[0], 2)
```

```
return render_template('manual.html', prediction_text=f'Для входного значения температуры {temperature_measurement1} температура на выходе: {output} градусов')
```

```
if __name__ == "__main__":
```

```
app.run()
```

## Приложение В. Макеты страниц веб-приложения

На рисунке В.1 приложения В представлен макет главной страницы.

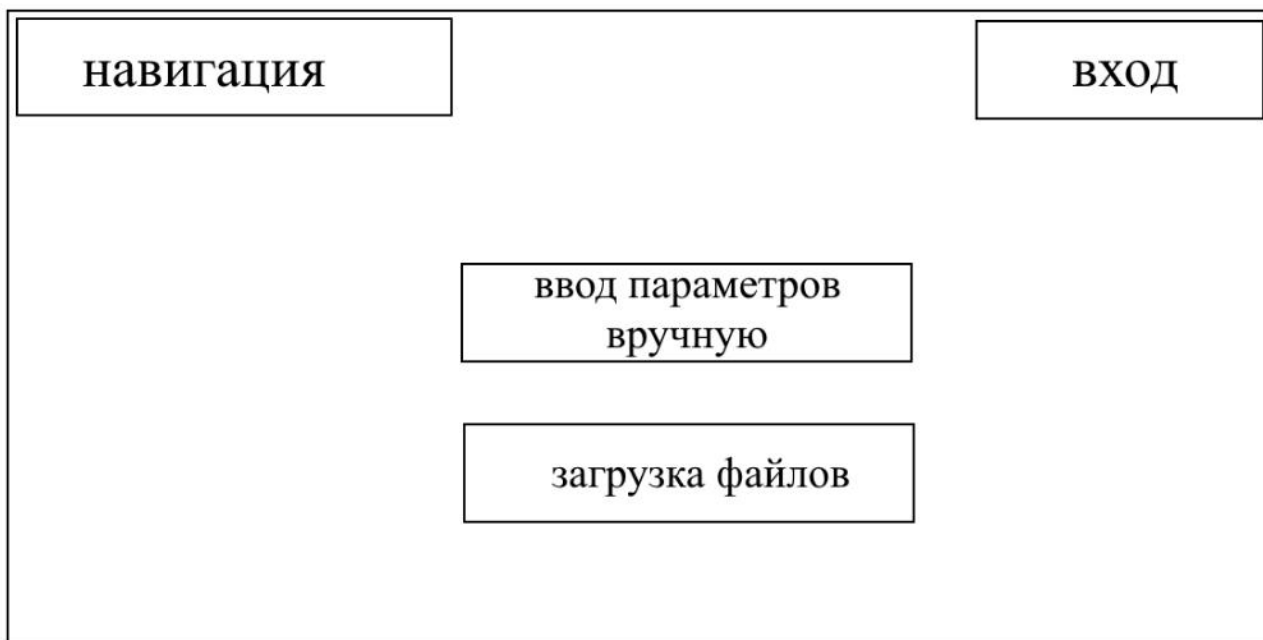


Рисунок В.1 – Макет главной страницы

На рисунке В.2 приложения В представлен макет страницы авторизации.

