

EISSN 1694-5050
ISSN 1694-5050

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК КЫРГЫЗСКОЙ РЕСПУБЛИКИ

ИНСТИТУТ МАШИНОВЕДЕНИЯ И АВТОМАТИКИ

ПРОБЛЕМЫ АВТОМАТИКИ И УПРАВЛЕНИЯ

№3 (45)



БИШКЕК

2022

ИМА НАН КР

ISSN 1694-5050 (печатная версия)
ISSN 1694-836X (электронная версия)

НАЦИОНАЛЬНАЯ АКАДЕМИЯ НАУК КЫРГЫЗСКОЙ РЕСПУБЛИКИ

ИНСТИТУТ МАШИНОВЕДЕНИЯ И АВТОМАТИКИ

ПРОБЛЕМЫ АВТОМАТИКИ И УПРАВЛЕНИЯ № 3 (45)

Журнал входит в РИНЦ, а также в перечень ВАК Кыргызской Республики рецензируемых научных журналов, в которых должны быть опубликованы основные результаты диссертаций на соискание ученых степеней доктора и кандидата наук

Журнал издается с 1996 года.

Зарегистрирован Министерством юстиции Кыргызской Республики
Свидетельство № 1503 от 24 марта 2020 г.



БИШКЕК

2022

ИМА НАН КР

П 78

Проблемы автоматизации и управления: Научно-технический журнал
/ Национальная академия наук Кыргызской Республики.–
Бишкек: ИМА, 2022.– №3 (45).– 226 с.

ГЛАВНЫЙ РЕДАКТОР:

д.т.н., профессор Брякин Иван Васильевич ✉
Институт машиноведения и автоматизации НАН Кыргызской Республики (Бишкек)

ОТВЕТСТВЕННЫЙ СЕКРЕТАРЬ:

к.т.н. Верзунов Сергей Николаевич ✉
Институт машиноведения и автоматизации НАН Кыргызской Республики (Бишкек)

РЕДАКЦИОННАЯ КОЛЛЕГИЯ:

д.т.н., академик РАН Евтушенко Юрий Гаврилович ✉
Федеральный исследовательский центр "Информатика и управление" РАН (Москва)

д.т.н., член-кор. АЭН РФ Карандаев Александр Сергеевич ✉
Южно-Уральский государственный университет (национальный исследовательский университет) (Челябинск)

д.ф.-м.н. академик НАН РК Калимолдаев Максат Нурдилович
Казахский национальный университет им. Аль-Фараби (Алматы)

д.т.н., член-кор. АЭН РФ Макаричев Юрий Александрович ✉
Самарский государственный технический университет (Самара)

д.т.н., профессор Храмшин Вадим Рифхатович ✉
Магнитогорский государственный технический университет им. Г.И. Носова (Магнитогорск)

д.т.н., профессор Еремин Евгений Леонидович ✉
Тихоокеанский государственный университет (Хабаровск)

д.т.н., профессор Лыченко Наталья Михайловна ✉
Кыргызско-Российский славянский университет (Бишкек)

д.т.н., профессор Исмаилов Бактыбек Исакаевич ✉
Агентство по аккредитации образовательных программ и организаций (Бишкек)

д.т.н., профессор Бочкарев Игорь Викторович ✉
Кыргызский государственный технический университет им. И. Раззакова (Бишкек)

д.т.н., профессор Галбаев Жалалидин Токтобаевич ✉
Кыргызский государственный технический университет им. И. Раззакова (Бишкек)

д.ф.-м.н., профессор Сатыбаев Абдуганы Джунусович ✉
Ошский технологический университет им. акад. М.М. Адышева (Ош)

д.т.н., профессор Бакасова Айна Бакасовна ✉
Институт машиноведения и автоматизации НАН Кыргызской Республики (Бишкек)

д.т.н. Янко Дмитрий Владимирович ✉
ОАО «Кыргызтелеком» (Бишкек)

к.ф.-м.н., с.н.с. Керимкулова Гулсаат Кубатбековна ✉
Институт машиноведения и автоматизации НАН Кыргызской Республики (Бишкек)

к.т.н. Аскалиева Гулзада Орозобаевна ✉
Институт машиноведения и автоматизации НАН Кыргызской Республики (Бишкек)

В журнале публикуются статьи по информационным технологиям и обработке информации, математическому и компьютерному моделированию динамических систем и процессов, автоматическому управлению динамическими системами и процессами

Контакты редакции:

720071, Бишкек, проспект Чуй, 265.

Институт машиноведения и автоматизации Национальной академии наук КР.

Телефон: +996 553 72-20-30.

E-mail: verzunov@hotmail.com

Полная электронная версия журнала: <http://pau.imash.kg/index.php/pau>

ISSN 1694-5050, 1694-836X © Институт машиноведения и автоматизации НАН КР, 2022

ОГЛАВЛЕНИЕ

АНАЛИЗ И СИНТЕЗ СИСТЕМ УПРАВЛЕНИЯ

СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ С ИЗМЕНЯЮЩЕЙСЯ КОНФИГУРАЦИЕЙ ДЛЯ
ДИСТИЛЛЯЦИОННОЙ КОЛОННЫ НА ОСНОВЕ AIS СО СТРУКТУРНОЙ ПЛАСТИЧНОСТЬЮ
О.И. Ширяева 5-14

РАЗРАБОТКА И АНАЛИЗ ИМИТАЦИОННОЙ МОДЕЛИ ДОРОЖНОГО ПЕРЕКРЕСТКА
Г.М. Адиева, А.Дж. Сатыбаев, Б.А. Жоробеков 15-21

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ СИСТЕМ И ПРОЦЕССОВ

ИССЛЕДОВАНИЕ ПУЛЬСАЦИЙ ДИНАМИЧЕСКОГО ДАВЛЕНИЯ В ОТКРЫТОМ БУРНОМ
ТУРБУЛЕНТНОМ ПОТОКЕ ВОДЫ (НА ОСНОВЕ ЭЛЛИПТИЧЕСКОГО ПРОФИЛЯ СКОРОСТИ
ВОДЫ)

Керимкулова Г.К., Пресняков К.А., Аскалиева Г.О. 22-26

МЕТОД УСТАНОВЛЕНИЯ ВЕРОЯТНОГО ВЛИЯНИЯ ВЫБОРА ОРИЕНТАЦИИ
КООРДИНАТНЫХ ОСЕЙ НА РЕЗУЛЬТАТЫ ИЗМЕРЕНИЙ КИНЕМАТИЧЕСКИХ
ХАРАКТЕРИСТИК ОТКРЫТОГО ТУРБУЛЕНТНОГО ПОТОКА ВОДЫ

Керимкулова Г.К., Пресняков К.А., Аскалиева Г.О., Першакова Е.Ю. 27-30

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЧИСЛЕННЫХ МЕТОДОВ МОДЕЛИРОВАНИЯ
МИКРОПОЛОСКОВОЙ АНТЕННЫ

А.У. Касимова 31-41

ТЕХНИЧЕСКИЕ СРЕДСТВА СИСТЕМ КОНТРОЛЯ, ДИАГНОСТИКИ И УПРАВЛЕНИЯ

ЭКСПЕРИМЕНТАЛЬНО-РАСЧЕТНОЕ ОПРЕДЕЛЕНИЕ ЭЛЕКТРОМАГНИТНОЙ
ОБСТАНОВКИ ПС 110 КВ "АНАНЬЕВО"

А.К. Асанов, А.А. Омокеева 42-51

УСТРОЙСТВО ДЛЯ КОНТРОЛЯ СОСТОЯНИЯ СИЛОВОГО ТРАНСФОРМАТОРА

Т. Кабаев, Г.Ш. Эралиева, Ч.М. Абдыбаева, Ч.М. Бузурманкулова 52-58

МОДЕРНИЗАЦИЯ РАБОЧЕГО КОЛЕСА БИРОТОРНОГО МИКРО-ГЭС (БМГЭС)

Оразбаев К.Н., Медетбеков М.Т., Джоошибеков М.Б. 59-64

ЭЛЕКТРОННЫЙ МЕДИЦИНСКИЙ ТЕРМОМЕТР С БЕСПРОВОДНОЙ ПЕРЕДАЧЕЙ
ИНФОРМАЦИИ

Д.М. Ёдгорова, А.А. Хакимов, А.З. Рахматов, Р.Г. Закиров 65-73

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СИСТЕМАХ МОНИТОРИНГА ПРИРОДНЫХ И ТЕХНОГЕННЫХ КАТАСТРОФ

РАЗРАБОТКА WEB-ПОРТАЛА ЭКОЛОГИЧЕСКОЙ ИНФОРМАЦИИ КЫРГЫЗСКОЙ
РЕСПУБЛИКИ

В.В. Гайдамако, Б.К. Каныбеков, Н.М. Лыченко, Д.А. Текеев 74-83

СПОСОБЫ И СРЕДСТВА ОБНАРУЖЕНИЯ ГРОВОЙ АКТИВНОСТИ В ВОЗДУШНОМ
ПРОСТРАНСТВЕ

М.К. Олжабаев, К.Ж. Исабаев, Н.Б. Имансакипова, И.В. Брякин 84-95

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

СОЗДАНИЕ ЭЛЕКТРОННЫХ УЧЕБНЫХ ПОСОБИЙ НА ОСНОВЕ СЕРВИСОВ GOOGLE В
ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ

Ч.Ш. Омурзакова, Э.Т. Султангазиев 96-102

УПРАВЛЕНИЕ ИНФОРМАЦИОННЫМИ ТЕХНОЛОГИЯМИ И РЕСУРСАМИ В ВЫСШИХ
УЧЕБНЫХ ЗАВЕДЕНИЯХ

Аскарбеков Р.Н., Орускулов Т.Р., Раззаков М.И. 103-109

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ И ОБРАБОТКА ИНФОРМАЦИИ

МОДЕЛИРОВАНИЕ КРАТКОСРОЧНОГО ПРОГНОЗА ЗАГРЯЗНЕННОСТИ ВОЗДУХА
ТВЕРДЫМИ ЧАСТИЦАМИ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С
УЧЕТОМ ФАКТОРА ЗАГРЯЗНЕНИЯ

Л.И. Великанова, Н.М. Лыченко 110-118

СИСТЕМА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ДИАГНОСТИКИ COVID-19 ПО КТ-
СНИМКАМ

С.Н Верзунов 119-134

МЕТОДЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ В ЗАДАЧАХ ПРОГНОЗИРОВАНИЯ БОЛЕЗНЕЙ
РАСТЕНИЙ С ИСПОЛЬЗОВАНИЕМ ТРАНСФЕРНОГО ОБУЧЕНИЯ

Сабитов Б.Р., Сейтказиева Н.С., Картанова А.Дж. 135-144

ИДЕНТИФИКАЦИЯ БОЛЕЗНЕЙ ТОМАТОВ НА ОСНОВЕ МНОГОКЛАССОВОЙ
КЛАССИФИКАЦИИ

Б.Р. Сабитов, Н.С. Сейтказиева, А.Дж. Картанова 145-153

ФИЛЬТРАЦИЯ И АНАЛИЗ СНИМКОВ ВЕГЕТАЦИОННОГО ИНДЕКСА

Г.Т. Каримова 154-158

МОДЕЛИРОВАНИЕ ЭЛЕКТРОСТАТИЧЕСКОГО РАЗРЯДА И ИДЕНТИФИКАЦИЯ
ЭЛЕКТРОМАГНИТНОГО ПОЛЯ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ
СЕТЕЙ

А. Асан уулу, А. Б. Бакасова 159-166

К ВОПРОСАМ ПЛАНИРОВАНИЯ И БАЛАНСИРОВКИ НАГРУЗКИ В HADOOP MAPREDUCE

В.В. Гайдамако 167-176

К ВОПРОСУ ПАРСИНГА БОЛЬШИХ ОБЪЕМОВ ДАННЫХ ФОРМАТА JSON СРЕДСТВАМИ
ЯЗЫКА GO

Д.О. Авельцов 177-184

ПРИМЕНЕНИЕ ПРОТОКОЛА СЕРИАЛИЗАЦИИ СТРУКТУРИРОВАННЫХ ДАННЫХ
PROTUBUF В МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ

Д.О. Авельцов 185-196

РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА АППАРАТНО-ПРОГРАММНОЙ
ИНФРАСТРУКТУРЫ ОРГАНИЗАЦИИ

В. В. Гайдамако, Е.Д. Киселев 197-205

ТЕХНОЛОГИЯ ПОСТРОЕНИЯ ГЕОИНФОРМАЦИОННОЙ МОДЕЛИ УДС ГОРОДА ОШ НА
ОСНОВЕ ДАННЫХ ИЗ ОТКРЫТЫХ ИСТОЧНИКОВ

З.Т. Маликова, Г.Т. Мамажакытова 206-212

МОДЕЛЬ ПРОГРАММНОГО ИНТЕРФЕЙСА ДЛЯ РАЗРАБОТКИ ПРОГНОСТИЧЕСКИХ
СИСТЕМ НА ПРИМЕРЕ РЕШЕНИЯ ЗАДАЧИ ПРОГНОЗА КЛАССА ИНДЕКСА КАЧЕСТВА
ВОЗДУХА

А.В. Сороковая 213-225

УДК 681.5.013

О.И. Ширяева, oshiryayeva@gmail.com*Институт информационных и вычислительных технологий КН МОН РК,
Казахстан***СИНТЕЗ СИСТЕМЫ УПРАВЛЕНИЯ С ИЗМЕНЯЮЩЕЙСЯ
КОНФИГУРАЦИЕЙ ДЛЯ ДИСТИЛЛЯЦИОННОЙ КОЛОННЫ
НА ОСНОВЕ AIS СО СТРУКТУРНОЙ ПЛАСТИЧНОСТЬЮ**

В данной статье представлены результаты исследования условий построения искусственных иммунных систем с использованием модифицированных алгоритмов на основе принципа структурной пластичности, которая учитывает различные режимы функционирования и изменяющуюся конфигурацию процессов очистки газа. Сформулирована задача синтеза регулятора с переменной структурой для многоконтурной системы с целью поиска экстремума, выбранного интегрального квадратичного критерия качества. В ходе функционирования системы управления происходит переключение между различными типами регуляторов: PID, I-PD и PI-D, что обеспечивает регулятору переменную структуру. Приведены условия использования алгоритма искусственных иммунных систем со свойством структурной пластичности или адаптивности. Это обуславливает перспективное использование данных свойств для сложных систем управления с переменным регулятором, в которых необходимо учитывать влияние взаимосвязей.

Ключевые слова: искусственная иммунная система, модифицированные алгоритмы, принцип структурной пластичности, изменяющаяся конфигурация, сложная система.

Введение

Синтез систем управления с изменяющейся конфигурацией занимает важную часть при решении промышленных задач с применением алгоритмов искусственного интеллекта. Успешное внедрение промышленного искусственного интеллекта обуславливает его интеграцию, например, и в нефтегазовой отрасли, отличающейся высокими технологическими требованиями к процессам, набором больших данных и использованием микропроцессорной техники [1]. Это является следствием интенсивного развития интеллектуальных алгоритмов для эффективного решения различных задач производства [2]. В частности, для обработки данных, для оптимального управления производственными процессами, для решения задач прогнозирования на основе технологий умного производства, методов машинного обучения, цифровых двойников [3].

Одно из важнейших перспективных направлений внедрения искусственного интеллекта в нефтегазовой отрасли – это моделирование и оптимизация различных процессов. То есть использование искусственного интеллекта для управления технологическими процессами, в том числе оптимального. Для решения данных задач используются оптимизационные алгоритмы искусственного интеллекта, широко представленные в современной литературе [4]. Многие из интеллектуальных алгоритмов эффективно используются для решения задач синтеза оптимальных типовых регуляторов систем автоматического управления: Dragonfly Algorithm (DA) [5], Genetic Algorithm (GA) [6], Ant Colony Optimization (ACO) [7], Particle Swarm Algorithm (PSA) [8], Cuckoo Search (CS) [9] и т.д.

Современные интеллектуальные алгоритмы разнообразны – каждый из алгоритмов имеет свои преимущества и недостатки, поэтому должен рассматриваться непосредственно в соответствии с конкретным технологическим процессом нефтегазовой отрасли, желаемыми требованиями к динамике и качеству систем, задачей оптимизации. В частности, интеллектуальные алгоритмы используются для параметрической настройки типовых PID регуляторов, чтобы синтезировать оптимальное управление с целью минимизации выбранного критерия качества [5-9].

Алгоритмы искусственного интеллекта широко применяются для настройки PID регуляторов одномерных систем (SISO), но представляет интерес использование алгоритмов искусственного интеллекта для настройки регуляторов с переменной структурой [10]. В состав таких регуляторов входят I-PD, PD-PI и PI-D регуляторы. Для решения задачи синтеза регуляторов в настоящее время разработаны методы синтеза I-PD и PD-PI регуляторов на основе таких интеллектуальных алгоритмов искусственного интеллекта, как Particle swarm optimization (PSO) [11], Cuckoo Search (CS) [12], Swarm Intelligence [13].

Успешный выбор интеллектуального алгоритма обеспечивает эффективную настройку регуляторов, и среди таких алгоритмов хорошо себя зарекомендовали алгоритмы искусственной иммунной системы, AIS [14-16]. Их особенность в том, что они обладают свойством адаптации и биопластичности, что позволяет им лучше функционировать в условиях изменения конфигурации системы [17]. Это используется для SISO-систем, но особенно важно для МИМО-систем. В данной работе алгоритмы AIS впервые используются для настройки регулятора с переменной структурой МИМО-систем для нефтегазовой отрасли на примере технологических процессов дистилляционной колонны.

Постановка задачи

Рассмотрим задачу синтеза системы управления с изменяющейся конфигурацией для дистилляционной колонны очистки газа.

Принцип работы дистилляционной колонны заключается в следующих шагах (рис. 1):

- в колонну поступает неразделённая газосодержащая жидкость или смесь;
- неочищенный газ поступает в колонну, направляется через лотки вниз башни;
- поступает в ребойлер, где происходит нагрев смеси, и нагретая смесь переходит в форму пара;
- поступает в нижнюю часть колонны, пар, обладая более низкой плотностью и более высокой температурой, направляется вверх колонны;
- проходя через лотки, начинается процесс дистилляции поступающего неочищенного газа;
- насыщенный газом пар идёт в верхнюю часть колонны;
- поступает в конденсатор, в который постоянно поступает холодная вода и выходит горячая, это необходимо для того, чтобы насыщенный паром газ начал конденсироваться;
- после того как насыщенный газ конденсировался, он переходит в рефлюкс-барaban, где происходит следующее: часть полученного дистиллята выходит в виде готового продукта, а оставшийся дистиллят отправляется в верхнюю часть колонны для того, чтобы повысить эффективность колонны и тем самым повысить качество готового продукта [13].

Математическая модель, разработанная в [18], системы управления очистки газа в дистилляционной колонне основывается на структуре баланса энергии. В этой структуре входами являются скорость потока жидкости, L , и скорость потока пара, V , а выходами – концентрации продуктов χ_D и χ_B .

Модель дистилляционной колонны может быть представлена системой с двумя входами и двумя выходами [19], что соответствует двумерной системе управления:

$$\begin{cases} \frac{d\chi_D(t)}{dt} = -\frac{1}{1.9588} \cdot \chi_D(t) + \frac{0.0042}{1.9588} \cdot L(t) - \frac{0.0062}{1.9588} \cdot V(t), \\ \frac{d\chi_B(t)}{dt} = -\frac{1}{1.9588} \cdot \chi_B(t) + \frac{0.0072}{1.9588} \cdot V(t) - \frac{0.0052}{1.9588} \cdot L(t) \end{cases} \quad (1)$$

где L – скорость потока жидкости, V – скорость потока пара, χ_D – концентрация жидкости в дисцилляторе, χ_B – концентрация жидкости в нижней части колонны. Введём в рассмотрение два вектора-столбца входных, MV_1, MV_2 , и выходных, PV_1, PV_2 , сигналов:

$$\begin{bmatrix} MV_1(s) \\ MV_2(s) \end{bmatrix} = \begin{bmatrix} L(s) \\ V(s) \end{bmatrix}, \quad \begin{bmatrix} PV_1(s) \\ PV_2(s) \end{bmatrix} = \begin{bmatrix} \chi_D(s) \\ \chi_B(s) \end{bmatrix}. \quad (2)$$

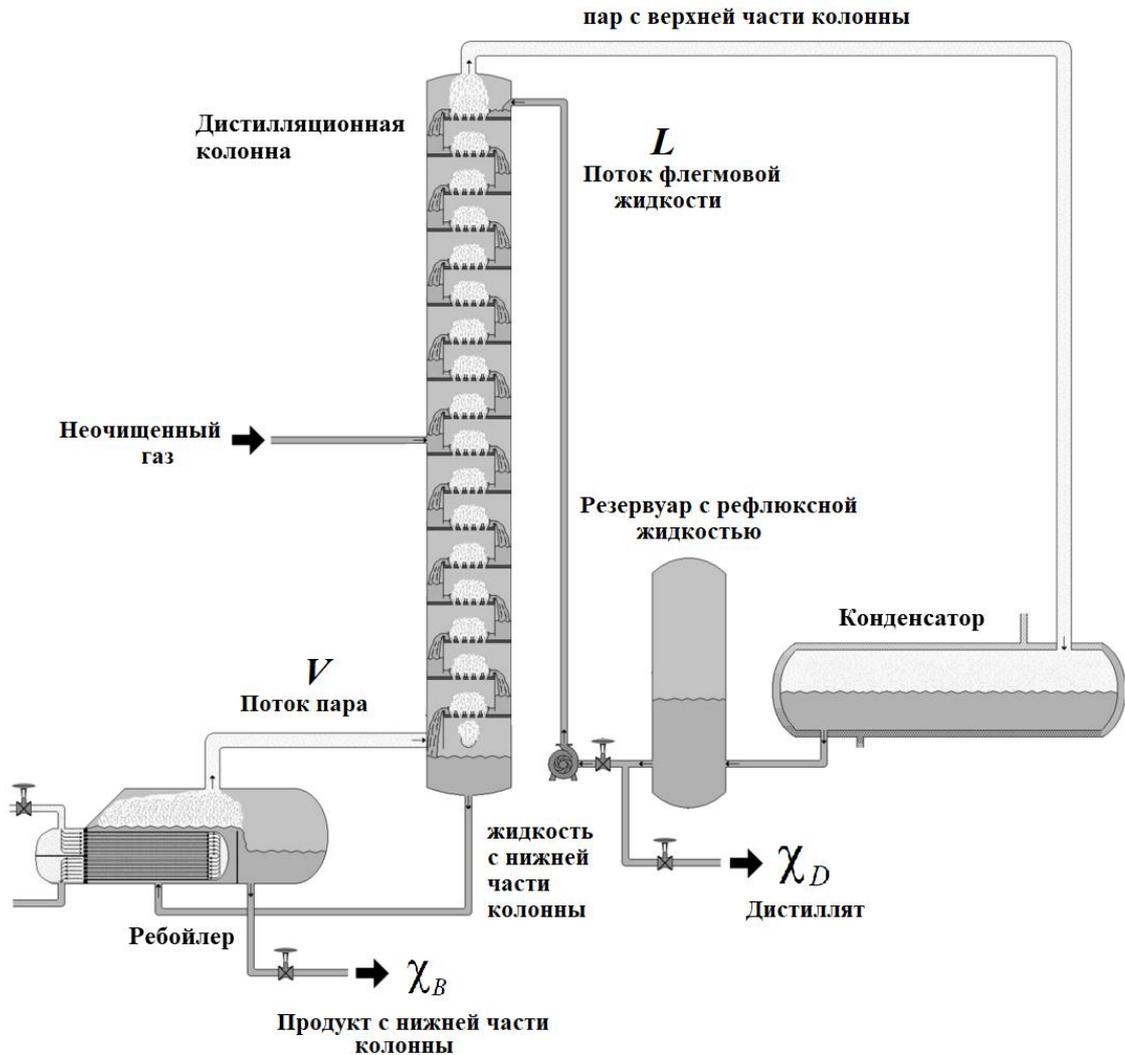


Рисунок 1 – Схема дистилляционной колонны

В соответствии с введёнными векторами выражение модели сложной системы управления процессами дистилляционной колонны имеет вид:

$$\begin{bmatrix} PV_1(s) \\ PV_2(s) \end{bmatrix} = \begin{bmatrix} \frac{0.0042}{1.9588s + 1} & \frac{-0.0062}{1.9588s + 1} \\ -0.0052 & \frac{0.0072}{1.9588s + 1} \end{bmatrix} \cdot \begin{bmatrix} MV_1(s) \\ MV_2(s) \end{bmatrix}. \quad (3)$$

Пусть $G_{11}(s), G_{12}(s), G_{21}(s), G_{22}(s)$ – передаточные функции объекта, связывающие входы и выходы:

$$\begin{bmatrix} PV_1(s) \\ PV_2(s) \end{bmatrix} = \begin{bmatrix} G_{11}(s) & G_{12}(s) \\ G_{21}(s) & G_{22}(s) \end{bmatrix} \cdot \begin{bmatrix} MV_1(s) \\ MV_2(s) \end{bmatrix}, \quad (4)$$

где G_{11}, G_{22} – передаточные функции изолированных подсистем, G_{12}, G_{21} – передаточные функции взаимосвязей между подсистемами (рис.2). Вопрос влияния и компенсации взаимосвязей за счёт устройств H_1 и H_2 рассматривался в работе [20].

В состав автоматизированной системы управления входит пропорционально-интегрально-дифференциальный (PID) регулятор, $C(s)$, с переменной структурой, что обуславливает рассмотрение данной системы как системы с изменяющейся конфигурацией. Законы управления таких регуляторов включают не только ошибку рассогласования, E , но и текущее значение выходного сигнала, PV : I-PD и PD-PI регуляторы (рис. 3).

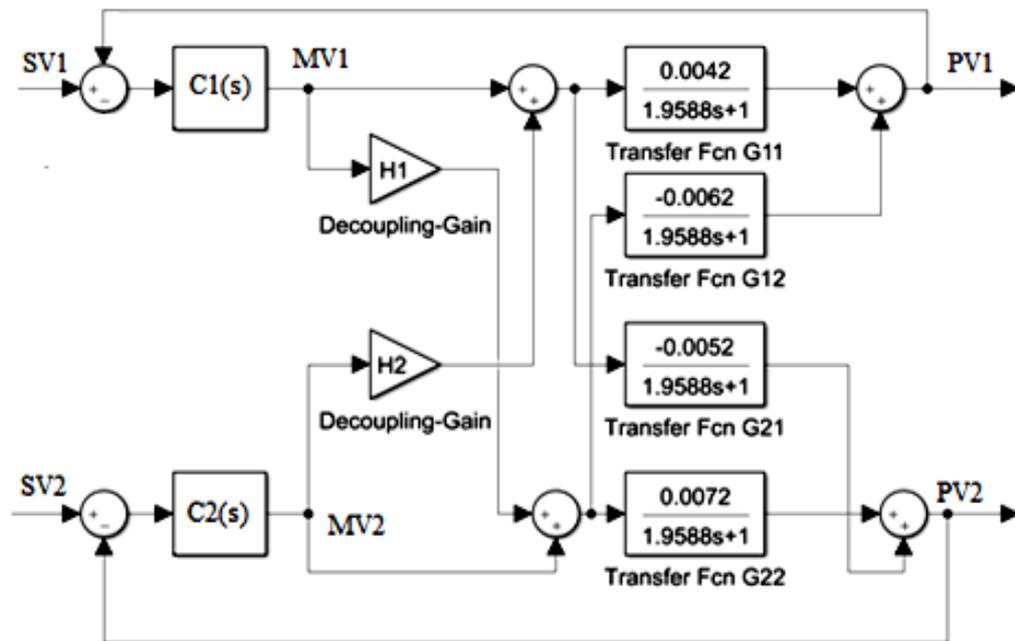


Рисунок 2 – Схема моделирования дистилляционной колонны в среде MATLAB

Такая конфигурация регулятора обеспечивает повышение быстродействия системы при выработке необходимого управляющего воздействия, MV , на объект управления с целью достижения заданных режимов, SV .

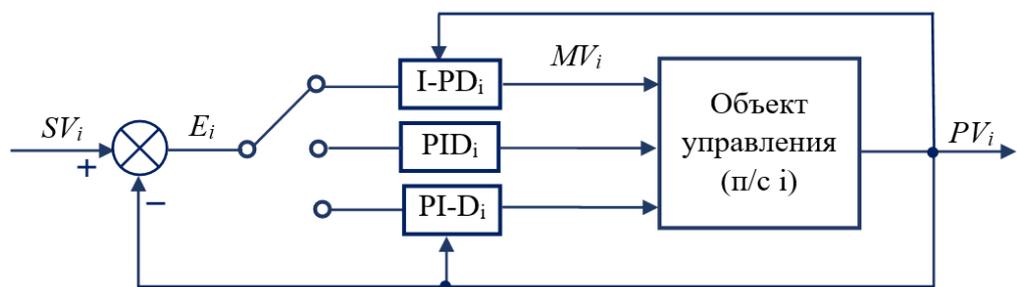


Рисунок 3 – Регулятор с переменной структурой

В ходе функционирования системы управления происходит переключение между различными типами регуляторов: PID, I-PD и PD-PI, что обеспечивает регулятору переменную структуру.

Задача синтеза переменного регулятора

В соответствии со структурой регулятора, представленного на рисунке 4, I-PD закон управления PCY имеет вид:

$$MV(t) = K_p \cdot K_s \cdot \left\{ PV(t) + \frac{1}{T_i} E(t) + T_D \frac{dPV(t)}{dt} \right\}, \quad (5)$$

где K_p – коэффициент усиления пропорциональной составляющей регулятора; K_s – коэффициент отношения разницы верхнего и нижнего предела управляющего сигнала, MV , и разницы верхнего и нижнего предела текущего значения выходного сигнала, PV ; T_i – постоянная времени интегрирующей составляющей регулятора; T_D – постоянная времени дифференцирующей составляющей регулятора.

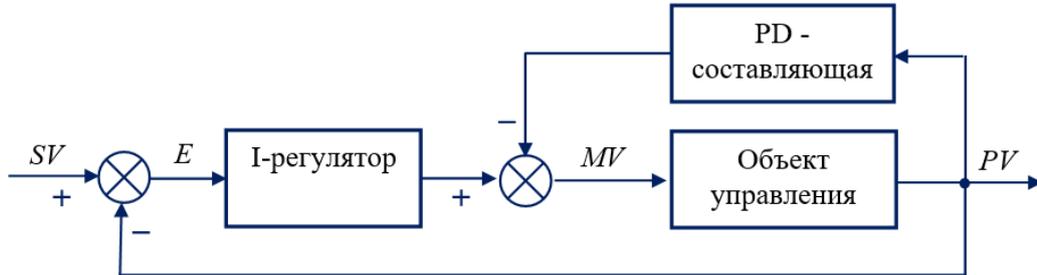


Рисунок 4 – Схема системы с I-PD регулятором

Передаточная функция замкнутой системы управления с регулятором (5) формируется в соответствии со схемой, представленной на рисунке 4:

$$T_{I-PD}(s) = \frac{C_I(s) \cdot G(s)}{1 + G(s) \cdot (C_I(s) + C_{PD}(s))}, \quad (6)$$

где $G(s)$ – передаточная функция объекта управления; C_I – передаточная функция I регулятора в прямой цепи структурной схемы системы управления, на вход которого подаётся ошибка рассогласования; C_{PD} – передаточная функция PD регулятора в цепи обратной связи объекта управления, на вход которого подаётся текущее значение выходного сигнала.

Закон PI-D управления, в котором управляющее воздействие формируется на основе ошибки рассогласования, E , и изменения выходного сигнала PV (рис.5):

$$MV(t) = K_p \cdot K_s \cdot \left\{ E(t) + \frac{1}{T_i} \int_{t=0}^{t_1} E(t) dt - T_d \frac{dPV(t)}{dt} \right\}. \quad (7)$$

Передаточная функция соответствует схеме системы с PI-D регулятором (рис.5).

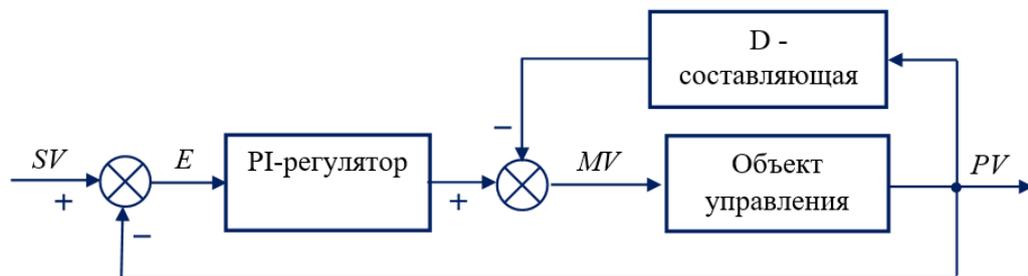


Рисунок 5 – Схема системы с PI-D регулятором

В этом случае выражение передаточной функции замкнутой системы:

$$T_{PI-D}(s) = \frac{C_{PI}(s) \cdot G(s)}{1 + G(s) \cdot (C_{PI}(s) + C_D(s))}. \quad (8)$$

Постановка задачи

На основе алгоритмов AIS для настройки регулятора с переменной структурой необходимо найти параметры регуляторов I-PD и PI-D, которые обеспечивают экстремум

целевой функции – квадратичной интегральной оценки качества:

$$ISE_i = \int_{t=0}^{t_1} E_i^2(t) dt = \int_{t=0}^{t_1} (SV_i(t) - PV_i(t))^2 dt \rightarrow \min. \quad (9)$$

Диапазон интегрирования ограничен временем моделирования технологического процесса, t_1 .

Алгоритм AIS со структурной пластичностью

В соответствии с работами по биологии нервная и иммунная системы различных организмов обладают свойством бипластичности, то есть двойной пластичности, что отражает способность иммунной системы к адаптации на основе параметрических и структурных изменений [20].

Свойство параметрической пластичности интерпретируется, как свойство системы изменять параметры функционирования для повышения её эффективности. Структурная пластичность соответствует свойству адаптации системы управления. Свойство структурной пластичности AIS интерпретируется, как свойство системы изменять свою структуру. Это обуславливает перспективное использование данных свойств для сложных систем управления с переменным регулятором, в которых необходимо учитывать влияние взаимосвязей.

Алгоритм AIS для решения задачи оптимизации критерия качества (9) и синтеза переменного регулятора (рис. 6) представлен ниже.

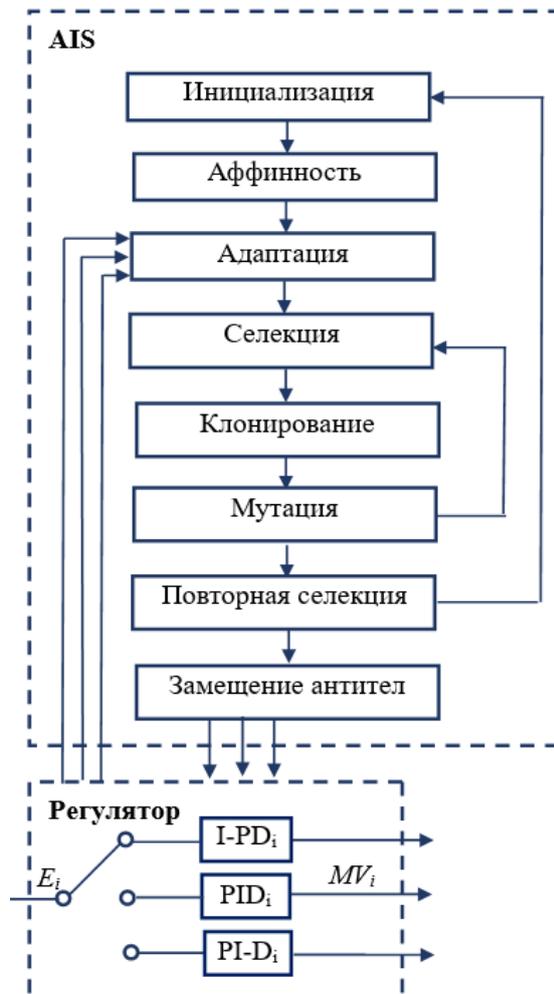


Рисунок 6 – Настройка переменного регулятора на основе алгоритма AIS с адаптацией

1 шаг. Инициализация: генерирование начальной популяции антител – параметров регуляторов.

2 шаг. Аффинность: вычисление значения критерия качества (9).

3 шаг. Адаптация: на основе свойства бипластичности производится изменение структуры системы на основе изменения структуры переменного регулятора (рис. 3).

4 шаг. Селекция: выбор подмножества антител с лучшей аффинностью – параметров регуляторов, обеспечивающих экстремум критерию качества.

5 шаг. Клонирование: получение популяции клонов.

6 шаг. Мутация: получение популяции из изменённых клонов.

7 шаг. Повторная селекция: выбор лучшего клона.

8 шаг. Замещение антител клонами с высокой аффинностью.

В соответствии с [21] алгоритм AIS включает в себя стандартные шаги поискового эволюционного алгоритма, основанного на функционировании биологической иммунной системы с формированием оптимального иммунного ответа: инициализация с формированием первоначальных значений параметров регуляторов, вычисление аффинности параметров регуляторов критерию качества, селекция, клонирование, мутация, выбор лучшего клона для поиска параметров регуляторов, обеспечивающих минимум критерия качества.

Представленные шаги алгоритма обеспечивают решение задачи оптимизации критерия качества (9) с учётом структурных изменений в системе управления, в частности изменений структуры регулятора. Изменение достигается за счёт того, что среди стандартных шагов популяционного цикла интеллектуального алгоритма AIS имеет место шаг, связанный со свойством бипластичности – адаптация.

В соответствии со свойством пластичности и адаптируемости данный алгоритм обеспечивает изменение структуры регулятора в соответствии с желаемыми требованиями к технологическому процессу. Процедура переключения осуществляется по схеме, представленной на рисунке 6, и в соответствии с данными переключениями производится изменение структуры регулятора и закона управления. В зависимости от выбранного закона управления изменяется критерий качества, так как его выражение включает в себя передаточную функцию замкнутой системы регулирования по ошибке рассогласования и законы управления.

Практическое значение и выводы

Сформированный алгоритм был апробирован проведёнными экспериментами на базе среды моделирования процессов очистки газа дистилляционной колонны, на платформе распределённой системы управления Exregion PKS [22, 23].

Приведём результаты моделирования системы (3) с регуляторами PID и I-PD, настроенными на основе алгоритма AIS с целью минимизации критерия качества (9).

Общими настройками алгоритма AIS выбраны:

$N=100$ – размер популяции;

$Ab=cadeia(n,44)$ – популяция антител;

$gen=50$ – число генераций;

$pm=0.5$ – вероятность мутации;

$d=0.3$ – популяция, которая подвергается случайным перестановкам;

$beta=0.5$ – множитель, контролирующий количество клонов.

В ходе реализации алгоритма проводится адаптация критерия качества (9) под изменяющуюся структуру регулятора от PID к I-PD следующим образом. Критерии качества используются в теории искусственной иммунной системы в качестве антигенов:

$$Ag = f(E, MV). \quad (10)$$

Подмножество антигенов тождественно выражению, включающему параметры регуляторов:

$$Ag = f\left(K_P, \frac{1}{T_I}, T_D, MV\right). \quad (11)$$

При изменении структуры регулятора в алгоритме производится адаптация критерия качества за счёт настройки подмножества (11). В качестве начальной популяции в таком случае берутся значения параметров PID регуляторов, настроенные на предыдущей итерации.

В среде MATLAB, для системы (3), состоящей из двух контуров, были получены (рис. 7):

- удовлетворительные оценки эффективности применения алгоритма AIS для минимизации критериев качества;
- параметры PID и I-PD регуляторов.

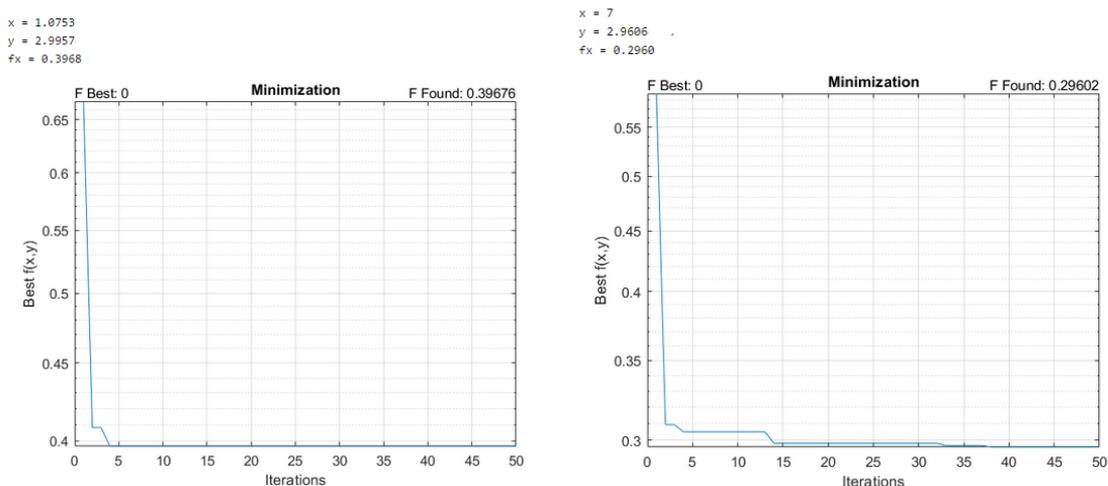


Рисунок 7 – Оценка работы алгоритма AIS при настройке регуляторов

Результаты моделирования двух контуров (3) с PID и I-PD регуляторами представлены на рис. 8.

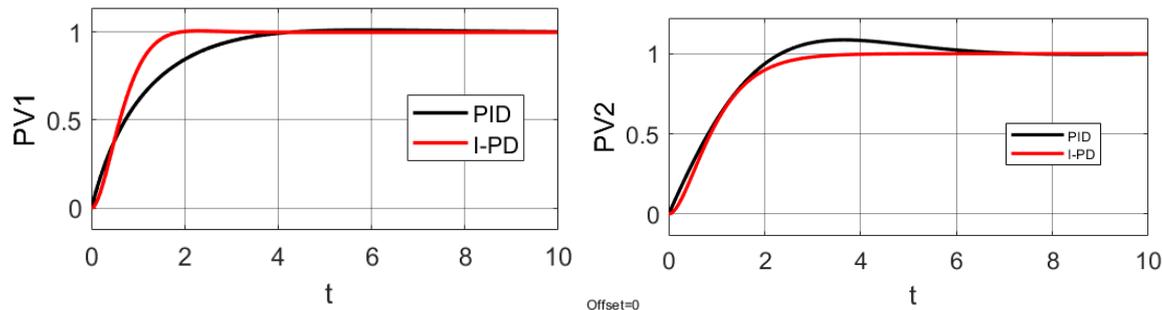


Рисунок 8 – Переходные процессы системы управления с регуляторами

В соответствии с полученными результатами моделирования можно сделать вывод о том, что параметры регуляторов в двух контурах системы, настроенные на основе алгоритма AIS, обеспечили системе желаемую динамику.

Заключение

Получены результаты формирования алгоритма AIS со свойством структурной пластичности для многоконтурной системы управления дистилляционной колонны. Данная система характеризуется изменяющейся конфигурацией за счёт регулятора с переменной структурой. Алгоритм AIS на основе адаптации позволяет настраивать такие регуляторы, как переключение между различными типами регуляторов: PID, I-PD и PI-D, что обеспечивает регулятору переменную структуру.

Данное исследование было профинансировано Комитетом науки Министерства образования и науки Республики Казахстан (грант № AP09258508).