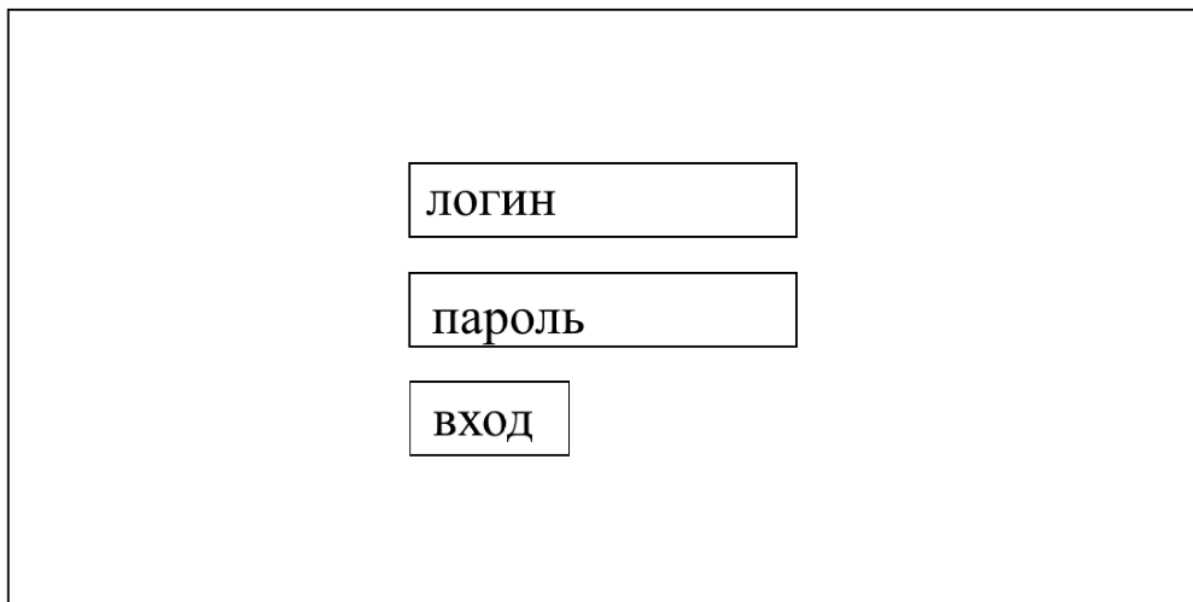


Рисунок В.2 – Макет страницы авторизации

На рисунке В.3 приложения В представлен макет страницы ввода данных для расчета вручную.



Рисунок В.3 – Макет страницы ввода данных

На рисунке В.4 приложения В представлен макет страницы загрузки файла с промышленными данными.

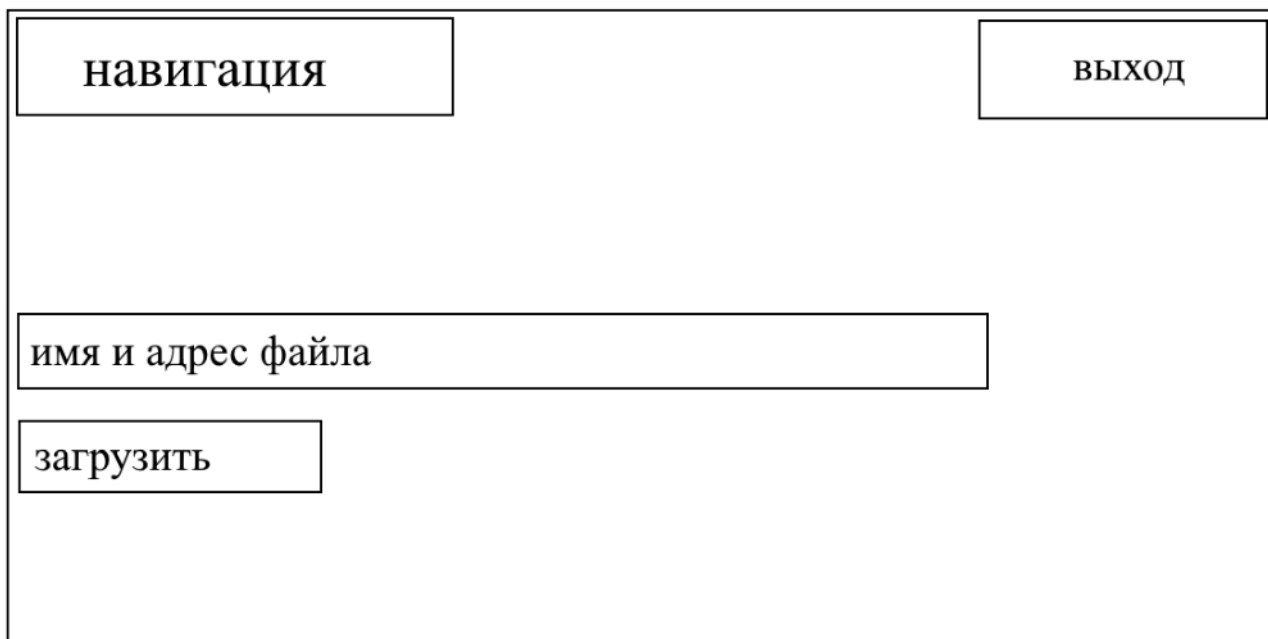


Рисунок В.4 – Макет страницы загрузки файла

## Приложение Г. Интерфейс веб-приложения

На рисунке Г.1 приложения Г представлен веб-интерфейс главной страницы.