Литература

1. Koroteev, D., Tekic, Z. Artificial intelligence in oil and gas upstream: Trends, challenges, and scenarios for the future / D. Koroteev, Z. Tekic // *Energy and AI*. – 2021. – № 3. – 10 p.
2. Malik, H. Applications of Artificial Intelligence Techniques in Engineering. Advances in intelligent systems and computing / H. Malik // Springer Nature Singapore. – 2019. – Vol. 1. – 698 p.
3. Lee, J. Industrial Artificial Intelligence for industry 4.0-based manufacturing systems / J. Lee // *Manufacturing Letters*. – 2018. – Vol. 18. – P. 20-23.
4. Xing B. Innovative computational intelligence: A rough guide to 134 clever algorithms. – Cham: Springer, 2014. – 451 p.
5. Kouba, N., Mena, M., Hasni, M., Boudour, M. A novel optimal combined fuzzy PID controller employing dragonfly algorithm for solving automatic generation control problem / Kouba, N., Mena, M., Hasni, M., Boudour, M. // *Electric Power Components and Systems*. – 2018. – Vol. 46. – Issue 19-20. – P. 2054–2070.
6. Slavov, T. Application of Genetic Algorithm to Tuning a PID Controller for Glucose Concentration Control / T. Slavov // *WSEAS Transactions on Systems*. – 2012. – Vol. 11. – Issue 7. – P. 223–233.
7. Amar, M. N. Optimization of WAG process using dynamic proxy, genetic algorithm and ant colony optimization / M. N. Amar // *Arabian Journal for Science and Engineering*. – 2018. – Vol. 42. – Issue 11. – P. 6399–6412.
8. Wang J. The Application of the Particle Swarm Algorithm to Optimize PID Controller in the Automatic Voltage Regulation System. – Springer Singapore: *Advanced Computational Methods in Energy, Power, Electric Vehicles and Their Integration*, 2017. – P. 529–536.
9. Sethi, R. Cuckoo search algorithm based optimal tuning of PID structured TCSC controller / R. Sethi // *Computational Intelligence in Data Mining*. – 2015. – Vol. 1. – P. 251–263.
10. Sahraoui, M. Application of artificial immune algorithm-based optimisation in tuning a PID controller for nonlinear systems / M. Sahraoui // *International Journal of Automation and Control*. – 2015. – No. 3 (9). – P. 186–200.
11. Kumar, D. Design of PSO based I-PD Controller and PID Controller for a Spherical Tank System / D. Kumar // *Indian Journal of Science and Technology*. – 2016. – Vol. 9. – Issue 12. – P. 1–5.
12. Puangdownreonga D. Optimal Design of I-PD Controller for DC Motor Speed Control System by Cuckoo Search // *International Electrical Engineering Congress, iEECON2016*. – Thailand, 2016. – P. 83–86.
13. Jain, T. Optimization of PD-PI Controller using Swarm Intelligence / T. Jain // *Journal Of Theoretical And Applied Information Technology*. – 2008. – No. 11 (4). – P. 1013–1018.
14. Padmanabhan, S., Chandrasekaran, M., Ganesan, S., Khan, M., Navakanth, P. Optimal Solution for an Engineering Applications Using Modified Artificial Immune System / S. Padmanabhan, M. Chandrasekaran, S. Ganesan, M. Khan, P. Navakanth // *Materials Science and Engineering*. – 2017. – Vol. 183. – P. 1–65.
15. Saleh, M., Saad, S. Artificial Immune System based PID Tuning for DC Servo Speed Control / M. Saleh, S. Saad // *International Journal of Computer Applications*, 2016. — Vol. 155, №2. – P. 23-26.
16. Wang, M., Feng, S., He, Ch., Li, Zh., Yu, X. An Artificial Immune System Algorithm with Social Learning and Its Application in Industrial PID Controller Design / M. Wang, S. Feng, Ch. He, Zh. Li, X. Yu // *Mathematical Problems in Engineering*. – 2017. – Vol. 2017. – 13 p.

17. Люпина, Ю.В., Орлова, А.Ш., Горностаев, Н.Г., Карпова, Я.Д. Пластичность нервной и иммунной систем у различных организмов: роль протеасом / Ю. В. Люпина, А. Ш. Орлова, Н. Г. Горностаев, Я. Д. Карпова // Журнал общей биологии. – 2014. – № 1 (75). – С. 3–24.
18. Minh V. T., Rani A. A. Modeling and control of distillation column in a petroleum process. – Perak.: Hindawi Publishing Corporation, 2009. – 14 p.
19. Самигулина, Г.А., Самигулин, Т.И. Разработка Smart-технологии для управления сложным объектом с использованием алгоритма муравьиной колонии / Г. А. Самигулина, Т. И. Самигулин // Вестник Алматинского университета энергетики и связи. – 2019. – №1 (44). – 106–113с.
20. Ширяева О.И. Анализ использования свойств бипластичности AIS для решения задач моделирования, анализа и синтеза сложных систем управления // Materials of the XVII International scientific and practical Conference «Prospects of world science – 2021». – Sheffield: Science and education LTD, 2021. – P. 93–99.
21. Woodland, D. Plasticity in Adaptive Immunity / D. Woodland // Viral immunology. – 2016. – Vol. 26. – №5. – 301 p.
22. Samigulin, T.I., Shiryayeva, O.I. Development of a SMART-system for a complex industrial object control based on metaheuristic algorithms of swarm intelligence // WSEAS Transactions on Power Systems. – 2021. – V.16. – pp. 231–240.
23. Ширяева, О. И. Теоретические основы для синтеза сложных систем управления с использованием алгоритмов искусственной иммунной системы / О. И. Ширяева // Проблемы автоматки и управления. – 2021. – № 1(40). – С. 61-67. – EDN XXJKKB.

УДК 004(075.8)

*Адиева Г.М., gulzinaadieva@gmail.com**Сатыбаев А.Дж. д.ф-м.н., профессор**Жоробеков Б.А. к.т.н., профессор**Ошский технологический университет им.академика М.Адышева, г.Ош***РАЗРАБОТКА И АНАЛИЗ ИМИТАЦИОННОЙ МОДЕЛИ ДОРОЖНОГО ПЕРЕКРЕСТКА**

Статья посвящена к разработке модели существующей ситуации по организации дорожного движения на перекрестке в городе Ош с применением онлайн системы Avenue 2.0. Актуальность проводимого исследования обоснована увеличением потока транспортных средств на магистралях, приводящих к перегруженности городских дорог. В исследовании применен метод компьютерного моделирования, для анализа – метод анализа транспортного потока. Проведен анализ управления дорожным движением и управления транспортными потоками.

Результат исследования дает о необходимости изменение режима работы светофоров с учетом интенсивности потока, а также о перераспределении направления движения транспортных потоков со стороны улицы Монуева.

Ключевые слова: Перекресток, очереди автомобилей, концептуальная модель, имитационное моделирование, система управления светофорами.

Введение

Увеличение числа автомобилей отрицательно сказывается на пропускной способности дорожной сети городов. Необходимо предпринять меры для решения проблем автодорожных пробок. Современные программные комплексы и математические методы становятся наиболее эффективными инструментами для решения такого рода задач. “Интеграция различных методов средств позволяет с разных сторон посмотреть на возникающие проблемы” [1с.225]. Моделирование с помощью программных средств дает возможность проанализировать общую ситуацию, выявлять загруженные места на улицах для принятия мер по их устранению.

Автомобильные пробки приводят к увеличению затрат времени передвигаясь пассажиров в пути. Также заторы приводят к износу транспортных средств, повышению расходов на топлива и выброс вредных веществ. Поэтому целесообразно решать данную проблему и повышать эффективность управления транспортной системы.

“Признанным методом анализа транспортных сетей является имитационное моделирование. Преимущество имитационных моделей по сравнению с аналитическими состоит в принципиальной возможности отобразить объект на любом уровне детализации” [2.с.56]. “Мероприятия по использованию имитационного компьютерного моделирования транспортных и пешеходных потоков могут быть реализованы в кратчайшее время с минимальными затратами по сравнению с дорожным строительством, что позволит максимально использовать возможности имеющейся УДС и учитывать потребности участников дорожного движения” [9].

Транспортный поток, состоящий из различных технических средств (автобусов, автомобилей, мотоциклов) является основным объектом управления в системе управления дорожным движением. В то же время поведение водителей автомобилей не всегда предсказуемо, они на дороге ведут себя по-разному и на различные события реагируют неодинаково, что на много затрудняет анализ такой системы [10].

В зарубежных странах разработано множество программных средств, которые помогают эффективно управлять транспортной системой города. Примерами таких программ являются VISSIM, Paramics, MITSIMLab, TRANSIMS, EMME/2, SATURN и др. Самым популярным программным продуктом служит PTV Vision, разработанный немецкими производителями. Они используют концепцию VISSIM VISUM. Применяемая мною онлайн сервис Avenue 2.0 в основном предназначена для моделирования

транспортного потока, расчета оптимальной длительности цикла светофорного регулирования, создания и оптимизации планов координаций движения на дорогах и их пересечениях.

На сегодняшний день существует множество различных способов моделирования транспортных потоков.

В работах О.А. Евдокимова описана имитационная модель перекрестка в системе имитационного моделирования ANYLOGIC 7.1. Приведены методы работы адаптивной и неадаптивной систем управления светофорами на перекрестке [2].

Автором статьи [3] для анализа регулируемого перекрестка предложена динамическая имитационная микромодель в среде PTVVISIONVISSIM. В модели автором не учитывается скорость движения автомобиля.

Автор статьи [4] применил метод «узел – функция – объект» (УФО). Данный подход основан на алфавите элементов, связей систем, подвергающихся анализу, а также их библиотек. Описывается протяжённость потока, количеством транспорта, а также параметры разрешающие и запрещающие движения на различные направления.

В научных статьях Владимирова С.Н., Кадырова А.С., Токашева Н.С. приведен обзор разработанных методов борьбы с дорожными заторами в мире [7], [8].

1. Концептуальная модель перекрестка

В представленной работе рассматривается перекресток проспекта Масалиева и улицы Монуева в городе Ош. При разработки данной модели в качестве основы была взята растровое изображение перекрестка из открытых источников «Open Street Map». Данный перекресток состоит из различных конфигураций, пешеходных переходов, количество полос и направлений движения, способов размещения светофоров.



Рисунок 1 – Перекресток проспекта А. Масалиева и улицы И. Монуева. Снимок с ресурса Open Street Map

В настоящее время данный перекресток является самым загруженным. «Подъездной сектор дороги ОСИ (Ош– Сары-Таш – Иркештам) смыкается с улицей Масалиева (основная северо-южная ось» [5.с.70]. Оттуда едет большой поток транспортных средств. Данный перекресток подключен к системе «умный город», который фиксирует нарушения правил дорожного движения и оценивает поток автомобилей.

Разработанная модель позволяет оценивать общее состояние данного перекрестка, визуализировать её структуру и осуществляет графический анализ, что позволяет ясно представить полученные решения и донести заложенные в него идеи.

Для получения данных о загруженности перекрестка города Ош был проделан ручной подсчёт количества автомобилей, проезжающих перекресток (пересечение проспекта Масалиева и улицы Монуева) в часы пик в течение часа (см. рис. 2).

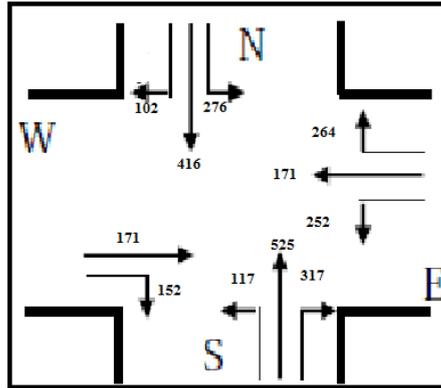


Рисунок 2 – Интенсивность транспортного потока

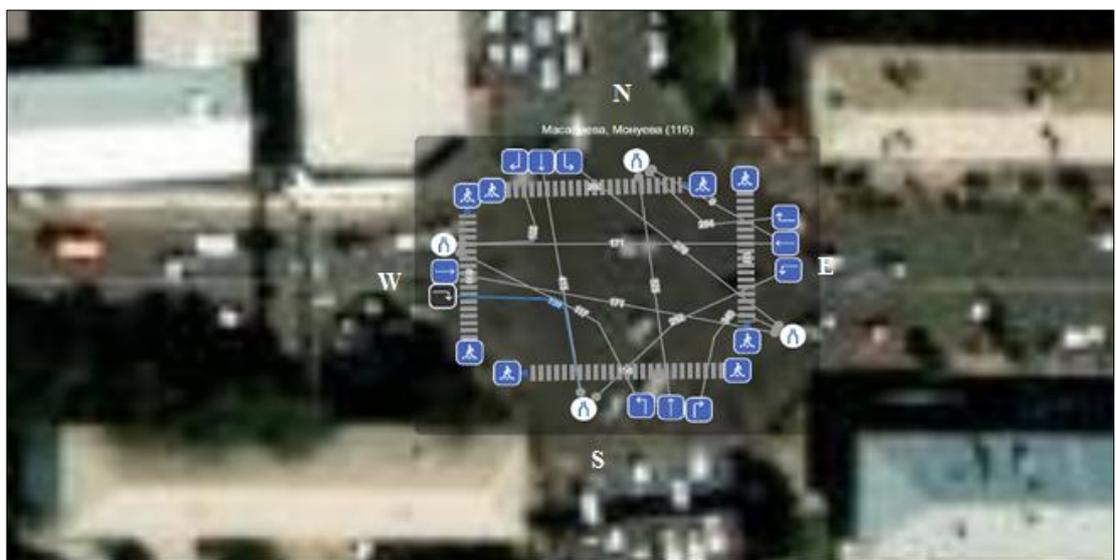


Рисунок 3 – Модель перекрестка

В модели осуществляется движение по четырем потокам: 1, 2, 3, 4. Каждый поток имеет по две полосы, в котором полосы могут осуществлять движения в одном направлении. Также для полос указана интенсивность движения автомобилей за час. На данном перекрестке движением управляют четыре светофора. Регулирование движений состоит из четырех фаз. Цикл светофорного регулирования составляет 120с.

В первой фазе (рис. 5) движение разрешено по проезжим частям S и N. Текущая фаза цикла подсвечена активными линиями. Диаграмма выдает информацию о направлениях движения по фазам светофорного цикла.

Толщина стрелки нормирована по интенсивностям всей модели, т.е. самая широкая стрелка обозначает самое большое значение потока по всей модели. Цвет стрелки показывает коэффициент насыщения. Бледно-зеленый – малонасыщенный поток. Ярко-зеленый – насыщение стремится к 100%, перенасыщенные потоки обозначаются черным или темно-зеленым цветом. Как видно из результатов моделирования по первой фазе, два прямонаправленных потока являются самыми насыщенными.

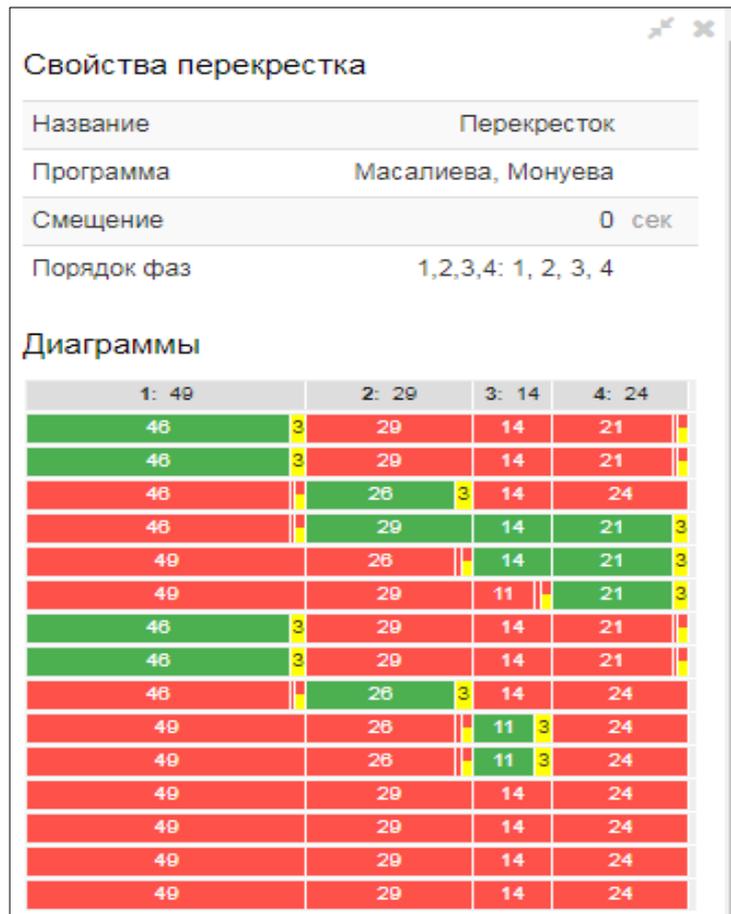


Рисунок 4 – Диаграмма перекрестка по фазам



Рисунок 5 – Первая фаза цикла

Поток направления с юга на север обозначен черной линией. Нажимая на линию, можно построить модель данного потока (рис. 6).

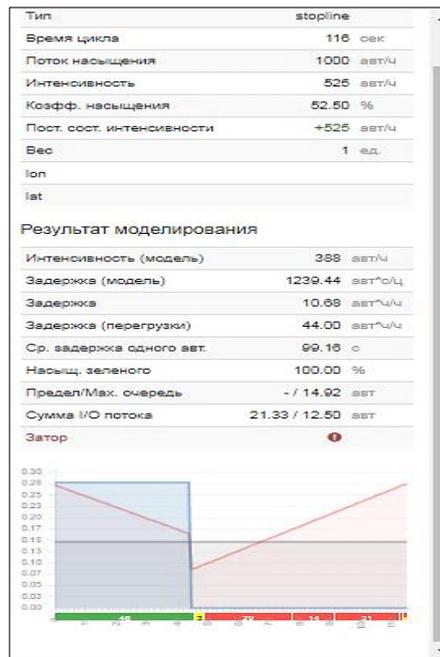


Рисунок 6 – Затор, полученный по результатам моделирования

По результатам полученных данных насыщение потока составляет 1000 автомобилей в час, это означает, что очередь будет разгружаться со скоростью 1 машина в 2 секунды. Интенсивность составляет 525 автомобилей за час. Коэффициент насыщения – 52%. По результатам моделирования видно, что по данной линии образовался затор. В диаграмме розовая линия показывает время горения красного сигнала, серая линия – интенсивность входящего потока, синяя линия означает интенсивность исходящего потока.

Во время работы светофоров во второй фазе (рис. 7, рис. 8) движение разрешено по проезжим частям S и N по направлениям W и E. В модели насыщенная линия обозначена черным цветом. По остальным потокам перегруз не наблюдается, потоки линий свободны.



Рисунок 7 – Модель второй фазы цикла

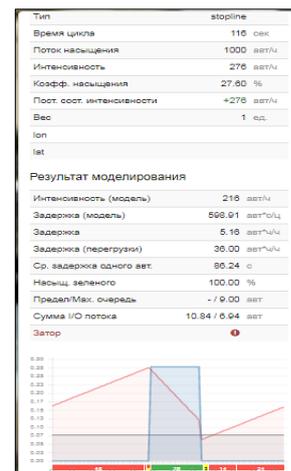


Рисунок 8 – Диаграмма заторной линии

Во время работы светофоров в третьей фазе (рис.9) движение разрешено по проезжим частям W и E. В модели насыщенная линия обозначена черным цветом. По остальным потокам перегруз не наблюдается, потоки линий свободны.



Рисунок 9 – Третья фаза цикла

Во время работы светофоров в четвертой фазе (рис.10) движение разрешено по проезжей части Е. Все потоки движения разгружены, заторов не наблюдается.

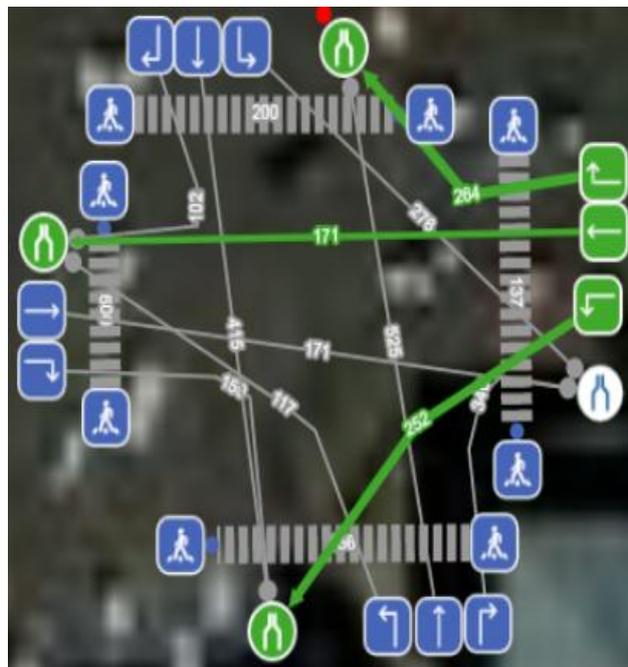


Рисунок 10 – Четвертая фаза цикла

Таким образом, исследования показали, что в отличие от других моделей, имитационная модель является более эффективной при определении пропускной способности перекрестка, выявлении заторов и проблем на магистралях. Применение такой модели в системах управления дорожным движением может способствовать минимизации заторовых ситуаций на улицах, а также дальнейших работ по изменению конфигураций светофоров для улучшения условий движения на улицах.

Заключение

Во всех больших городах мира остро стоят проблемы организации дорожного движения. Не является исключением и город Ош. Система Avenue 2.0 предназначена для моделирования транспортных потоков и редактирования, расчета фаз светофорного регулирования.

Предлагаемая модель перекрестка может быть применена в дальнейших исследованиях по оценке простоя автомобилей в очередях, также для оптимизации транспортного потока изменяя алгоритмов управления светофорами.

На регулируемом перекрестке проспекта Масалиева – ул. Монуева исследования показывают низкую скорость движения 8км/ч и 10км/ч соответственно из-за превышения пропускной способности, которая определяется как “пробки”. Замедление движения обусловлено множеством причин, таких, как ожидание сигнала светофора, беспорядочность движения транспорта, пешеходы, остановки маршруток, такси вблизи перекрестка и въезд и выезд с парковок на обочинах дороги.

По результатам исследования делаем выводы о необходимости изменение интервалов времени светофоров с учётом интенсивности потока, а также о перераспределении направления движения транспортных потоков со стороны улицы Монуева. Поэтому для улучшения дорожной ситуации города Ош необходимо изменение параметров светофорных объектов, предоставление приоритетного проезда городскому пассажирскому транспорту, увеличение количество муниципальных автобусов большой вместимости, рассмотрение возможности ограничения въезда в город из приграничных районов. Для предотвращения неправильных парковок автомобилей на обочинах дорог, необходимо строительство платных стоянок в загруженных улицах города.

Литература

1. Живоглядов, В.Г. Теория движения транспортных и пешеходных потоков[Текст] / В. Г. Живоглядов. Омский научный вестник № 1 (117). – 2013– С.225–229.
2. Евдокимов, А.О. Имитационная модель перекрестка с системой интеллектуального управления [Текст] / А.О.Евдокимов, А.В.Горохов, Д.Я.Лугов. Международный научно-исследовательский журнал №5 (36). – 2015. – Часть 2. – С.55–60.
3. Аширова, С. Р. Моделирование заторовых ситуаций на регулируемых пересечениях в системе транспортного моделирования PTVVISSIM / С. Р. Аширова, И. Н. Горячкина. // Молодой ученый. — 2017. — № 11.3 (145.3). — С. 1–3.
4. Маторин С.И., Жихарев А.Г., Зайцева Н.О., Брусенская И.Н. Имитационное моделирование транспортных потоков с применением УФО-подхода // Научные ведомости Белгородского государственного университета. Серия: Экономика. Информатика. – 2013. – № 22–1 (165).
5. Адиева, Г.М. Анализ общественного транспорта города Ош.[Текст] / Г.М. Адиева. Проблемы автоматизации и управления. –2021. – № 1 (40). – С. 68–74.
6. Яцкив, И. В. Использование возможностей имитационного моделирования для анализа транспортных узлов [Текст] / М.В. Яцкив, Е. А. Юршевич, Н. В. Колмакова // Имитационное моделирование. Теория и практика (ИММОД-2005): Сб. докл. 2-й Всерос. конф. – Т. 2. – СПб.: ЦТ СС, 2005. – С. 237 – 245.
7. Владимиров С. Н. Транспортные заторы в условиях мегаполиса // Известия МГТУ. – 2014. – №1 (19) – С. 77–84.
8. Кадыров А.С., Токашева Н.С. Анализ и предложения по проблеме устранения автомобильных пробок (дорожных заторов) // Международный научно-исследовательский журнал. – 2014. –№ 3–2 (22). – С. 69–72.
9. Ведяшкин, В.И. Моделирование транспортных и пешеходных потоков при помощи программного комплекса PTV VISSIM в г. Барнауле / В.И. Ведяшкин // Педагогическое образование на Алтае. – 2014. – № 2. – С. 219–220.
10. Адиева, Г. М. Анализ общественного транспорта города Ош / Г. М. Адиева // Проблемы автоматизации и управления. – 2021. – № 1(40). – С. 68-74. – EDN IGTKZXO.

МАТЕМАТИЧЕСКОЕ И КОМПЬЮТЕРНОЕ МОДЕЛИРОВАНИЕ ДИНАМИЧЕСКИХ СИСТЕМ И ПРОЦЕССОВ

УДК 631.6. (575.2)

*Пресняков Константин Александрович, д.т.н., с.н.с.,
Керимкулова Гүлсаат Кубатбековна, к.ф.-м.н., e-mail: gulsaat@mail.ru
Аскалиева Гулзада Орозобаевна, к.т.н., e-mail: 87guzya@mail.ru
Институт машиноведения и автоматики*

ИССЛЕДОВАНИЕ ПУЛЬСАЦИЙ ДИНАМИЧЕСКОГО ДАВЛЕНИЯ В ОТКРЫТОМ БУРНОМ ТУРБУЛЕНТНОМ ПОТОКЕ ВОДЫ (НА ОСНОВЕ ЭЛЛИПТИЧЕСКОГО ПРОФИЛЯ СКОРОСТИ ВОДЫ)

Установлены математические соотношения для распределений динамического давления и его пульсаций по глубине потока воды с использованием эллиптического профиля осредненной скорости воды. Выявлены значения силового воздействия динамического давления и его пульсаций в придонной части потока. Предварительно установлено существенное влияние выбора осредненного профиля скорости воды на распределение динамического давления и его пульсаций.

Ключевые слова: динамическое давление и его пульсации, силовое воздействие.

Введение

В работе [1], исходя из аксиомы для открытого турбулентного потока воды [2], были установлены соотношения для распределений динамического давления и его пульсаций по глубине потока воды, которые имеют следующий вид:

$$P_d = \rho \cdot u^2 + 1,05^2 \cdot u_*^2 \cdot \psi^2(\tilde{y}), \quad (1)$$

где P_d – динамическое давление, Па; ρ – плотность воды, кг/м³; u – продольная компонента скорости воды, м/с; u_* – динамическая скорость потока воды, м/с; $\psi(\tilde{y})$ – координатная часть распределения пульсаций вертикальной компоненты скорости по глубине потока [3]

$$\psi(\tilde{y}) = \frac{0,677 \cdot \tilde{y}}{\tilde{y} + 0,036} \cdot (\tilde{y}^2 - 2 \cdot \tilde{y} + 2), \quad (2)$$

где \tilde{y} – относительная вертикальная координата;

$$P'_d = 2 \cdot \rho \cdot (u(\tilde{y}) \cdot 2,10 \cdot u_* \cdot \phi(\tilde{y}) + 1,05^2 \cdot u_*^2 \cdot \psi^2(\tilde{y})), \quad (3)$$

где P'_d – пульсации динамического давления воды, Па; $u(\tilde{y})$ – распределение продольной компоненты скорости воды по глубине потока, м/с; $\phi(\tilde{y})$ – координатная часть распределения пульсаций продольной компоненты скорости воды по глубине потока [2]

$$\phi(\tilde{y}) = 0,25 + \left(\frac{0,21}{0,28 + \tilde{y}} \right). \quad (4)$$

Понятно, что как динамическое давление, так и его пульсации зависят, в частности, от выбора конкретного распределения продольной компоненты скорости воды по глубине потока.

Постановка задачи исследования

Известна [4, 5] для турбулентного потока воды эллиптическая зависимость осредненной скорости воды по глубине потока:

$$u(\tilde{y}) = u_m \cdot \sqrt{1 - p \cdot (1 - \tilde{y})^2}, \quad (5)$$

$$p = 0,57 + \frac{3,3}{C}, \quad (6)$$

где u_m – максимальная скорость потока воды, м/с; p – параметр, зависящий от коэффициента Шези; C – коэффициент Шези, $m^{1/2}/c$.

Представляет интерес исследование распределений динамического давления и его пульсаций по глубине потока воды с использованием эллиптического профиля скорости.

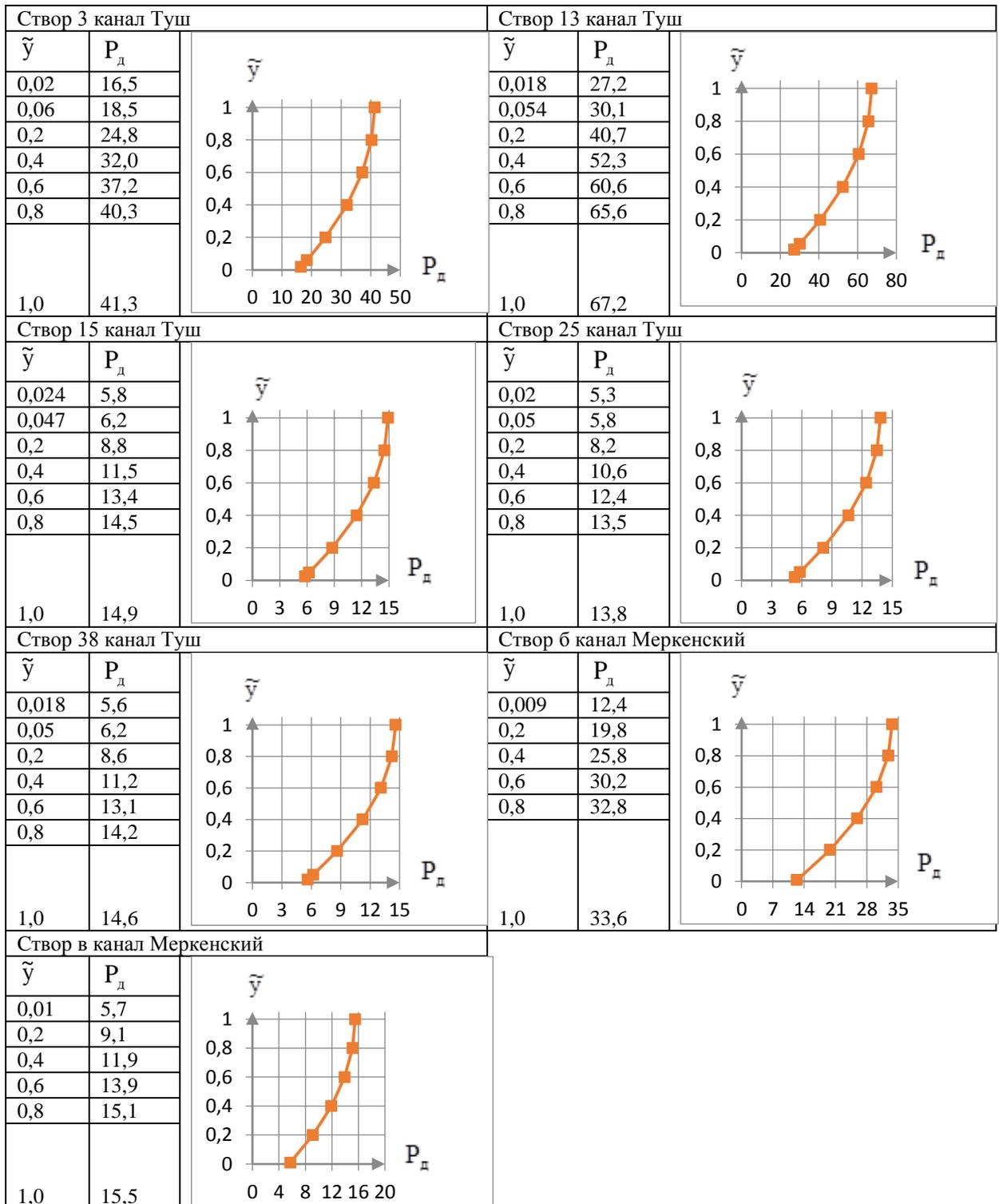


Рисунок 1 – Распределения динамического давления открытого бурного турбулентного потока воды по его глубине.

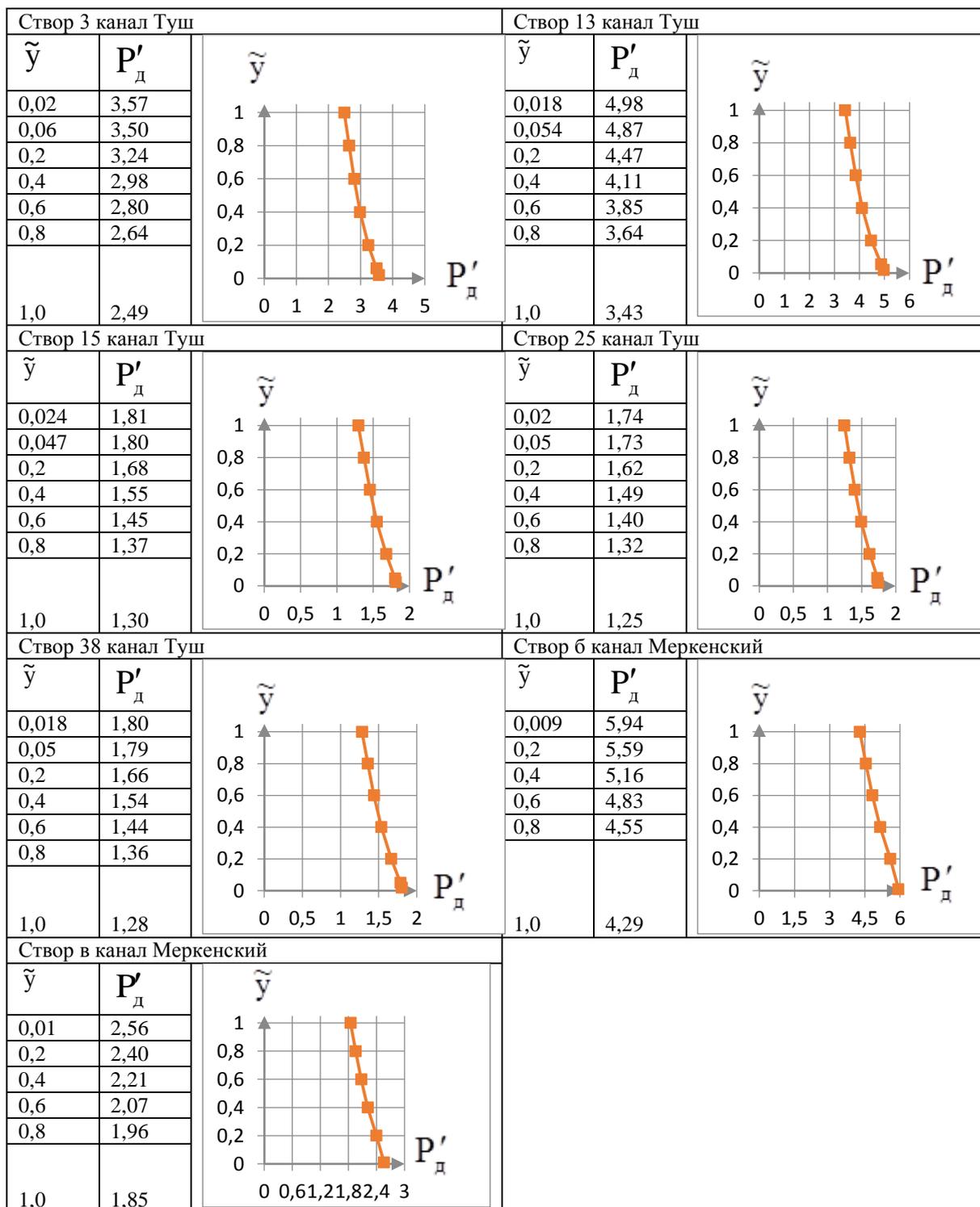


Рисунок 2 – Распределения пульсаций динамического давления открытого бурного турбулентного потока воды по его глубине.

Задача исследования

Найти выражения для поля динамического давления и поля его пульсаций в открытом бурном турбулентном потоке воды с использованием эллиптического профиля скорости воды.

Решение поставленной задачи

I. Распределение динамического давления и его пульсаций по глубине потока воды.

Расчетные формулы для искомым характеристик запишутся в следующем виде:

– для динамического давления

$$P_d = \rho \cdot \left(u_m \cdot \sqrt{1-p \cdot (1-\tilde{y})^2} \right)^2 + 1,05^2 \cdot u_*^2 \cdot \psi^2(\tilde{y}); \quad (7)$$

– для пульсаций динамического давления

$$P'_d = 2 \cdot \rho \cdot \left(u_m \cdot \sqrt{1-p \cdot (1-\tilde{y})^2} \cdot 2,10 \cdot u_* \cdot \phi(\tilde{y}) + 1,05^2 \cdot u_*^2 \cdot \psi^2(\tilde{y}) \right) \quad (8)$$

С использованием эмпирических данных для канала Туш (створы 3, 13, 15, 25, 38) и канала Меркенский (створы б, в) [6,5] произведены расчеты искомых характеристик по формулам (7) и (8). Результаты расчетов приведены на рисунке 1 и 2.

II. Нормы рассматриваемых характеристик

Динамическое давление распределяется по глубине потока воды согласно кривой [1], имеющей максимум в приповерхностном слое потока воды (рис. 1).

При этом значение рассматриваемого параметра меняется в пределах от 13,8 до 67,2 кПа.

Среднее значение нормы динамического давления равно 30 ± 20 кПа.

Распределения пульсаций динамического давления по глубине потока воды описываются кривыми [1], имеющими максимум на выступе шероховатости дна водотока (рис. 2).

Норма пульсаций динамического давления (на выступе шероховатости дна водотока) составляет ряд значений от 1,74 до 5,94 кПа.

Среднее значение указанной нормы равно 3 ± 1 кПа.

III. Оценка силы, обусловленной динамическим давлением и его пульсациями

Исходя из определения давления [7] как отношения силы к площади, запишем выражения для:

– силы, обусловленной наличием динамического давления в придонной части потока воды

$$F_{P_d(\tilde{\Delta})} = P_d(\tilde{\Delta}) \cdot 1m^2; \quad (9)$$

– силы, обусловленной наличием пульсаций динамического давления на выступе шероховатости

$$F_{P'_d(\tilde{\Delta})} = P'_d(\tilde{\Delta}) \cdot 1m^2. \quad (10)$$

Оценка силового воздействия дает следующий результат:

– для динамического давления (в придонной части потока воды) от 5,3 до 27,2 кН и среднее значение 11 ± 6 кН;

– для пульсаций динамического давления (на выступе шероховатости) от 1,7 до 5,9 кН и среднее значение 3 ± 1 кН.

Отношение этих характеристик равняется от 2,08 до 5,59 и среднее значение 3 ± 1 .

Следовательно, силовое воздействие динамического давления (в придонной части потока) больше силового воздействия пульсаций динамического давления (на выступе шероховатости) в 3 раза.

IV. Относительные распределения динамического давления и его пульсаций по глубине потока воды

Относительные распределения динамического давления и его пульсаций по глубине потока воды имеют, в частности, значение при сравнительной оценке экспериментальных результатов, полученных в самых разнообразных условиях, зачастую в, казалось бы, несопоставимых условиях.

Имеет смысл сопоставление указанных распределений, полученных с применением параболического и эллиптического профилей осредненной скорости воды.

Подобное сопоставление дает следующие результаты:

➤ для отношения $\tilde{P}_d^3 / \tilde{P}_d^{\text{пар}}$ от $1 \pm 0,2$ до $1 \pm 0,3$; т.е. расхождение между сопоставляемыми характеристиками достигает 30%;

➤ для отношения $\tilde{P}'_d / \tilde{P}'_{d\text{ пар}}$ от $0,6 \pm 0,4$ до $0,8 \pm 0,1$; т.е. расхождение между сопоставляемыми характеристиками достигает 67%.

Поэтому напрашивается предварительный вывод о существенном влиянии выбора профиля осредненной скорости воды на значения динамического давления и его пульсаций. Окончательное мнение по рассматриваемому вопросу может быть сформировано только после прохождения цепочки известных полуэмпирических профилей скорости воды.

Заключение

Установлены математические соотношения с использованием эллиптического профиля осредненной скорости воды:

- распределения динамического давления и его пульсаций по глубине потока воды в абсолютных единицах (Па);
- те же распределения в относительном виде.

В результате произведенной оценки силового воздействия динамического давления и его пульсаций в придонной части потока воды выявлены следующие их значения – для динамического давления 11 ± 6 кН, а для пульсаций динамического давления 3 ± 1 кН, т.е. превышение силового воздействия динамического давления над его пульсациями составляет 3 раза.

Предварительно установлено существенное влияние выбора осредненного профиля скорости воды на распределение динамического давления (до 30%) и его пульсаций (67%).

Литература

1. Пресняков К.А., Керимкулова Г.К., Аскалиева Г.О. Исследование пульсаций динамического давления в открытом бурном турбулентном потоке воды (на основе параболического профиля скорости воды) // Проблемы автоматизации и управления. – 2022. – №1(43) – С. 61–71.
2. Пресняков К.А., Керимкулова Г.К., Аскалиева Г.О. Сравнительный анализ пульсаций динамического давления и компонент скорости в открытом турбулентном потоке воды // Проблемы автоматизации и управления. – 2021. – №3(42) – С. 58– 66.
3. Никитин И.К. Турбулентный русловой поток и процессы в придонной области. – Киев: Изд-во АН УССР, 1963. –138с.
4. Караушев А.В. Проблемы динамики естественных водных потоков. – Л.: ГИМИЗ, 1960. – 392 с.
5. Аскалиева Г.О., Пресняков К.А, Керимкулова Г.К. Модель и алгоритмы идентификации параметров открытых водотоков. – LAPLAMBERT, 2019 г. – 177с.
6. Пресняков К.А., Керимкулова Г.К. Установление среднеинтегрального профиля скорости воды частично изученного открытого водотока на основе реконструкции эмпирических данных //Н.-т. журнал ИАИТ НАН КР «Проблемы автоматизации и управления». – Бишкек, 2012. –№2.– С. 47–56.
7. Чугаев Р.Р. Гидравлика. – Л.: Энергия (Лен.отделение), 1976.

УДК 631.6. (575.2)

Аскалиева Гулзада Орозобаевна, к.т.н., 87guzya@mail.ru

Пресняков Константин Александрович, д.т.н., с.н.с.

Керимкулова Гүлсаат Кубатбековна, к.ф.-м.н., gulsaat@mail.ru

Першакова Елена Юрьевна, hellper64@mail.ru

Институт машиноведения и автоматизации НАН КР

МЕТОД УСТАНОВЛЕНИЯ ВЕРОЯТНОГО ВЛИЯНИЯ ВЫБОРА ОРИЕНТАЦИИ КООРДИНАТНЫХ ОСЕЙ НА РЕЗУЛЬТАТЫ ИЗМЕРЕНИЙ КИНЕМАТИЧЕСКИХ ХАРАКТЕРИСТИК ОТКРЫТОГО ТУРБУЛЕНТНОГО ПОТОКА ВОДЫ

Предложен метод, который позволяет конкретизировать варианты выбора ориентации координатных осей путем использования понятий правовинтовой и левовинтовой декартовых систем координат и произвести сравнительный анализ измерений пульсаций продольной и вертикальной компонент скорости открытого турбулентного потока воды и сформулировать на основе указанных измерений вывод о степени и характере упомянутого влияния.