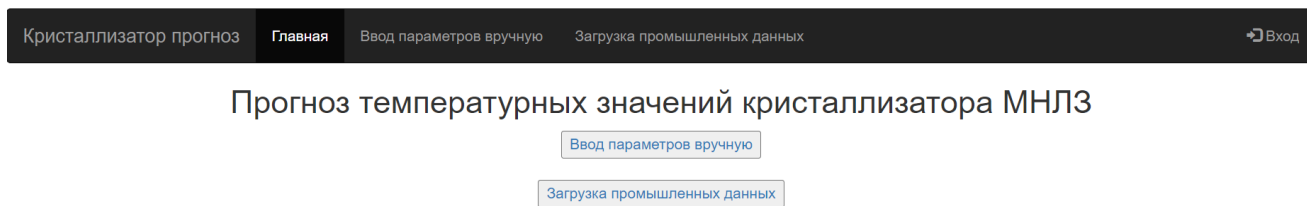


Рисунок Г.1 – Главная страница

На рисунке Г.2 приложения Г представлен веб-интерфейс формы авторизации.

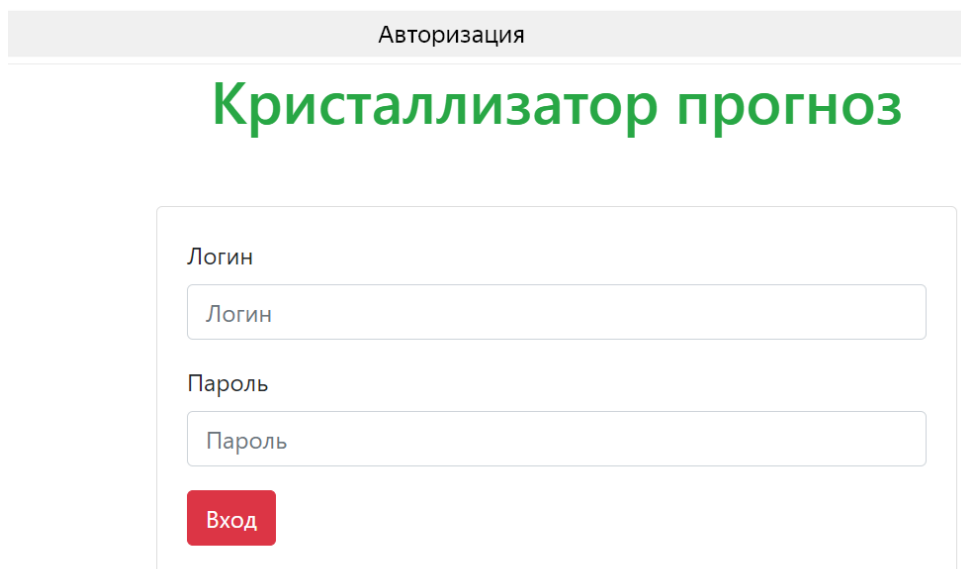


Рисунок Г.2 – Форма авторизации

На рисунке Г.3 приложения Г представлен веб-интерфейс страницы загрузки файла с промышленными данными.

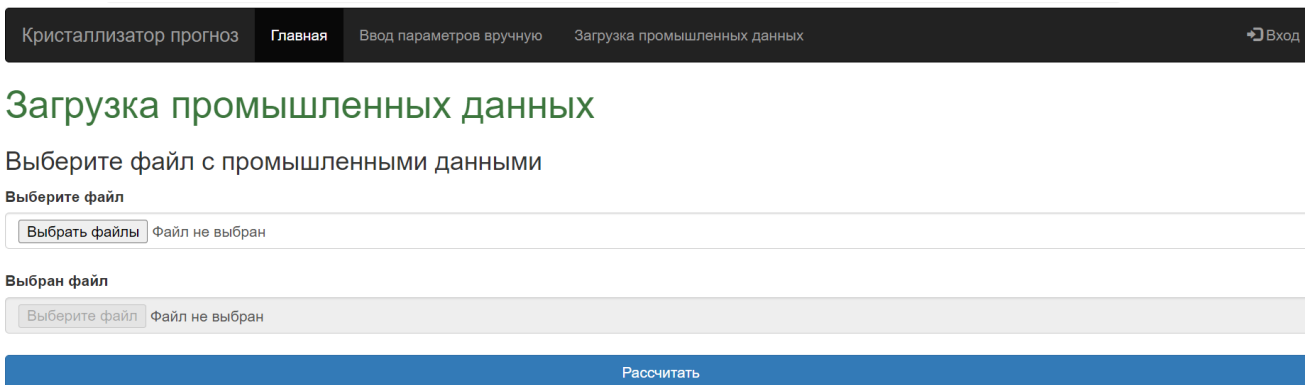


Рисунок Г.3 – Страница загрузки файла с промышленными данными

На рисунке Г.4 приложения Г представлен веб-интерфейс страницы ввода данных для прогнозирования тепловых параметров кристаллизатора.

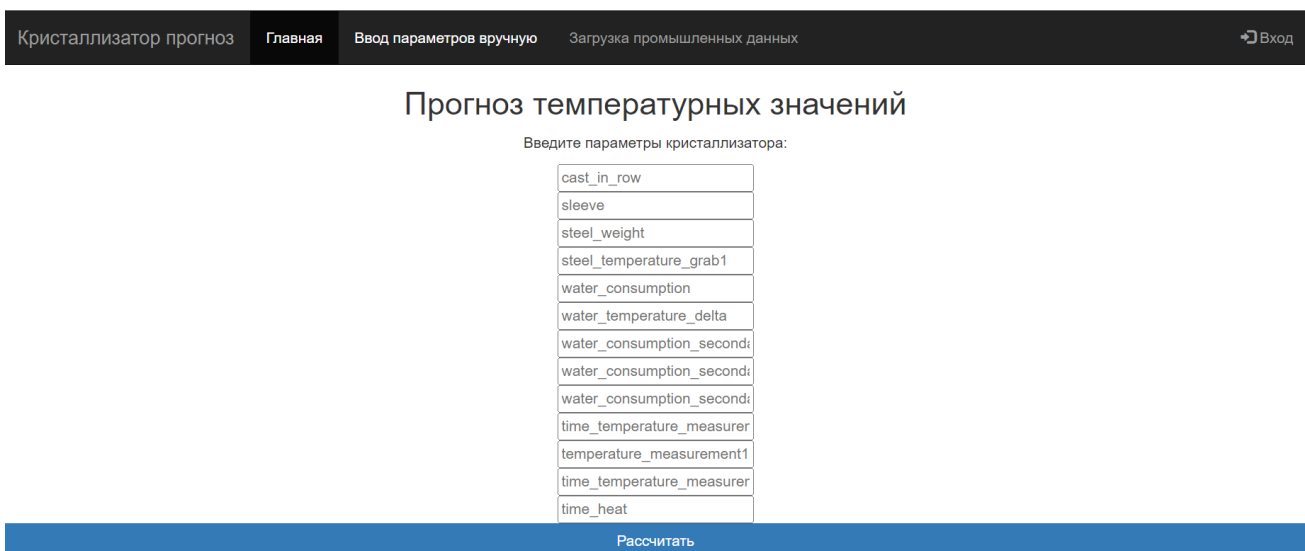


Рисунок Г.4 – Страница ввода данных