Ключевые слова: ориентация координатных осей, правовинтовая и левовинтовая декартовы системы координат, пульсации продольной и вертикальной компонент скорости воды, открытый турбулентный поток воды

Введение

В литературе содержится утверждение М.А. Великанова, которое гласит: все статические характеристики в их числовом выражении не зависят от ориентации координатных осей [1, стр. 249]. Подобное утверждение несет в себе идею метода установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды. Искомый метод (согласно М.А. Великанову) можно сформулировать следующим образом: метод установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды, заключающийся в выборе объекта исследований, назначении вариантов ориентации координатных осей, измерении кинематических характеристик турбулентного потока воды согласно выбранным вариантам, сравнении результатов указанных измерений и установлении степени и характера вероятного влияния выбора ориентации координатных осей на результаты упомянутых измерений.

Основным недостатком этого метода является установление вероятного влияния выбора ориентации координатных осей на осредненные, а не на пульсационные характеристики турбулентного потока воды.

Задачи исследований – установить степень и характер вероятного влияния выбора ориентации координатных осей на результаты измерений пульсаций продольной и вертикальной компонент скорости открытого турбулентного потока воды.

Решение поставленной задачи. Метод установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды [2] заключается в выборе установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды, заключающемся в выборе объекта исследований, назначении вариантов ориентации координатных осей, измерении кинематических характеристик турбулентного потока воды согласно выбранным вариантам, сравнении результатов указанных измерений и установлении степени и характера вероятного влияния выбора ориентации координатных осей на результаты упомянутых измерений, в котором в качестве объекта исследований выбирают прямолинейный открытый турбулентный поток воды, в качестве вариантов ориентации координатных осей выбирают правовинтовую и левовинтовую системы декартовых

координат, в качестве измеряемых кинематических характеристик назначают пульсации продольной и вертикальной компонент скорости воды, производят измерения упомянутых характеристик в праввинтовой и леввинтовой системах декартовых координат, проводят сравнение результатов упомянутых измерений и формируют вывод о степени и характере вероятного влияния выбора ориентации координатных осей на результаты проведенных измерений.

В литературе известно определение праввинтовой и леввинтовой декартовых систем координат, которое звучит следующим образом: «прямоугольная или косоугольная система координат в пространстве называется правой (левой), если ее базис является правой (левой) тройкой векторов» [3]. Согласно указанному определению, праввинтовую и леввинтовую системы декартовых координат можно изобразить в следующем виде (рис. 1).

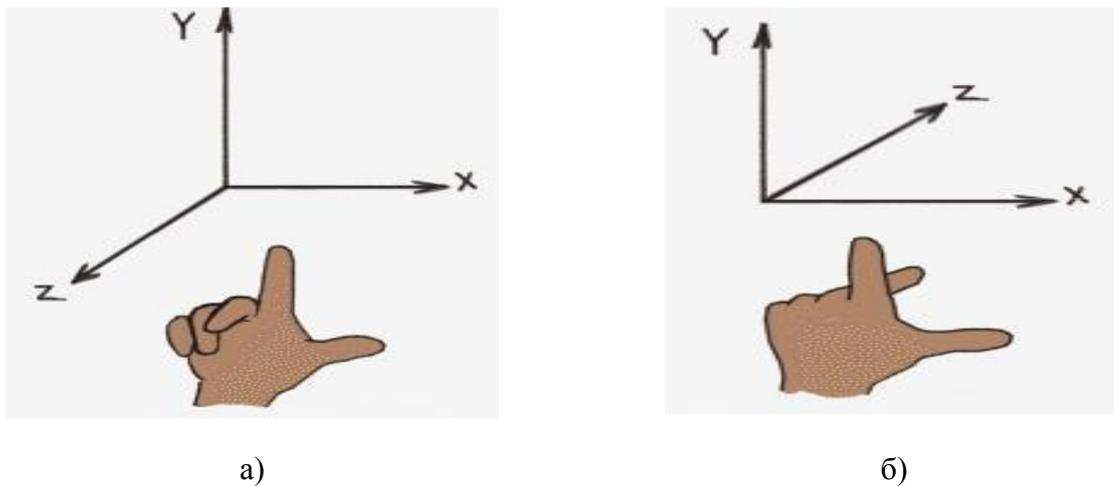


Рисунок 1– Изображение праввинтовой а) и леввинтовой б) систем декартовых координат

Теперь используем свойство выбранного объекта исследований, а именно «прямолинейный открытый турбулентный поток воды», то обстоятельство, что прямолинейность потока означает отсутствие в нем поперечных циркуляционных сечений. Это означает в свою очередь, что в обоих вариантах ориентации координатных осей ось z выпадает из нашего рассмотрения, тогда указанные выше две схемы ориентации координатных осей совпадают друг с другом. Поэтому отсюда можно сделать вывод, что результаты измерений кинематических характеристик турбулентного потока воды в нашем случае не зависят от ориентации координатных осей.

При апробации предлагаемого метода для других вариантов ориентации координатных осей и для других объектов исследований вполне возможно получение других результатов измерений.

Техническую схему реализации метода установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды иллюстрируют чертежом (рис. 2).



Рисунок 2 – Техническая схема метода установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды

Структура технической схемы предлагаемого метода установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды состоит из 3 блоков:

- ✓ блок 1 – выбор объекта исследований;
- ✓ блок 2 – выбор вариантов ориентации координатных осей;
- ✓ блок 3 – измерения кинематических характеристик для указанных вариантов и проведение сравнительного анализа полученных результатов.

Метод установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды осуществляют следующим образом (рис. 3):

- ✓ выбирают объект исследования (блок 1);
- ✓ конкретный объект исследования с указанием его основных параметров (блок 2);
- ✓ выбирают варианты ориентации координатных осей (блок 3);
- ✓ правовинтовая и левовинтовая декартовы систем координат как варианты ориентации координатных осей (блок 4);
- ✓ измеряют пульсации продольной и вертикальной компонент скорости воды в выбранных вариантах ориентации координатных осей (блок 5);
- ✓ сравнивают результаты указанных измерений (блок 6);
- ✓ анализируют полученные результаты (блок 7);
- ✓ формируют вывод о степени и характере вероятного влияния выбора ориентации координатных осей на результаты упомянутых измерений (блок 8).



Рисунок 3 – Структурная блок-схема реализации метода установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды

Заключение

Предложенный метод позволяет конкретизировать варианты выбора ориентации координатных осей путем рассмотрения правовинтовой и левовинтовой декартовых систем координат и произвести сравнительный анализ измерений пульсаций продольной и вертикальной компонент скорости открытого турбулентного потока воды. Установлено, что результаты измерений кинематических характеристик турбулентного потока воды в нашем случае не зависят от ориентации координатных осей.

Литература

1. М.А.Великанов. Динамика русловых потоков. – Т.1 Структура потока. – М.: Госиздат техн. -теор. лит. 1954. – 323 с.
2. Аскалиева Г.О., Пресняков К.А., Керимкулова Г.К., Першакова Е.Ю. Способ установления вероятного влияния выбора ориентации координатных осей на результаты измерений кинематических характеристик открытого турбулентного потока воды / Заявка №3756 от 29.04.2022 г.–Бишкек: Кыргызпатент.
3. Векторное и смешанное произведения [Электронный ресурс] / – Режим доступа: <http://math.phys.msu.ru/data/24/Lecture04.pdf> – Загл. с экрана

УДК 004.942

А.У. Касимова, adina.kasimova.99@mail.ru
Институт машиноведения и автоматизации ИАИ КР

СРАВНИТЕЛЬНЫЙ АНАЛИЗ ЧИСЛЕННЫХ МЕТОДОВ МОДЕЛИРОВАНИЯ МИКРОПОЛОСКОВОЙ АНТЕННЫ

В статье приводится сравнение численных методов моделирования антенн, таких, как метод моментов (МоМ), метод конечных элементов (FEM), а также метод конечных интегралов (FIT). Для более быстрого моделирования антенн рекомендуется использовать метод моментов, однако при моделировании антенн малых размеров лучше подходят методы конечных элементов и конечных интегралов. В данной работе показаны результаты моделирования микрополосковой антенны с применением ячеек из метаматериала на основе комплементарных разомкнутых колец CSRR [1] в программе CST Microwave Studio (CST MWS) и Matlab Antenna Toolbox (AT). С помощью программы CST Microwave Studio могут быть рассчитаны многие характеристики антенны, такие как полоса пропускания, коэффициент усиления, обратные потери и другие [1]. Также в работе сравниваются результаты моделирования микрополосковой антенны, полученные в программах CST Microwave Studio и Matlab Antenna Toolbox (AT).

Ключевые слова: метод моментов (МоМ), метод конечных элементов (FEM), метод конечных интегралов (FIT), микрополосковые антенны, оптимизация, моделирование антенн, CST Microwave Studio (CST MWS), Matlab Antenna Toolbox (AT).

Введение

В настоящее время область применения микрополосковых антенн довольно широка: они применяются в подповерхностной радиолокации (индикация неоднородностей естественного и искусственного происхождения, различных полостей, пластиковых и металлических труб, подземных кабелей, дефектов в объектах из диэлектрика), в геологии (измерение глубины залегания полезных ископаемых, минерализованного дна торфяных залежей, толщины ледников, снежного покрова); в археологии; в медицине; в экологии (обнаружение протечек в трубопроводах); в создании радаров со сверхвысоким разрешением; в WIFI-системах беспроводного доступа в Интернет [1, 3]. Однако микрополосковые антенны имеют ряд недостатков: узкая рабочая полоса частот, невысокая эффективность, в особенности для короткозамкнутых микрополосковых антенн, а также сложность реализации некоторых способов питания (апертурный, бесконтактный). В настоящее время применение метаматериалов может быть перспективным для создания микрополосковых антенн с улучшенными характеристиками: меньшие размеры по сравнению со стандартными микрополосковыми антеннами, увеличение коэффициента усиления антенны, расширение рабочей полосы антенны [4].

В работе [2] был предложен новый способ поиска оптимальной конструкции ячеек-резонаторов для микрополосковой антенны, позволяющий целенаправленно оптимизировать параметры ячеек-резонаторов, значительно улучшающих ее электрические характеристики.

Всё возрастающие требования к основным показателям качества антенн привели к развитию методов и средств проектирования антенн. Широкое распространение получили численные методы решения электродинамических задач [5]. На сегодняшний момент для разработки антенн используются специализированные EDA-программы для EM-расчетов. В основе работы подобных программных продуктов лежит численное решение уравнений Максвелла в интегральной или дифференциальной форме [6]. Выбор того или иного метода влияет на эффективность и точность расчетов компонентов антенн. Поэтому в данной работе представлены результаты сравнения трех численных методов моделирования: метод моментов (МоМ), метод конечных элементов (FEM) и метод конечного интегрирования (FIT), которые используются в программных пакетах, таких, как CST Microwave Studio и Matlab Antenna Toolbox (AT) для расчета EM-полей.

CST MICROWAVE STUDIO – мощная программа трехмерного моделирования электромагнитного поля. Программа использует различные методы расчета (расчет переходного процесса во временной области, анализ в частотной области, метод нахождения собственных частот) для моделирования сложных СВЧ-структур: фильтров, антенн, циркуляторов, аттенуаторов и т.д.[7].

Antenna Toolbox предоставляет функции и приложения для проектирования, анализа и визуализации антенных элементов и решеток. В ней можно проектировать автономные антенны, а также создавать массивы антенн, используя predetermined элементы с параметризованной геометрией или произвольные планарные элементы.

Antenna Toolbox использует метод моментов (MoM) для вычисления входных свойств (импеданс), поверхностных свойств (распределение тока и заряда) и свойств поля (диаграмма направленности в ближней и дальней зоне). Также возможна визуализация геометрии антенны и результаты анализа в 2D и 3D [8].

Метод моментов (MoM)

Для моделирования антенн широко применяется метод моментов (MoM), использующийся для расчёта токов в металлических и диэлектрических структурах при излучении в свободном пространстве [9]. При использовании данного метода все металлические части антенны заменяются поверхностными электрическими токами, после чего решается задача возбуждения электромагнитного поля этими токами [10]. Метод моментов – это решение уравнений Максвелла в интегральной форме в частотной области. Достоинство метода моментов заключается в том, что он является «методом источника», т.е. дискретизируется только интересующая структура, а не свободное пространство, как при решении уравнений для нахождения поля в объеме. В математике такое решение получило название функции Грина. Если функция Грина может быть записана аналитически в простой форме, то метод моментов оказывается весьма эффективным [11]. При этом не требуются граничные условия, а используемая память пропорциональна геометрии задачи и частоте [9]. Функция Грина достаточно просто может быть найдена только для ограниченного числа структур. В частности, к ним относятся свободное пространство и плоскостные структуры. Метод моментов применяется в следующих системах: MicrowaveOffice, FEKO, ADS, а также MatlabAntennaToolbox [11].

Решение задачи возбуждения среды осуществляется с помощью аппарата тензорных функций Грина. После того как задача возбуждения решена и найдено электрическое поле, используются граничные условия на металлических элементах. В результате задания граничных условий в дискретных точках формируется система линейных алгебраических уравнений (СЛАУ) относительно коэффициентов при базисных функциях, которые имеют смысл амплитуд токов, текущих в пределах элементарной площадки. Эти условия используются для определения эквивалентных токов [12]. Количество уравнений в СЛАУ равно числу элементарных площадок N , которое неизбежно растёт при увеличении размеров анализируемого объекта или при увеличении частоты. Важным моментом решения является разбиение поверхности металла на элементарные площадки и аппроксимация электрического тока в пределах площадки. Пример разбиения показан на рисунке 2 [9].

Точность MoM зависит от размеров площадки разбиения: чем меньше размер площадки разбиения, тем точнее данный метод [12].

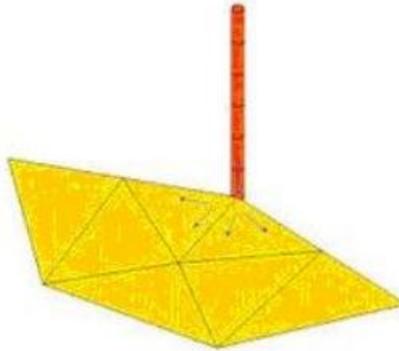


Рисунок 1 – Разбиение источника и поверхности на сегменты и треугольники

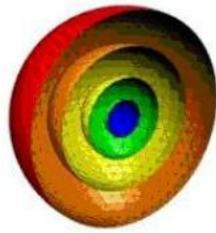


Рисунок 2– Специальные функции Грина для плоской многослойной среды



Рисунок 3 – Принцип эквивалентных объемов (тетраэдральная сетка разбиения)

В работе [12,13] показаны примеры решения СЛАУ методом моментов:

$$\sum_{m=1}^N C_m (L\varphi_m, \psi_m) = (e, \psi_m), 1 \leq n \leq N, \quad (1)$$

$$j = \sum_{m=1}^N C_m \varphi_m. \quad (2)$$

Решив систему уравнений (1), можно найти коэффициенты C_m и по уравнению (2) приближенное значение плотности тока, соответствующее заданному числу N [13].

Метод конечных элементов (ФЕМ)

Метод конечных элементов применяется при моделировании больших или неоднородных диэлектрических объектов, которые невозможно эффективно моделировать другим методом. Метод конечных элементов позволяет вести разработку численных алгоритмов с высокой степенью универсальности. Эти алгоритмы оказываются весьма эффективными для широкого круга таких задач, как анализ волноводных и полосковых структур, моделирование антенн и расчет сложных невзаимных устройств, содержащих гиротропные среды [11].

В данном методе какое-то определенное пространство разбивается на тетраэдры (элементы), диэлектрические свойства которых изменяются от элемента к элементу [9].

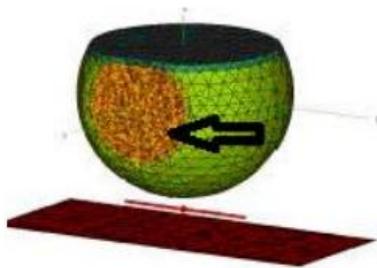


Рисунок 4 – Область, которая рассчитывается методом конечных элементов

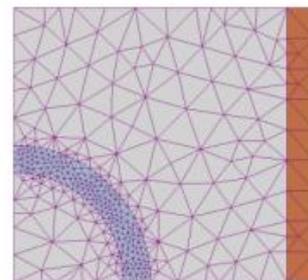


Рисунок 5 – Разбиение на конечные элементы

Размер элементов можно менять, уменьшая его вблизи интересующей области и увеличивая для снижения затрат процессорного времени [9].

Процесс разбиения пространства на элементы включает определенную последовательность шагов. Первым шагом является дискретизация области, то есть построение сетки и определение материала элементов. Область, в которой решается задача, аппроксимируется непересекающимися подобластями – конечными элементами, вершины (узлы) которых предназначены для описания геометрии элемента и для задания компонент решения [14]. Компоненты решения в узле называются степенями свободы, число которых зависит от рассматриваемых задач. В качестве степеней свободы могут фигурировать как узловые значения неизвестной функции, так и ее производные по пространственным координатам в узлах.

Вторым шагом в процессе конечно–элементного анализа является выбор базисных функций, которые чаще всего выбираются в виде полиномов. Они могут иметь различный порядок: линейный, квадратичный, кубический и т.д. [14].

Третий шаг – составление системы линейных алгебраических уравнений, количество которых равно количеству неизвестных значений в узлах, на которых ищется решение исходной системы, прямо пропорциональное количеству элементов, и ограничивается только возможностями ЭВМ [9]. Затем решается система уравнений, и определяются расчетные величины в элементах, которые чаще всего являются производными от неизвестных функций.

В работе [15] была разработана вычислительная модель для электромагнитного моделирования. Здесь уравнения Максвелла дискретизируются методом конечных элементов на тетраэдрической сетке. В [15] рассмотрен тонкий провод длиной 41 м и радиусом 10 мм. Провод дискретизируется 42 тетраэдрами с длиной ребра 1,08 м. Антенна возбуждается в средней точке источником напряжения. Была проведена проверка метода при различных относительных положениях провода и сетки.

На рисунке 6 показана частотная зависимость входного сопротивления тонкого провода при различных сетках и в сравнении с результатами, полученными методом моментов. «Хорошо видно, что от сетки результаты не зависят, в то время как с повышением частоты расхождения с методом моментов увеличиваются» [16].

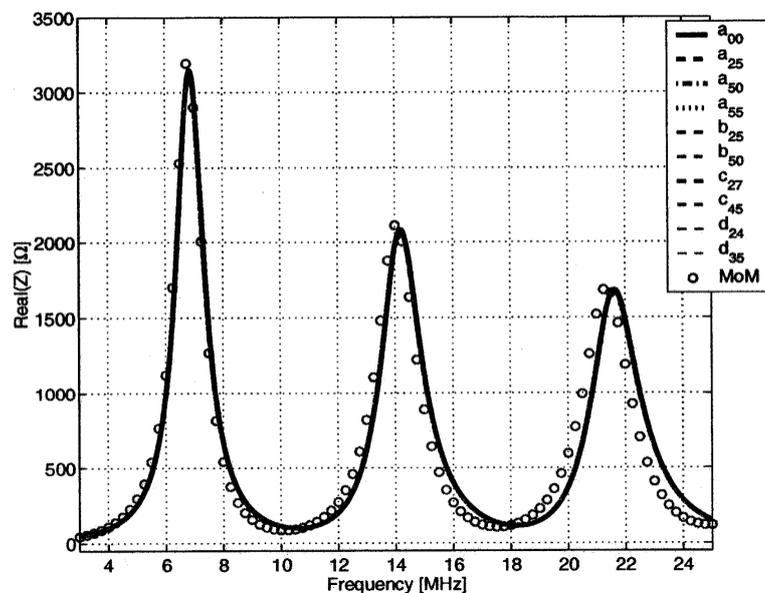


Рисунок 6 – Частотная зависимость входного сопротивления тонкого провода [16] при различных сетках (сплошная кривая) и в сравнении с результатами, полученными методом моментов (кружочки) [15,16]

Метод конечного интегрирования (FIT-метод)

Программа CST Microwave Studio базируется на методе конечного интегрирования, заключающегося в пространственной дискретизации пространства, то есть пространство разбивается на дискретные ячейки (сетку). Если применяется декартова система координат, то FIT-метод может рассматриваться как метод конечных разностей во временной области. Данный метод является частным случаем метода конечного интегрирования. Очень важной особенностью решения во временной области является зависимость требуемых вычислительных ресурсов от размеров структуры [7].

Вычисления во временной области позволяют выполнять электродинамический анализ трехмерных структур для всего диапазона частот за одну итерацию, что делает этот метод быстрым для решения сверхширокополосных задач. Кроме того, в пакет CST MWS включен вычислительный модуль, основанный на методе конечных элементов (FEM) и использующий пирамидальную сетку, а также модуль для вычислений собственных мод (Eigenmode) [6].

Метод конечных интегралов имеет высокую эффективность в задачах, в которых необходим анализ нестационарных процессов в неоднородном, анизотропном пространстве для объектов с произвольной формой границ. Этот численный метод обеспечивает универсальную схему пространственной дискретизации, которую можно применять к различным случаям, начиная со статического поля и заканчивая высокочастотными расчетами во временной и частотной областях.

В отличие от других численных методов, FIT-метод рассматривает уравнения Максвелла не в дифференциальной форме, а в интегральной:

$$\oint_L \vec{E} \cdot d\vec{l} = - \iint_S \frac{\partial \vec{B}}{\partial t} \cdot d\vec{s}, \oint_L \vec{H} \cdot d\vec{l} = \iint_S \left(\frac{\partial \vec{D}}{\partial t} + \vec{j} \right) \cdot d\vec{s}, \quad (3)$$

$$\oiint_S \vec{D} \cdot d\vec{s} = \iiint_V \rho dV, \oiint_S \vec{B} \cdot d\vec{s} = 0. \quad (4)$$

Для решения этих уравнений необходимо определить область расчета. После определения расчетной области ее необходимо разбить на конечное число ячеек, представляющих собой тетраэдральные (четырёхгранные) или гексагональные (шестигранные) ячейки. При этом необходимо, чтобы ячейки плотно прилегали к друг другу так, чтобы пересечение двух различных ячеек либо отсутствовало, либо представляло собой двумерный многоугольник общей одномерной гранью обеих ячеек или точкой. Это разбиение и дает конечную группу ячеек, представляющую собой расчетную сетку, называемой первичной. Помимо первичной сетки, которую можно в программе CST Microwave Studio, есть еще одна, вторичная или двойная, которая строится ортогонально первичной [17].

Пространственная дискретизация уравнений Максвелла производится на этих двух ортогональных сетках. После чего уравнения Максвелла отдельно записываются для каждой грани ячейки. На рисунке 7 показана дискретизация первого уравнения (3):

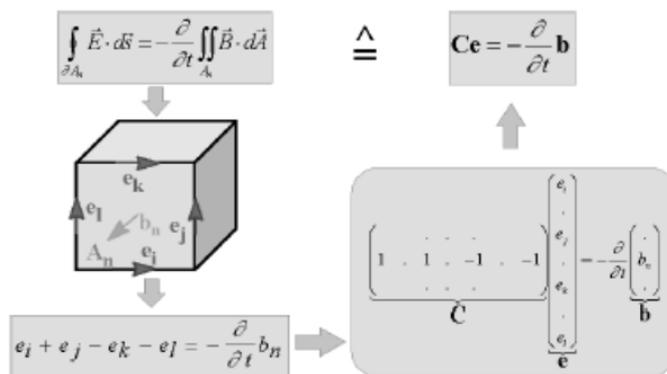


Рисунок 7 – Дискретизация уравнения (3), соответствующего правилу Фарадея

Реализация метода конечного интегрирования более подробно рассмотрена в работе [10]. В работе [17] авторы исследовали возможность метода конечного интегрирования при прогнозировании распространения радиоволн в неоднородных анизотропных средах. Использование данного метода показало лишь небольшое ухудшение точности расчета уровня сигнала (0,3 дБ). По результатам проведенных расчетов и экспериментальных измерений, приведенных в таблице 1, авторы показали, что метод конечных интегралов обеспечивает высокую точность прогнозирования и позволяет обеспечить детализацию расчета среднего локального уровня поля вплоть до пространственных масштабов длины волны.

Таблица 1– Среднеквадратическое отклонение результатов расчета и экспериментальных измерений

Частота	Модель расчета	1 положение передатчика, СКО (дБ) для доли площади измерений			2 положение передатчика, СКО (дБ) для доли площади измерений		
		80%	90%	100%	80%	90%	100%
400 МГц		80%	90%	100%	80%	90%	100%
	CST Microwave Studio	2,08	2,55	3,63	2,93	3,89	5,17
	CST Microwave Studio, малая модель	2,33	2,68	3,66	Нет данных		
900 МГц		80%	90%	100%	80%	90%	100%
	CST Microwave Studio	3,68	4,34	5,20	3,27	3,91	4,63

В работе [6] авторы привели сравнение результатов СВЧ-моделирования двух микрополосковых антенн, выполненных с помощью шести предназначенных для этих целей, широко распространенных коммерческих программных продуктов (ADS Momentum, MAGMAS 3D, IE3D, FEKO Suite, HFSS, CST MWS). Все эти программы используют различные численные методы для задач электродинамического моделирования. Ниже приведено сравнение полученных результатов моделирования микрополосковых антенн.

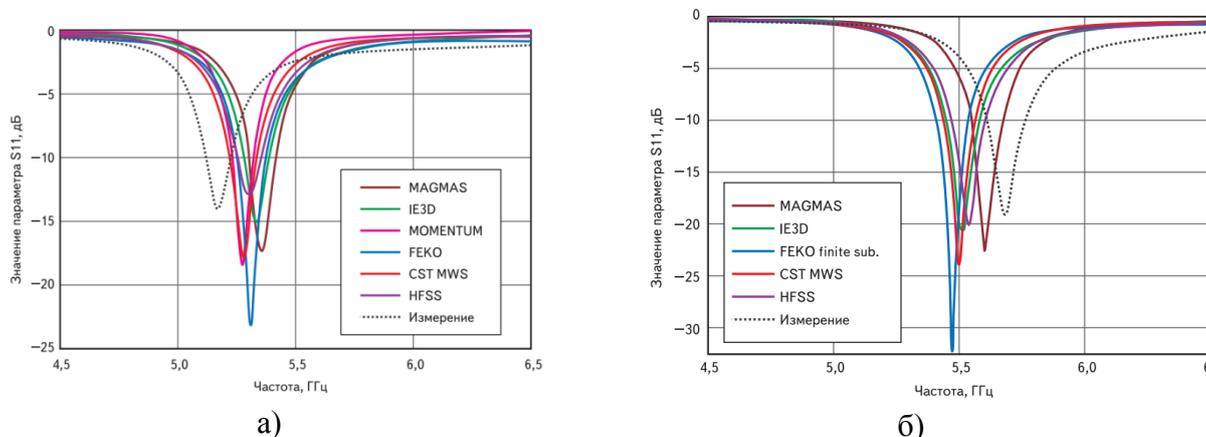


Рисунок 8 – Рассчитанные зависимости параметров S11 для микрополосковой антенны: а) с однородной подложкой, б) с неоднородной подложкой

По результатам сравнения этих программ можно сделать выводы о том, что «классические» микрополосковые антенны рассчитываются каждым пакетом с разбросом результатов не более чем 1,5%.

Таким образом, программные пакеты, использующие метод моментов, имеют высокую скорость моделирования, а также низкую стоимость. Однако программы, использующие методы FIT и FEM, способны моделировать более сложные трехмерные структуры [6].

Также адаптивное наложение сетки в HFSS и CST MWS позволяет лучше отслеживать сложные индуктивные связи в анализируемой модели в автоматическом режиме, а программы на основе МоМ требуют более внимательного контроля при построении сетки разбиения. Также практическая тенденция, направленная на

миниатюризацию антенн, уменьшает преимущества использования программ, использующих метод моментов, так как при малых размерах подложки дифракция и отражения на ее границах становятся ярко выраженными. Таким образом, программы, использующие метод конечных элементов и метод конечного интегрирования на текущий момент, лучше подходят для моделирования антенн малых размеров [6].

Сравнение результатов моделирования микрополосковой антенны в программах CST Microwave Studio и Matlab Antenna Toolbox

В данной статье приводятся результаты сравнения моделирования оптимизированной микрополосковой антенны в программах CST Microwave Studio и Matlab Antenna Toolbox (AT). В работе [1] автором предложен новый способ поиска оптимальной конструкции ячеек-резонаторов для микрополосковой антенны, позволяющий целенаправленно оптимизировать параметры ячеек-резонаторов, значительно улучшающих ее электрические характеристики [1]. Новый метод объединяет модель регрессии, использующую глубокую сеть на основе полносвязных нейронных слоев, с поиском на основе алгоритма условной оптимизации для нахождения наилучших конструктивных параметров ячеек – резонаторов[1]. Данный метод состоит из двух шагов, показанных на рисунке 9.

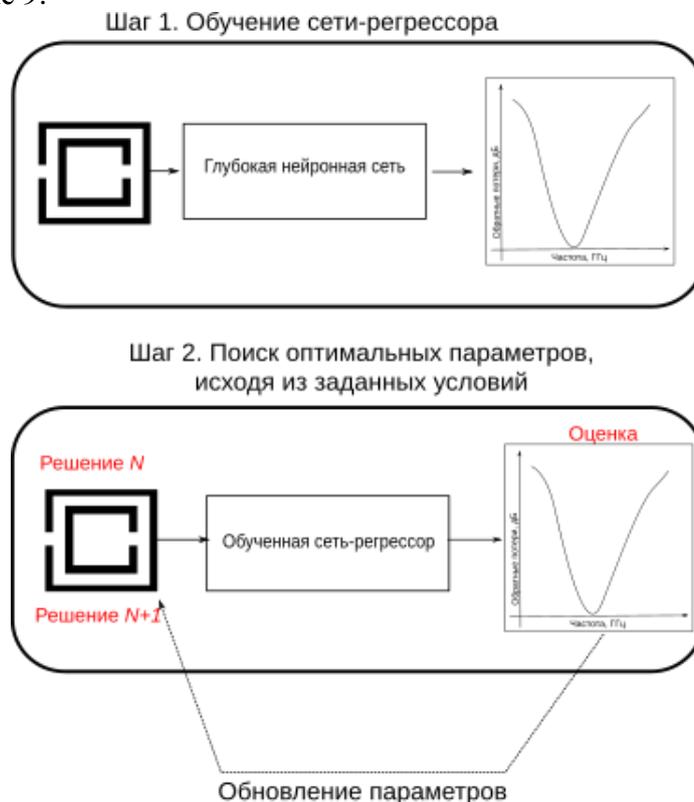


Рисунок 9 – Общая схема поиска оптимальной конструкции CSRR-ячеек микрополосковой антенны

Более подробно о новом способе, позволяющем оптимизировать параметры CSRR-ячеек резонаторов, можно ознакомиться в работе [1], в которой автор более подробно рассматривает обучение сети-регрессора для определения электрических характеристик микрополосковой антенны с CSRR-ячейками.

На рисунке 10 показана ячейка метаматериала на основе комплементарных разомкнутых колец CSRR, которая была добавлена в конструкцию антенны для улучшения ее электрических характеристик.

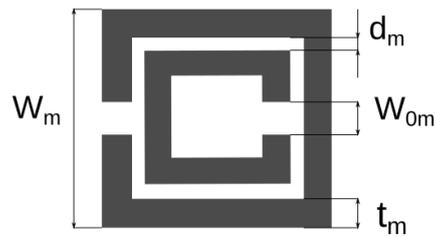


Рисунок 10 – Параметры CSRR-ячейки резонатора

Моделирование оптимизированной микрополосковой антенны проводилось автором в программе Matlab Antenna Toolbox (AT), которая для моделирования может задействовать только одно ядро CPU, в отличие от CST MWS, которая может использовать параллельную обработку на множестве ядер CPU и GPU [1]. На рисунке 11 показана конструкция микрополосковой антенны, выполненная в программах CST Microwave Studio (рис.12 а) и Matlab Antenna Toolbox (рис.12 б).

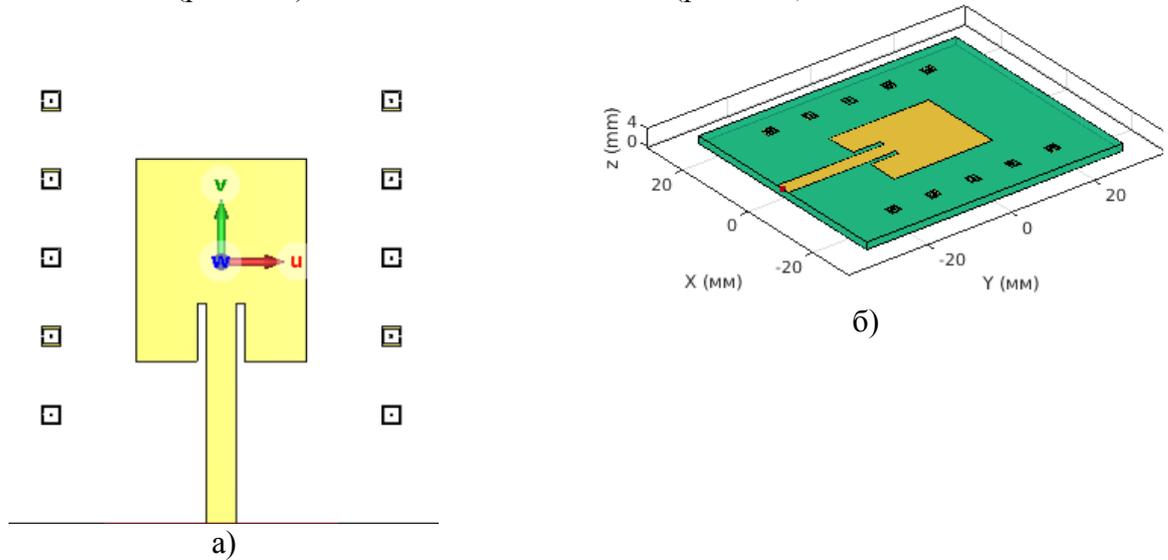
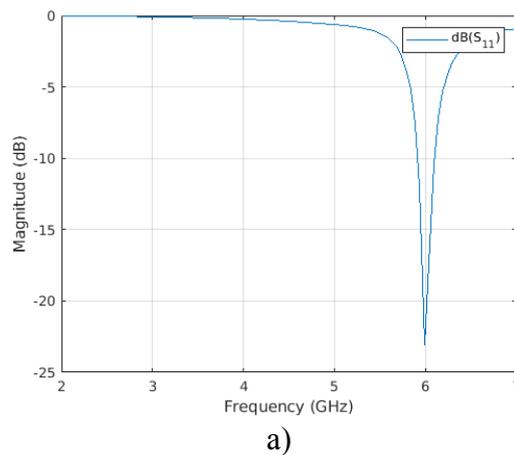
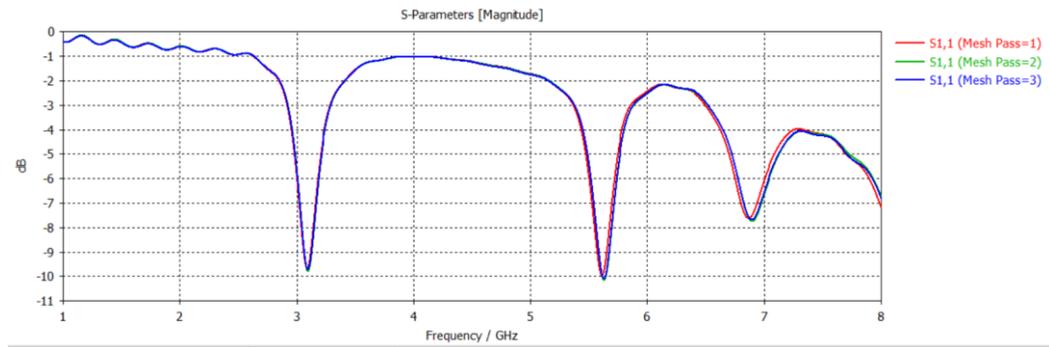


Рисунок 11 – Микрополосковая антенна с CSRR-ячейками (внешний вид): а) в программе CSTMWS , б) в программе MatlabAT

На рисунках 12,13 приведены результаты моделирования данной антенны в программах CST Microwave Studio и Matlab Antenna Toolbox (AT).

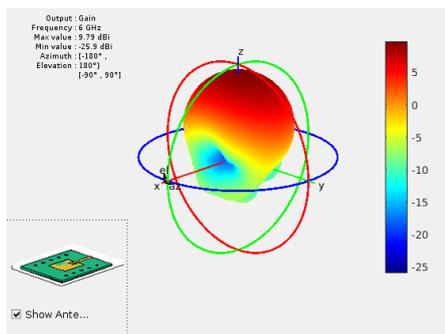


а)

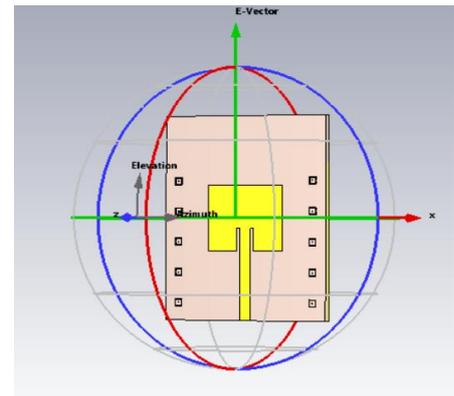


б)

Рисунок 12 – Кривая обратных потерь для микрополосковой антенны, полученная в программе MatlabAT (а) и CSTMWS (б)



а)



б)

Рисунок 13 – Диаграмма направленности микрополосковой антенны в программах MatlabAT (а) и CSTMWS (б)

Заключение

Сравнительный анализ численных методов моделирования показал, что каждый метод имеет ряд недостатков и преимуществ. Для более быстрого моделирования антенн, имеющих довольно протяженные размеры, лучше использовать метод моментов. Так как микрополосковые антенны имеют сравнительно малые размеры, что является их преимуществом в настоящее время, метод моментов теряет свое преимущество, так как происходит дифракция и отражение на границах диэлектрической подложки. Поэтому здесь лучше использовать такие методы моделирования, как метод конечных элементов и метод конечного интегрирования, которые, помимо этого, позволяют моделировать более сложные трехмерные модели.

Как видно из представленных рисунков 12,13, результаты моделирования, полученные программой MatlabAT, сильно отличаются от результатов, которые были получены в программе CST Microwave Studio. Данные отличия можно объяснить тем, что программные пакеты Matlab Antenna Toolbox и CST Microwave Studio для моделирования используют разные численные методы: Matlab Antenna Toolbox базируется на методе моментов в частной области, тогда как в основе программы CST Microwave Studio лежит метод конечного интегрирования во временной области, а также метод конечных элементов. Помимо этого, нельзя исключать возможные погрешности в расчетах при построении конструкции антенны и ячеек-резонаторов, а также необходимо учесть большие затраты компьютерных ресурсов. В дальнейшем планируется получить результаты экспериментальным путем.

Литература

1. Касимова, А. У. Обзор и анализ современных методов измерения диэлектрической проницаемости горных пород / А. У. Касимова, С. Н. Верзунов // Проблемы автоматизации и управления. – 2022. – № 1(43). – С. 33-49. – EDN BLNKOG.
2. Верзунов С.Н. Способ оптимизации конструктивных параметров ячеек-резонаторов микрополосковых антенн на основе интеллектуального анализа данных // Электротехнические системы и комплексы. – 2022. –№ 3(56). – С. 54–64. URL: [https://doi.org/10.18503/2311-8318-2022-3\(56\)-54-64](https://doi.org/10.18503/2311-8318-2022-3(56)-54-64)
3. Будагян И. Ф. Инновационные технологии моделирования сверхширокополосных антенных систем // Т-Comm. 2014. №11. URL: <https://cyberleninka.ru/article/n/innovatsionnye-tehnologii-modelirovaniya-sverhshirokopolosnyh-antennyh-sistem> (дата обращения: 01.10.2022).
4. Хамед Ебрахим Абдо Махьюб, Кисель Наталья Николаевна. Исследование характеристик микрополосковой антенны с управляемым метаматериалом // Известия ЮФУ. Технические науки. – 2019. – №3 (205). URL: <https://cyberleninka.ru/article/n/issledovanie-harakteristik-mikropoloskovoy-antenny-s-upravlyаемым-metamaterialom> (дата обращения: 10.11.2022).
5. Квасников А.А., Демаков А.В., Иванов А.А., Ключкин Д.В., Комнатнов М.Е., Куксенко С.П. Система компьютерного моделирования антенн методом моментов // Системы управления, связи и безопасности. 2022. №1. URL: <https://cyberleninka.ru/article/n/sistema-kompyuternogo-modelirovaniya-antenn-metodom-momentov> (дата обращения: 01.11.2022).
6. Александр Васильченко, Ив Схольц, Вальтер Де Раад, Ги Ванденбош Качественная оценка вычислительных методов электродинамики на примере программных продуктов для высокочастотного моделирования микрополосковых антенн // Технологии в электронной промышленности. 2008. №3. URL: https://www.tech-e.ru/pdf/2008_03_52.pdf (дата обращения 10.11.22)
7. Курушин А.А., Пластиков А.Н. Проектирование СВЧ-устройств в среде CST MicrowaveStudio. – М. Изд-во МЭИ, 2011. – 155 с. URL: https://kurushin.ucoz.ru/_ld/0/1_CST_kniga_MEI_8.pdf (дата обращения 10.11.22)
8. <https://www.mathworks.com/products/antenna.html>(дата обращения 10.11.22)
9. https://storage.tusur.ru/files/8666/%D0%A0%D0%A2%D0%A1-1303_%D0%A1%D0%A0%D0%90%D0%92%D0%9D%D0%98%D0%A2%D0%95%D0%9B%D0%AC%D0%9D%D0%AB%D0%99%20%D0%90%D0%9D%D0%90%D0%9B%D0%98%D0%97%20%D0%9C%D0%95%D0%A2%D0%9E%D0%94%D0%9E%D0%92%20%D0%AD%D0%9B%D0%95%D0%9A%D0%A2%D0%A0%D0%9E%D0%9C%D0%90%D0%93%D0%9D%D0%98%D0%A2%D0%9D%D0%9E%D0%93%D0%9E%20%D0%9C%D0%9E%D0%94%D0%95%D0%9B%D0%98%D0%A0%D0%9E%D0%92%D0%90%D0%9D%D0%98%D0%AF%20%20%D0%9C%D0%95%D0%A2%D0%9E%D0%94%20%D0%9C%D0%9E%D0%9C%D0%95%D0%9D%D0%A2%D0%9E%D0%92%20%D0%98%20%D0%9C%D0%95%D0%A2%D0%9E%D0%94%20%D0%9A%D0%9E%D0%9D%D0%95%D0%A7%D0%9D%D0%AB%D0%A5%20%D0%AD%D0%9B%D0%95%D0%9C%D0%95%D0%9D%D0%A2%D0%9E%D0%92.pdf (дата обращения 01.10.22)
10. https://studbooks.net/2195070/tehnika/programma_modelirovaniya_vysokochastotnyh_elektromagnitnyh_poley_microwave_studio(дата обращения 10.11.22)
11. Гончаров В. А. Методы моделирования электромагнитных полей в вычислительных средах // Научный электронный архив. URL: <http://econf.rae.ru/article/5166> (дата обращения: 10.11.2022).
12. Банков С.Е., Грибанов А.Н., Курушин А.А. Электродинамическое моделирование антенных и СВЧ-структур с использованием FEKO. – М., One-Book,2013. – 423 стр.

- URL: https://kurushin.ucoz.ru/_ld/0/8_20131001__FEKO_.pdf (дата обращения 01.10.22)
13. Чугунов М.В. Анализ методов численного моделирования антенн радиоволновых средств обнаружения // Вестник УлГТУ. – 2010. – №1 (49). URL: <https://cyberleninka.ru/article/n/analiz-metodov-chislennogo-modelirovaniya-antenn-radiovolnovykh-sredstv-obnaruzheniya> (дата обращения: 10.11.2022).
 14. Ю. А. Сагдеева, С. П. Копысов, А. К. Новиков. С 138. Введение в метод конечных элементов: Метод. пособие. – Ижевск: Изд-во «Удмуртский университет». 2011. 44 с. URL: <http://elibrary.udsu.ru/xmlui/bitstream/handle/123456789/6649/2011272930.pdf> (дата обращения 10.11.22)
 15. F. Edelvik, G. Ledfelt, P. Lotstedt and D. J. Riley, "An unconditionally stable subcell model for arbitrarily oriented thin wires in the FETD method," in IEEE Transactions on Antennas and Propagation, vol. 51, no. 8, pp. 1797-1805, Aug. 2003, doi: 10.1109/TAP.2003.814750. URL: <https://ieeexplore.ieee.org/document/1219587> (дата обращения: 10.11.22)
 16. Кудин В. П. Метод конечных элементов в прикладной электродинамике / В. П. Кудин // Известия Гомельского государственного университета имени Ф. Скорины. Сер. Естественные науки. — 2017. — № 3. — С. 153—159. URL: <https://elib.gstu.by/handle/220612/18353> (дата обращения 10.11.22)
 17. Михайлов Е.В., Дудов Р.А., Захаров П.Н., Козарь А.В., Королев А.Ф. Исследование возможностей метода конечных интегралов при прогнозировании распространения радиоволн в неоднородных анизотропных средах // Сборник докладов III Всероссийской конференции “Радиолокация и радиосвязь”.– С. 601–606. URL: <http://jre.cplire.ru/jre/library/3conference/pdf/p013.pdf> (дата обращения 10.11.22)

ТЕХНИЧЕСКИЕ СРЕДСТВА СИСТЕМ КОНТРОЛЯ, ДИАГНОСТИКИ И УПРАВЛЕНИЯ

УДК 621.316.9.

А.К. Асанов, asanov_ak@mail.ru

А.А. Омокеева, aiza9@mail.ru

Кыргызский государственный технический университет им. И. Раззакова

ЭКСПЕРИМЕНТАЛЬНО-РАСЧЕТНОЕ ОПРЕДЕЛЕНИЕ ЭЛЕКТРОМАГНИТНОЙ ОБСТАНОВКИ ПС 110 КВ "АНАНЬЕВО"

Статья содержит экспериментально-расчетное определение электромагнитной обстановки ПС 110 кВ «Ананьево», расположенной в Иссык-Кульской области. На основе измеренных данных произведен расчет и моделирование параметров электромагнитной совместимости основного оборудования с системами вторичной коммутации. Произведены измерения удельных электрических сопротивлений грунта и сопротивления растеканию заземляющего устройства, на основе которых при коротких замыканиях и молниевом разряде определены опасности воздействия разностей потенциалов на вторичное оборудование. Произведен расчет среднего количества разряда молнии на территорию объекта. Произведены анализ и оценка состояния заземляющего устройства, сечений и коррозионного состояния элементов существующего заземляющего устройства, размещений вторичных цепей. Определены наличия и качества металlosвязей оборудования с заземляющим устройством объекта. Оценка результатов расчета сопоставлена с требованиями нормативно-технических документаций.

Ключевые слова: заземляющее устройство, удельное электрическое сопротивление грунта, сопротивление растеканию заземляющего устройства, вторичные цепи, короткое замыкание, импульсные помехи, молниезащита.

Введение

Система заземления и уравнивания потенциалов является важной частью современного энергообъекта. Характеристики ЗУ должны отвечать требованиям обеспечения электробезопасности обслуживающего персонала и надежной работы оборудования электроустановки в нормальных и аварийных условиях. ЗУ должно обеспечивать следующие эксплуатационные функции электроустановки:

- защиту людей от поражения напряжением шага и напряжением прикосновения при КЗ в электроустановках;
- действие релейных защит от замыкания на землю;
- действие защит от перенапряжений – разрядников, ОПН;
- отвод в грунт токов молнии с молниеприемников объекта и подходящих к нему тросовых молниеотводов;
- отвод рабочих токов (токов нулевой последовательности, тяговых токов и т.д.);
- снижение электромагнитных влияний на вторичные цепи и оборудование, в том числе импульсных перенапряжений при ударах молнии, протекании через ЗУ ВЧ-составляющей тока КЗ и т.п.;
- защиту проводящих коммуникаций (экранов и оболочек кабелей, труб воздухопроводов и т.п.) от токовых перегрузок;
- защиту от статического электричества;
- снижение разностей потенциалов, приложенных к изоляции цепей связи, уходящих с территории объекта – при КЗ на землю и молниевых разрядах.

В современных условиях состоянию заземляющего устройства и системы молниезащиты должно уделяться повышенное внимание, поскольку они оказывают существенное влияние на электромагнитную обстановку на объекте. Устройства РЗА, АСУ ТП, АСКУЭ и связи, основанные на микропроцессорных элементах, имеют широкие функциональные возможности и ряд других преимуществ перед электромеханическими устройствами. Однако, в отличие от них, современные устройства обладают сравнительно

высокой чувствительностью к электромагнитным помехам. Энергообъекты (ПС, ТЭЦ, ГЭС, и т.п.) являются мощными источниками электромагнитных полей и помех, поэтому для нормального функционирования современных устройств защиты и автоматизации необходимо обеспечивать их электромагнитную совместимость (ЭМС) с электромагнитной обстановкой (ЭМО) на энергообъектах. Система заземления играет важную роль в обеспечении ЭМС аппаратуры защиты, автоматизации, управления, связи.

Таким образом, необходимо проводить обследование систем заземления и уравнивания потенциалов на действующих энергообъектах для обеспечения электробезопасности и ЭМС. По результатам должны разрабатываться мероприятия по модернизации заземляющих устройств.

Целью настоящей работы является обследование существующего контура заземляющего устройства ПС 110/10 кВ «Ананьево» с выявлением нарушения электромагнитной обстановки в части влияния на надежность работы первичного оборудования и систем вторичной коммутации.

Краткие сведения об объекте

Объект расположен рядом с селом Ананьево Иссык-Кульской области. ПС «Ананьево» является понижающей подстанцией, осуществляющей прием, преобразование и передачу электрической энергии. ПС эксплуатируется с постоянным дежурным персоналом на объекте.

Измерения производились на объекте в августе 2022 г. При выполнении работ использовались следующие документы: - нормальная схема электрических соединений ПС «Ананьево» на 2022 год; - журнал уставок РЗА; - документ «Токи короткого замыкания на шинах 6 –500кВ подстанций Национальной электрической сети Кыргызстана (НЭСК) в максимальных и минимальных режимах по состоянию на 2021 г.».

В районе расположения ПС уровень грозовой активности составляет ~40 – 60 часов в год[2]. Рядом с территорией ПС находятся опоры ВЛ 110 кВ и 10кВ.

ПС имеет два распределительных устройства напряжением выше 1 кВ: ОРУ-110 кВ, выполнено по схеме две рабочие системы шин и ремонтная перемычка; РУ-10 кВ, выполнено в виде КРУН-10 кВ с двумя секциями шин. На ПС установлен один силовой трансформатор мощностью 25 МВА110/35/10 кВ. Питание собственных нужд осуществляется от щита собственных нужд 0,4 кВ, расположенного в здании ОПУ. ЩСН получает питание от ТСН 10/0,4 кВ, мощностью 63 кВА, установленного в КРУН-10 кВ. Питание постоянным током осуществляется от аккумуляторной, расположенной в здании ОПУ. От ПС отходят 2 ВЛ 110 кВ, а также ВЛ 10 кВ.

Размещение вторичных цепей

На открытой территории ПС (ОРУ-110 кВ, трансформатор, участок между ОПУ и РУ выше 1 кВ) цепи проложены в железобетонных незаглубленных кабельных лотках. Трассы прокладки лотков указаны на рис. 2. Часть цепей от электроаппаратов до лотков выполнены в металлических трубах и в виде металлокоробов, а в некоторых частях отсутствует защитный покров цепей. Силовые и контрольные кабели проложены в лотках хаотично, без разнесения на существенное расстояние рис. 1.

Участки цепей от электроаппаратов до клеммных шкафов выполнены, как правило, в металлических трубах, металлокоробах или в ПВХ-кожухе. Клеммные шкафы ЭА расположены непосредственно на стойках ЭА, к которым они относятся.

Вторичные цепи в здании ОПУ и КРУН 10 кВ проложены в кабельных каналах и/или открыто по воздуху. В настоящее время на подстанции применяется МП-аппаратура АСКУЭ и связи.



Рисунок 1 – Расположение силовых и контрольных кабелей в кабельных лотках

Принимаемые параметры токов КЗ

В таблице 1 и 2 приведены параметры по принятым в данной работе параметрам КЗ в сетях с заземленной и изолированной нейтралью. В качестве исходных данных использовались значения токов КЗ[3], а также справочные данные по применяемому трансформаторному оборудованию.

Таблица 1 – Параметры токов КЗ в сетях с заземленной нейтралью

U, кВ	Ток КЗ, кА		Токи в нейтральных трансформаторов, кА	Время отключения КЗ, сек(основная защита / с учетом УРОВ)	Параметры ВЧ-составляющей тока КЗ	
	Трехфазного	Однофазного			Ток, А	Частота, кГц
110	2,057	1,58	1Т/2Т	Не предоставлено	0,1 / 0,7	1000* / 1000*

* - параметры ВЧ-составляющих токов КЗ по стороне 110 кВ выбраны в соответствии с [9].

Таблица 2 – Параметры токов замыканий в сетях с изолированной нейтралью

Напряжение, кВ	Ток КЗ, кА		Отключение однофазных замыканий	Время отключения КЗ, сек (основная/резервная защита)
	Трехфазного	Двухфазного		
10кВ	8,97	7,648	Не предусмотрено	0,5/ 1,5

Параметры вторичных цепей

Согласно [4], прочность изоляции вторичных цепей на промышленной частоте принимается равной 1,2 кВ (действующее значение).

Согласно [4], прочность изоляции вторичных цепей при воздействии импульсных перенапряжений принимается равной 6 кВ (IV категория перенапряжения).

МП-аппаратура, предназначенная для применения на электрических станциях и подстанциях, должна удовлетворять требованиям [5]. В соответствии с этим принимаем устойчивость аппаратуры к микросекундным импульсным помехам большой энергии – 4 кВ и однократным колебательным затухающим помехам – 2,5 кВ.

В случае протекания по экранам кабелей вторичных цепей части тока КЗ допустимый нагрев экранов определяется согласно [1, п. 1.4.16] и составляет 150⁰ С (с учетом начальной температуры).

1. Результаты измерений и расчетов

Определение состояния ЗУ

ЗУ ПС образовано заглубленными искусственными заземлителями. Заземлители проложены преимущественно на глубине 0,5 – 0,8 метра, что удовлетворяет требованиям [1, п. 1.7.92]. Частичное растекание токов возможно по грозотросам ВЛ 110 кВ. Внешняя ограда ПС не заземлена. Ограда ПС выполнена при помощи ж/б стоек, электрически соединенных с металлическими звеньями ограды (сетка «рабица»).

В связи с отсутствием проектной и исполнительной схемы ЗУ ОРУ-110 кВ, с учетом измеренного значения сопротивления растеканию ЗУ, была построена адекватная модель ЗУ (рис. 2) с помощью ПО «Контур»[11, 12].

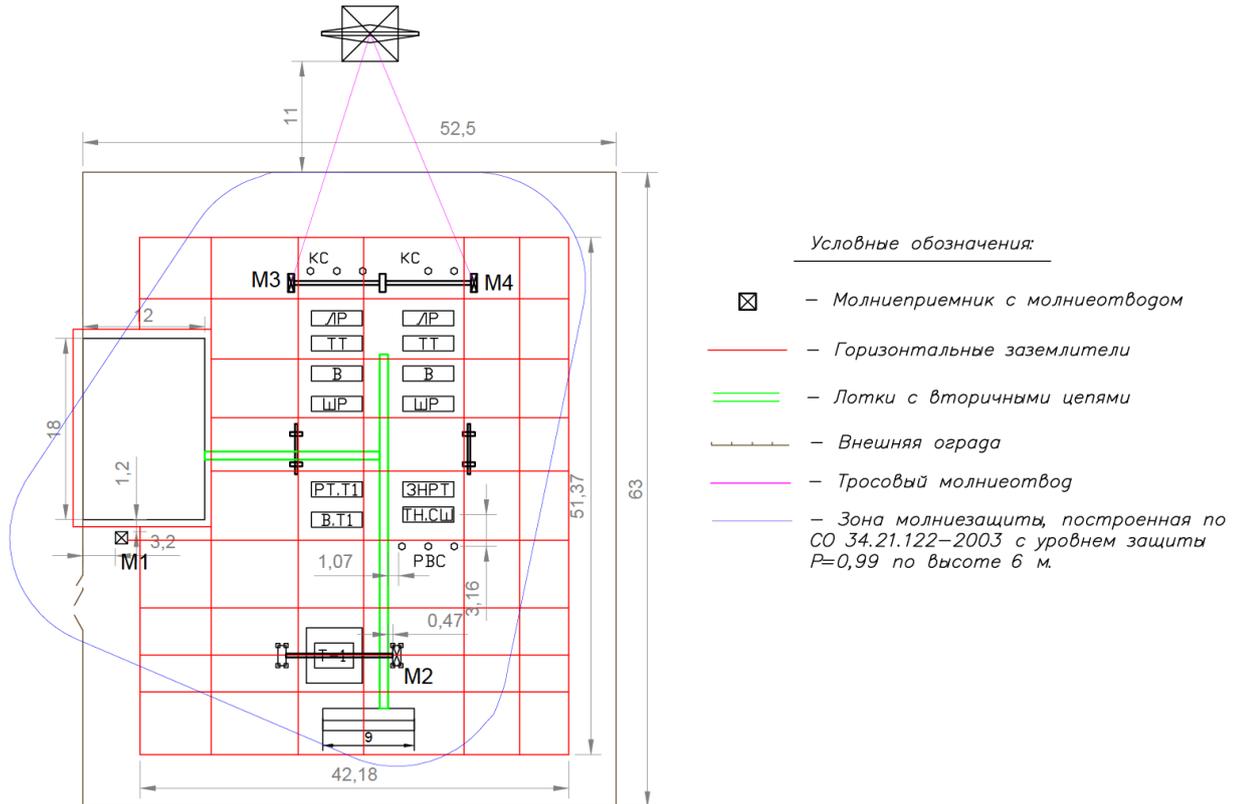


Рисунок 2 – Расчетная модель ЗУ с планом размещения основного оборудования и вторичных цепей

КРУН-10кВ представляет собой металлическую конструкцию, поэтому элементы самого здания используются в качестве СУП. Вокруг здания ОПУ проложены горизонтальные заземлители. СУП в помещениях РЩ ОПУ образована закладными элементами в полу помещения, а также специально проложенными перемычками (ШУП). В ходе обследования отклонений от требований НТД не выявлено.

Растекание существенной части тока с ЗУ объекта может происходить по тросовым молниеотводам.

Оценка сечений и коррозионного состояния элементов существующего ЗУ

Согласно [6], контролировалось сечение и коррозионное состояние элементов ЗУ. Места проведенных вскрытий грунта для контроля коррозии и состояния заземляющих проводников и заземлителей показаны на рисунках 2 и 3. Результаты обследования коррозионного состояния ЗУ представлены в таблице 3.

Таблица 3 – Сечение и коррозионное состояние заземлителей

Место	Материал	Форма	Сечение, мм ²	Коррозия (% от сечения)	Эффективное сечение, мм ²	Нагрев при КЗ, °С	Критерий по [1, п. 1.7.111-112]	Итог
ОРУ 110 кВ	Сталь	Квадрат 16x16 мм	256	5-30	179	44	400°С	Удовл.



Рисунок 3 – Коррозийное состояние заземлителей

Оценка удельного сопротивления грунта

Измерения удельного сопротивления грунта проводились как на территории ПС, так и за ее пределами по методу вертикального электрического зондирования. Глубина промерзания грунта в зимний период составляет не более 2,0 м (согласно карте нормативных глубин промерзания грунтов) [1].

Согласно результатам измерений и аппроксимации, проведённой в соответствии с [7], данным [1, 6], принимаются следующие параметры эквивалентной модели грунта:

- летний период: двухслойный грунт с удельным сопротивлением 350 Ом*м (верхний слой) и 980 Ом*м (нижний слой). Глубина раздела слоев – 2м;
- зимний период: однослойный грунт с удельным сопротивлением 980 Ом*м.

Определение сопротивления растеканию ЗУ

Сопротивление растеканию является одной из главных характеристик ЗУ электроустановок. Сопротивление растеканию измерялось с помощью классического метода «амперметра-вольтметра» интегрированным прибором MRU-120. Токовый и потенциальный зонды были вынесены за территорию ПС на расстояния порядка 250 и 300 метров соответственно. Данные расстояния примерно в 3 раза превышают размер ЗУ ПС по диагонали.

Измеренное сопротивление растеканию с учетом естественных заземлителей составляет 2,2 Ом.

На основании результатов измерений был проведен расчет распределения токов по отходящим коммуникациям. Распределение токов имеет следующий вид:

- 38% токов растекается по тросовым молниеотводам;
- 62% тока растекается с ЗУ ПС.

По результатам расчетов, сопротивление растеканию ЗУ объекта в зимний период составит около 2,6 Ом.

Согласно [1], сопротивление растеканию ПС не должно превосходить 0,5 Ом в любое время года с учетом естественных заземлителей [1, п.1.7.90], но с учетом удельного сопротивления грунта $\rho=980$ Ом*м допускается превысить значение на $\rho*0,002$ раза [1, п.1.7.108], , при этом сопротивление растеканию ЗУ составит 0,98 Ом. Таким образом, сопротивление растеканию ЗУ ПС не соответствует требованиям НТД.

Определение наличия и качества металловязей оборудования с ЗУ объекта

С целью контроля электрической целостности заземляющего устройства предусматривается контроль качества электрической связи с ЗУ аппаратов и конструкций в пределах обследуемой части объекта. Контроль проводится путем измерения сопротивления связи по четырехпроводной схеме [6]. При необходимости контролировалась доля тока в проводящих коммуникациях, отходящих от электроаппаратов и конструкций.

Также измерялись сопротивления связи с ЗУ ПС мест заземления нейтралей трансформаторов. При этом, согласно [6, п.2.2], критерий для сопротивления связи с ЗУ ПС нейтралей трансформаторов составляет 0,2 Ом, а для остальных ЭА такого критерия в НТД нет. Поэтому для остальных ЭА будет использоваться следующий критерий для сопротивления связи ЭА с ЗУ ПС:

- Если значение тока КЗ на ОРУ не превышает 5 кА, в качестве максимально допустимого значения будет приниматься величина 0,2 Ом (по аналогии с требованиями [6, п.2.2] для нейтралей трансформаторов).
- Если значение тока КЗ на ОРУ превышает 5 кА, то в качестве максимально допустимого значения будет приниматься такая величина сопротивления, при которой в случае КЗ на ЭА разность потенциалов на частоте 50 Гц, приложенная к изоляции вторичных цепей, не будет превышать 1,2 кВ (согласно требованиям [4]).

Таким образом (согласно данным по токам КЗ – таблица 1 и 2), для ПС «Ананьево» принят критерий связи ЭА с ЗУ ПС – 0,2 Ом.

Измеренные значения сопротивления связи лежат в диапазоне от 0,08 до 4,9 Ом (типично 0,12 – 0,16 Ом).

Выявлены конструкции, связи которых с ЗУ ПС нарушены. Эти конструкции указаны ниже в таблице 4.

Таблица 4 – Аппараты и конструкции с нарушением связи с ЗУ ПС

№	Название	Сопротивление, Ом
1	М1	4,9
2	КРУН 10 кВ	2,8

Определение наличия и качества связей оборудования с заземлителями

В ходе проведенного обследования контролировалось состояние связи оборудования с заземлителями. Измерения проводились в соответствии с [4], согласно которому сопротивление связи оборудования с заземлителями не должно превышать 0,05 Ом.

Результаты измерений показали соответствие НТД.

2. Определение опасности коротких замыканий

Протекание через заземляющее устройство сверхтоков при КЗ на землю в сетях выше 1 кВ создает опасность воздействия разностей потенциалов на вторичное оборудование.

Моделирование КЗ в сетях выше 1 кВ

Согласно [8], проводилось имитационное моделирование КЗ. Для выполнения имитационного моделирования использовался цифровой прибор MRU-120. Измерения данным прибором производятся на частоте 128 Гц.

С целью исключения погрешности за счет частотной зависимости характеристик ЗУ дополнительно проводились расчеты на частоте 50 Гц с помощью ПО «Контур». Кроме того, данное ПО используется для определения сезонной зависимости разностей потенциалов, поскольку позволяет проводить расчеты для различных грунтовых условий.

Результаты измерений и расчетов приведены в таблице 5. Сравнение результатов измерений и расчетов на частоте 128 Гц показывает, что построенная модель ЗУ достоверна (максимальная погрешность не превышает 15%).

Таблица 5 – Токи и напряжения промышленной частоты при коротком или двойном замыкании на землю

Трасса кабеля	Место приложения воздействия	Расчетно-экспериментальные воздействия			Время воздействия, с	Допустимый уровень воздействия		Выводы
		Наибольшее напряжение на кабеле устройстве (f=128 Гц), кВ		Наибольшее напряжение на кабеле или устройстве (расчет, f=50 Гц), кВ		Наибольшее напряжение на кабеле или устройстве, кВ	Наибольшая температура нагрева экрана, оболочки или брони кабеля, 0С	
		Изм.	Расч.					
ОПУ – ОРУ-110	ТТ	1,87	1,92	1,54	0,7	1,2	150	Неудовл.
ОПУ – РУ-10	Т и КРУН-10	11,3	11,79	8,79	1,5	1,2	150	Неудовл.

Согласно результатам измерений, имитационного моделирования и расчетов, разности потенциалов при КЗ в сети 110 и 10 кВ будут представлять опасности для вторичного оборудования и МП-аппаратуры.

3. Система молниезащиты объекта

Система молниезащиты объекта организована мачтой молниеприемника М1, молниеприемниками на порталах М2-М4, а также грозотросами, заходящими на входные порталы 110кВ. Нумерация и места расположения указаны в соответствии с рисунком 2.

Расчет среднего количества разрядов молнии на территорию объекта за год

Поражение объекта молнией носит вероятностный характер и зависит от характеристик грозовой активности в регионе, геометрических параметров рассматриваемого объекта и характеристик окружающей местности. Поэтому выбор принимаемого в расчетах значения тока молнии будет определяться ожидаемым средним числом поражений объекта за год.

Расчет количества молниевых разрядов выполняется в соответствии с методикой [10].

Уровень грозовой активности [2] в районе расположения объекта: $T_d = 40 - 60$ часов в год.

Плотность ударов молнии на 1 кв. км в год рассчитывается по формуле:

$$N_g = 6.7 \cdot T_d / 100. \quad (1)$$

Если принять верхнее значение $T_d = 60$ часов в год, то $N_g = 4,02$.

Ожидаемое количество разрядов молнии в защищаемую территорию за год определяется по формуле:

$$N_D = N_g \cdot A_d \cdot C_d \cdot 10^{-6}. \quad (2)$$

Здесь A_d – площадь сбора разрядов для рассматриваемой территории, C_d – коэффициент, учитывающий влияние относительного местонахождения защищаемого объекта [9]. Вокруг ПС отсутствуют объекты равной или большей высоты, поэтому $C_d = 1,0$.

Ожидаемое количество поражений молнией территории открытой части объекта определяется согласно описанной методике. Принимается: $A_d = 26200$ кв.м.

Для принятой грозовой активности $N_D = 0,1054$, что соответствует примерно 1 удару в ~10 лет.

По данным [2], молниевые разряды с амплитудой тока 15 кА и выше будут составлять 15 % от общего числа молниевых разрядов.

Результаты моделирования импульсного сопротивления ЗУ молниеотводов

Результаты моделирования показывают, что наибольшая разность потенциалов на ЗУ при молниевых разрядах будет возникать при ударах в М1, М2, М3, М4 (табл. 6).

Таблица 6 – Обследование элементов системы молниезащиты

Обозначение	Тип элемента СМЗ	Наличие цепей до 1 кВ, отходящих от элемента СМЗ, Тип цепей:	Импульсное сопротивление ЭСМЗ, Ом	Расстояние от элемента СМЗ до:				Как выполнено заземление:	Нарушения НТД
				трасс прокладок и сторонних вторичных цепей, м	до ЭА с вторичными цепями, м	клеммных шкафов, м	зданий с МП аппаратурой, м		
М1	мачта с молниеотводом	-	6,2	10	18	18	2,5	Аппарат имеет плохую связь с ЗУ	Не соответствует требованиям ПУЭ-7 (п. 4.2.135)
М2	Портал с молниеприемником	Цепи питания прожекторов ~220В, питание ~380В.	3,7	0,5	7	12	23	растекание в 2 стороны	
М3	Портал с молниеприемником	-	3,2	10	2	7	20	растекание в 2 стороны	
М4	Портал с молниеприемником	-	3,2	10	2	6	30	растекание в 2 стороны	
РВС-ТНС Ш	Ограничитель перенапряжений	-	-	1	3,5	3,5	20	растекание в 2 стороны	

Анализ системы молниезащиты по условиям воздействия на вторичное оборудование

При ударах молнии в молниеотводы объекта возможно воздействие импульсных разностей потенциалов и наводок на вторичные цепи. Кроме того, опасность может представлять стекание импульсных токов с заземления высоковольтных ОПН. С целью выявления средств молниезащиты, протекание тока молнии через которые представляет опасность для вторичного оборудования, произведен предварительный визуальный осмотр системы молниезащиты. Результаты осмотра приведены в таблице 6. Список элементов системы молниезащиты, протекание тока молнии через которые представляет наибольшую опасность для первичного оборудования и систем вторичной коммутации, представлен в таблице 7.

Таблица 7 – Элементы системы молниезащиты, протекание тока молнии через которые представляет наибольшую опасность для вторичных цепей

Тип цепей	ЭСМЗ, представляющие наибольшую опасность	Обоснование
Цепи СН	М2	Прожектор установлен на конструкции с молниеприемником
Цепи РУ-10кВ		Расстояние до кабельного лотка не более 5 м
Цепи на ОРУ-110		Расстояние до лотков не более 0,5 м
Цепи в ОПУ	М1	Расстояние элемента СЗМ до ОПУ не более 2,5 м

Определение импульсных помех и перенапряжений во вторичных цепях при молниевом разряде

Определение уровней импульсных помех и перенапряжений во вторичных цепях производилось расчетным методом на ПО «Контур». Результаты приведены в таблице 8. Анализ таблицы показывает, что молниевые разряды в элементы СЗМ будут создавать разности потенциалов, опасные для всех групп цепей, гальванически связанных с ЗУ ПС.

Таблица 8 – Импульсные напряжения, воздействующие на контрольные кабели при ударах молнии

Обозначение ЭСМЗ	Трасса кабелей (N трассы)	Место заземления цепи	Ток молнии, кА/ток генератора, А	Параметры импульса, мкс	Разность потенциалов вдоль трассы, кВ	Импульсная помеха, кВ со стороны аппаратуры	Коэффициент ослабления, о.е.	Допустимый уровень воздействия, кВ приложенное к кабелю	Выводы
МЗ, М4	КС-110 - ОПУ	КС-110	15/1	10/350	9,2	9,2	1	6	Неудовл.
ПМ2	Цепи СН	Прожектор на ПМ2	15/1	10/350	26,7	26,7	1	6	Неудовл.
РВС-ТН СШ	ТНСШ – ОПУ	ТН-СШ	10/1	10/350	16,1	16,1	1	6	Неудовл.

Заключение

Электромагнитную обстановку на ПС «Ананьево» следует считать крайне жесткой. Наибольшую опасность представляют помехи при КЗ на ОРУ 110 кВ и импульсные помехи при молниевых разрядах.

Схема СУП в помещении КРУН-10кВ удовлетворяет требованиям НТД и ЭМС. Схема СУП в помещении ОПУ также удовлетворяет требованиям НТД.

Площадь сечения заземлителей и заземляющих проводников удовлетворяет требованиям НТД.

Удельное сопротивление грунта в месте расположения объекта является неблагоприятным по условиям растекания тока с заземлителей.

Сопротивление растеканию ЗУ ПС не удовлетворяет требованиям ПУЭ в любое время года. Значительный вклад в растекание тока вносят естественные заземлители, в первую очередь тросовые молниеотводы.

На объекте(ОРУ-10 кВ, РУ-10) выявлены конструкции с нарушением локальной металловязи с ЗУ ПС.

Разности потенциалов на промышленной частоте при КЗ в сетях выше 1 кВ представляют опасность для вторичных цепей.

Помехи и перенапряжения при молниевом разряде представляют опасность для вторичных цепей.

Литература

1. Правила устройства электроустановок. 7-е изд. – М.: Изд-во НЦ ЭНАС, 2002.
2. Чичинский М.И. Особенности грозопоражаемости в горных районах и их учет при выборе грозозащиты воздушных линий электропередачи. Дис.... на соискание ученой степени канд. техн. наук/Чичинский М.И. – Фрунзе, 1984.
3. Данные ОАО «Национальная электрическая сеть Кыргызстана» с 2012 по 2022г.
4. ГОСТ Р 50571-4-44-2011 (МЭК 60364-4-44:2007) Электроустановки низковольтные. Часть 4 – 44. Требования по обеспечению безопасности. Защита от отклонений напряжения и электромагнитных помех. – М.: Стандартинформ, 2012.
5. ГОСТ Р 51317.6.5-2006 (МЭК 61000-6-5-2001) Совместимость технических средств электромагнитная. Устойчивость к электромагнитным помехам технических средств, применяемых на электрических станциях и подстанциях. Технические требования и методы испытаний.
6. Методические указания по контролю заземляющих устройств электроустановок. РД 153-34.0-20.525-00, М. СПО ОРГРЭС, 2000.
7. Коструба С.И. Измерение электрических параметров земли и заземляющих устройств. – М. Энергоатомиздат, 1983.
8. Методические указания по определению электромагнитной обстановки и совместимости на электрических станциях и подстанциях. СО 34.35.311-2004. – М.: Изд-во МЭИ, 2004.
9. Методические указания по обеспечению электромагнитной совместимости на объектах электросетевого хозяйства. СТО-56947007-29.240.044-2010.
10. IEC 62305 Lightning Protection (МЭК 62305 Молниезащита).
11. Отчет НИР за 2022г. «Исследования электромагнитной обстановки объектов электроэнергетики и обеспечение электромагнитной совместимости устройств в электрических сетях» // НИИ Энергетики и связи при КГТУ им. И.Раззакова. – Бишкек, 2022. – 95 с.
12. Исследование влияния сопротивления грунта на сопротивление заземляющих устройств (на примере Нарынской области) / А. К. Асанов, Б. Ж. Джолдошбеков, Н. К. Джусупбекова, М. К. Аркарчиева // Проблемы автоматизации и управления. – 2022. – № 2(44). – С. 70-76. – EDN OUYGDG.

УДК 621.314.212

*Т. Кабаев,**Г.Ш. Эралиева, eralieva.gulmira@mail.ru,**Ж.К. Абдыбаева,**Ч.М. Бузурманкулова**Кыргызский государственный технический университет им. И.Раззакова*

УСТРОЙСТВО ДЛЯ КОНТРОЛЯ СОСТОЯНИЯ СИЛОВОГО ТРАНСФОРМАТОРА

Предложено устройство, позволяющее проводить оперативный контроль содержания газовых пузырьков в масле силового трансформатора (СТ), реагировать на бросок масла в сторону расширителя и на критическое превышение температуры масла. Конструктивно оно так же, как газовое реле, врезано в трубопровод между баком СТ и расширителем и защищено от внешних электромагнитных воздействий. В отличие от газового реле реагирует на изменение диэлектрических параметров масла в процессе эксплуатации СТ. Устройство состоит из трех конденсаторов с плоскими электродами. В первом конденсаторе, в зависимости от состояния СТ, диэлектриком может оказаться масло или масло с газовыми пузырьками. Диэлектриком второго и третьего конденсаторов является одна цельная плоская пластина из сегнетоэлектрика, которая может перемещаться в промежутке между электродами этих конденсаторов. При перемещении пластина все время находится между электродами второго конденсатора, но она может входить или выходить из промежутка между электродами третьего конденсатора. Эта сегнетоэлектрическая пластина с одного конца жестко прикреплена к верхнему торцу поршня из фторопласта, а другой торец поршня прижат к пружине. При этом на подвижный поршень, с торца прижатый к пружине, может воздействовать возникший в трубопроводе скачок давления масла. Каждый из конденсаторов является элементом, подключенным к плечам соответствующих четырех плечевых мостовых схем, к двум вершинам которых подается от сети отфильтрованное синусоидальное напряжение, а с других двух вершин сигналы подаются к соответствующим аналоговым входам АЦП информационно-управляющей системы СТ.

Ключевые слова: мониторинг; состояние масляного силового трансформатора; газовые пузырьки; трансформаторное масло; температура, сегнетоэлектрический конденсатор.

Введение и постановка задачи

Общепринятым в мировой практике является внедрение систем контроля параметров трансформаторов в процессе эксплуатации, по результатам которых вырабатываются меры по предупреждению внутренних повреждений (неблагоприятного развития дефектов) и аварийного отключения [1].

В настоящее время устройством, реагирующим на все повреждения внутри бака трансформатора, при которых происходит выделение газа, ускоренное перетекание масла или смеси масла с газом из бака в расширитель, а также на снижение уровня масла, является газовое реле, которое используется только как защитное устройство силового трансформатора (СТ) [2].

Имеются работы [3, 4], где показаны способы расширения функциональных возможностей существующих информационно-управляющих систем мониторинга силовыми масляными трансформаторами (СМТ) с помощью разработанных новых устройств. Эти устройства не реагируют на ускоренное перетекание масла или смеси масла с газом из бака в расширитель.

В работе сделана попытка разработки устройства, выполняющего, как и газовое реле, защитные функции от внутренних повреждений в СТ и в отличие от газового реле функцию мониторинга температурного состояния СТ.

Решение задачи

На рис.1, 2 представлены результаты решения поставленной задачи.

На рис.1 показаны все основные конструктивные элементы устройства, выполняющие как защитные функции СТ, так и функции мониторинга.

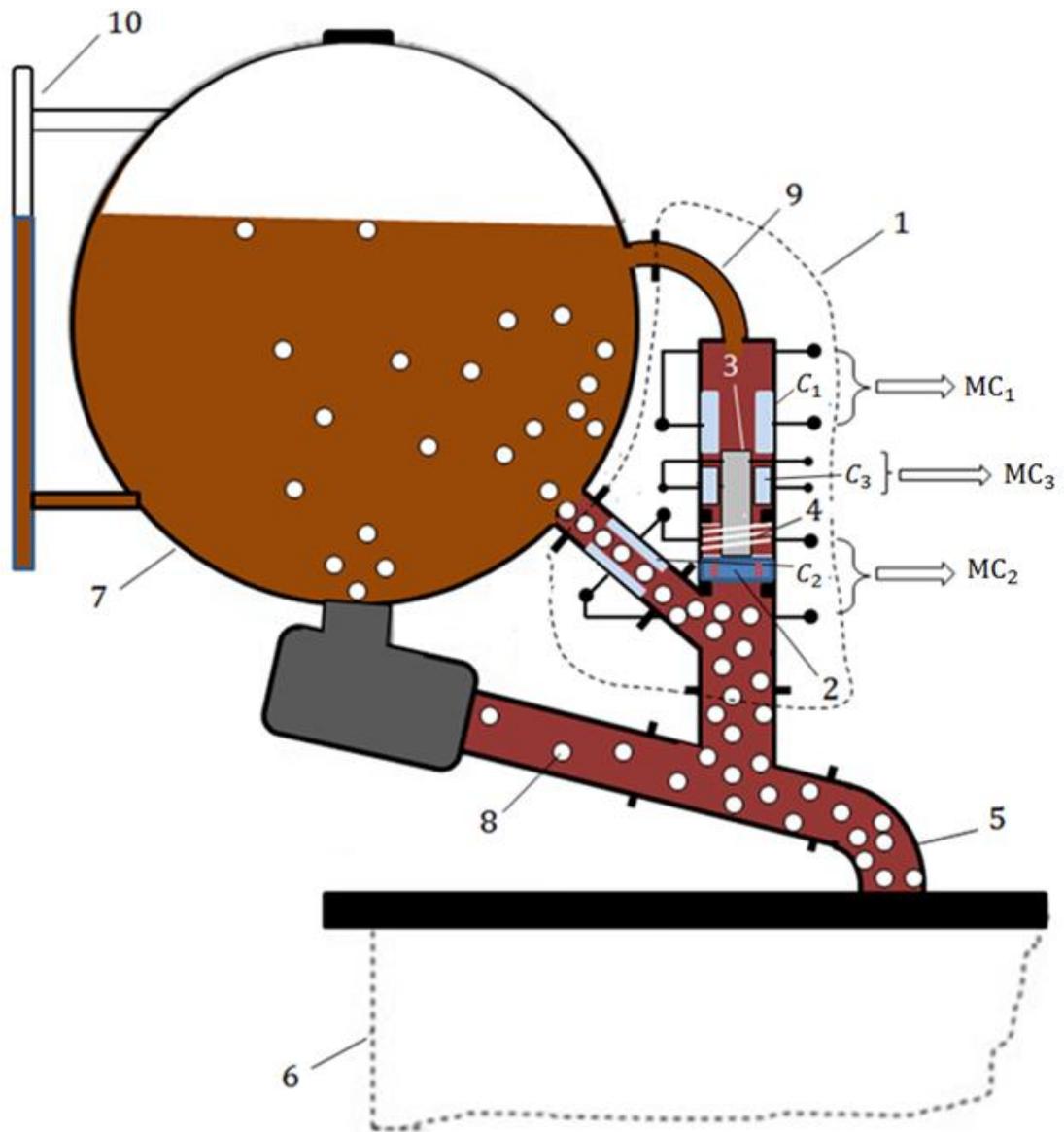


Рисунок 1— Устройство для оперативного контроля температуры масла, содержания газовых пузырьков в нем и регистрации броска масла в сторону расширителя силового трансформатора

На рис.1 предлагаемое устройство выделено пунктирными линиями, где 2 – подвижный поршень из фторопласта со сквозными каналами небольших диаметров для пропуска масла; 3 – плоская пластина из сегнетоэлектрика, жестко скрепленная с поршнем 2; 4 – пружина; 9 – соединительная трубка из механически прочного маслостойкого диэлектрического материала, соединяющая диэлектрический корпус устройства 1 с расширителем 7; 8 – газовые пузырьки, возникающие при некоторых внутренних повреждениях в СТ; C_1 – конденсатор с плоскими электродами, в межэлектродный промежуток которого может входить или выходить плоская пластина из сегнетоэлектрика; C_2 – конденсатор с плоскими электродами, вмонтированный в диэлектрическую трубку, в промежутке между электродами которого в зависимости от состояния СТ может быть или масло, или масло с газовыми пузырьками; C_3 – конденсатор с плоскими электродами, диэлектриком которого служит сегнетоэлектрическая пластина; MC_i – мостовые схемы. Остальные элементы – типовые и не относятся к предлагаемому устройству, т.е. 5 – трубопровод, соединяющий бак с расширителем 7; 6 – бак СТ; 7 – расширитель; 10 – указатель уровня масла.

В обоснование того, что конденсатор C_2 может выполнять функцию емкостного датчика пузырьков, обратимся к работе [5] (табл.1 – характеристики емкостного датчика,

полученные в результате эксперимента), где отмечено, что благодаря простоте изготовления и возможности минимизации (сконструирован на плате) и нашли применение в ряде отраслей промышленности.

Таблица 1

Емкость (нФ)	Диаметр пузырьков воздуха (мм)	Выходное напряжение (мВ)
43.50	0	18.60
42.63	0.82	20.30
42.54	1.00	22.01
41.07	2.97	40.50
40.44	3.55	43.70
39.38	4.00	47.90

Из формулы для расчета емкости конденсатора с двумя плоскими параллельными электродами видно, что когда масло, имеющее более высокую диэлектрическую проницаемость, чем газ, попадет в промежуток между электродами, то емкость конденсатора увеличивается по сравнению с той, когда пузырьки газа отсутствовали:

$$C = \frac{\varepsilon S}{d}, \quad (1)$$

где ε – диэлектрическая проницаемость диэлектрической среды; S – площадь поперечного сечения электродов; d – расстояние между ними.

Возникшие внутри бака трансформатора повреждения, приводящие к выделению газа, попадающего по трубопроводу в промежуток между электродами C_2 , уменьшают его ёмкость, а выходное напряжение повышается. Следовательно, рассматриваемый конденсатор выполняет функцию датчика пузырьков. При этом датчик реагирует и на размер пузырька (таб.1).

Конденсатор C_1 выполняет функцию датчика, реагирующего на значительные повреждения в СТ, когда вследствие бурного образования газа происходит бросок масла (или смеси масла с газом) в расширитель. В этом случае под воздействием потока масла поршень 2 (рис.1, 2) совместно с сегнетоэлектриком перемещается в промежуток между электродами конденсатора, значительно увеличивая его емкость, так как сегнетоэлектрик имеет высокую диэлектрическую проницаемость. Этот конденсатор представлен параллельной нелинейной схемой замещения.

В промежутке между электродами конденсатора C_3 постоянно находится сегнетоэлектрик (титанат бария $BaTiO_3$). Так как диэлектрическая проницаемость этого материала в значительной степени зависит от температуры [6] (рис.3), то этот конденсатор может выполнять функцию датчика температуры масла, так как сегнетоэлектрик находится в этой среде. Этот конденсатор представлен параллельной нелинейной схемой замещения.

Рассматриваемые ёмкостные датчики C_1, C_2, C_3 по существу являются элементами мостовых схем Шеринга (MC_1, MC_2, MC_3) [7], которые представляют собой одинарные четырехплечевые мосты, имеющие в одном из плеч подстроечные конденсаторы C_4, C'_4, C''_4 и параллельно присоединённые к ним постоянные активные сопротивления, R_4, R'_4, R''_4 , в противоположные плечи включены испытуемые объекты, в нашем случае C_1, C_2, C_3 , в третьи плечи включены подстроечные активные сопротивления R_3, R'_3, R''_3 , а в четвёртые — эталонные конденсаторы C_N, C'_N, C''_N . Отметим, что конденсаторы C_1 , и C_3 , у которых диэлектриками являются сегнетоэлектрики нелинейные.

На входы 1 и 2 (диагональ питания) всех МС подаются сетевое напряжение через фильтры, что избавляет их от паразитных гармоник. С выходов 3 и 4 (измерительная диагональ) сигналы подаются на соответствующие входы АЦП информационно-управляющей системы (ИУС) силового трансформатора. Все соединительные провода экранированы (рис.2).

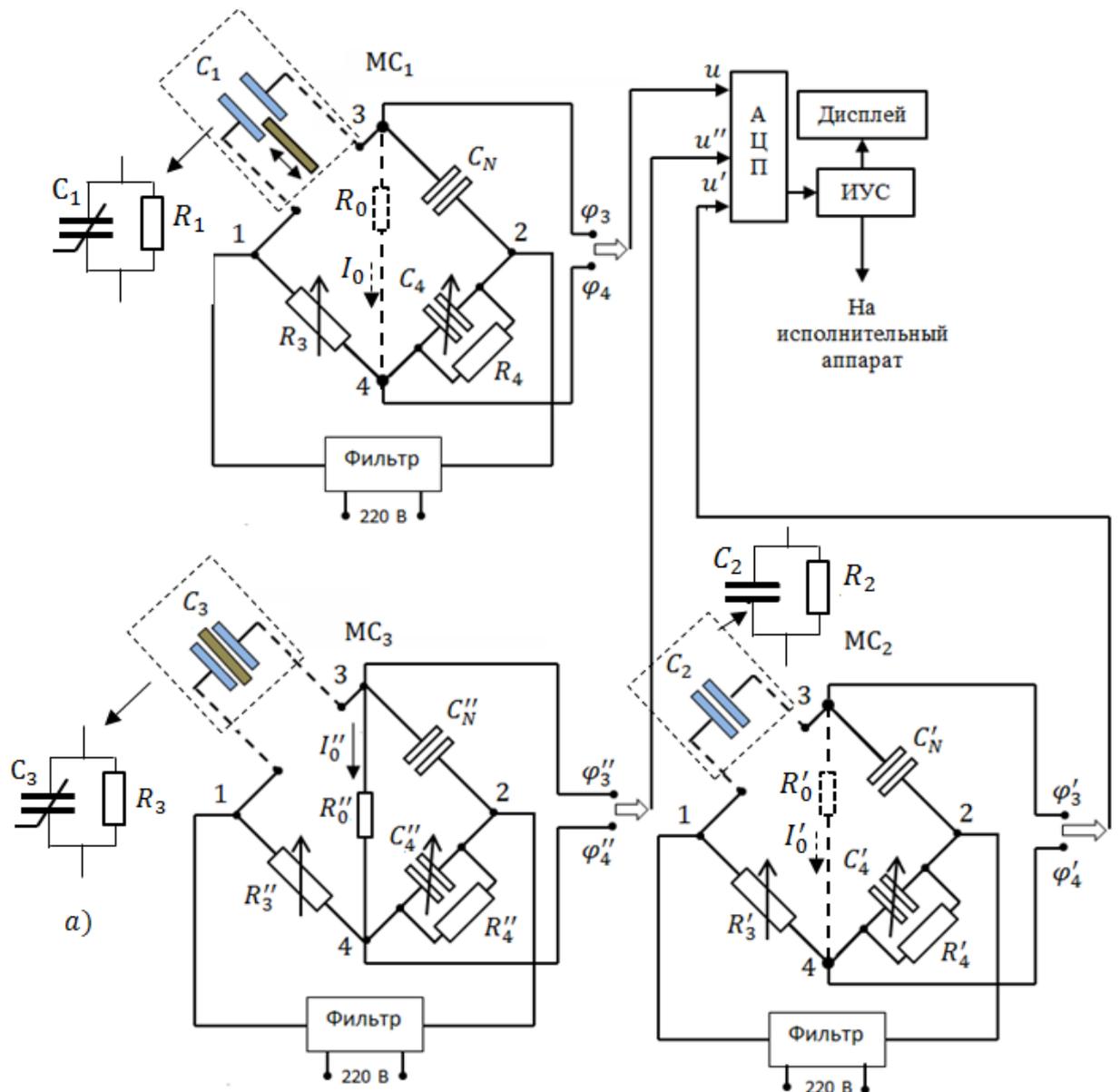


Рисунок 2 – Схемы подключения емкостных датчиков C_1, C_2, C_3 к плечам четырехплечевых мостовых схем, сигналы от которых подаются к соответствующим аналоговым входам АЦП, информационно-управляющей системы СТ

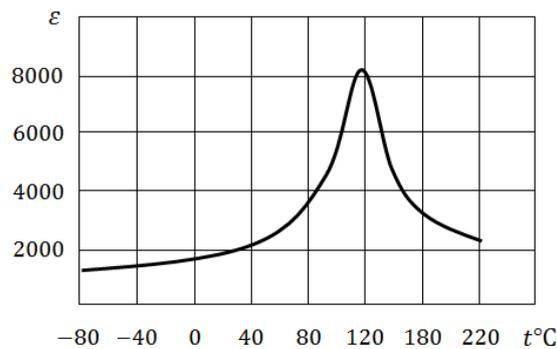


Рисунок 3 – Зависимость ϵ сегнетоэлектрика от температуры [6]

Как было отмечено выше, значения емкостей конденсаторов C_1 и C_2 существенно изменяются только при возникновении соответствующих внутренних повреждений в СТ. Поэтому в исходном состоянии при отсутствии внутренних повреждений параметры мостов MC_1, MC_2 подобраны таким образом (из условия равновесия моста), что эти мосты

сбалансированы. Следовательно, разности потенциалов на измерительных диагоналях МС ($\varphi_3 - \varphi_4, \varphi'_3 - \varphi'_4$) равны нулю и их равновесное состояние сохраняется до возникновения внутренних повреждений. При возникновении внутренних повреждений мосты МС₁, МС₂ выходят из равновесного состояния и падения напряжений в резисторах R_0, R'_0 от токов I_0, I'_0 , протекающих по ним, подаются на соответствующие входы АЦП информационной системы, сигнализируя о возникших повреждениях. Однако сбалансированный в начале мост МС₂ выходит из равновесия не только во время появления газовых пузырьков между его обкладками из-за внутренних повреждений в СТ, но при длительной его эксплуатации, когда масло начинает стареть и существенно повышается его тангенс угла диэлектрических потерь $tg(\delta)$. Следовательно, по характеру изменения тока I'_0 от времени эксплуатации ИУС могут выдавать информацию о степени старения масла.

Следует отметить, в мостовой схеме МС₂ конденсатор C_2 также представлен параллельной схемой замещения, однако в отличие от C_1 и C_3 имеет линейную характеристику.

Иная ситуация с режимом функционирования МС₃.

Предположим, что этот мост вначале настроен в равновесие. Однако это состояние не может оставаться стабильным, так как температура масла СТ в процессе эксплуатации изменяется случайным образом и, следовательно, также изменяются и значения емкости C_3 вследствие изменения диэлектрической проницаемости сегнетоэлектрика (рис.3). При этом на измерительной диагонали МС₃ возникает ток I''_0 , и падение напряжения на резисторе R''_0 подается на соответствующий вход АЦП (рис.2). Следовательно, по величине этого сигнала можно производить мониторинг температурного состояния СТ.

Режим функционирования моста МС₃ при изменении емкости конденсатора C_3 , связанный с изменением температуры масла, могут производить, анализируя характер изменения тока I''_0 в измерительной диагонали 3-4 (рис.2). Как известно, этот ток определяется по формуле [8]:

$$I''_0 = \dot{U} \frac{Z_1 Z_4 - Z_2 Z_3}{Z_1 Z_2 (Z_3 + Z_4) + Z_3 Z_4 (Z_1 + Z_2) + Z_0 (Z_1 + Z_2) (Z_3 + Z_4)}, \quad (2)$$

где Z_0 – сопротивление диагонали 3-4, Z_1, Z_2, Z_3, Z_4 , соответственно сопротивление плеч 1-3, 2-3, 1-4, 2-4 (рис.2). Значения этих сопротивлений определяем по следующим формулам:

$$Z_1 = \frac{R_3}{1 + j\omega C_3 R_3}; \quad (3)$$

$$Z_2 = \frac{1}{j\omega C''_N}; \quad Z_3 = R''_3; \quad Z_4 = \frac{R''_4}{1 + j\omega C''_4 R''_4}. \quad (4)$$

Формула (3) соответствует параллельной схеме замещения сопротивления плеча 1-3, (рис.2, а).

Из выражения (2) вытекает условие равновесия моста:

$$Z_1 Z_4 = Z_2 Z_3, \quad (5)$$

так как ток в диагонали будет отсутствовать, если выражение $(Z_1 Z_4 - Z_2 Z_3)$ в числителе (2) обратится в нуль.

Как было отмечено выше, в процессе эксплуатации СТ условие (5) для мостовой схемы МС₃ не выполняется из-за изменения температуры масла СТ. Поэтому по величине сигнала с выхода МС₃, который подается в ИУС через АЦП, можно проводить мониторинг температурного состояния СТ.

На рис. 4, а, б представлены соответственно компьютерная модель третьей мостовой схемы (МС₃) и результаты моделирования при изучении режима его функционирования. Кривые 1 и 2 соответствуют законам изменения падения напряжения на сопротивлении диагонали 3-4 МС₃, при этом 1 – соответствует значению C_3 , когда мост сбалансирован, 2 – несбалансированное состояние моста, когда значение емкости C_3 увеличивается в

четыре раза за счет увеличения температуры масла до 120 °С. Например, при температуре масла в 40 °С диэлектрическая проницаемость диэлектрика конденсатора C_3 равна 2000 (рис.3), а при 120 °С – 8000, следовательно, емкость, согласно формуле (1), увеличится в 4 раза.

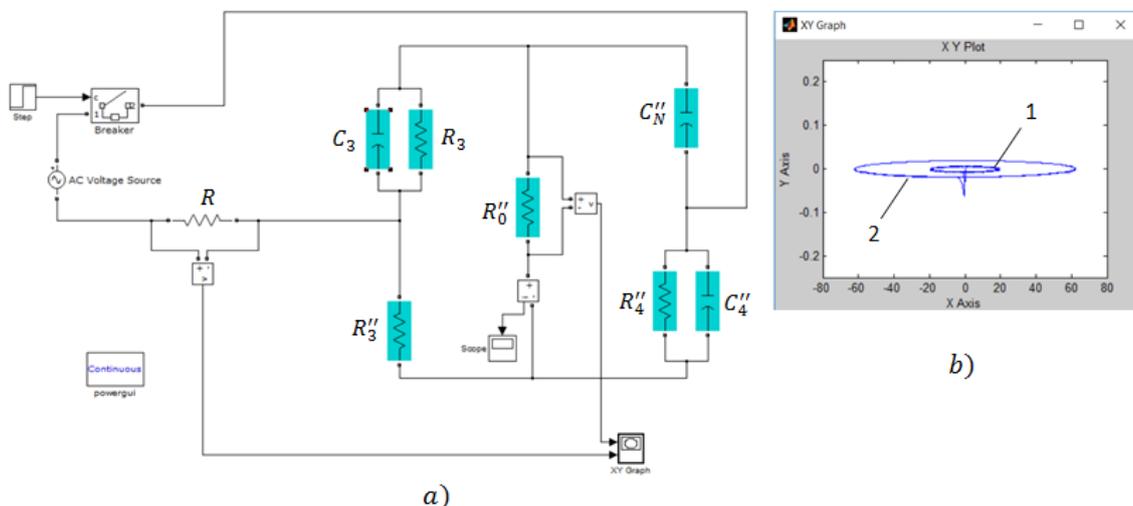


Рисунок 4 – а) - Компьютерная модель MC_3 ; б) – результаты моделирования, 1- соответствует такому значению C_3 , когда мост сбалансирован, 2 – несбалансированное состояние моста

Заключение

В заключение отметим, что предлагаемое устройство, кроме его возможности выполнять защитную функцию, как изложено выше, также может контролировать и степень старения масла по величине сигнала, поступающего от моста MC_3 в ИУС. Кроме того, ИУС может сигнализировать о приближении температуры масла к предельной границе допустимого превышения температуры, т.е. к значению 115 °С по ГОСТ [9].

Литература

1. СО 34.46.305—2005. Инструкция по применению системы мониторинга при изготовлении и ремонте силовых трансформаторов. – 2-е изд. — М.: ОАО «ЦКБ Энергоремонт», 2007.
2. Защитные устройства трансформаторов–газовое реле // <https://leg.co.ua/info/transformatory/zaschitnye-ustroystva-transformatorov-2.html>. (Дата обращения 17.09.2022).
3. Устройства для информационно-управляющей системы по мониторингу масляных силовых трансформаторов / Г. Н. Ниязова, К. Сатаркулов, Т. К. Кабаев, Р. А. Мырзаканова // Проблемы автоматики и управления. – 2020. – № 2(39). – С. 43-50. – EDN LSJJUN.
4. Калматов, У. А. Диагностирование силовых трансформаторов по характеру распределения температурного поля в верхних слоях масла / У. А. Калматов, К. Сатаркулов // Проблемы автоматики и управления. – 2019. – № 1(36). – С. 53-57. – DOI 10.5281/zenodo.3252996. – EDN XLPXLG.
5. Датчик пузырьков // https://translated.turbopages.org/proxy_u/en-ru.ru.5e408560-62fb7cea-06b3ed21-74722d776562/https/en.wikipedia.org/wiki/Bubble_sensor. (Дата обращения 17.09.2022).
6. Желудев И.С. Электрические кристаллы. – М.: Наука, 1969. – 213 с.
7. Сви П. М. Методы и средства диагностики оборудования высокого напряжения. – М.: Энергоатомиздат, 1992. – 239 с.

8. Каплянский А.Е., Лысенко А.П., Полотовский Л.С. Теоретические основы электротехники. – М.: Высшая школа, 1972. – 442 с.
9. Переходные тепловые процессы в силовых трансформаторах при стохастическом характере изменения их коэффициентов загрузки / Г. Ш. Эралиева, Т. Кабаев, Ч. Б. Бузурманкулова, Н. Суйунтбекова // Проблемы автоматизации и управления. – 2022. – № 1(43). – С. 4-12. – EDN EANCUA.

УДК 519.711.2: 621.311.212

К. Н. Оразбаев, Kazbek_ozarbaev@mail.ru
М.Т. Медетбеков, kambar@list.ru
М. Б. Джоошибеков, msocha@mail.ru
Институт машиноведения и автоматизации НАН КР

МОДЕРНИЗАЦИЯ РАБОЧЕГО КОЛЕСА БИРОТОРНОГО МИКРО-ГЭС (БМГЭС)

В данной работе рассмотрен бироторный генератор (БМГЭС), смонтированный на рабочее колесо, который приводится во вращательное движение на одном валу. Особенности этой конструкции заключаются в том, что на лопасть рабочего колеса сконструирована дополнительная водозаборная емкость – резервуар, который производит забор воды в емкости и создаёт дополнительный момент инерции для крутящего момента силы на рабочее колесо. Эта лопасть с водозаборным резервуаром намного эффективнее чем простая лопасть без дополнительных резервуаров. Исходя из этого, и используя теорию маятника Обербека, выведено рабочее уравнение и вычислен момент инерции лопастей рабочего колеса. Используя момент инерции, вычислен момент силы рабочего колеса. Были проведены анализы по проведенным расчетам, построены графики зависимостей величин от массы лопасти с забором воды и без забора воды на емкости лопасти.

Ключевые слова: бироторный генератор, лопасть, рабочее колеса, емкость-резервуар, момент инерции, момент силы.

Введение

Правительство Республики Кыргызстан предпринимает значительные усилия по освоению гидроэнергетических ресурсов страны, рассматривая это направление в качестве средства удовлетворения потребности страны в электроэнергии, а также увеличения объёмов её экспорта. В связи с этим рассмотрена необходимость и значение малой гидроэнергетики для удаленных населённых пунктов от центрального электроснабжения и горных труднодоступных местностях.

Практически все реки Кыргызстана имеют горное происхождение, что обуславливает значительную скорость их течения, которая имеет большой гидродинамический потенциал, бывают местности где реки низконапорные, т.е. гидродинамический потенциал реки слабый, и не выгодно строить гидроэлектростанции в этом месте. И поэтому одной из важных и актуальных научных задач при решении этих проблем является поиск и разработка новых, высокоэффективных, надежных и экономически привлекательных конструкций микро- и мини- ГЭС. В данной работе предлагается новый способ, гидротурбин бироторной микро-ГЭС (БМГЭС) модернизируя лопастную часть рабочего колеса используя методику [1,2,3, 4, 5], патентные источники [6,7,8] и этим получить повышенную выходную мощность, надежную в работе, и упрощенную конструкцию турбинного узла. Рабочее колесо гидротурбины БМГЭС представлено на фиг.1. фронтальном и продольном разрезе. На рабочее колесо монтирован бироторный генератор МГЭС который приводится во вращательное движение на одном валу. Особенности этой конструкции заключаются в том, что на лопасти рабочего колеса сконструирована дополнительная водозаборная емкость – резервуар который производит забор воды в емкости и создаёт дополнительный момент инерции для крутящего момента силы на рабочее колесо.

Камера гидротурбин состоит из рабочего колеса, насаженного на него БМГЭС на одном валу, корпус статора генератора служит как рабочее колесо гидротурбин, на корпус статора насажена втулка рабочего колеса гидротурбин. Бироторный генератор, ротор и статор вращаются противоположно относительно друг друга, из-за чего стала возможной сборка рабочего колеса гидротурбин с генератором энергии, как одно целое т.е. статор вместе вращается с рабочим колесом. Благодаря группе шестерни планетарной модели, шестерни сателлита и центральной, шестерни ротора приводятся во вращательное движение. С помощью шестерни сателлита, в которой установлен между шестерни

планетарной модели и центральной шестерни приводит во вращение статора и ротора в противоположном направлении. На фиг.1. приведен схематический рисунок в разрезе.

Обычно гидрокoleso с лопастями вращается без дополнительных емкостей на лопастях под напорным давлением.

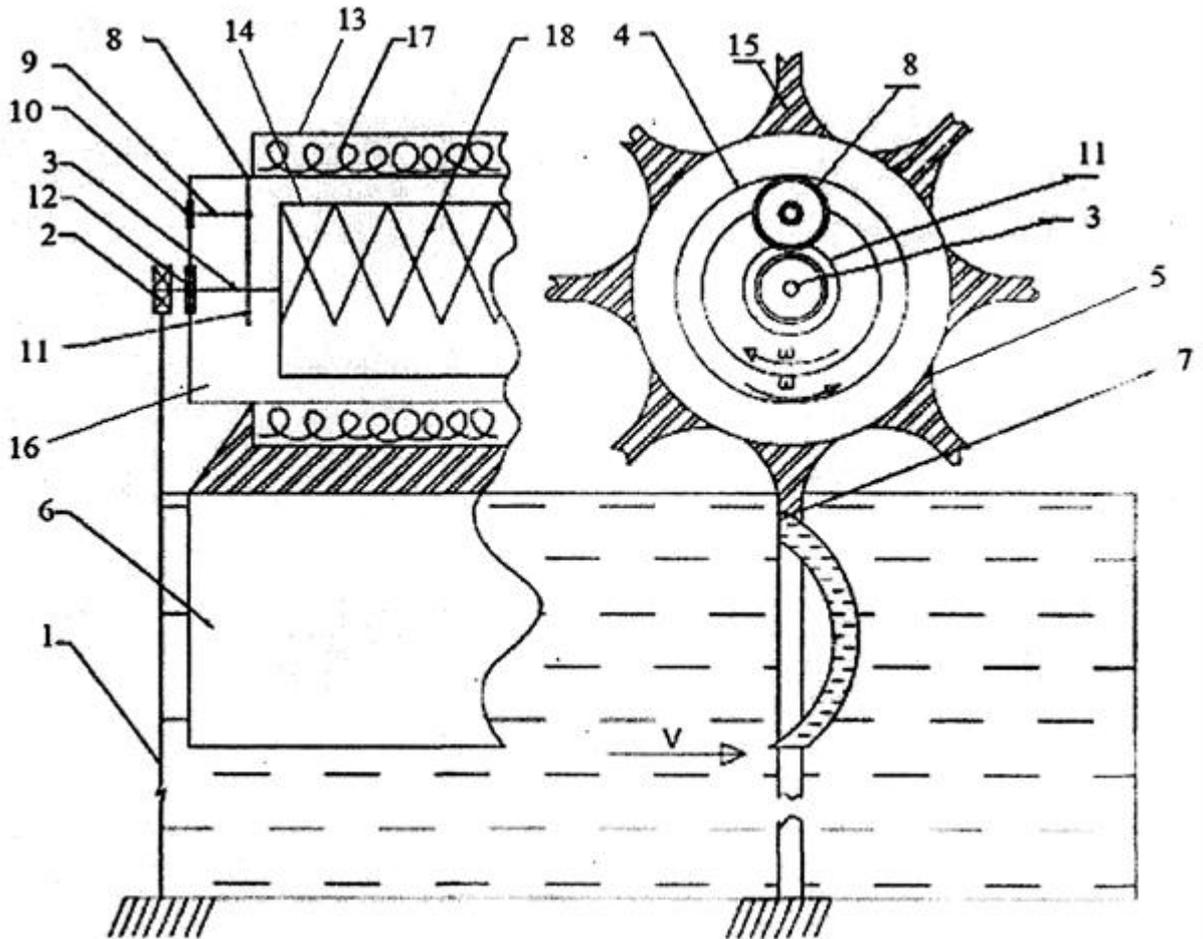


Рисунок 1 – Фронтальный и продольный разрез микрогидроэлектростанции.

Идея заключается в том, что с дополнительной емкостью, устроенной на лопастях для забора воды, в которой вода заполняет резервуар, создают своим весом тяжесть, тем самым создает дополнительную гравитационную силу, и при этом создают дополнительный момент инерции для крутящего момента силы рабочего колеса. Рабочее колесо гидротурбины это колесо с лопастями поэтому его надо рассматривать как не однородное физическое тело имеющее общую массу тела, закрепленные на оси вращения. Его надо рассматривать как лопасть отдельного стержня с грузиками (грузики — это резервуар) который напоминает маятник Обербека и поэтому воспользуемся теорией маятника Обербека [9,10,11]. Как известно момент инерции всего маятника относительно оси вращения равен сумме моментов инерции четырех стержней с цилиндрическими грузами на них, момент инерции втулки, в которой крепятся стержни, и момента инерции барабана:

$$J = J_{вт} + J_{б} + 4(J_{см} + J_{гр}) \quad (1)$$

где $J_{вт}$ - момент инерции втулки; $J_{б}$ - момент инерции барабана; $J_{см}$ - момент инерции стержня; $J_{гр}$ - момент инерции груза;

У втулки и барабана ось вращения проходит через их центр масс, это цилиндрические тела, поэтому:

$$J_{б} = \frac{1}{2} m_{б} r_{б}^2, \quad J_{вт} = \frac{1}{2} m_{вт} r_{вт}^2. \quad (2)$$

J_6 – момент инерции барабана и $J_{вт}$ – момент инерции втулки,

$$J_{gp} = J_{гр} = \frac{1}{2} m_{гр} r_{гр}^2. \text{ момент инерции груза.} \quad (3)$$

Момент инерции стержня длиной l_{cm} , относительно оси, проходящей через его конец и перпендикулярный ему

$$J_{cm} = \frac{1}{3} m_{cm} l_{cm}^2. \quad (4)$$

Таким образом, водозаборный резервуар на лопасти заполняясь водой создает дополнительную потенциальную энергию рабочего колеса, и вода со своей весовой тяжестью усиливает крутящий момент рабочего колеса турбины. Это даёт дополнительную кинетическую энергию рабочего колеса и в свою очередь легко приводит во вращательное движение бироторный генератор.

Чтобы решить данную задачу мы будем рассматривать резервуар на лопасти как материальную точку по сравнению с рабочим колесом массой m , только тогда можно решить эту задачу.

Вычисление моментов инерции рабочего колеса гидротурбин.

Для вывода рабочей формулы начертим схематический рисунок для ясного представления задачи рис.2.

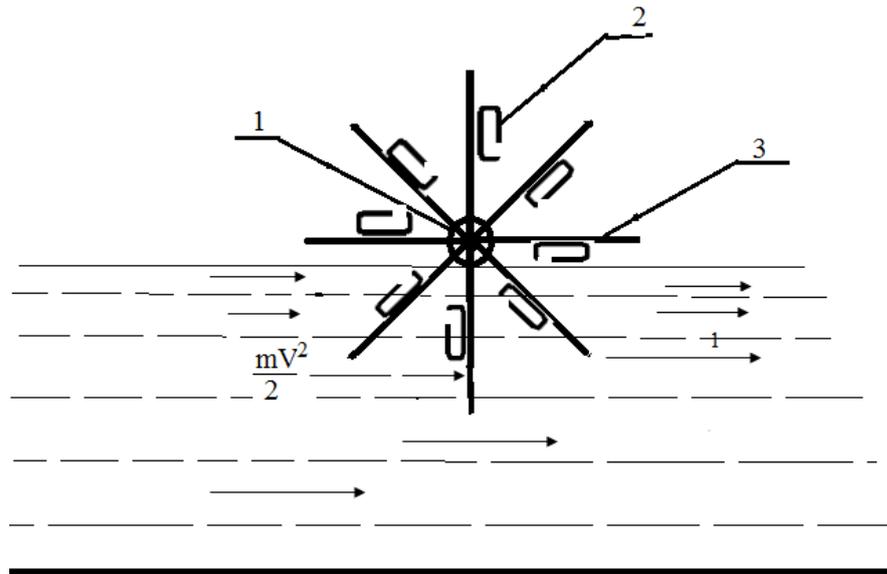


Рисунок 2 – Схематический рисунок гидроколеса микро-ГЭС с водозаборным резервуаром на лопасти в проточной воде. 1-ось, 2-резервуар для забора воды, 3- лопасть вместе с резервуаром.

Если рабочее колесо гидротурбины где лопасть с дополнительным водозаборным резервуаром загружается в воду только одной лопастью, а остальные нет, тогда исходя из этого определим момент инерции для одной загруженной лопасти. Тогда общий момент инерции состоит из суммы моментов инерции рабочего колеса и резервуара. Исходя из этого определения можем написать следующее выражение:

$$J = 4J_{\text{лоп.кол.}} + 8J_{\text{емкост.}}. \quad (5)$$

где $J_{\text{лоп.кол.}} = \frac{1}{12} \cdot (m_{\text{лоп.кол.}} \cdot r_{\text{лоп.кол.}}^2)$ момент инерции рабочего колеса,

$J_{\text{емкост.}} = J_{\text{емкост.}} + m_{\text{емкост.}} \cdot b^2$ момент инерции резервуара.

Согласно (1) можно записать общие уравнения для моментов инерции гидроколеса,

$$J = \frac{1}{12} \cdot (m_{\text{лоп.кол.}} \cdot r_{\text{лоп.кол.}}^2) + J_{\text{емкост.}} + (m_{\text{емкость}} + m_{\text{воду}}) \cdot b^2 \quad (6)$$

$$J = \frac{1}{12} \cdot 4(m_{\text{лоп.кол.}} \cdot r_{\text{лоп.кол.}}^2) + \frac{1}{2} \cdot 8(m_{\text{кол.}} \cdot r_{\text{кол.}}^2) + (m_{\text{емкость}} + m_{\text{воду}}) \cdot b^2 \quad (7)$$

$$J = \frac{1}{3} (m_{\text{лоп.кол.}} \cdot r_{\text{лоп.кол.}}^2) + 4[(m_{\text{кол.}} \cdot r_{\text{кол.}}^2) + (m_{\text{емкость}} + m_{\text{воду}}) \cdot b^2] \cdot \varepsilon \quad (8)$$

Что бы привести уравнение к общему виду, уравнения умножим на обе стороны равенства ε от этого ничего не изменится, наоборот видоизменится уравнение, и упростится;

$$J \cdot \varepsilon = \frac{1}{3} (m_{\text{лоп.кол.}} \cdot r_{\text{лоп.кол.}}^2) + 4[(m_{\text{кол.}} \cdot r_{\text{кол.}}^2) + (m_{\text{емкость}} + m_{\text{воду}}) \cdot b^2] \cdot \varepsilon \quad (9)$$

И так, мы знаем момент силы кручения рабочего колеса равно моменту инерции колеса, умноженного на угловое ускорение.

$$M = J \cdot \varepsilon,$$

где J -момент инерции, ε - угловая ускорения,

Вставляя свои значение получим следующий вид уравнения, которое включает момент силы лопасти колеса, и момент силы самого колеса, и водозаборную емкость - резервуара

$$M = \frac{1}{3} M_{\text{лоп.кол.}} + 4(M_{\text{кол.}} + M_{\text{емкость}}) \quad (10)$$

Таким образом, приходим к выводу, что момент силы для кручения рабочего колеса M равен сумме момента силы рабочего колеса плюс момента силы воды на резервуар и плюс момента силы свободной части от нагрузки лопасти, на которую действует кинетическая энергия протекающей воды.

Для рассмотрение количественных картин влияние массы воды для крутящего момента силы проверки проведем расчет по следующим значениям гидротурбин.

Основные данные гидроколеса:

- диаметр колеса $d = 0,5$ м
- масса колеса $m = 16$ кг
- масса лопасти $m = 2$ кг
- длина лопасти $l = 0,25$ м
- радиус колеса $r = 0,25$ м
- масса емкости $m = 1$ кг
- масса воды $m = 1$ кг
- расстояние от оси вращения до емкости $b = 0,20$ м

$$J = \frac{1}{3} (m_{\text{лоп.кол.}} \cdot r_{\text{лоп.кол.}}^2) = \frac{1}{3} (2 \cdot (0,25)^2) = 0,041 \text{ [кг} \cdot \text{м}^2] \text{ момент инерции лопасти.}$$

$$J = (m_{\text{кол.}} \cdot r_{\text{кол.}}^2) = 16 \cdot (0,25^2) = 16 \cdot 0,0625 = 1 \text{ [кг} \cdot \text{м}^2] \text{ момент инерции колеса.}$$

$$J = (m_{\text{емкость}} + m_{\text{воду}}) \cdot b^2 = (1+1) \cdot (0,20)^2 = 0,08 \text{ [кг} \cdot \text{м}^2] \text{ момент инерции емкости и воды.}$$

$$J = 0,041 + 4(1 + 0,08) = 4,121 \text{ с водой емкости. [кг} \cdot \text{м}^2]$$

$$J = 4,121 \text{ [кг} \cdot \text{м}^2]$$

$$\text{Момент инерции без набора воды } J = 0,041 + 4(1 + 0,04) = 4,081 \text{ [кг} \cdot \text{м}^2]$$

$$J = 4,081 \text{ [кг} \cdot \text{м}^2]$$

Разница момента инерции без набора воды и с набором равно

$$\Delta J = 4.121 - 4.081 = 0.04$$

С другой стороны, момент силы равен:

$$M = \varepsilon \cdot J$$

Известно, что угловая скорость при равномерном вращение записывается в виде

$$\omega = \omega_0 + \varepsilon \cdot t,$$

где $\omega_0 = 2\pi\nu_0$ – начальная угловая скорость.

Отсюда угловое ускорение равно $\varepsilon = \frac{\omega_0}{T} = \frac{2\pi\nu_0}{T}$. Подставляем уравнение момента силы и получаем рабочее уравнение:

$$M = \frac{2\pi\nu_0}{T} \cdot \frac{(m_{\text{КОЛ}} \cdot r_{\text{КОЛ}}^2)}{2}$$

С другой стороны, для крутящего момента силы колеса равны вычислениям мощности генератора:

$$M = \frac{9550 \cdot N}{n}, \quad \text{тогда}$$

Таким образом, нами показаны значение J, M, N которые зависят от массы и массовых характеристик турбин.

Если считать, что генератор выдаёт только ≈ 100 Вт. тока, то скорости тока реки примерно равно $V = 1 \text{ м/с}$. В этом случае надо увеличить уклон русла реки или напор воды. В нашей Республике реки текут в больших и малых количествах с большимим уклонами и с большим напором, такой БМГЭС можно поставить в любом месте.

Таблица 1. В таблице приведены данные зависящие от массы воды момент инерции, момент силы и мощность генератора электроэнергии.

m(масса воды)[кг]	J(момент инер.) [кг·м ²]	M(момент силы)[нм]	N(мощность генер.)10 ² кВт.
1,2	4,841	25,33	13,2
1,4	5,641	29,52	15,4
1,6	6,441	33,71	17,6
1,8	7,241	37,89	19,8
2,0	8,041	42,08	22,2

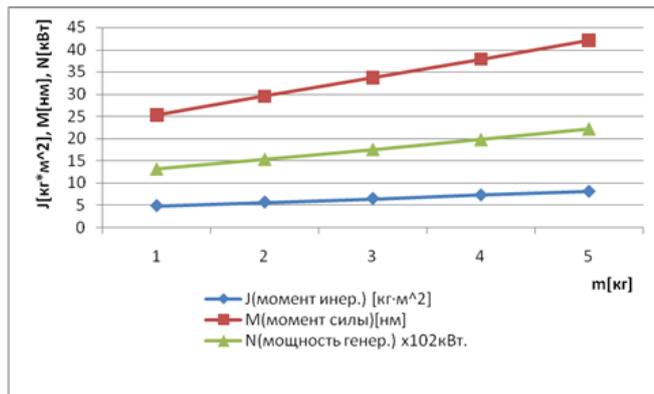


Рисунок 3 – Приведена зависимость момента инерции и мощности генератора электроэнергии от массы лопасти воды.

Из приведенных зависимостей можно сделать вывод, что по расчетным данным ростом массы воды на дополнительной водозаборной лопасти которая дает дополнительный импульс силы со своей весовой тяжестью во время вращения рабочего колеса можно построить такой микро –ГЭС для мало мощных рек. Зависимости показывают по восходящей линии т.е. повышается момент силы и мощность ГЭС.

Выводы

Бироторный генератор в свое время разработан для повышения мощности генератора за счет дополнительного механизма, вращаясь в противоположном направлении, у ротора и статора увеличивается скорость вращения т.е. и частота вращения.

В данной работе предлагается повышение мощности гидротурбин, для этого сконструирована дополнительно на лопасти водозаборная емкость для повышения крутящего момента гидротурбин. Это водозаборная емкость по идее должна создавать дополнительный момент инерции для гидроколеса и силу крутящего момента тем самым увеличить мощность гидротурбин. Для этого проводилось теоретическое исследование по методу Обербека.

1. Проведен вывод уравнения для расчета момента инерции по методу Обербека для пустой водозаборной емкости.
2. Далее вывод уравнения для расчета момента инерции по методу Обербека с набором воды на водозаборную емкость.
3. Вычислен момент силы рабочего колеса используя момент инерции.

Таким образом, можно сделать вывод, что по проделанной работе анализа по графику рис.3. показывают, что с увеличением массы воды на лопасти рабочего колеса силы крутящего момента возрастают за счет момента инерции гидроколеса и в свою очередь увеличивается мощность генератора тока бироторной микро-ГЭС (БМГЭС). На данную работу была подана заявка патент на изобретение «Кыргызпатенту» и получено решение о выдаче патента от 02.06.2022г.

Литература

1. Гуревич М.И. Теория струй идеальной жидкости. –М.: Наука, 1979.
2. Кривченко Г.И. Гидравлические турбины и насосы. –М.: Энергоатомиздат, 1983.
3. Дейч М.Е., Зарянкин А.Е. Гидрогазодинамика. –М.: Энергоатомиздат, 1984. – 384.
4. Юренков В.Н., Иванов В.М., Клейн Г.О., Блинов А.А., Родивилина Т.Ю., Иванова П.В. Методика расчёта обтекания лопасти водяного колеса. //Вестник АлтГТУ им. И.И.Ползунова, №2, 2006. –С.143-150.
5. Бакасова, А. Б. Расширенные варианты использования гидро - ветряной электроустановки и автоматическая стабилизация режимов ее работы / А. Б. Бакасова, М. С. Асанов, К. Сатаркулов // Проблемы автоматизации и управления. – 2021. – № 3(42). – С. 4-14. – EDN QCWZWE.
6. Патент KG №1506, С1, кл. F03B 13/00 (2012.01), Обозов А.Дж, Акпаралиев Р.А., Исаев Р.Э. и т.д.
7. Патент KG № 1748, С1, кКл. F03B 13/00 (2015.01), F03B 3/04 (2015.01), Обозов А.Дж, Акпаралиев Р.А., Исаев Р.Э. и т.д.
8. Патент KG № 1808, С1, кКл. F03B 3/02 (2015.01), F03B 13/00 (2015.01), Ураимов Р.Ж., Обозов А.Дж, Кенжаев И.Г., Жамалов А.Ж., Акпаралиев Р.А., Медеров Т.Т.
9. Савельев И.В. Курс общей физики. Т.1. Изд. «Лань». СПб. 2005
10. Яворский Б.М., Пинский А. А. Основы физики, т. I. М.: Наука, 1974, 496с.
11. Волькенштейн В.С. Сборник задач по общему курсу физики. Изд. «Книжный мир». СПб. 2003.

УДК 621.315

*Ёдгорова Дилбар Мустафаевна¹, yodgorova@uzsci.net**Хакимов Алим Адилевич¹, alimad54@gmail.com**Рахматов Ахмад Зайнидинович², rahmatov@oaofoton.uz**Закиров Руслан Гильмуллаевич³, zrg1980@mail.ru*¹*Физико-технический институт АН РУз*²*АО «ФОТОН», Республика Узбекистан, 100047, г.Ташкент, ул.Амира Темура.*³*Научно-исследовательский институт Физики полупроводников и микроэлектроники при Национальном университете Узбекистана*

ЭЛЕКТРОННЫЙ МЕДИЦИНСКИЙ ТЕРМОМЕТР С БЕСПРОВОДНОЙ ПЕРЕДАЧЕЙ ИНФОРМАЦИИ

В статье приводится вариант беспроводного медицинского термометра на основе инновационного термодатчика. Предложенный термометр отличается от существующих более простой схемой включения термодатчика, не требующей стабилизации тока, повышенной точностью измерений и малым энергопотреблением. В качестве беспроводной линии связи применена технология Bluetooth. Все элементы термометра размещены в герметичном, влагостойком, противоударном корпусе, что делает его использование безопасным. Экономическая эффективность достигается за счет использования распространенной, недорогой элементной базы.

Ключевые слова: термодатчик, беспроводная линия связи, дифференциальный усилитель, медицинский термометр, контроллер

Введение

В связи с пандемией COVID-19 многие медицинские учреждения мира столкнулись с нехваткой медицинского персонала, что привело к увеличению числа врачебных ошибок.

Вместе с тем развитие беспроводных технологий значительно увеличивает возможности предоставления качественного медицинского обслуживания большему числу пациентов. В последнее время задача беспроводной передачи данных о параметрах организма человека представляла значительную проблему для разработчиков медицинских датчиков. В связи с этим данная область исследований является на сегодняшний день очень востребованной [1].

Необходимость усовершенствования медицинских датчиков широко проявляется во всех областях медицины, где проводное соединение приборов представляет большое неудобство для врачей, пациентов и обслуживающего персонала. Также наблюдается тенденция к необходимости периодического измерения температуры тела у сотрудников организаций. При этом используемые в настоящее время бесконтактные системы измерения температуры, принцип действия которых основан на регистрации интенсивности инфракрасного излучения, имеют погрешность более 0,5°C [2]. Такая погрешность может быть оправдана и пригодна только для приблизительного измерения температуры у большого количества людей, и проблема быстрого получения информации о температуре тела каждого сотрудника с высокой точностью остается нерешенной. Решение данной проблемы возможно путем применения системы, позволяющей производить считывание информации с датчиков температуры, зафиксированных непосредственно на теле человека.

Для решения этих задач применение систем беспроводной связи является очевидным преимуществом. Оно позволяет ничем не ограничивать мобильность человека относительно сервера, на котором расположена база данных. На основе вышеизложенного можно заключить, что исследования по созданию высокоскоростных систем измерения температуры тела с возможностью беспроводной передачи информации являются на сегодняшний день актуальными и востребованными [3].

Постановка задачи

В 2013 году под эгидой ООН в Японии была принята Минаматская конвенция по ртути, которая с 2020 года запрещает производство, экспорт и импорт нескольких видов продукции, содержащей ртуть, в том числе ртутных градусников. В связи с этим сократилось предложение ртутных термометров аптечной сети [4]. В качестве аналога предлагаются различные типы электронных термометров. Однако, по мнению потребителей, из стран СНГ данный тип термометров имеет большую погрешность измерений, что объясняется следующим фактором. Электронные термометры начали массово выпускать во второй половине прошлого века в странах Западной Европы и в Америке. Особенностью применения термометров в этих регионах является измерение температуры оральным способом, т.е. через рот, а не подмышкой, как принято во многих странах СНГ. Во рту другие показатели нормальной температуры: не 36,6°C, а 37,0°C, и другая скорость нагрева датчика термометра. Температура тела, измеренная в подмышечной впадине, как правило, на 0,3 – 0,5 градуса ниже температуры, измеренной в полости рта, и на 0,5 – 0,9 градуса ниже ректальной [5]. В связи с этим температура, измеренная подмышкой электронным термометром, рассчитанным на измерение температуры во рту, имеет высокую погрешность измерения. Это объясняется различием в алгоритмах обработки измерительной информации различными типами электронных термометров. Однако при оральном способе измерения температуры возникает необходимость в значительно более тщательной, по сравнению с измерением температуры подмышкой, дезинфекции термометра перед применением, что приводит к дополнительным затратам времени [6].

Таким образом причиной высокой погрешности электронных термометров является их использование не по назначению, а не конструктивный недостаток. Также преимуществом применения электронных термометров является возможность передачи информации по беспроводным линиям данных без ограничения свободы передвижения пациента.

При анализе рынка СНГ на предмет предложения электронных термометров с возможностью измерения температуры подмышкой и передачи информации по беспроводной линии связи было выявлено, что единственным прибором, удовлетворяющим данным требованиям, является RELSIB WT50. Основным недостатком этого термометра является его высокая цена – порядка 25 долларов США за один прибор [7].

В связи с вышеизложенной актуальной проблемой на сегодняшний день является разработка электронных термометров с возможностью передачи данных по беспроводной линии связи, с низкой стоимостью производства и эксплуатации.

В статье приводятся сведения по разработке электронного термометра и на базе термодатчика. Новизна предложенной разработки заключается в применении инновационного термодатчика, подробно описанного в [8]. Благодаря этому предложенный в статье медицинский термометр отличается от существующих повышенной точностью и пониженным энергопотреблением.

Выбор технологии беспроводной связи

В настоящее время в связи с интенсивным развитием Интернета вещей большое распространение получили различные стандарты ближней беспроводной связи, такие как Zigbee, RFID, NFC, Bluetooth и др. [9].

При анализе применимости технологий беспроводной связи были учтены подходящие характеристики в области энергопотребления, обеспечения безопасности передаваемых данных, дальности действия. В результате проведенного анализа для реализации беспроводной линии связи в предлагаемом термометре была выбрана технология беспроводной передачи стандарта Bluetooth Low Energy (BLE), соответствующая стандарту IEEE 802.15.4.

Изначально стандарт BLE был разработан фирмой Nokia. BLE имеет высокую безопасность, надёжность, низкую задержку при подключении и низкую потребляемую мощность. Еще одной важной особенностью стандарта является адаптивность перестройки частоты, то есть происходит коррекция ошибок [10] при передаче сигнала, BLE быстро изменяет свою рабочую частоту, выбирая наиболее оптимальную для устранения помех, проблем переполнения и для снижения интерференции (наложения волн).

В связи с изложенными выше преимуществами стандарта передачи данных BLE, в предлагаемом варианте медицинского термометра этот стандарт применен для передачи цифровых данных на приемную станцию. При этом первичная обработка измерений может осуществляться непосредственно в датчике.

Разработка структурной схемы термометра

В разрабатываемом термометре предлагается использовать инновационный термодатчик [11], представляющий собой аналоговое устройство, выходным сигналом которого является напряжение, линейно зависящее от температуры тела. Зависимость приведена на рис. 1 для напряжения питания 3В.

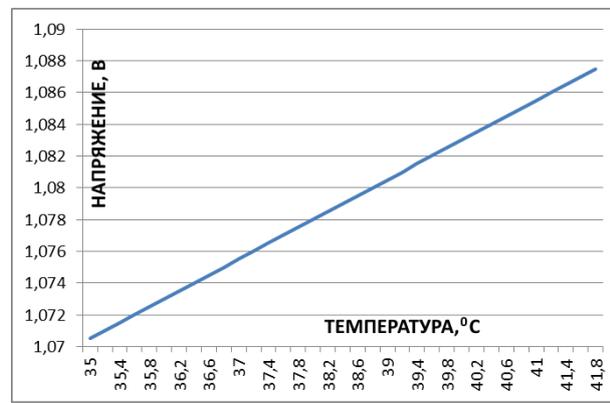


Рисунок 1 – Зависимость выходного сигнала термодатчика от температуры тела

Как видно из графика, зависимость линейная, в диапазоне 35–42°C напряжение выходного сигнала варьируется от 1,070 до 1,088В. Таким образом при изменении температуры $\Delta T = 7^\circ\text{C}$ изменение напряжения $\Delta U = 0,018\text{В}$. Чувствительность датчика при этом составляет:

$$S = \frac{\Delta U}{\Delta T} = \frac{0,018 \text{ В}}{7 \text{ }^\circ\text{C}} = \frac{18 \text{ мВ}}{7 \text{ }^\circ\text{C}} \approx 2,6 \text{ мВ}/^\circ\text{C}.$$

С учетом линейной зависимости изменения напряжения от температуры изменение температуры на $0,1^\circ\text{C}$ вызовет изменение напряжения сигнала на $0,26\text{мВ}$. С учетом того, что чувствительность АЦП с наиболее высокой разрешающей способностью составляет $5...10\text{мВ}$, требуется предварительное усиление сигнала термодатчика минимум в 20 раз.

На основе вышеизложенного предлагается структурная схема медицинского термометра для передачи информации по беспроводной линии связи. При разработке структурной схемы учтено требование минимизации количества компонентов с целью повышения надежности и уменьшения цены термометра. Для этого структурная схема содержит компоненты, реализуемые на одном кристалле Bluetooth приемопередатчика.

Структурная схема включает в себя следующие элементы: источник питания постоянного напряжения 3В, контроллер BLE, термодатчик.

Контроллер BLE имеет встроенный аналого-цифровой преобразователь и Bluetooth-модуль, преобразующий цифровой сигнал в соответствующий формат передачи данных.

Предлагаемая схема термометра представлена на рис.2.

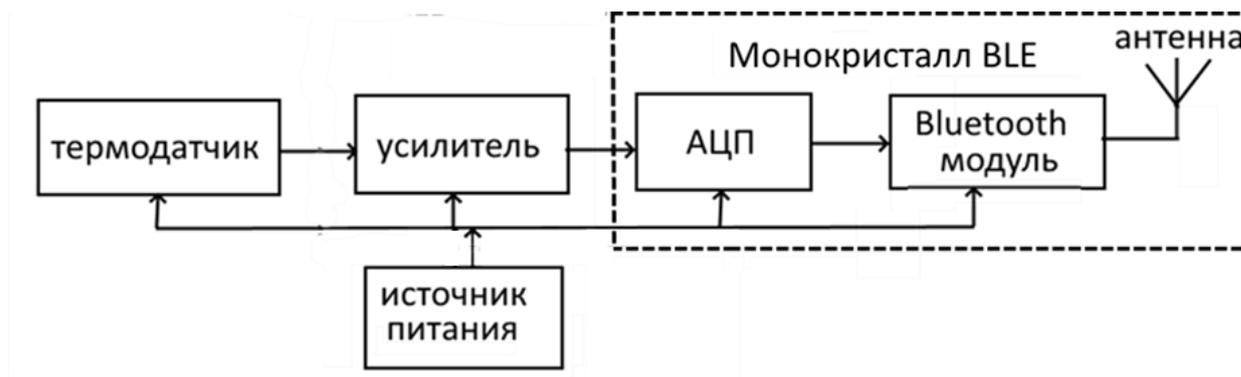


Рисунок 2 – Структурная схема беспроводного термометра

Усилитель, исходя из возможностей существующей элементной базы BLE-контроллера, целесообразно выполнить в виде отдельного модуля.

Выбор элементной базы

Для реализации термометра необходимы следующие элементы: термодатчик, дифференциальный усилитель, BLE-контроллер, элементы коммутации и индикации. В настоящее время на рынке представлен широкий ассортимент аналоговых термодатчиков. Характеристики, наиболее распространенных из них приведены в таблице 1 [12].

Таблица 1– Примеры аналоговых датчиков температуры

Модель	Диапазон измерений, °C	Точность, °C	Температурный коэффициент, мВ/°C	Производитель
LM35	от -55 до +150	±2	10	National Semiconductor
LM135	от -50 до +125	±1.5	10	National Semiconductor
LM335	от -40 до +100	±2	10	National Semiconductor
C1047	от -40 до +125	±2	10	Microchip
TMP37	от -40 до +125	±2	20	Analog Devices

Как следует из таблицы, общим недостатком известных полупроводниковых датчиков является их низкая точность измерения температуры. В связи с этим для реализации предложенного термометра используется термодатчик, разработанный в Физико-техническом институте Академии наук Республики Узбекистан [8].

В качестве усилителя сигнала термодатчика используется дифференциальный усилитель, схема включения которого приведена на рис. 3

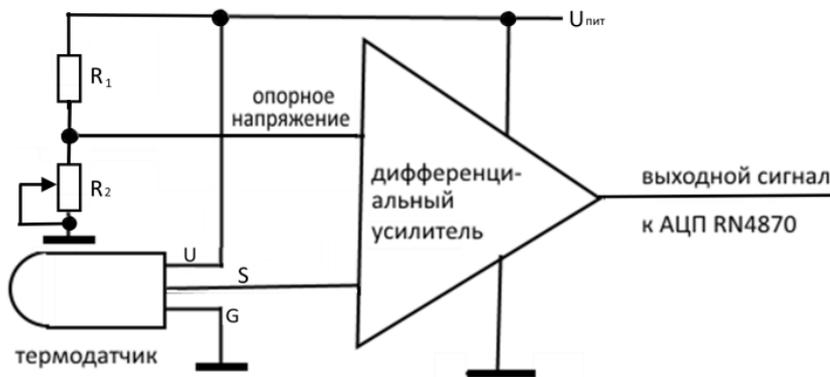


Рисунок 3 – Схема подключения термодатчика к дифференциальному усилителю

При выборе модуля дифференциального усилителя необходимо учитывать его стоимость и доступность, так как особенностью разработанного термодатчика является устойчивость к искажениям сигнала, вносимого усилителем. В связи с этим в качестве наиболее оптимального усилителя сигнала выбрана микросхема AD8132 Analog Devices

[13]. Опорное напряжение для дифференциального усилителя формируется путем деления напряжения питания при помощи делителя, образованного резисторами R1 и R2 [14].

При выборе BLE-контроллера в целях минимизации затрат на электронные компоненты при разработке термометра было учтено требование наличия в нем встроенного аналого-цифрового преобразователя. По этой причине был выбран модуль RN4870. Данный модуль объединяет в одном решении контроллер основной полосы частот Bluetooth, встроенный стек Bluetooth, цифровые и аналоговые входы / выходы, ВЧ-усилитель мощности и антенну. RN4870 имеет низкое энергопотребление-3В. RN4870 предоставляет интерфейс управления на основе команд ASCII, отправляемых по универсальной последовательной шине. Интерактивные команды ASCII позволяют настраивать RN4870 без сложных инструментов настройки. RN4870 поддерживает роли как периферийных, так и центральных общих профилей доступа, активно сканируя другие подключаемые устройства вместо ожидания входящих запросов на подключение [16].

Разработка принципиальной схемы термометра

На рис.3 приведена электрическая принципиальная схема термометра с возможностью передачи данных по беспроводной линии связи Bluetooth. Схема состоит из следующих элементов: термодатчика, дифференциального усилителя AD8132, модуля RN4870 Bluetooth Low Energy, источника постоянного напряжения 3,3В.

Схема может работать в двух режимах: эксплуатационном и режиме программирования. Переключение между режимами осуществляется с помощью переключателя «ПРОГ. /РАБОТА». В режиме программирования на плату подается напряжение 5В в режиме работы-3,3В. Для программирования Bluetooth-контроллера RN4870 применяется TTL адаптер CH340G [15], также показанный на схеме рис.4.

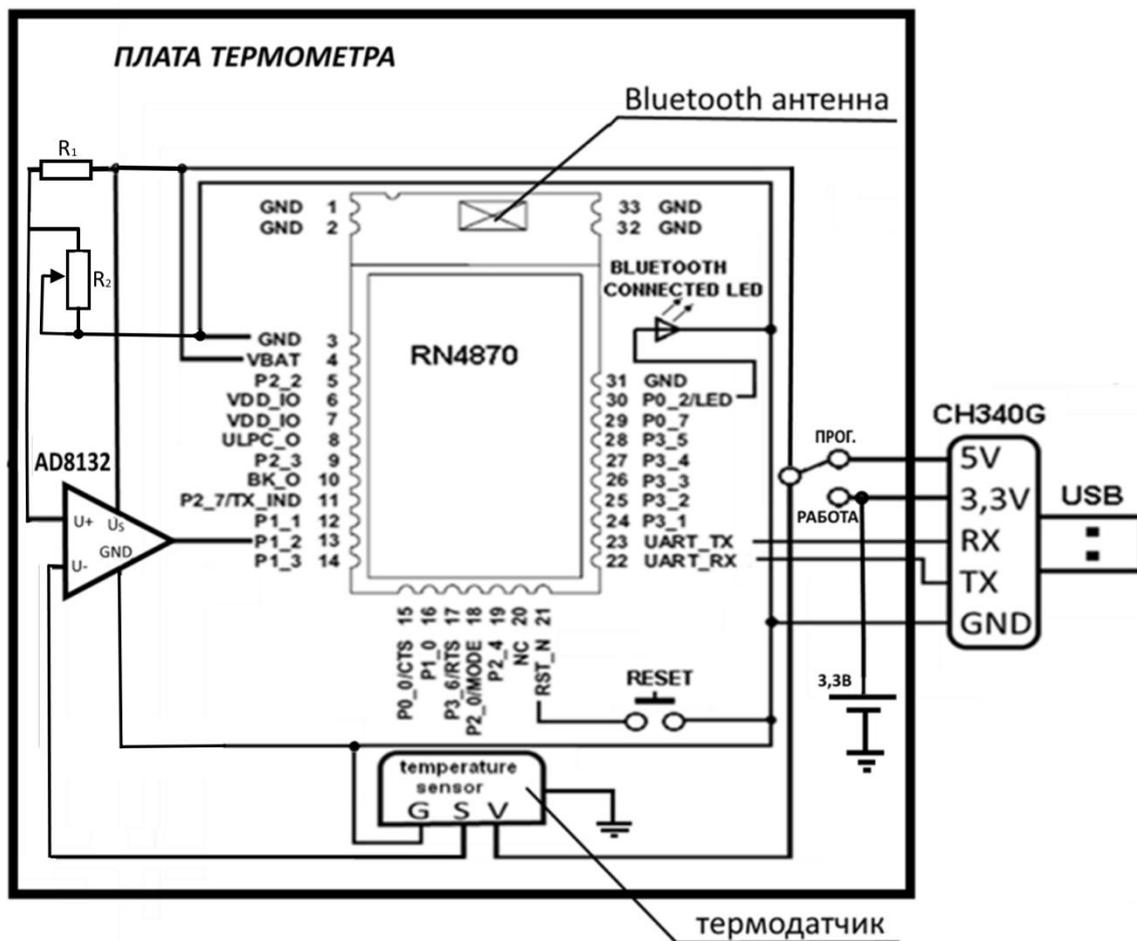


Рисунок 4 – Схема принципиальная электрическая термометра

В режиме передачи данных модулем RN4870 происходит переменное свечение светодиода, обозначенного на схеме Bluetooth connected LED.

Кнопка RESET предназначена для перезапуска термометра в случае сбоев его работы.

Как видно из схемы, отличием включения термодатчика от традиционных схем является отсутствие регулировки силы тока, протекающего через термодатчик. Это позволяет повысить надежность схемы, а также точность измерения температуры.

Как было рассчитано ранее, изменение температуры на 1°C вызывает изменение напряжения на термодатчике на 2,6мВ. Так как термометр является медицинским, к нему предъявляется требование иметь разрешающую способность не ниже $0,1^{\circ}\text{C}$. Такое изменение температуры вызывает изменение напряжения на 0,26мВ. Для дальнейшей обработки такой сигнал должен быть усилен минимум до 10 мВ.

Усиление осуществляется с помощью дифференциального усилителя AD8132, в качестве опорного напряжения в котором используется напряжение питания 3,3В.

Усиленный сигнал подается на аналоговый вход Bluetooth-контроллера RN4870, где преобразуется с помощью встроенного 10-разрядного аналого-цифрового преобразователя в сигнал Bluetooth. Далее преобразованный сигнал излучается в пространство с помощью, встроенной в модуль RN4870 антенны.

Для конфигурирования модуля RN4870 в него должна быть записана следующая программа.

Выбор конструкции термометра

Конструкция предлагаемого беспроводного термометра должна отвечать требованиям надежного контакта датчика температуры с телом человека. На этом основании была в качестве конструкции выбрана структура, показанная на рис.5.

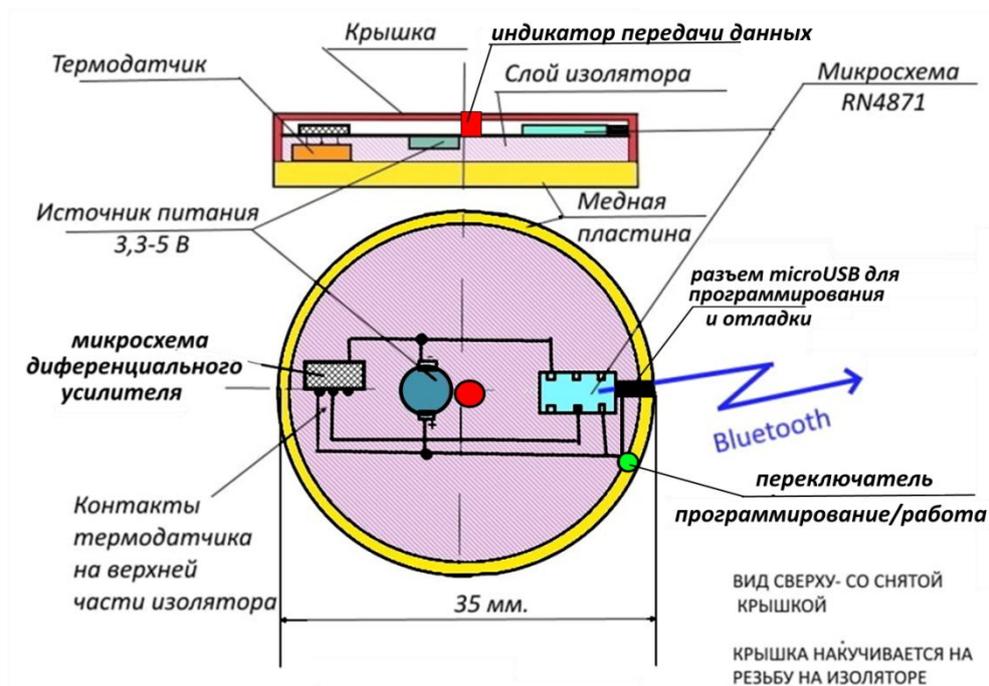


Рисунок 5 – Предлагаемая конструкция термометра

Конструкция термометра, показанная на рис.4, включает металлический (медный) диск, защищенный пластиковым корпусом, а также основные элементы термометра: термодатчик, элемент питания, микросхему дифференциального усилителя AD8132, микросхему контроллера Bluetooth RN4870, переключатель «программирование/работа», индикатор режима передачи данных по беспроводной линии связи. Медный диск должен плотно прилегать к полупроводниковому датчику температуры. Это достигается при

помощи теплопроводящего клея. Основным преимуществом предлагаемой конструкции беспроводного термометра является возможность интеграции всех остальных частей беспроводного датчика в конструкцию (внутри пластикового корпуса).

С целью уменьшения потерь сигнала датчика температуры микросхема дифференциального усилителя размещается в непосредственной близости от него. Переключатель «программирование/работа» располагается на торце корпуса термометра и не имеет выступающих за пределы корпуса частей.

Постановка эксперимента и испытания термометра

Целью проведения испытаний разработанного беспроводного термометра является проверка точности передачи температуры, замеренной аналоговым термодатчиком, на смартфон при помощи беспроводной линии связи Bluetooth. Схема экспериментальной установки приведена на рис.6. Как видно из схемы, в лабораторном макете использовались следующие компоненты:

Разработанный в рамках проекта термометр, имеющий разъем microUSB для сопряжения с персональным компьютером с целью программирования входящего в состав термометра контроллера RN4870 и калибровки. Переключатель «программирование/работа» при этом должен быть установлен в режим «программирование».

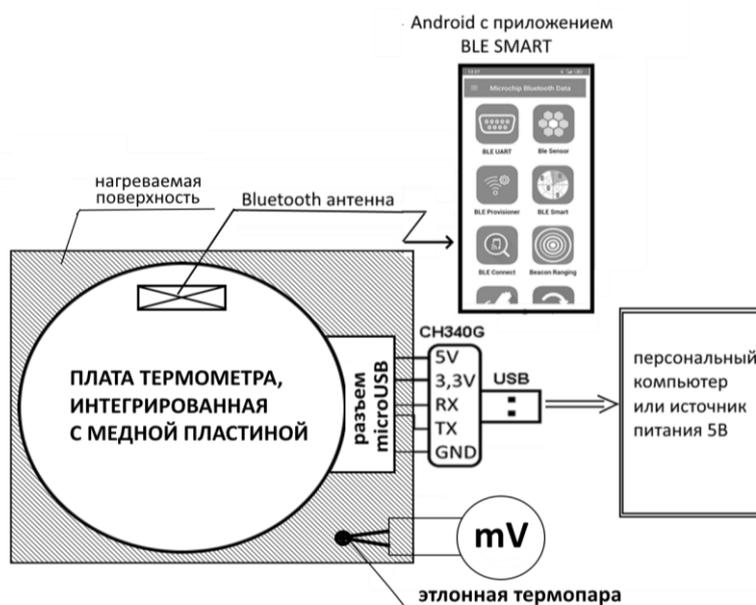


Рисунок 6 – Схема экспериментальной установки

Универсальный асинхронный приемопередатчик (UART) адаптер CH340G для сопряжения термометра с USB входом персонального компьютера.

Эталонная термопара медь-константан, поверенная в соответствии с требованиями ГОСТ 8.558-009-ГСИ «Государственная поверочная схема для средств измерений температуры». Термопара подключалась к вольтметру для измерения генерируемого напряжения и его пересчета в температуру.

Смартфон с операционной системой Android и установленным приложением BLE SMART компании MICROCHIP.

Аналоговый сигнал от термодатчика поступает в 16-битный встроенный АЦП контроллера RN4870. Чувствительность АЦП-0,001В (для изменения цифрового сигнала на 1 бит напряжение должно измениться на 0,01В). Данные измерений в диапазоне температур 34,2–38,2 градуса приведены в таблице 2.

Таблица 2– Результаты измерений температуры по линии связи Bluetooth

Данные контроллера	Напряжение термодатчика (В)	Температура по данным термопары (°C)
1A5	0,421	34,2
1A6	0,422	34,4
1A7	0,423	34,6
1A8	0,424	34,8
1A9	0,425	35,0
1AA	0,426	35,2
1AB	0,427	35,4
1AC	0,428	35,6
1AD	0,429	35,8
1AE	0,43	36,0
1AF	0,431	36,2
1B0	0,432	36,4
1B1	0,433	36,6
1B2	0,434	36,8
1B3	0,435	37,0
1B4	0,436	37,2
1B5	0,437	37,4
1B6	0,438	37,6
1B7	0,439	37,8
1B8	0,44	38,0
1B9	0,441	38,2

Данные контроллера – это запись в 16-ричной системе значения напряжения в милливольтках. Например, 1A5 переводим в десятичную систему, получаем 421 милливольт (0,421В). Все данные можно просмотреть как на экране персонального компьютера, так и на смартфоне.

В результате проведенного эксперимента была определена погрешность термодатчика, составившая 0,1°C. Такая погрешность является удовлетворительной для передачи сигнала температуры на оконечное устройство. Однако, учитывая разрешающую способность аналого-цифрового преобразователя Bluetooth модуля RN4870, необходимо уменьшить погрешность до 0,7°C – 0,8°C.

Выводы

Предложенный в статье термометр имеет ряд преимуществ по сравнению с другими моделями благодаря применению в нем инновационного термодатчика. Основными преимуществами термометра являются:

Безопасность. Так как термометр изготовлен из безопасного пластика, он не выйдет из строя даже при ударных нагрузках.

Точность. В предложенном электронном термометре погрешность составляет не больше 0,1 °C.

Влагостойкость. Предложенный термометр можно протирать влажными салфетками и дезинфицировать с помощью специальных растворов, что особенно актуально для медицинских учреждений.

Литература

1. Ющенко, М. А. Беспроводная передача информации в медицинском оборудовании / М. А. Ющенко. — Текст: непосредственный // Современные тенденции технических наук: Материалы III Междунар. науч. конф. (г. Казань, октябрь 2014 г.). — Т. 10. — Казань: Бук, 2014. — С. 29–32. — URL: <https://moluch.ru/conf/tech/archive/123/5260/> (дата обращения: 10.08.2022).
2. Бесконтактный умный инфракрасный термометр на Arduino и смартфоне. – [Электронный ресурс]. – Режим доступа: <https://microkontroller.ru/arduino->

- projects/beskontaktnyjumnyj-infrakrasnyj-termometr-na-arduino-i-smartfone/ (дата обращения 04.08.2022)
3. Bülent Bilgehan, Lara Kayed, Özlem Sabuncu, General probability distribution model for wireless body sensors in the medical monitoring system, Biomedical Signal Processing and Control, Volume 77, 2022
 4. Technical Background Report for the Global Mercury Assessment 2013-электронный ресурс <https://www.mercuryconvention.org/sites/default/files/2021-06/АмарУнерТехБакРепGMA2013.pdf.pdf> (дата обращения: 14.08.2022)
 5. Guodan Liu, Shuwei Liang, Songtao Hu, Calculation method of mean skin temperature weighted by temperature sensitivity of various parts of human body, Journal of Thermal Biology, Volume 100, 2021
 6. Правильная дезинфекция градусников: хранение и меры предосторожности Источник: <https://dezinfekc.ru/primenenie/dezinfekciya-gradusnikov> (дата обращения: 09.08.2022)
 7. Электронный ресурс <https://relsib.com/product/termometr-meditsinskij-tsifrovoy-elektronnyj-relsib-wt50> (дата обращения: 19.08.2022)
 8. Karimov A.V., Dzhuraev D.R., Kuliev Sh.M., Turaev A.A. Distinctive features of the temperature sensitivity of a transistor structure in a bipolar mode of measurement // Journal of Engineering Physics and Thermophysics, 2016. – Vol. 89, No. 2. – PP. 514-517.
 9. Никифоров Олег Юрьевич Базовые технологии Интернета вещей // Символ науки. 2015. №9-1. URL: <https://cyberleninka.ru/article/n/bazovye-tehnologii-interneta-veschey> (дата обращения: 23.08.2022).
 10. Амин Раад Джихад Алалван, Енин В.Д. Применение Bluetooth-технологии в интернете вещей // Международный журнал гуманитарных и естественных наук. 2018. №5-1. URL: <https://cyberleninka.ru/article/n/primenenie-bluetooth-tehnologii-v-internete-veschey> (дата обращения: 23.08.2022).
 11. Karimov A.V., Yodgorova D.M., Abdulhaev O.A., Kamanov B.M., Turaev A.A. Features of the temperature properties of a field-effect transistor in a current-limiting mode // Journal of Engineering Physics and Thermophysics, 2013. – Vol. 86, No. 1. – PP. 248-254.
 12. Электронный ресурс <https://mcucpu.ru/index.php/pdevices/datchiki/108-poluprovodnikovye> (дата обращения: 15.08.2022)
 13. Электронный ресурс <https://www.analog.com/media/en/technical-documentation/data-sheets/AD8132.pdf> (дата обращения: 13.08.2022)
 14. Артюшенко В. М., Пожидаев Г. И. Схемы подключения измерительного оборудования в электротехнических комплексах и системах // Электротехнические и информационные комплексы и системы. 2006. №1. URL: <https://cyberleninka.ru/article/n/shemy-podklyucheniya-izmeritelnogo-oborudovaniya-v-elektrotehnicheskikh-kompleksah-i-sistemah> (дата обращения: 23.08.2022).
 15. Электронный ресурс <http://ww1.microchip.com/downloads/en/DeviceDoc> (дата обращения: 23.08.2022)
 16. CH340G Datasheet -<https://www.alldatasheet.com/datasheet-pdf/pdf/1132618/ETC2/CH340G.html> (дата обращения: 18.08.2022)

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В СИСТЕМАХ МОНИТОРИНГА ПРИРОДНЫХ И ТЕХНОГЕННЫХ КАТАСТР

УДК 681.3

Гайдамако В.В., dolph@gmail.com,

Каныбеков Б.К., bektur1610@yandex.ru,

Лыченко Н.М., nlychenko@mail.ru,

Текеев Д.А., mr.tekeev14@gmail.com

Институт машиноведения и автоматизации НАН КР,

Кыргызско-Российский славянский университет, Бишкек, Кыргызстан

РАЗРАБОТКА WEB-ПОРТАЛА ЭКОЛОГИЧЕСКОЙ ИНФОРМАЦИИ КЫРГЫЗСКОЙ РЕСПУБЛИКИ

Представлены результаты проектирования и разработки Web-портала экологической информации Кыргызской Республики. Создание такого портала призвано привлечь внимание общественности к экологическим проблемам и вызовам, восполнить дефицит геоэкологической информации для исследователей, повысить заинтересованность в сборе экологических данных по Кыргызстану. Web-портал включает программный инструментарий для организации доступа к открытым источникам информации о концентрациях твердых частиц PM_{2.5}, PM₁₀ и индексе качества воздуха, сервисы для регистрации датчиков пользователей, сбора и сохранения измерительных данных, а также картографического отображения результатов мониторинга параметров окружающей среды.

Ключевые слова: сеть наблюдателей-добровольцев (crowdsourcing), концентрации твердых частиц PM_{2.5}, PM₁₀, интерфейс прикладного программирования (API, Application Programming Interface), Web-интерфейс, UML-диаграммы.

Введение

Одним из решений для обеспечения необходимого уровня безопасности человека при катастрофическом воздействии окружающей среды и минимизации экономического ущерба является разработка и внедрение комплексов аппаратно-программных средств мониторинга и оповещения о чрезвычайных геоэкологических ситуациях в Кыргызской Республике, в частности – Web-портала экологической информации.

К настоящему времени в КР нет ни одного сервиса для представления полной актуальной информации о состоянии окружающей среды в стране. В Интернете существует небольшое количество международных онлайн-сервисов, представляющих данные о метеопараметрах и загрязнении воздуха твердыми частицами, в том числе с территории КР, но при получении данных с этих сервисов возникают проблемы, связанные с ограничениями доступа и затрудненностью поиска необходимой информации.

Недостаток достоверной информации о текущем экологическом состоянии окружающей среды приводит к недостаточному вниманию к экологическим проблемам, связанным с загрязнением окружающей среды, и невозможности своевременного принятия мер по их решению.

Для решения указанных проблем необходима организация доступа к накопленным существующими в мире различными открытыми сервисами данным о состоянии окружающей среды [1, 2] через единое виртуальное окно – Web-портал. Кроме того, для анализа экологических ситуаций необходимо большее количество информации, что в принципе может быть достигнуто за счет формирования сети наблюдателей-добровольцев (crowdsourcing). В облачных информационно-измерительных системах предоставление датчиков в совместное использование стало новой парадигмой – измерения как сервис (Sensing-as-a-Service, SCaaS), датчики как сервис (SIaaS). Эта парадигма, суть которой в представлении данных датчиков и самих датчиков различным приложениям, поддержке

различных типов датчиков, обеспечении устойчивой масштабируемости и эластичности без изменения базовой инфраструктуры, использована при разработке Web-портала. Поставщик (владелец) датчиков представляет данные своих датчиков в общее пользование, а пользователь получает доступ к ним через запрос на конкретные данные [3].

Одной из самых больших экологических угроз в Кыргызской Республике, в частности в г. Бишкеке, стало загрязнение атмосферного воздуха. Почти не имея промышленности, наша столица занимает печальные первые места в мировых антирейтингах по загрязнению воздуха. В связи с этим разрабатываемый Web-портал экологической информации ориентирован прежде всего на представление информации о концентрациях твердых частиц PM_{2.5}, PM₁₀ и индексе качества воздуха AQI (World air quality index (AQI) [4]. Однако стоит отметить, что такая ориентация не ограничивает возможности портала в будущем представлять и другую геоэкологическую информацию в случае дополнения сервисов соответствующими датчиками и программными интерфейсами.

Таким образом, актуальность работы связана с необходимостью восполнения дефицита информации об экологической ситуации в КР, в частности, о загрязнении атмосферного воздуха твердыми частицами PM_{2.5} и PM₁₀, привлечения внимания общественности к экологическим проблемам в КР и повышения заинтересованности в сборе данных из сети наблюдателей-добровольцев. Доступ к этой информации позволит продолжить решать задачи прогноза экологического состояния КР, в частности – атмосферного воздуха г. Бишкека [5], с целью своевременного принятия решений по защите окружающей среды.

Основные задачи, реализуемые порталом:

- предоставить пользователю возможность для считывания оперативных и исторических данных с существующих в мире открытых источников экологической информации с целью формирования представления о возможных опасностях и рисках загрязнения окружающей среды в КР;
- предоставить возможность наблюдателям-добровольцам по всей территории КР включаться в совместную сеть наблюдателей для сбора, анализа и картографического отображения результатов мониторинга параметров окружающей среды.

Функционал системы

В соответствии с поставленными задачами разрабатываемый Web-портал, помимо традиционных информационных функций, включает в себя систему мониторинга параметров окружающей среды с датчиками, подключаемыми гражданами, и программно-сервисную часть для агрегации с открытыми источниками информации с последующей ее обработкой и представлением этой информации в понятном пользователю виде.

На рисунке 1 представлены основные группы функций системы в виде обобщенной диаграммы вариантов использования [6].

Выявлены 5 ролей (актеров) – активных сущностей, взаимодействующих с системой: *Гость* – незарегистрированный пользователь системы, имеет возможность просматривать всю доступную информацию на WEB-портале, в частности, данные с датчиков в виде форм на интерактивной карте и графиков; *Зарегистрированный пользователь* – имеет возможность не только просматривать информацию на WEB-портале, но и возможность регистрации нового датчика, подписки на один или несколько датчиков, возможность скачать данные за определенный период времени; *Администратор* – имеет возможность размещать новостную информацию на WEB-портале, подключать новые датчики к системе мониторинга, редактировать базы данных пользователей и датчиков, добавлять новые источники информации; *Физический датчик* – устройство, которое занимается сбором данных о состоянии окружающей среды и дальнейшей передачей их на сервер; *Внешние источники* – сервисы, представляющие

информацию из открытых источников данных о загрязнении атмосферного воздуха (сайты Luftdaten [1], AirNow [2]).

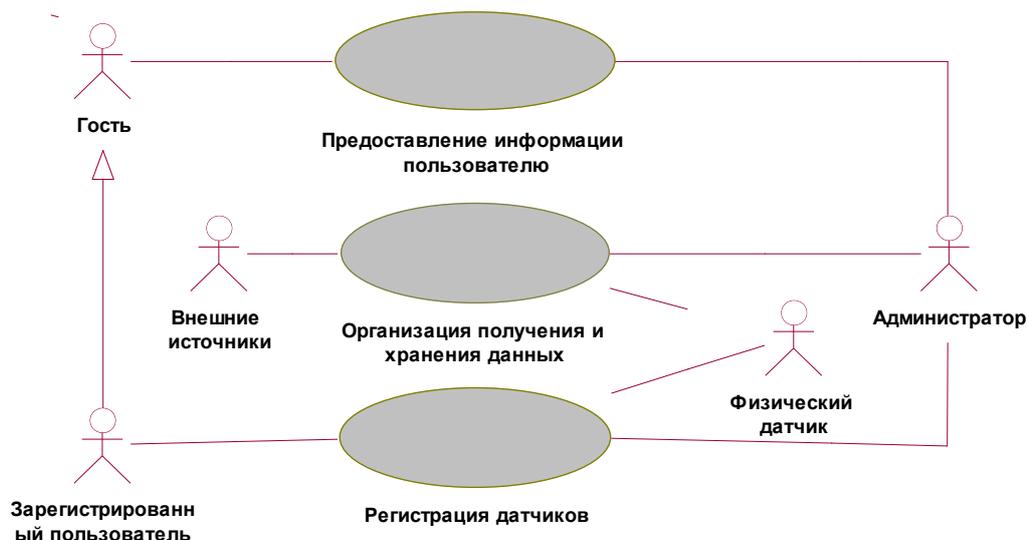


Рисунок 1 – Обобщенный функционал Web-портала

Группа функций «Представление информации пользователю» объединяет возможности Web-портала, связанные с представлением, добавлением и редактированием всей доступной на портале информации. Эта информация содержит: основные показатели датчиков и результаты их измерений в удобном для пользователя виде (картографически, в виде отдельных графиков), новостную ленту, контактную информацию, ссылки на открытые источники данных. То есть группа включает в себя функции для работы с информационной частью Web-портала (рисунок 2): просмотр информации на портале; редактирование информации на портале; настройка и отрисовка карты: масштабирование, получение данных, просмотр исторических данных, просмотр оперативных данных и параметров датчика, выбор периода наблюдений, регистрация пользователя, подписка на датчик, выбор датчика.

Гость портала может просмотреть всю доступную информацию, взаимодействовать с картой местности, где расположены датчики. При выборе определенного датчика *Гость* может увидеть представляемые датчиком данные – актуальные результаты измерений, например – концентрация в атмосферном воздухе твердых частиц PM_{2.5}, PM₁₀, индекса качества воздуха AQI, а также метаданные о самом датчике. *Гость* также имеет возможность просмотреть исторические данные, представленные в виде графиков временных рядов наблюдений. Для *Гостя* также доступна функция регистрации, после которой он становится *Зарегистрированным пользователем* с возможностью подписки на определенный датчик или группу датчиков. Для подписки на датчик *Пользователь* осуществляет поиск и выбирает физический датчик (выбор осуществляется по типу и другим параметрам физического датчика, а также по местоположению) и подписывается на него. Для *Пользователя* создается так называемый виртуальный датчик, соответствующий одному или нескольким физическим датчикам. *Пользователь* получает возможность использовать сенсорные данные с этого или этих физических датчиков [3].

Администратор имеет возможность добавлять, редактировать, удалять новостной поток, размещенный на портале.

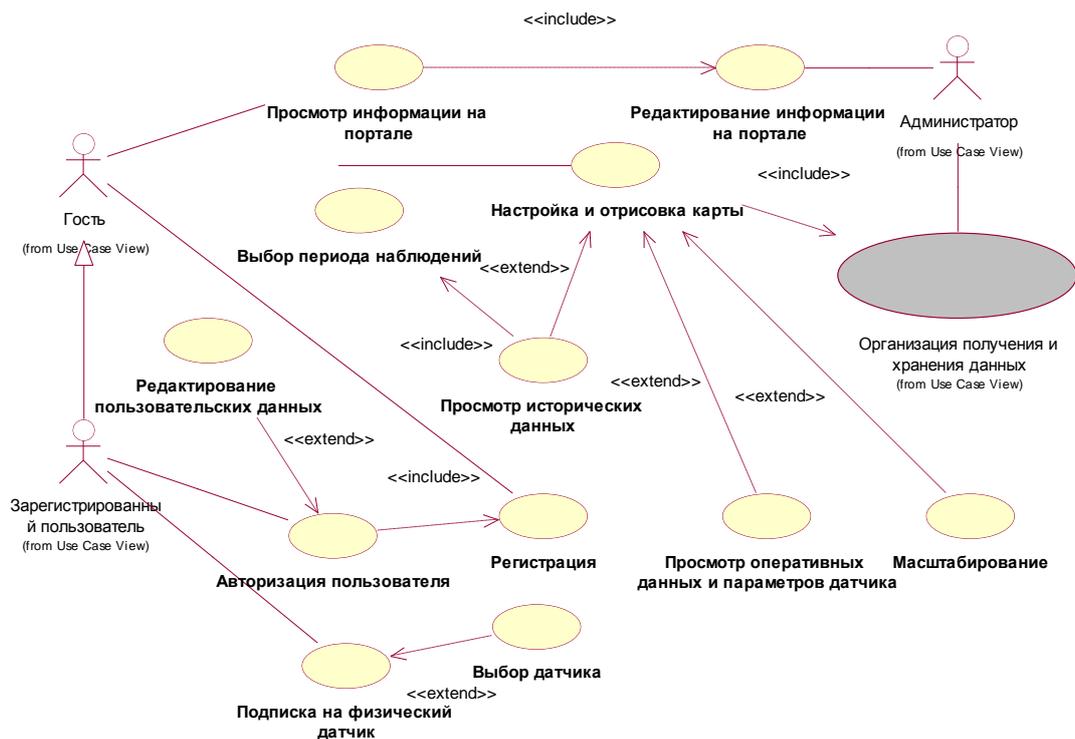


Рисунок 2 – Детализация группы функций «Представление информации пользователю»

Взаимодействие пользователя с системой при регистрации физического датчика описывает группа функций «Регистрация датчиков» (рисунок 3). Группа включает следующие варианты использования: «Регистрация датчика», «Редактирование метаданных датчика», «Просмотр базы данных (БД) датчиков и пользователей», а также действия с БД – «Редактирование», «Добавление», «Удаление».

Зарегистрированный пользователь может зарегистрировать свой датчик после описания датчика, включающего уникальный номер датчика, локацию датчика (широту, долготу, высоту, адрес, принадлежность датчика (частное лицо, компания, организация, другое). *Администратор* добавляет внесенные пользовательские данные о датчике в БД датчиков и пользователей.

Возможность получения результатов мониторинга загрязнения атмосферного воздуха твердыми частицами с внешних открытых источников (Luftdaten, AirNow, другие) и зарегистрированных в системе пользовательских датчиков (датчиков сети наблюдателей-добровольцев (crowdsourcing) реализуется через функции группы «Организация хранения и получения данных», детализация которой представлена на рисунке 4. На диаграмме показано, как происходит подключение к внешним источникам, получение данных о результатах мониторинга и дальнейшая их запись в БД, обработка клиентских запросов на получение результатов. Действующими акторами на данном уровне являются *Внешние источники* и *Администратор*.

Внешние источники представляют собой сущность, к которой обращается система при помощи параметризованных запросов для получения результатов мониторинга экологической ситуации. В нашем случае главным источником выступает Sensor. Community (Luftdaten) [1] – всемирная сеть сенсоров, информация с которой находится в открытом доступе, а также имеется в наличии документация для работы с предоставляемым API [11].

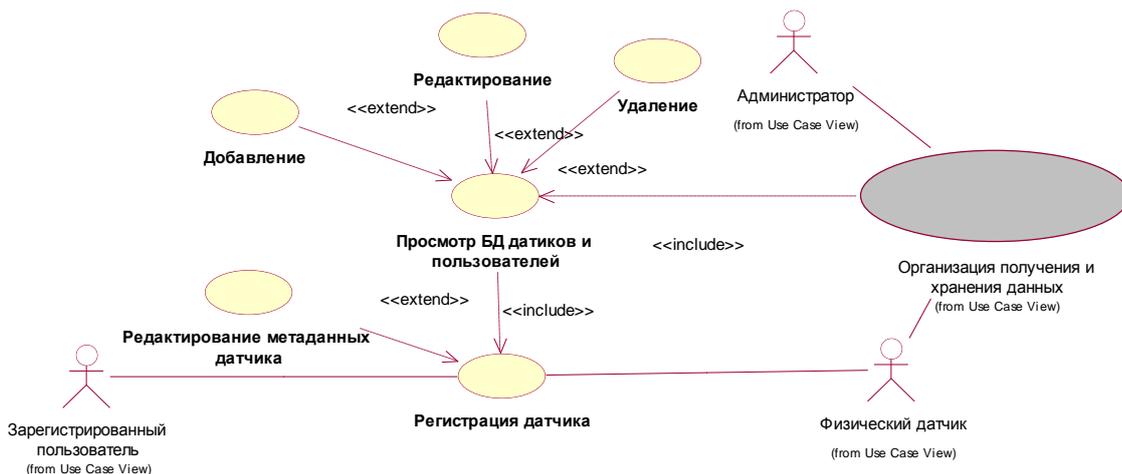


Рисунок 3 – Детализация группы функций «Регистрация датчиков»

В варианте использования «АПИ НАН КР» реализованы функции: получение данных с внешних источников, получение данных с физического датчика, получение пользовательских данных при регистрации, возврат ответа на клиентские запросы. Сохранение полученных данных о датчике, результатах мониторинга, пользовательских данных в БД для дальнейшего предоставления пользователям реализуется в варианте использования «Организация хранения данных».

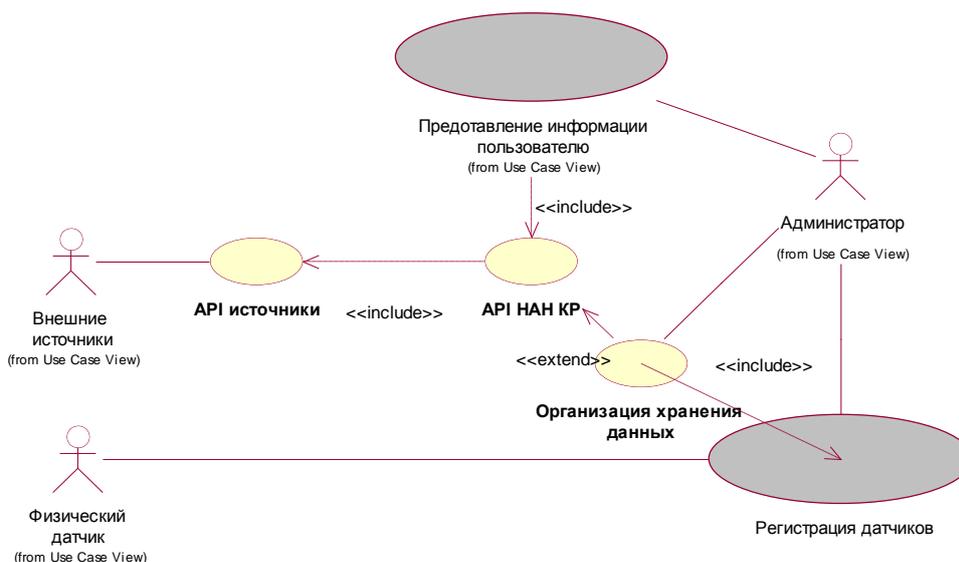


Рисунок 4 – Детализация группы функций «Организация получения и хранения данных»

Архитектура Web-портала

Архитектура Web-портала представлена на рисунке 5 в виде UML-диаграммы компонентов [6]. На этой диаграмме показаны взаимосвязи между основными компонентами системы:

- Client – клиентская среда, где пользователь контактирует с предоставляемым системой функционалом;
- WEB-Server – отвечает за формирование запросов на сервер;
- Server– представляет собой среду, где обрабатываются запросы;
- Parser– отвечает за графический интерфейс пользователя;

- External API – внешние источники информации, представляющие данные о датчиках, результаты мониторинга, новостной поток;
- Sensors – физические датчики, зарегистрированные в системе мониторинга экологической информации КР;
- Moder (ORM) – создание записей в базу данных из объектов классов и получение объектов из базы данных;
- DataBase – база данных, куда записываются все приходящие данные.

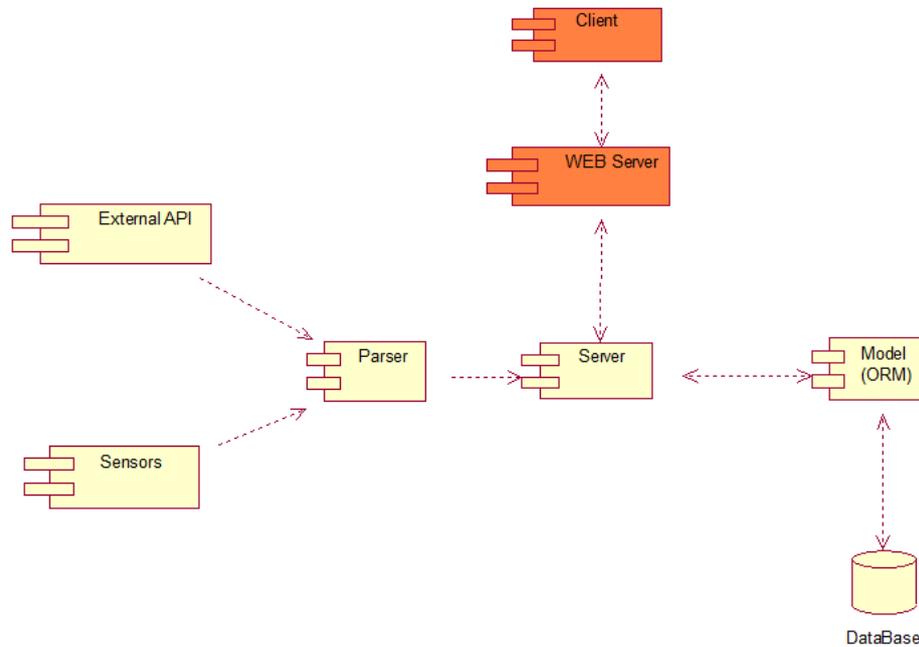


Рисунок 5 – Компоненты системы

Для взаимодействия с серверной частью использован REST (Representational State Transfer – передача репрезентативного состояния) способ создания API с помощью протокола HTTP. Принципы REST API [7]:

- Client-Server: отделения клиента от сервера по HTTP-протоколу;
- Stateless: отсутствие записи состояния клиента;
- Casheable (кэширование): В каждом запросе клиента должно быть указано, нужно ли кэшировать данные (сохранять в специальном буфере для частых запросов);
- Layered System (многоуровневая система): возможно существование промежуточных узлов, при этом каждый узел взаимодействует только с ближайшими уровнями и не связан запросами с другими;
- Uniform Interface (унификация): все данные должны запрашиваться через один URL-адрес стандартными протоколами;
- Code on Demand (код по запросу): сервер может отправлять скрипты для выполнения на клиенте.

Принципы REST API помогают наладить эффективную работу серверов для своих сайтов и приложений. REST отлично подходит при выстраивании серверной архитектуры с помощью других протоколов: HTTP, URL, JSON и XML [7].

Все модульные части Web-портала разбиты на классы. Дальнейшее развитие может обеспечиваться путем добавления новых классов, методов, интерфейсов и представлений. При разработке классов была выбрана модель Code First, при которой сначала пишет класс, затем по классу создаются база данных и таблицы. В соответствии с этим подходом нужно только определить модель, которая будет храниться в базе данных, и класс контекста. На рисунке 6 показана диаграмма классов, реализованных в рамках проекта «Web-портал экологической информации КР», на которой отображены следующие классы и интерфейсы взаимодействия с клиентскими запросами:

- классы для разрабатываемой системы: Sensor, SensorType, Location, Measurement, FavoriteSensor, User, Admin, News;
- интерфейсы для взаимодействия с клиентскими запросами: SensorController, SensorTypeController, MeasurementController, LocationController, FavoriteController, UserController, AdminController.

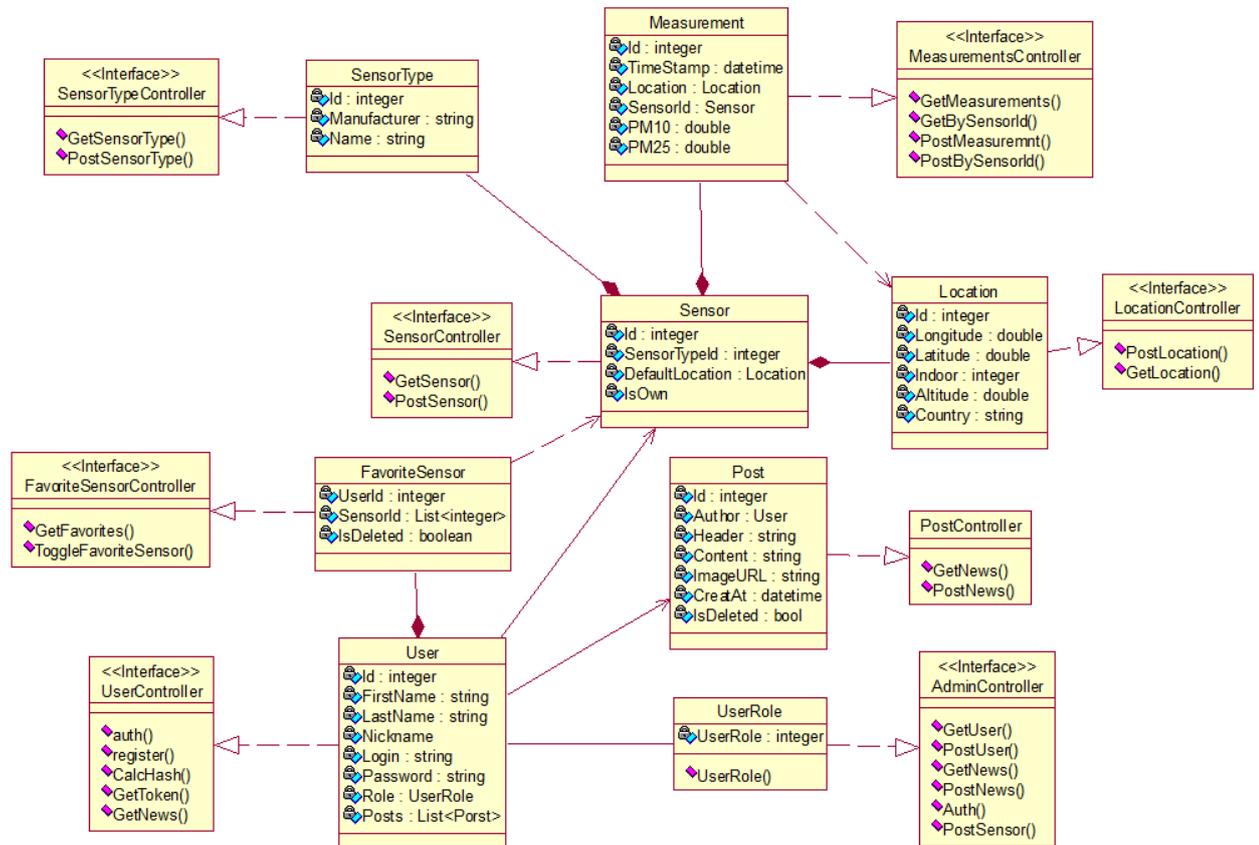


Рисунок 6 – Диаграмма классов

Реализация системы

Для реализации были использованы следующие средства разработки: Microsoft Visual Studio 2019; языки программирования: C# [8]; доступ к данным: ORM Entity Framework Core 5; СУБД: PostgreSQL [9]; контроль версий: GitHub; контейнеризатор приложений: Docker [10]. В качестве аппаратной базы для организации сбора данных были выбраны плата сбора данных ESP8266 NodeMcu и лазерный датчик SDS011 [11] – сенсор, измеряющий количество частиц пыли категорий PM2.5 (диаметром меньше 2,5 мкм) и PM10 (диаметром от 2,5 до 10 мкм) в воздухе.

Было проведено тестирование программного продукта, при этом результаты тестирования подтвердили, что программный продукт отвечает всем функциональным требованиям и требованиям качества.

После входа в портал пользователь взаимодействует с картой местности, где расположены датчики (рисунок 7). При выборе датчика при помощи клика мыши пользователю становится доступна основная часть информации о датчике и результаты мониторинга в режиме реального времени – последние измерения уровня загрязнения воздуха твердыми частицами (рисунок 8). Зарегистрированный пользователь при этом может подписаться на датчик.

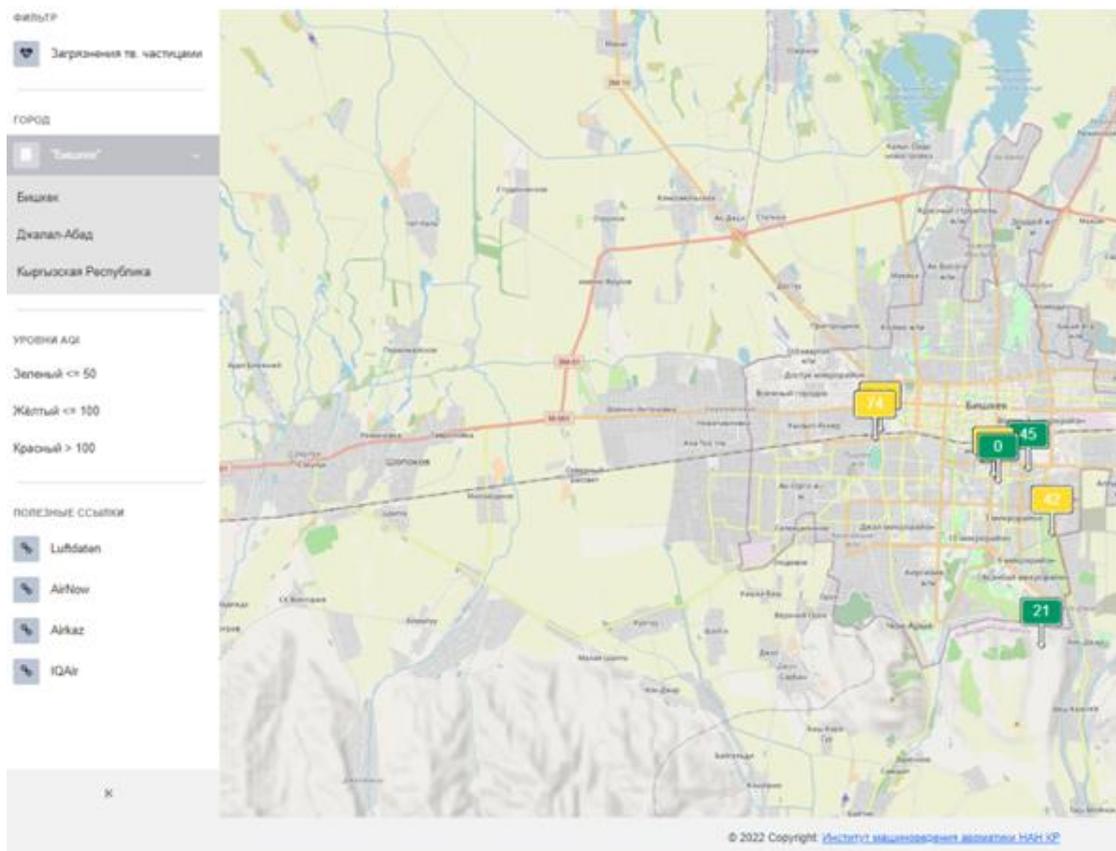


Рисунок 7 – Страница карты с маркерами датчиков

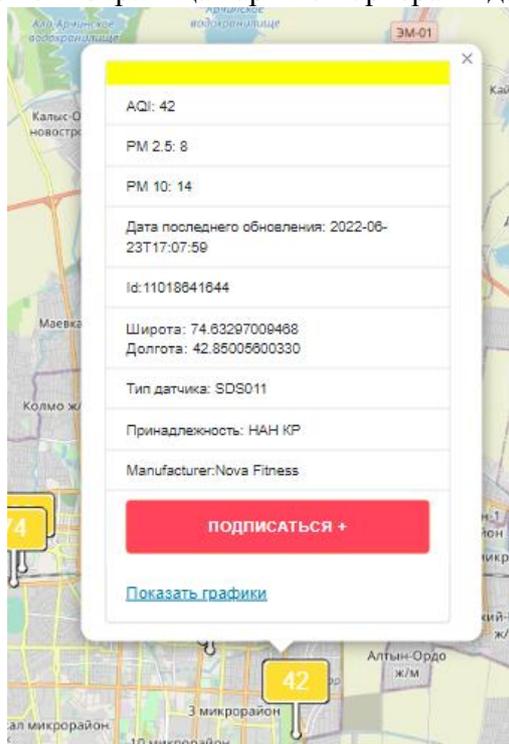


Рисунок 8 – Форма с информацией об актуальных измеренных данных и о датчике

Для регистрации собственного датчика зарегистрированному пользователю необходимо перейти на соответствующую страницу (рисунок 9). После всех операций датчик появится на карте, что будет говорить о том, что датчик прошел регистрацию и получил новый ID.

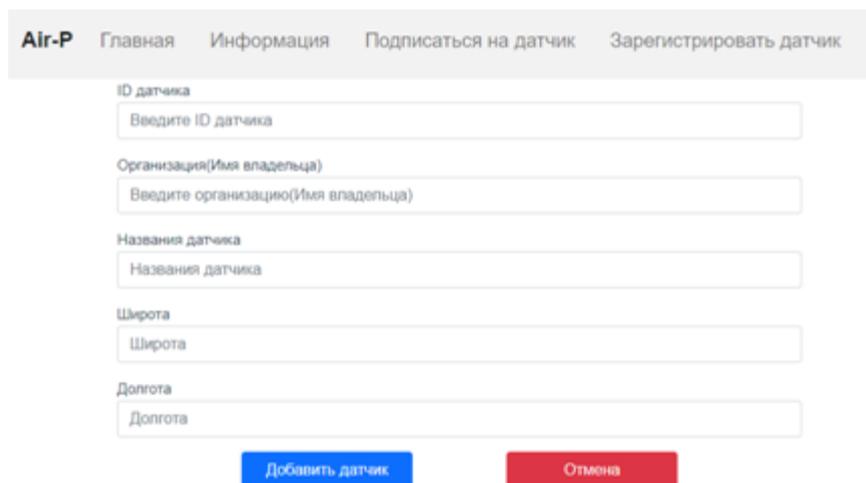


Рисунок 9 – Страница с регистрацией датчика

После того как данные с датчиков будут загружены, на карте появятся маркеры с местоположением датчиков. При выборе маркера и функции “Показать графики” произойдет переход на страницу с графиками за весь период измерения датчика, где уже используется GET-запрос с параметрами выбранного датчика. Также возможно получение результатов измерений этого датчика за указанный период времени.

Заключение

Таким образом, в работе представлены функциональные требования к Web-порталу экологической информации, архитектура и общая модель системы и некоторые скриншоты реализации. Web-портал предназначен для организации сбора, хранения и представления данных мониторинга параметров окружающей среды, в частности, концентраций твердых частиц PM2.5, PM10 и индекса качества воздуха AQI с внешних открытых источников данных, а также для подключения датчиков пользователей с помощью HTTP REST API запросов с дальнейшей возможностью подписки на эти датчики других пользователей. Web-портал имеет информативный и удобный интерфейс пользователя, представленный в виде интерактивной карты с маркерами датчиков, графиками, контекстными формами с данными.

В дальнейшем портал будет дополняться другими возможными источниками открытых геоэкологических данных, а также новыми сервисами и услугами, что позволяет рассматривать разработанный Web-портал как эволюционный прототип облачной информационно-измерительной системы.

Литература

1. Web-сайт для сбора данных LuftDaten. URL: <https://luftdaten.info> (дата обращения 15.10.2022).
2. AirNow Department of State. URL: https://airnow.gov/index.cfm?action=airnow.global_summary#U.S._Department_of_State_Bishkek (дата обращения: 30.05.2022).
3. Гайдамако В.В. Инфраструктура Sensor-Cloud – облачные информационно-измерительные системы // Проблемы автоматизации и управления – 2018. – №2 (35). – С. 109–118.
4. Air Quality Index (AQI) - A Guide to Air Quality and Your Health. US EPA. 9 December 2011.
5. Модели прогноза уровня загрязнения атмосферного воздуха г. Бишкек / Н. М. Лыченко, Л. И. Великанова, С. Н. Верзунов, А. В. Сороковая // Вестник Кыргызско-Российского Славянского университета. – 2021. – Т. 21. – № 4. – С. 87–95. – EDN IRIZPF.

6. Гома Х. – UML-проектирование систем реального времени, параллельных и распределенных приложений – 2011. Addison wesley. 259 стр.
7. REST API. URL: <https://blog.skillfactory.ru/glossary/rest-api/> (дата обращения: 30.05.2022).
8. Язык программирования С# и платформа .NET. URL: <https://metanit.com/sharp/> (дата обращения: 30.05.2022).
9. Руководство по PostgreSQL. URL: <https://metanit.com/sql/postgresql/> (дата обращения: 30.05.2022).
10. Docker docs. URL: <https://docs.docker.com> (дата обращения: 06.06.2022).
11. Всемирная сеть сенсоров окружающей среды Sensor.Community <https://sensor.community/ru/sensors/airrohr/>

УДК551.508.94

М.К. Олжабаев murat16091973@gmail.com*К.Ж. Исабаев* rtv_nk@mail.ru*Военно-инженерный институт радиоэлектроники и связи, г. Алматы,
Республика Казахстан**Н.Б. Имансакипова* nurlanaimansakipova@mail.ru*Satbayev University**И.В. Брякин* bivas2006@yandex.com*Институт машиноведения и автоматизации НАН КР*

СПОСОБЫ И СРЕДСТВА ОБНАРУЖЕНИЯ ГРОВОЙ АКТИВНОСТИ В ВОЗДУШНОМ ПРОСТРАНСТВЕ

В настоящей статье анализируются особенности возникновения грозовой активности в воздушном пространстве и способы, средства ее обнаружения. Надежная и точная система обнаружения молний на большой территории и представления этой информации в режиме реального времени пилотам, операторам аэродромов и службам воздушного движения является большим преимуществом для безопасности, так как грозовая активность представляет значительную опасность воздушным судам как во время полета, так и на земле, работе аэронавигационного оборудования, систем и комплексов противовоздушной обороны. Особо отмечается, что грозовая активность высокой интенсивности так же может привести к чрезвычайным ситуациям природного и техногенного характера. Следовательно, исследователи могут анализировать законы распределения грозовых кластеров во времени и в пространстве и сравнить их с географическими законами за счет структуры векторов-признаков грозовых разрядов, выделенных в грозовой очаг. Главным образом определить среди грозовых объектов наиболее опасные, угрожающие жизни человека и наносящие ущерб многим сферам его деятельности.

Ключевые слова: гроза, молния, воздушное пространство, угрозы, активность, атмосфера, заряд, сеть, обнаружение

Введение

Изучение гроз и их действия на авиационную технику и сооружения на земле является доминирующей темой последних исследований в ветротехнике. Это связано с тем, что методы, используемые для определения действия ветра на конструкции, все еще в основном основаны на моделях для стационарных явлений синоптического масштаба, которые происходят в нейтральных атмосферных условиях, с профилями скоростей в равновесии с планетарным пограничным слоем (ППС). Грозовая активность представляет опасность для эксплуатации воздушных судов (в полете и на земле), работы аэропорта и предоставления услуг воздушного движения. Молния может указывать на наличие других метеорологических опасностей для полетов. В некоторых ситуациях она может быть единственным надежным средством обнаружения этих других опасностей.

Понятие грозовой активности в воздушном пространстве. Гроза – это природное явление, возникающее между или внутри облаков электрических разрядов, сопровождающийся громом [1- 11, 12].

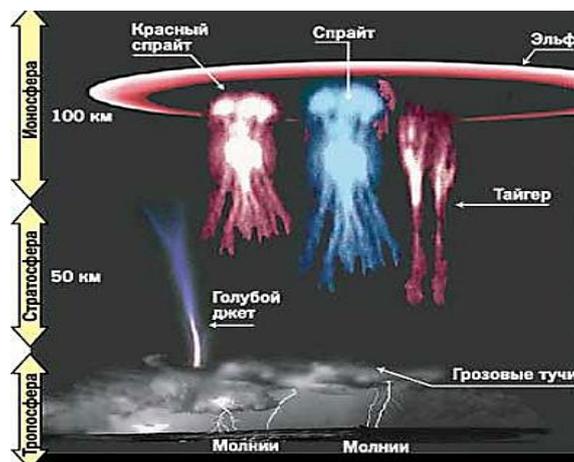


Рисунок 1 – Природа грозовых разрядов

Молния – электрический световой разряд в атмосфере, который происходит из или внутри облака или реже из высоких сооружений на земле или в горах. Можно выделить три основных типа молний. Кроме того, существуют и другие формы светящихся электрических проявлений. Отметим также, что молнии иногда можно наблюдать вместе со шлейфами вулканического пепла во время извержений (рис.1) [1].

Особенности возникновения грозовой активности в воздушном пространстве. Грозы связаны с кучево-дождевыми облаками и чаще всего сопровождаются осадками, которые, достигая земли, выпадают в виде ливня, снега, снежных гранул, мелкого града или града [1].



Рисунок 2 – Громоотвод



Рисунок 3 – Облачный разряд

Этот тип молнии (рис. 2), называемый в народе «громоотвод» или «молния из облака в землю», возникает между облаком и землей. Обычно кажется, что она следует извилистым путем и обычно разветвляется вниз от четко выраженного основного канала (полосатая, вилообразная или ленточная молния). Разряд в земле обычно инициируется, когда движущийся вниз отрицательно заряженный «ступенчатый лидер» соединяется с потоком положительного заряда, тянущимся вверх [2]. После создания электропроводящего канала следует мощный электрический разряд. Это «обратный удар» и наиболее яркая и заметная часть разряда молнии. Большинство вспышек молнии между облаками и землей состоят из нескольких ударов, что вызывает эффект мерцания или стробоскопического света [1, 2].

Хотя это случается гораздо реже, чем молнии, инициированные движущимися вверх стримерами, молнии «облако-земля» могут быть инициированы движущимся вниз положительно заряженным стримером. Обычно он возникает в верхней части грозового облака, а не в нижней. Положительный разряд от облака к земле обычно очень яркий по сравнению с другими молниями. Он также может пройти много километров по горизонтали и ударить в землю в виде так называемого «синего болта» (молния в чистом воздухе, или наковальня). Разряды «земля-облако», инициированные движущимся вверх лидером, иногда могут исходить от объектов на земле, таких как высокие башни и небоскребы (рис. 3).

Этот тип молнии (рис. 4), который в народе называют «листовой молнией», поскольку он освещает небо листовым светом, возникает внутри грозового облака (внутриоблачная молния) или из одного облака в другое (облачно-облачная молния, или межоблачная молния). Как правило, она создает рассеянное освещение без четкого канала. К этому типу молний относятся так называемые тепловые молнии, состоящие из диффузного света, вспыхивающего от далеких гроз, видимых на горизонте [3]. Иногда разряды молнии, возникающие под наковальней или внутри нее, можно увидеть горизонтально движущимися на некоторое расстояние, образуя многочисленные ветви, похожие на деревья. Такие разряды известны как «ползущие наковальни» [4].



Рисунок 4 – Воздушный разряд

Этот тип молнии, который также иногда называют «полосатой молнией», возникает в виде извилистых разрядов, проходящих из грозового облака в воздух и не ударяющих в землю. Разряд часто разделяется на ветви, но с четко выраженным главным каналом. Он часто включает длинную квазигоризонтальную часть [5].

К другим формам светящихся электрических разрядов относятся следующие.



Рисунок 5 – Спрайты



Рисунок 6 – Джеты



Рисунок 7 – ELVES, эльфы

Короткоживущие светящиеся электрические проявления, формирующиеся в верхних областях атмосферы над крупными грозовыми облаками [3, 4].

Крупные грозовые облака способны производить электрические явления высоко в атмосфере. Они редко наблюдаются визуально, и то в основном с помощью чувствительного фотографического оборудования и в силу своей слабости только ночью. К переходным светящимся явлениям относятся спрайты, джеты и ELVES.

Спрайты (рис. 5). Это крупномасштабные электрические разряды, возникающие высоко в атмосфере на высоте около 50–90 км, над крупными грозовыми системами. Обычно они возникают одновременно с мощными положительными разрядами от облаков к земле. Они проявляются в виде большой, но слабой вспышки, обычно красного цвета. Спрайты длятся не более нескольких секунд. Их редко можно увидеть человеческим глазом и только ночью над отдаленными крупными грозовыми системами [4].

Джеты (рис. 6). К ним относятся голубые струи, голубые стартеры и гигантские струи. Голубые струи — это узкие конусообразные вспышки голубого цвета, выходящие из вершины грозового облака и поднимающиеся на высоту около 40–50 км. Голубые струи длятся всего долю секунды, но их наблюдали пилоты самолетов. Голубые звезды более слабые и короткие, чем голубые струи, и достигают высоты около 20 км. Гигантские струи были сфотографированы на высоте около 70 км [5].

ELVES (рис. 7) (излучение света и возмущения очень низкой частоты, вызванные источниками электромагнитных импульсов) появляются на высокочувствительных видеозаписях в условиях низкой освещенности в виде тусклого, уплощенного, расширяющегося свечения диаметром до 400 км и продолжительностью, как правило, менее 0,001 секунды.



Рисунок 8 – Шаровая молния



Рисунок 9 – Огонь святого

Иногда наблюдается светящийся шар (рис. 8) вскоре после разряда на землю. Этот шар, диаметр которого, как сообщается, обычно составляет от 10 до 20 см, но иногда достигает 100 см, известен как шаровая молния. Она медленно движется в воздухе или по земле и обычно исчезает с сильным взрывом [6].

Огонь святого Эльма - это непрерывный светящийся электрический разряд (рис. 9) неяркой или средней интенсивности в атмосфере, происходящих над поверхностью Земли (например, молниеотводов, ветряных лопастей и корабельных мачт) или от летящих самолетов (например, кончиков крыльев). Это явление можно наблюдать, когда электрическое поле вблизи поверхности объектов становится сильным [1]. Оно часто проявляется в виде фиолетовых или зеленоватых шлейфов, или сполохов, хорошо видимых ночью. Хотя электрические условия, порождающие огонь святого Эльма, существуют в связи с грозами, он также может развиваться в электрических полях, усиленных обильным снегом и сильным ветром без гроз, а также при пролете самолетов через вулканический пепел [2, 3].

Анализ способов и средств обнаружения грозовой активности в воздушном пространстве. Далее рассмотрим способы и средства обнаружения грозовой активности в воздушном пространстве. Сети обнаружения молний отслеживают развитие, интенсивность и движение гроз и используются для выпуска предупреждений и прогнозов суровой погоды. Сети обнаружения молний существуют во многих странах и могут быть объединены для работы на континентальной основе. Зачастую они управляются (или финансируются) государством, их цель многообразна, но в первую очередь это защита населения и некоторых отраслей промышленности посредством предупреждений, прогнозов и развертывания спасательных и аварийных групп. Типичные отрасли промышленности, помимо авиации, которые полагаются на молнии и связанные с ними сообщения о неблагоприятной погоде, включают производство электроэнергии, лесное хозяйство, опасные материалы, спорт и отдых. Исторические архивные данные также используются для проведения исследований, прогнозирования будущего и расчета страховых рисков [7].



Рисунок 10 – Внешний вид и демонстрация работы вспомогательного модуля ERL-10

Сеть обнаружения молний состоит из сети неподвижных антенн, которые записывают электрические сигналы, излучаемые разрядами молний среди облаков и из облаков в землю. Для триангуляции точного местоположения требуется не менее трех антенн. Эта наземная сеть может также поддерживаться мобильными наземными детекторами и детекторами космического базирования. Однако ни один из этих двух вариантов не повысит стандарт обслуживания, требуемый авиацией – точность и скорость: мобильные устройства, работающие в одиночку, могут вносить неоднозначность в определение местоположения, а спутниковым данным требуется несколько минут, чтобы достичь пользователя [8, 17].

Европейский проект «Ветер и порты», выполненный автором и его исследовательской группой, предлагает вклад в расширение знаний о грозах. В рамках проекта была создана сеть мониторинга, проведено численное моделирование ветровых полей, анализ ветрового климата и внедрены алгоритмы для среднесрочного (1–3 дня) и краткосрочного (0,5–2 часа)

прогноза ветра в портах Генуи, Савоны, Специи, Ливорно и Бастии. Новый проект «Ветер, порты и море» в настоящее время финансируется Европейским сообществом для продолжения и расширения предыдущего проекта [9].

Сеть мониторинга, реализованная в рамках проекта «Ветер и порты», состоит из 22 ультразвуковых анемометров. Другие 11 ультразвуковых анемометров были установлены местными портовыми властями. В рамках проекта «Ветер, порты и море» 7 ультразвуковых анемометров и 3 лидара. Сеть мониторинга ветра, состоящая из 40 ультразвуковых анемометров и 3 лидаров (рис. 10) запущена с 2015 года; к анемометрической сети добавляются другие приборы, включая термометры, барометры и гигрометры [4, 5].



Рисунок 11 – Сеть мониторинга

Набор серверов, размещенных в каждом порту, принимает данные о мерах, разрабатывает основные статистические данные за 10-минутные периоды и отправляет информацию на центральный сервер в DICCA. Этот сервер хранит необработанные данные и статистические значения в центральном наборе данных после систематической проверки и подтверждения данных. Была реализована процедура для извлечения и разделения различных интенсивных ветровых событий. Хотя в литературе обычно применяется бинарное разделение стационарных гауссовских синоптических событий и нестационарных негауссовских грозных явлений, зарегистрированные данные указывают на существование стационарных негауссовских промежуточных событий, которые делают вышеуказанное разделение критически важным вопросом (рис. 11).

Каждая запись скорости грозы разлагается на сумму ее медленно меняющейся средней части, усредненной по скользящему среднему за 30 с, и остаточного колебания, определяемого произведением ее меняющегося во времени среднеквадратичного значения на случайный стационарный гауссовский сигнал с нулевым средним и единичным среднеквадратичным значением [9]. Основные параметры каждой записи и ее частей собираются и анализируются для создания реалистичной параметризованной модели грозы. Модель распределения экстремальных скоростей ветра в грозах находится на стадии изучения. Динамическая реакция сооружений на грозу оценивается двумя альтернативными методами. Первый основан на моделировании методом Монте-Карло ветровых полей, согласованных с вышеуказанной моделью грозы. Второй, ориентированный на инженерные и кодовые приложения, использует эволюцию метода спектра отклика, широко распространенного в сейсмическом секторе [1]. Этот метод сначала был разработан для систем с одной степенью свободы, подверженных идеально когерентным полям ветра (рис. 12), а затем распространен на системы с несколькими степенями свободы, подверженные частично когерентным полям ветра. Сравнение результатов, полученных двумя методами, указывает на почти полное совпадение. Метод спектра реакции приводит к простому определению эквивалентных статических воздействий грозы. Метод спектра отклика и эквивалентные статические действия гроз включены в новый подход, называемый методом независимой ветровой нагрузки, направленный на создание нового критерия для оценки ветровой нагрузки на сооружения в смешанном климате. Вместо классического уникального условия ветровой нагрузки, объединяющего различные ветровые явления по смешанной

статистике, данная методика дает столько независимых условий ветровой нагрузки, сколько ветровых явлений характеризует смешанный климат, прежде всего синоптические и грозовые явления [7, 8].

Этот подход надежно поддерживается глубоким разнообразием этих явлений в отношении стационарного или нестационарного и гауссового или негауссового характера скорости ветра, формы профилей скорости, параметризации полей ветра в отношении длины шероховатости и термической стратификации, интенсивности, размера, продолжительности и частоты [10].

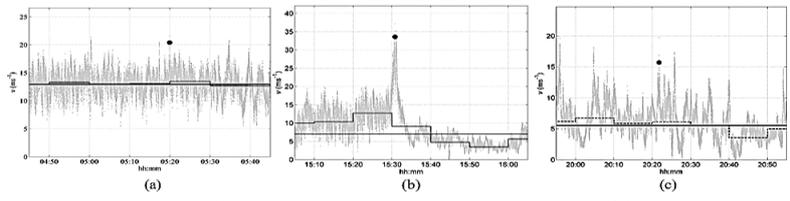


Рисунок 12 – (а) синоптическое событие; (б) гроза; (с) промежуточное событие

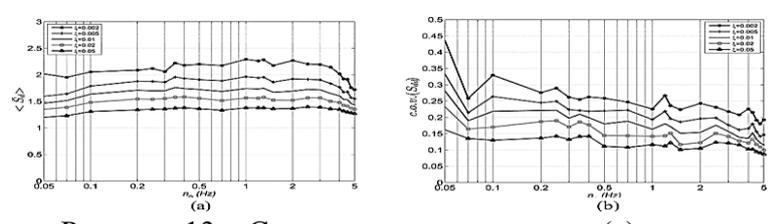


Рисунок 13 – Спектр среднего отклика (а) и его коэффициент вариации (б) как функции основной частоты n_0 и коэффициента демпфирования ξ

Это замечание о различной интенсивности, размере, продолжительности и частоте синоптических и грозовых явлений указывает на то, что они не могут быть представлены уникальным набором частичных и комбинированных коэффициентов, обычно принимаемых в отношении классического уникального условия ветровой нагрузки. Диверсификация этой нагрузки в набор независимых условий ветровой нагрузки приводит к пересмотру фактических правил сочетания нагрузок, по крайней мере, к введению диверсифицированных наборов частичных и комбинированных факторов ветровой нагрузки [9].

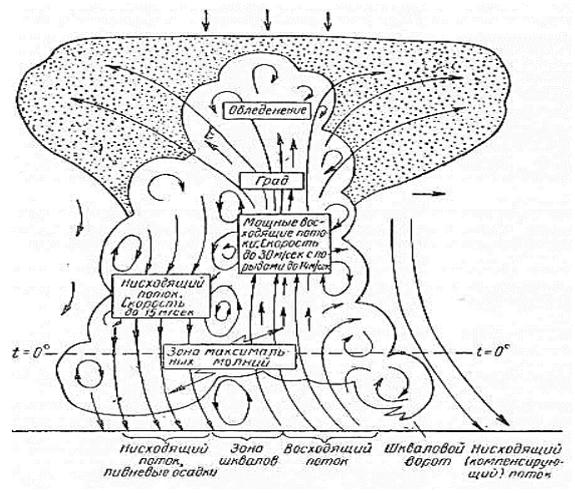


Рисунок 14 – Грозовое облако в максимальной фазе

Грозовое облако в среднем существует около 5 ч. За один час и менее могут «усложниться» все три стадии развития облака, есть вероятность, что грозовое облако может

сохраняться до 10 ч и более. Во время образования облака происходит его электризация. Это электризация опасно для авиации, так как после накопления электрических зарядов и достижения между облаками или облаком и землёй напряжённости электрического поля, превышающей пробивную напряжённость, возникают молнии [7, 8].

Грозовые облака могут развиваться неодинаково, это зависит от следующих факторов: влагосодержания воздушной массы, контраста температур в зоне атмосферного фронта и рельефа местности, над которой проходят кучево-дождевые облака. Обычно грозы делят на внутримассовые и фронтальные [9].

Спутники. Мы можем наблюдать грозы с помощью различных инструментов. Большинство районов Земли можно увидеть с помощью метеорологических спутников. Спутники регулярно фотографируют Землю из космоса, сообщая нам, где находятся облака. Метеорологи наблюдают за этими снимками в течение долгого времени, чтобы увидеть быстро растущие облака, что является признаком возможной грозы. Спутники также могут сообщить нам температуру облаков. Облака с холодными вершинами обычно находятся очень высоко в атмосфере, и это может означать, что облако достаточно высокое, чтобы стать грозой. Метеорологи также отслеживают движение облаков, чтобы определить, какие районы будут затронуты грозой в следующий раз [10].

Радары. Погодный радар очень важен для метеорологов, поскольку он может обнаружить дождь и суровую погоду даже в облачную или темную погоду.

Допплеровский радар посылает электромагнитные волны, которые могут отражаться от предметов в воздухе, таких как осадки. Количество отраженной энергии может сказать нам, насколько сильным может быть дождь или град. Допплеровский радар также может показать нам, как дует ветер вблизи и внутри шторма. Это помогает понять, какие виды опасности может представлять гроза (торнадо, микровзрывы, фронты порывов и т.д.). Это также помогает нам понять, как гроза питает сама себя [6, 16].

Дополнительные средства мониторинга грозовой активности. Канадская компания Voltek один из лидеров среди производителей программного обеспечения и оборудования для получения быстрой и аналитической информации о грозовых разрядах. Современные системы детектирования молний Voltek установлены по всему миру. Они решают задачи штормового оповещения как самостоятельно, так и в составе комплексов мониторинга окружающей среды совместно с метеорологическими радиолокаторами.

Электростатические флюксометры Voltek. Эти системы EFM-100. EFM-100 ближнего радиуса действия (дальность до 38 км), а также модернизированная версия EFM-100M. EFM-100M определяют грозовые разряды и состояние высокой напряжённости электрического поля, предшествующее первым разрядам молний.

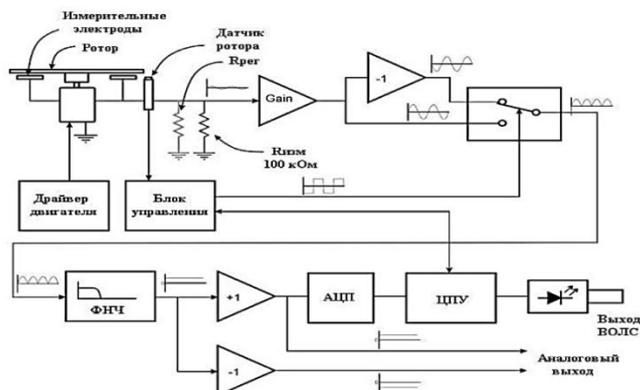


Рисунок 15 – Функциональная схема флюксометра EFM-100

Для увеличения дальности передачи используют протокола связи RS-485, а с помощью GSM/GPRS модема передают сигнал тревоги на мобильный телефон или электронную почту. В диапазоне от -20 до +20 В показывает в аналоговой форме результат измерения., так

Максимальная напряженность 20 кВ/м равен сигналу с амплитудой 20 В, 12 кВ/м – 12 В и т.д., так как напряжение на аналоговом выходе прямо пропорционально напряженности электрического поля.



Рисунок 16 – Внешний вид флюксметра EFM-100 и способы его монтажа

Корпус флюксметра EFM-100 изготавливают из нержавеющей стали и алюминия, для повышения коррозионной стойкости. Выводы соединительных разъемов флюксметра EFM-100 имеют золотое покрытие, рабочий диапазон устройства от -40 до +60°C.

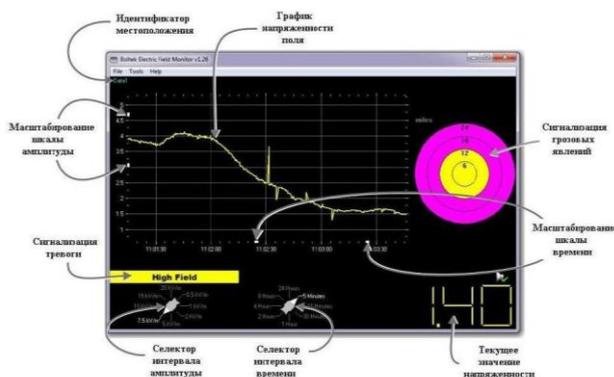


Рисунок 17 – Внешний вид основного окна

Графический интерфейс Voltek Electric Field Monitor используют для визуализации и наглядного представления полученных результатов. За счет программного обеспечения возможно увидеть в виде графиков, в режиме реального времени напряженность электрического поля атмосферы, а также регистрировать и сохранять в log-файлы случаи грозных разрядов (рис. 17). Прилагается подробная инструкция. Сигналы предупреждения и сигнализации создаются на основе настраиваемых параметров - это дальности возникновения гроз, длительности и уровне повышенной напряженности, периодичности и т.д. Возможно подключить от 1-го до 4-х устройств к одному компьютеру, объединенных по локальной или глобальной сети, с установленным программным обеспечением.

Грозопеленгаторы Voltek. Это системы дальнего радиуса действия (до 1200 км) к ним обязательно применяют грозопеленгаторы LD-250 (LD-250M), LD-350 или StormTracker. Для определения азимута и дальности до грозы, применяют самостоятельно, либо в качестве дополнения к системам ближнего радиуса действия однопунктовые грозопеленгаторы-дальномеры.

Пеленгатор Stormtracker похож LD-250, такая же дальность, но не является автономным устройством [6]. Состоит из PCI-карты, вставляемую в слот материнской платы компьютера, подключается к внешней антенне и устанавливается программное обеспечение (рис. 18).



Рисунок 18 – Внешний вид грозопеленгаторов

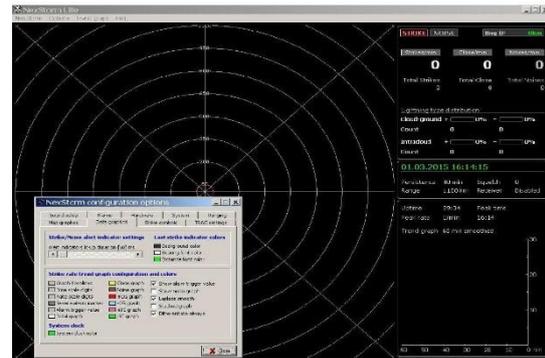


Рисунок 19 – Внешний вид основного окна

Программное обеспечение NexStorm (Astrogenic) обеспечивает отображение информации. Она позволяет определить местоположение разрядов молнии в полярных координатах, где в центре системы координат располагается приемник.

На мониторе отображается информация о пространственной локализации каждого из разрядов, позволяющая отследить скорость, направление перемещения грозового фронта и количество разрядов, зарегистрированных за последнее время. Главное окно NexStorm содержит индикаторы, метки и счетчики, объединенные в три группы и располагаемые справа от карты (рис. 19).

Угрозы грозовой активности в воздушном пространстве. Карта молниевой опасности отображает местную угрозу молний для определенных районов. Она в значительной степени основана на вероятности возникновения грозовых молний из облаков в землю в сочетании с ожидаемой частотой вспышек [5, 19].

Таблица 1– Угрозы грозовой активности в воздушном пространстве

Уровень угрозы	Характеристика
Экстремальный	«Чрезвычайная угроза жизни и имуществу от молнии». В радиусе 12 миль от места расположения, умеренная вероятность возникновения молний CG (или 50% вероятность грозы), с грозами, способными вызвать чрезмерное количество молний CG. И/ИЛИ... высокая вероятность появления молний CG (или вероятность грозы от 60 до 70 %), при этом грозы могут сопровождаться частыми молниями CG. И/ИЛИ... очень высокая вероятность молний CG (или вероятность грозы от 80% до 90%), при грозах, способных к случайным молниям CG
Высокий	«Высокая угроза жизни и имуществу от молнии». В радиусе 12 миль от места расположения, низкая вероятность молний CG (или 30%–40% вероятность грозы), с грозами, способными вызвать чрезмерное количество молний CG. И/ИЛИ... умеренная вероятность появления молний CG (или 50% вероятность грозы), при этом грозы могут сопровождаться частыми молниями CG. И/ИЛИ... высокая вероятность молний CG (или 60%–70% вероятность грозы), с грозами, способными к случайным молниям CG
Умеренный	«Умеренная угроза жизни и имуществу от молнии». В пределах 12 миль от местоположения очень низкая вероятность появления молний CG (или вероятность грозы от 10 до 20 %), при этом грозы способны вызвать чрезмерное количество молний CG. И/ИЛИ... низкая вероятность появления молний CG (или вероятность грозы от 30% до 40%), при этом грозы могут сопровождаться частыми молниями CG.

	И/ИЛИ... умеренная вероятность молний CG (или 50% вероятность грозы), с грозами, способными периодически освещать молнии CG
Низкий	«Низкая угроза жизни и имуществу от молнии». В пределах 12 миль от места расположения очень низкая вероятность возникновения молний CG (или вероятность грозы от 10% до 20%), при этом грозы могут быть частыми. И/ИЛИ... низкая вероятность появления молний CG (или вероятность грозы от 30 до 40 %), при этом грозы могут сопровождаться молниями CG время от времени
Очень низкий	«Очень низкая угроза жизни и имуществу от молнии». В радиусе 12 миль от местоположения очень низкая вероятность появления молний CG (или вероятность грозы от 10 до 20 %), при этом грозы могут иногда приводить к появлению молний CG
Без угроз	«Отсутствие заметной угрозы жизни и имуществу от молнии». В радиусе 12 миль от местоположения условия окружающей среды не способствуют возникновению молнии CG

Перспективы и значимость развития способов и средств обнаружения грозовой активности в воздушном пространстве. Необходимо разработать рекомендации по защите от электрических разрядов, в первую очередь сделать анализ возникающих при этом процессов и оценить величины наведенных токов и напряжений, так как грозовая активность опасно влияет на авиацию, энергетику и других отраслей экономики и населения, а также возникновение лесных пожаров [15].

В современных условиях, для решения важной задачи, необходимо изучение грозовой активности и развитие системы грозомониторинга [2]. Программно-аппаратные комплексы позволяют за короткое время развернуть систему наблюдения за грозовой активностью, которая отличается высокой надежностью и простой эксплуатации. Большой выбор функциональных возможностей и дополнительных опций помогает получить оптимальное решение для заданных условий эксплуатации [4]. Развитие высокотехнологической системы грозомониторинга, помимо научно-прикладных целей, позволит решать множество важнейших задач для всех отраслей экономики: уменьшение экономического ущерба от грозовых явлений, грамотное проектирование любых инженерно-технических сооружений и обеспечение безопасной эксплуатации различных объектов [6].

Заключение и выводы. Грозы являются и будут являться опасным природным явлением, которое приводит к опасным ситуациям, в результате которых гибнут люди и наносится ущерб многим сферам человеческой деятельности, таким как электроэнергетика и лесное хозяйство [13, 20]. Для своевременного выявления закономерностей эволюции грозовой активности и прогнозирования особенностей развития и интенсивности грозовой деятельности, необходимо совершенствовать методы выделения грозовых объектов и методы регистрации грозы [21]. Проведенный анализ показывает, что при обзоре научных публикаций, посвященных задаче выделения грозового очага, не существует аналитического выражения, описывающего развитие грозового очага.

Чтобы выделять грозовой очаг методами многомерного статистического анализа данных, необходимо исследовать грозовые процессы. Необходимо применять методы кластерного анализа и визуализация данных, где грозовой очаг задается множеством векторов-признаков, для того чтобы исследовать структуры совокупности объектов. Для улучшения точности определения местоположения грозового разряда при однопунктовых и многопунктовых методах, необходимо качественное выделение грозового очага из массива данных о грозовых разрядах, так как результаты грозорегистраторов содержат не правильные данные: «шумы» и погрешности определения координат грозового разряда в пункте наблюдения.

Следовательно, совокупность векторов-признаков грозовых разрядов, выделенная в грозовой очаг, позволит исследователям систематизировать законы распределения грозовых кластеров в пространстве и во времени и соотнести их с географическими законами, а также выявить среди грозовых объектов наиболее опасные, угрожающие жизни человека и наносящие ущерб многим сферам его деятельности.

Литература

1. Белоцерковский, А. В. Активно-пассивная радиолокация грозовых и грозоопасных очагов в облаках / А. В. Белоцерковский [и др.] // СПб. : Гидрометеиздат. – 2020. – 215 с.
2. Будуев, Д. В. Однопунктовая система пассивного мониторинга грозовой деятельности: Дис. ... канд. техн. наук: 05.13.18 / Будуев Денис Владимирович. – Челябинск, 2021. – 125 с.
3. Ершова, Т. В. Параметры молниевой активности по инструментальным измерениям / Т. В. Ершова, В. П. Горбатенко // Вестник Томского государственного педагогического университета. – 2020. – №. 5. – 150–154 с.
4. Иньков, В. К. Широтные особенности грозовой деятельности / В. К. Иньков, Л. Г. Махоткин // Труды ГГО. Атмосферное электричество. – 2020. – Вып. 442. – С.34–38 .
5. Кононов, И. И. Радиотехнические методы местоопределения грозовых очагов / И. И. Кононов, И. А. Петренко, В. С. Снегуров. – Л.: Гидрометеиздат, 2021. – 221 с.
6. Муллаяров В.А., Каримов Р.Р., Козлов В. И., Мурзаева Н. Н. Связь грозовой деятельности с солнечной активностью по наблюдениям фонового ОНЧ-излучения // Метеорология и гидрология. – 2020. – N 8. – С. 48–56 .
7. Панюков, А. В. Системы пассивного мониторинга грозовой деятельности / А. В. Панюков, Д. В. Будуев, Д. Н. Малов // Вестник Южно-Уральского государственного университета. Серия: Математика, Физика, Химия. – 2021. – №. 8. – 24 с.
8. Раков, В. А. Современные пассивные радиотехнические системы местоопределения молний / В. А. Раков // Метеорология и гидрология. – 2020. – №1. – С.118–123.
9. Снегуров, А. В. Методика и результаты наблюдений за грозами, системами пеленгации гроз и МРЛ / А. В. Снегуров, А. В. Снегуров, Г. Г. Щукин // Ученые записки Российского государственного гидрометеорологического университета. – 2021. – №. 12. – С.38–49.
10. Трусковский, П. Анализ методов и средств однопозиционной пассивной радиолокации грозовых очагов / П. Трусковский // ProceedingsofInternationalConference RelStat'04. TransportandTelecommunication. – Vol.6. – №3. – 2021. – 431–437 с.
11. Авиационные факторы риска, программа обучения и подготовки ETR - №20 ВМО – ТД - No1390 // Секретариат Всемирной метеорологической организации Женева – Швейцария – 2007. – С.39–43.
12. Ефимов В.В., Основы авиации. Часть I. Основы аэродинамики и динамики полета летательных аппаратов / В.В Ефимов, М.Г Ефимова.// Учебное пособие. – М.: МГТУ ГА, 2012. – С.42–58.
13. Лободин Т.В. К вопросу об ущербе, наносимом грозами // Труды ГГО. – Вып. 486. 1986. – С. 70–71.
14. Иньков В.К., Махоткин Л.Г. Широтные особенности грозовой деятельности // Труды ГГО. Атмосферное электричество. –1981. – Вып.442. – С. 34–38.

15. Раков В.А. Современные пассивные радиотехнические системы местоопределения молний // Метеорология и гидрология. 1990. – № 11. – С.118–123.
16. Кашпровский В. Е. Определение местоположения гроз радиотехническими методами. — М.: Наука. 1984. – 220 с.
17. Панюков А.В., Будуев Д.В. Алгоритм определения расстояния до местоположения молниевго разряда // Электричество. – № 4. – 2001. – С. 10–14.
18. Степаненко В. Д., Гальперин М. С. Радиотехнические методы исследования гроз. — Л.: Гидрометеиздат. 1983. – 204 с.
19. Дружин Г.И., Козлов В.И. Экспериментальные исследования влияния трассы распространения при регистрации излучения мировых очагов гроз // Геомагнетизм и аэрономия. –1994. –Т.34. – N 6. – С. 174–176.
20. Лободин Т.В. К вопросу об ущербе, наносимом грозами // Труды ГГО. – Вып. 486. – 1986. – С. 70–71.
21. Брякин, И. В. Анализ основных свойств ЕЭМП КНЧ-ОНЧ диапазонов / И. В. Брякин, Н. Г. Мамытов // Проблемы автоматки и управления. – 2022. – № 1(43). – С. 50-60. – EDN XGVKIG.

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ОБРАЗОВАНИИ

УДК: 681.14:004

Ч.Ш. Омурзакова, shailoobekovna@list.ru**Э.Т. Султангазиев, esentur_sultanga@mail.ru***Кыргызский национальный аграрный университет им. К.И.Скрябина,
Кыргызстан, г. Бишкек.***СОЗДАНИЕ ЭЛЕКТРОННЫХ УЧЕБНЫХ ПОСОБИЙ НА ОСНОВЕ
СЕРВИСОВ GOOGLE В ОБРАЗОВАТЕЛЬНОМ ПРОЦЕССЕ**

В наше время в образовательных учреждениях огромное внимание уделяется компьютерному сопровождению профессиональной деятельности преподавателя-предметника. Опубликование предполагает сокращенное обобщение навыка разработки электронных образовательных изданий (словарей, лекций направлений, презентаций и др.). Для решения данной проблемы одними из самых удобных, понятных и доступных в использовании являются сервисы Google с большим количеством инструментов для коллективной и индивидуальной работы. Однако, помимо этого, рассматриваются возможности интернет-сервиса Google с целью формирования электронных образовательных ресурсов. Замечается то, что инструментарий Google выделяется относительно несложностью применения, он современен, также снимает многочисленные сложности по применению мультимедийных материалов в силу оптимальной сочетаемости входящих в пакет сервисов.

Ключевые слова: внеаудиторная самостоятельная работа, образовательный процесс, обучение, электронное издание, электронное учебно-методическое пособие, электронное обучение, электронные образовательные ресурсы, мультимедиа-технологии, Google, Google-формы, Google-сайты.

Введение

Данная тема считается актуальной, так как она дает возможность составить перечень методического обеспечения с целью повышения эффективности урока. Для того чтобы занятие было интересным для учащихся, необходимо использовать новейшие формы деятельности. Для решения отмеченной задачи целесообразно практиковать в учебном процессе ИКТ, интегрируя их с остальными современными образовательными технологиями. В наше время преподавателю надо ориентироваться в просторном спектре инновационных технологий.

Одним из перспективных направлений развития современной образовательной системы считается применение электронных учебных пособий. Значительную помощь в подготовке и проведении уроков оказывают электронные приложения к пособиям. Это дает возможность разнообразить домашние задания, сделать их наиболее интересными, помогает студенту работать в индивидуальном темпе. А также оказывает существенное влияние на деятельность не только учащихся, но также самого преподавателя. Преподаватель часто выступает в роли консультанта, помощника, который стимулирует динамичность, активность, а также индивидуальность студентов.

Электронное учебное пособие (ЭУП) — это программно-методический обучающий комплекс, предназначенный для самостоятельного изучения студентом учебного материала по определенным дисциплинам [1,3].

Как правило, электронные учебные пособия строятся по модульному принципу и включают всю необходимую информацию, содержат в себе несколько частей [3]:

теоретическую часть – в основе данной части содержится текст, графика (статические схемы, чертежи, таблицы и рисунки), анимация, натурные видеозаписи, а также интерактивный блок;

практическая часть – там должно быть представлено пошаговое решение типичных задач и упражнений по данному учебному курсу с содержанием минимальных пояснений;

контрольная часть — содержит набор тестов, контрольных вопросов по теоретической части, но также и решение задач и упражнений по практике;

справочная часть – может включать в себя предметный указатель, таблицы, размерности и другую необходимую информацию в графической, табличной или любой другой форме [2].

Применение электронных пособий в учебном процессе позволяет сделать аудиторские и самостоятельные занятия более интересными, динамичными и убедительными, а огромный поток изучаемой информации легкодоступным, а также предоставляет преподавателю большой резерв технической и технологической поддержки, высвобождающей значительную часть его времени для общения со студентами [4,5].

На сегодняшний день создано множество электронных пособий на абсолютно различные темы, начиная с простых небольших программ, завершая сложными и значительными проектами, которые формировались годами. Причина во всеобщей компьютеризации общества; в наше время в образовательных учреждениях огромный интерес уделяется компьютерному сопровождению профессиональной деятельности преподавателя-предметника.

Среди множества информационных ресурсов, которые можно применять в преподавательской практике при формировании электронных кейсов в условиях реализации очного и дистанционного обучения, считаются сервисы Google. Google Sites – это простой, понятный конструктор сайтов, интерфейс и прочие особенности, которые почти идентичны другим сервисам Google: Docs, Sheets и т.д. Сервис Google-сайты позволяет любому участнику образовательного процесса получить доступ к необходимой информации в удобное время, обеспечить мобильность образовательного процесса. Режим обучения может быть традиционным (со стационарного компьютера или ноутбука) или мобильным, при котором доступ к ЭОР открывается через планшет, телефон, смартфон или другое мобильное устройство.

Основные преимущества применения сервисов Google в образовании с точки зрения пользователя – это минимальные требования к аппаратному обеспечению (неотъемлемое требование – присутствие допуска в сеть Интернет). В настоящее время информационные технологии стали неотъемлемой частью в функционировании современного общества и инфраструктуры. С появлением децентрализованной сети Интернет и ростом числа пользователей появилась потребность в передаче большого количества данных и в системах передачи, которые способны обслуживать большое количество запросов [17-18]. Google-технологии не требуют от пользователя современных технических знаний, затрат на приобретение, также сервис специального программного обеспечения (доступ к приложениям можно получить через окно интернет-браузера), поддерживают все операционные системы и клиентские программы, применяемые преподавателями, учащимися и учреждениями образования, все инструменты Google бесплатны. С его помощью мы сможем быстро создать, наполнить содержимым и опубликовать набор шаблонных интернет-страничек, настроенных для наших целей. Нам не потребуется писать ни строчки программного кода. Конструктор позволяет гибко редактировать веб-странички через визуальный редактор по способу drag and drop, прибавляя туда блоки с различными компонентами (картинка, видеоматериал, документ).

Среди сервисов Google наиболее востребованными, а также способствующими решению основных образовательных задач можно отметить Google-формы (Google Forms), Google-презентации, Google-документы (Google Docs), Google-диск (Google Drive).

Отметим, кроме того, что Google-формы не считается единственным сервисом с необходимым функционалом, но он является более удобным как в силу простоты усвоения его возможностей пользователем, так и в силу того, что у многих пользователей сети Интернет, в особенности у владельцев устройств с ОС Android, имеется аккаунт Google, необходимый с целью использования Google-формами; в случае, если аккаунт отсутствует, необходимо зарегистрироваться в Google (нужен Google-аккаунт), а также открыть собственный Google-диск. После регистрации откроется собственный почтовый ящик, где уже будут ждать приветственные письма от команды Google. Нажав на кнопку “Сервисы” в верхнем правом углу, необходимо будет подобрать с раскрывшегося списка пиктограмму “Диск” (рис. 1).

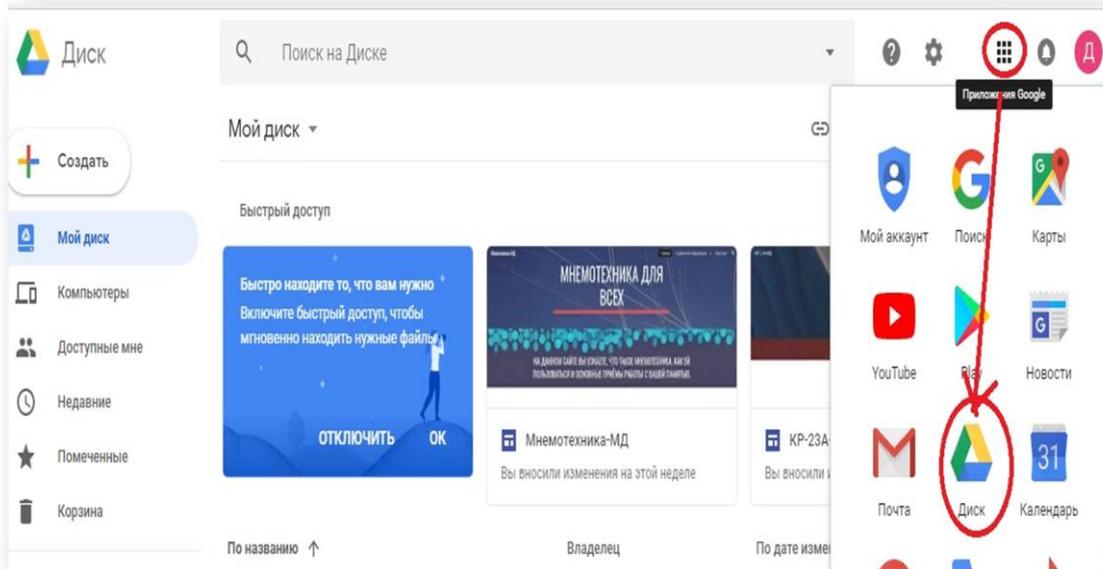


Рисунок 1– Страница Google-диск

Появится консоль, позволяющая создать свой сайт. Данная программа позволяет, минуя браузер, работать с документами Google. Интерфейс Google-диска знаком и понятен. Важно, что любым файлом, хранящимся на Google-диске, можно поделиться в Google+. Это означает, что студенты смогут просмотреть информацию прямо в своей ленте. Публиковать можно документы, таблицы, презентации и формы Google, загрузить форму опроса, применяя Google Docs, видеоролики из YouTube, можно разместить календарь и карты Google, а также использовать большое количество готовых гаджетов из iGoogle.

Наше подготовленное пособие Google-сайт после прохождения процедуры согласований стало использоваться на занятиях со студентами 2-го курса кафедры “Прикладная информатика и информационные технологии” Кыргызского национального аграрного университета имени К.И. Скрябина. Структура пособия соответствует традиционной схеме создания: введение, теоретическая часть (лекции, презентации), практическая часть, проверочные (тестовые) разделы, заключение, глоссарий.

Начальная страничка этого электронного учебника представлена в рисунке 2.

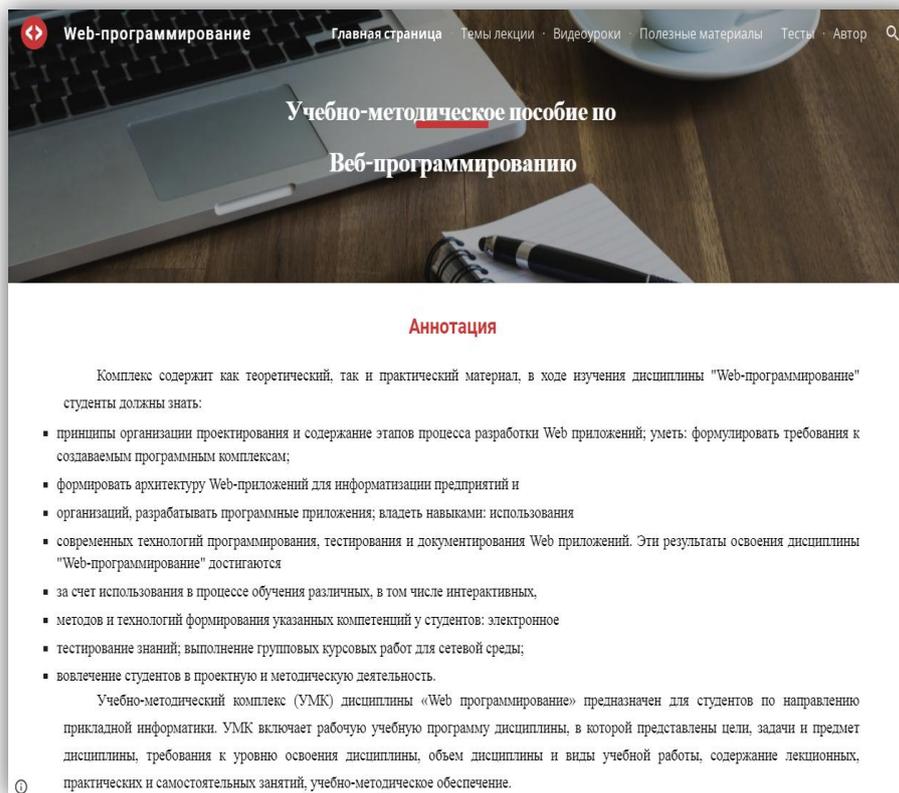


Рисунок 2 – Главная страница электронного пособия

В основной страничке располагается меню с целью перехода к использованному материалу в варианте кнопок (рис. 2). В меню содержатся ссылки: «Раздел «Об электронном учебнике»», «Лекции», «Комплект презентаций для сопровождения занятий», «Практические работы», «Тест», «Глоссарий».

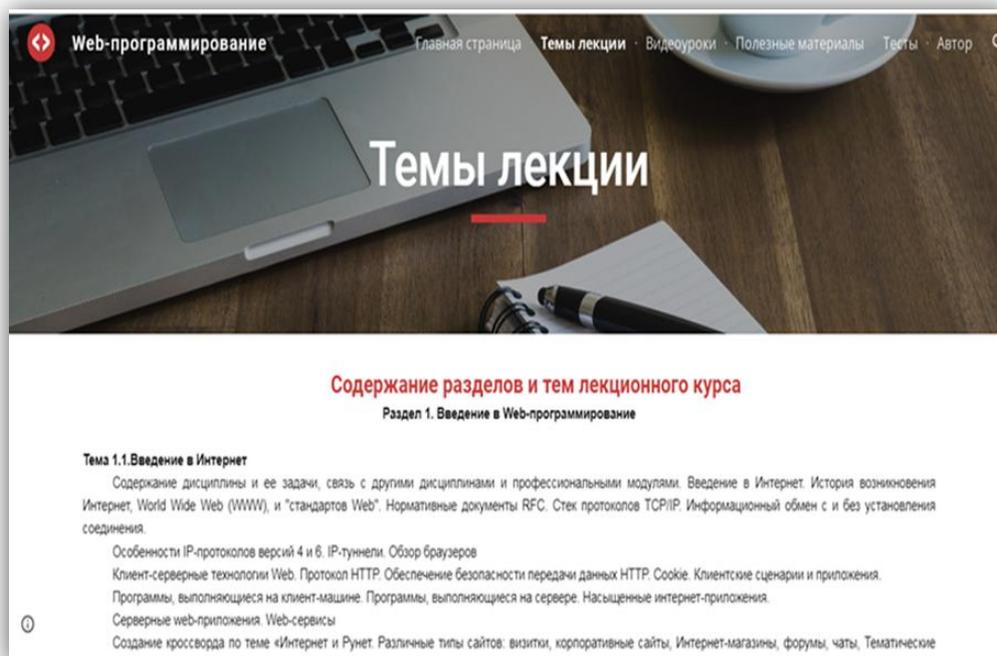


Рисунок 3 – Страница “Тематика лекции”

Далее можем помещать информацию в основную страничку.

Также можем дополнить или документ, или картинку, встроить ссылочку в сторонний сайт, использовать любые файлы из Интернета, с собственного Google-диска либо с компьютера. Можно добавить в собственный сайт видеоматериал, больше увлекательных гиперссылок, создавая привлекательный дизайн.

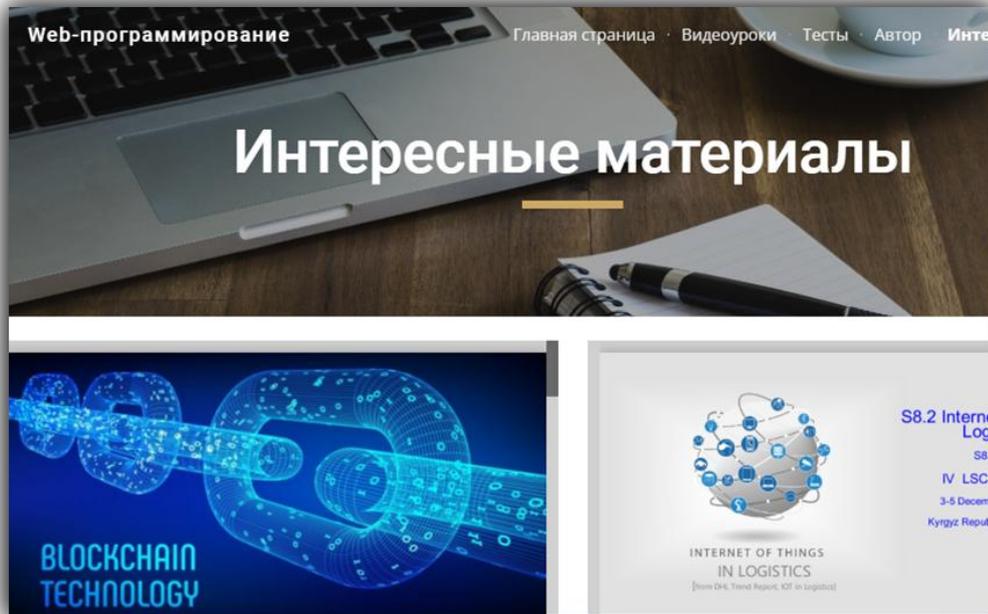


Рисунок 4 – Материалы электронного пособия

Создадим небольшой тест по подготовке. Табличка с ответами создается автоматически в той же папке в Google-диске, где находится тест.

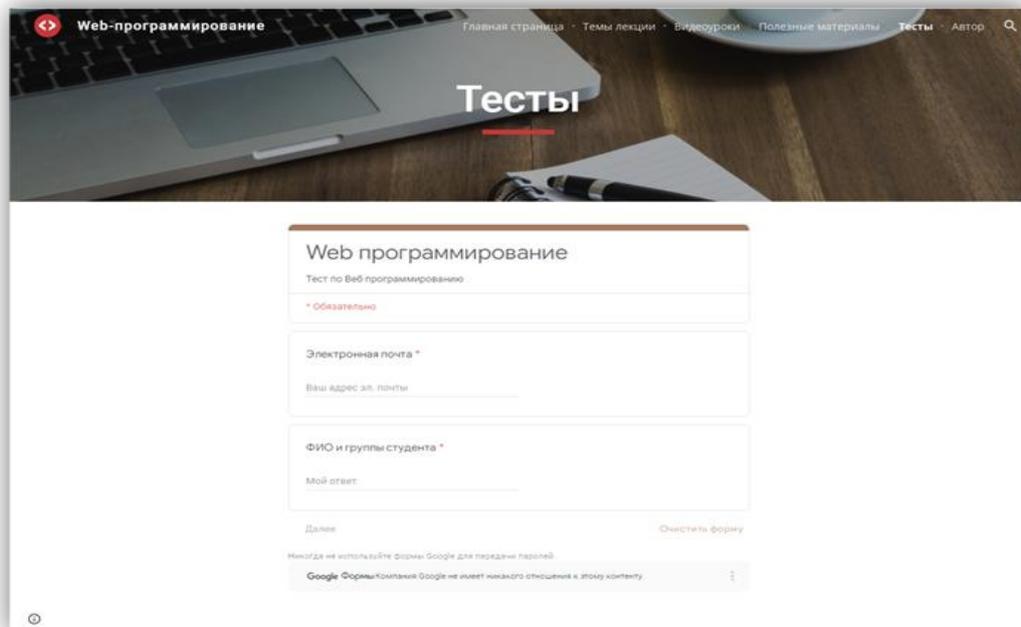


Рисунок 5– Страница «Тестовые вопросы по контролю знаний»

Заключение

Подводя итоги, можно отметить, что создание, использование сайта, блога, диска и календаря Google позволяют создать значительный электронный интерактивный и мультимедийный образовательный ресурс, использование которого возможно как в формате гибридного, так и смешанного преподавания.

Практическая значимость статьи заключается в том, что сформированы методические рекомендации для разработки и внедрения платформы Google Sites в образовательный процесс. Представленный материал может быть применен преподавателями по всем дисциплинам в ходе обучения.

Литература:

1. Спецкурс «Технологии создания электронных учебных пособий» – как средство обучения разработке электронных учебных изданий /Акимова, И.В. Губанова, О.М., Леонова, Т.Ю., Титова, Н.В. // Современные проблемы науки и образования. – 2019. – № 4.
2. Панкратова, О.П. Использование электронных учебных пособий в учреждениях профессионального образования — [Электронный ресурс] — <http://www.moluch.ru/conf/ped/archive/72/4050/>
3. Пискунова, А.И. История педагогики и образования. От зарождения воспитания в первобытном обществе до конца XX в./ Учебное пособие для педагогических учебных заведений. – М.: Эфесс, 2007. — 496 с.
4. Омурзакова, Ч.Ш. Повышение качества образования через ИТ. [Текст] / Ч.Ш. Омурзакова // Социально-экономическое, социально-политическое и социокультурное развитие регионов. Vedeckovydavatelскеentrum«Sociosfera-CZ». – Прага, 2014. – С. 130–135.
5. Омурзакова Ч.Ш. Научно-методические основы формирования социально-коммуникативной компетентности учащихся с применением информационных технологий / Ч.Ш. Омурзакова // Материалы междунаучной конференции «Научно-инновационные технологии: идеи, исследования и разработки». 60-й Юбилейный научный форум молодежи в КГТУ имени И. Раззакова . – Бишкек, 2018. – № 1. – С. 56–62.
6. Омурзакова, Ч.Ш. Уровни социально-коммуникативной компетентности учащихся [Текст] / Ч.Ш. Омурзакова, С.К. Калдыбаев // Международный журнал экспериментального образования. – Москва, 2016. – №5. Часть 2. – С. 20–24.
7. <https://www.google.ru> – сеть поисковых систем и сервис Google Drive;
8. http://nsportal.ru/sites/default/files/2015/02/18/sozdanie_akkaunta_v_google.pdf – инструкция обучающимся по созданию аккаунта в Google (авторская разработка) в локале
9. http://nsportal.ru/sites/default/files/2015/02/19/rabota_s_servisom_google_drive.pdf – методическая разработка Т.В. Сиволовой «Создание документов в сервисе Google Drive».
10. Ермолаева, М.Г. Современный урок: анализ, тенденции, возможности. [Текст]: Учеб.-метод. пособ. / М.Г. Ермолаева. – СПб.: КАРО, 2011. – 160 с.
11. Сероштанова И.Г. Создание совместных учебных объектов с помощью Google-сервисов. [Электронный ресурс] / И.Г. Сероштанова // Режим доступа: <https://docs.google.com/document/d/19TRX6kZw6Zl12LVMLFT5QPSRTMtdw-Jm1fBXN2mS86I/edit#!>

12. Google Apps Education Edition [Электронный ресурс] // Режим доступа: <http://www.google.com/a/help/intl/en/edu/index.html>
13. Шаповалова-Спиридонова, М.В. Использование Google презентаций как инструмента для создания успешных групповых творческих проектов учащихся. [Электронный ресурс] / М.В. Шаповалова-Спиридонова // Режим доступа: <http://iyazyki.ru/2014/04/presentations-googletool/>
14. Рождественская, Л.В. Google Docs для создания коллективной презентации. [Электронный ресурс] / Л.В. Рождественская // Режим доступа: <https://edugalaxy.intel.ru/index.php?automodule=blog&blogid=8&showentry=910>
15. Сервисы Google. [Электронный ресурс] // Режим доступа: <http://www.google.com>.
16. Сотрудничество в среде Google [Электронный ресурс] / Веб-сайт. Режим доступа: <http://sites.google.com/a/pednn.ru/ged/Home>.
17. Корякин, С.В. Современные тенденции развития систем информационной безопасности. [Текст] / С.В. Корякин // Проблемы автоматике и управления: Научно-технический журнал. – Бишкек: Илим, 2017. – №2(33). – С. 82–92.
18. Гайдамако В.В. Инфраструктура Sensor-Cloud –облачные информационно-измерительные системы // Проблемы автоматике и управления. – 2019. – №1(36). – С. 131–140.

УДК 681.518.3:651.011.42:378.14

*Аскарбеков Р.Н., askarbekovu@gmail.com**Орускулов Т.Р.,**Раззаков М.И. razzakoff@mail.ru**Кыргызский государственный технический университет им. И.Раззакова*

ИНФОРМАЦИОННЫЕ ТЕХНОЛОГИИ В ДЕЛОПРОИЗВОДСТВЕ УЧЕБНОГО ЗАВЕДЕНИЯ

Эффективное управление любым учебным заведением требует большого количества информации, которая должна неким образом фиксироваться, обрабатываться и управляться. Необработанные данные или информация в различных формах постоянно поступают из разных уголков, и образовательные учреждения используют автоматизированные инструменты для их обработки, помимо ручных или устаревших методов. В нынешнюю цифровую эпоху университетские сообщества сильно зависят от свободного доступа к информации. В основе этой статьи лежит наблюдение о том, что в университетах имеется некоторая отсталость менеджмента на основе современных информационных технологий для обработки данных. После пандемии 2020 года стало понятно, что информационные технологии в высших учебных заведениях (вуз) должны развиваться и отвечать требованиям времени и ожиданиям студентов. Уже сейчас многие вузы нашей республики испытывают затруднения по заполнению данных. Все виды отчетности и дополнительная информация обрабатываются преподавателями, что вызывает чрезмерную загруженность. Появилась острая необходимость эффективного менеджмента информацией внутри вуза и избавление преподавателей от нагрузки, связанной с заполнением и оформлением бумаг и дополнительных отчетностей.

Ключевые слова: информационные технологии, управление и администрирование вузов, блокчейн, высшее образование, бумажная работа.

Введение

Сегодня многие вузы превратились в место по развитию бюрократии, где процветают администраторы. Проблема в том, что профессорско-преподавательскому составу делегируется избыточная и часто бесполезная бумажная работа. Заполнение ящиков и написание десятков отчетов, и всё это, кажется, служит для того, чтобы доказать администраторам, что преподаватели знают, как выполнять свою работу, и в то же время продемонстрировать, что административная масса незаменима.

Следовательно, отношения между преподавателями и администраторами в ряде университетов сегодня полностью изменились, и не в лучшую сторону. Многие читатели в этот момент задумаются о крайне печальных случаях отчаяния преподавателей, вызванных чрезмерной бумажной работой, невозможностью выполнить требования по продвижению по службе и атмосферой неуважения, если не сказать больше. Мы воздаем должное им и их семьям.

Уже сегодня ученые, имеющие опыт исследований, преподавания и администрирования, предлагают ряд изменений в управлении вузами. Появляющиеся технологии и связанные с ними методы и приемы, а также приложения открывают множество возможностей для того, чтобы сделать процесс управления и администрирования вузов более эффективным, прозрачным и совместным [1, 2]. Технологии, конечно, не панацея. В статье предлагаются вашему вниманию некоторые методы и модели задач для улучшения управления вузом.

Новые технологии и высшее образование

За последние несколько десятилетий система высшего образования (ВО) претерпела кардинальные изменения, что вызвало нагрузку на высшие учебные заведения (вузы) во всем мире. Если бы нам нужно было определить единственную причину этих событий, мы должны были бы указать на негласный, но все же происходящий процесс перехода к

новой модели роста и развития во всем мире и вытекающий из этого поиск новых моделей роста в регионах, на национальном и местном уровнях. Этот процесс вызывает ряд опасений, ограничений и соображений, которые в контексте ВО и вузов выражаются в сокращении государственных ресурсов и возможностей финансирования государственного высшего образования, вытекающих из этого требований к большей самодостаточности вузов и необходимости адаптации вузов к изменяющимся требованиям на рынке труда, возросшей мобильности студентов и преподавателей.

Поскольку эти процессы обуславливают изменения в обществе, включая образ жизни и модели воспитания детей, мировоззрение, образ мышления, устремления, мотивацию, дополнительным вызовом, с которым приходится сталкиваться вузам, являются удовлетворение ожиданий нового поколения студентов и преподавателей, способы общения, обучения, преподавания, приобретения навыков. На этом фоне необходимо переосмыслить вопрос управления и администрирования вуза сегодня. Достижения в области информационных и коммуникационных технологий (ИКТ) добавляют дополнительный слой к этой и без того сложной реальности. Однако, как следует из этого тома, новые технологии, такие, как искусственный интеллект (ИИ) и когнитивные вычисления (КВ), дополненная реальность (ДР), виртуальная реальность (VR) и смешанная реальность (СР), Интернет вещей (ИВ) и блокчейн; несут обещание эффективной помощи администраторам вузов в навигации их учреждений в период глубоких изменений.

В связи с этим необходимо решить три вопроса: кто участвует в управлении и администрировании вуза? О каких технологиях идет речь и что они могут предложить? Что поставлено на карту? Цель этой данной статьи — ответить на эти вопросы.

Управление и администрирование вуза: кто участвует?

Типичный разговор об управлении и администрировании вузов колеблется вокруг таких вопросов, как занимать и удерживать студентов, как привлекать и поддерживать преподавателей, как способствовать качеству образования, создавать бренд и сохранять хорошее имя на рынке образовательных услуг, как создать имидж университета в национальных и международных рейтингах университетов, как быть привлекательным для студентов, предлагая учебные программы, которые соответствуют спросу на рынке труда и потребностям общества в целом, и как обеспечить финансовую устойчивость учебного заведения.

Управление и администрирование вузов — это область, которая требует участия различных заинтересованных сторон. В их число входят не только администраторы, такие как президенты, вице-президенты, проректоры, группы управления персоналом, офис-ассистенты, но и, конечно же, деканы, заведующие кафедрами и даже студенты [5].

Одной из таких областей является прозрачность и распространение информации внутри данного вуза. Этот вопрос имеет много аспектов. То есть это может касаться таких простых вопросов, как информирование в режиме реального времени командой по проведению мероприятий других офисов и преподавателей о возможных событиях и недоступности помещений. Это также может относиться к предоставлению равного доступа к информации о мероприятиях, приглашениям на местные конференции, стипендиям и другим предметам, которые преподаватели считают привилегиями. Слишком часто информация такого рода не разглашается, что лишает равный доступ к некоторым группам профессорско-преподавательского состава, тем самым вызывает негодование тех, кто не имеет доступа к информации. Очевидно, что это проблема, затрагивающая функцию управления человеческими ресурсами, которая в вузе, возможно, больше, чем в любой другой организации, требует особенно умелого управления талантами. С этой точки зрения достижения в области ИКТ обещают обойти несколько проблем, которые проявляются в современных моделях управления и администрирования. Следующий раздел проливает свет на этот вопрос.

Какие технологии и с какой целью?

В контексте поворота к большим данным и квантовых вычислений несколько технологий должны изменить лицо высшего образования и, в частности, администрирования и управления вузами. Эти технологии включают искусственный интеллект (ИИ) и когнитивные вычисления (КВ); технологии: дополненная реальность (ДР), виртуальная реальность (ВР) и смешанная реальность (СР); Интернет вещей (ИВ) [2] и блокчейн. Эти технологии и связанные с ними методы (такие, как интеллектуальный анализ данных, обработка естественного языка, машинное обучение, исследование, анализ пожеланий и компьютерная лингвистика) и приложения (такие, как социальные сети или облачные вычисления) открывают ряд возможностей для разнообразных заинтересованных сторон, участвующих в создании высшего образования, чтобы более эффективно, чем когда-либо, повлиять на вуз.

Способность – обращенная к большим данным – управлять огромными наборами данных и осмысливать их означает, что наши аналитические и предписывающие возможности существенно расширились за последние годы. Последствия этого имеют основополагающее значение для изучения, понимания и использования проблем и возможностей, присущих внешнему и внутреннему контексту вуза [10]. Это может включать: анализ демографических тенденций в сочетании с вероятными потоками прибывающих студентов, анализ пожеланий и интересов и предпочтений студентов в отношении выбора их основных и дополнительных специальностей, анализ тенденций и тенденций в отношении предпочтений в отношении обучения за рубежом, анализ макроэкономических изменений и формирующиеся в результате модели деловой активности в сочетании с перспективными требованиями рынка труда. Очевидно, что детальное понимание этих вопросов позволит администраторам вузов управлять вузами с большой точностью, используя возможности и предупреждая возможные риски. Хорошая новость заключается в том, что методы работы с большими данными позволяют нам интерпретировать огромные наборы данных и извлекать информацию по целому ряду вопросов, отмеченных выше. Проблема остается в том, что сам процесс сбора и анализа данных, как правило, является дорогостоящим, и, следовательно, очень немногие вузы могут себе это позволить.

Что касается внутреннего контекста вуза, использование больших данных создает возможность получить четкое и подробное представление об управлении студенческим потоком, расписании занятий, экзаменов, специальных учебных активностей, мероприятий, встреч и т.д. Учитывая, что современные тенденции в образовании способствуют постоянному более индивидуальному опыту студентов, разнообразие вариантов, которые администраторы вуза должны учитывать и управлять, выходит за рамки когнитивных способностей человека. В результате некоторые ресурсы, имеющиеся в вузах, часто бывают бездействующими, тогда как существующие синергии остаются неиспользованными.

Рассмотрим все те случаи, когда менеджеры высшего звена не знают о талантливых способностях (включая исследования, достижения, интересы, работу и карьерные мотивы), присущих профессорско-преподавательскому составу. В ряде случаев неоправданное невежество заставляет их искать другие конкретные навыки за пределами вуза, что увеличивает стоимость содержания вуза и снижает работоспособность преподавателей. Такие примеров очень много. Суть аргумента заключается в том, что никогда раньше администраторы вузов не имели доступа к такому количеству инструментов, облегчающих их повседневную работу, как сегодня. Искусственный интеллект и когнитивные вычисления, могут в этом контексте служить мощным подспорьем для администраторов, соответственно либо будут и дальше вести свою работу на основе принятия некоторых стандартизированных решений или будут обеспечены всей информацией, необходимой для принятия обоснованного решения по данному вопросу. Пока рано делать выводы о том, используют ли вузы эту возможность; скорее, для вузов это остается проблемой.

Технология блокчейн — еще один пример того, как новые ИКТ революционизируют. Исследования в области блокчейна процветают, а его приложения распространяются [6, 9]. В контексте вузов блокчейн можно использовать различными способами, в том числе для хранения записей о студентах и предоставления к ним доступа для различных заинтересованных сторон, имеющих на это разрешение.

Это касается не только внутренних сторон, но и третьих лиц, например, вузов, в которые поступают студенты (по мобильности). С другой стороны, благодаря поддержке согласованных протоколов хранения, соблюдению правил и законов о конфиденциальности и защите конфиденциальных данных собранные данные таким образом могут быть доступными и пригодными для крупномасштабных аналитических целей. В меньшем масштабе, очевидно, путем создания индивидуальных электронных портфолио каждого студента, их записи будут неизменяемыми, что облегчит процесс поступления студентов в другие университеты, на работу и т.д. Это также ограничит масштабы лжесвидетельства, когда речь идет об академической деятельности [16, 17].

Опять же, можно было бы умножить примеры из самых разных областей. Например, рассмотрите возможность использования блокчейна для расширения библиотечных услуг, которые вузы предлагают своим студентам и преподавателям. И, наоборот, подумайте, как блокчейн может помочь в работе деканатов.

Любой, кто когда-либо видел количество файлов, проходящих через руки людей, непосредственно вовлеченных в этот процесс, понимает, какое благо было бы, если бы данные можно было извлекать прямо из системы, систематизировать, форматировать для того, чтобы администраторы могли готовые использовать или принимать решение. Как и в случае с библиотеками, блокчейн предлагает возможность создания обучающих платформ, таким образом позволяя удаленно взаимодействовать преподавателям и студентам.

Управление и администрация вуза и новые технологии: что поставлено на карту?

Обсуждая перспективы использования потенциала новых технологий для повышения эффективности управления и администрирования вуза, необходимо подчеркнуть, что технология, отделенная от целенаправленной деятельности использующего ее индивидуума, обречена на бесполезность. Кроме того, применение технологий в отрыве от более широкого социального контекста, в который они должны вносить свой вклад, нецелесообразно. Ожидается, что в контексте администрирования и управления ВО новые технологии, как подчеркивается в этой работе, повысят общую эффективность вузов, что включает в себя множество вопросов, начиная с разработки учебных программ и заканчивая управлением объектами. Тем не менее это также подразумевает более пристальное внимание к обеспечению профессионального развития, благополучия, участия, вовлеченности и приверженности преподавателей [13, 15, 19].

Короче говоря, новые технологии несут в себе огромные перспективы и потенциал, и, если их использовать наиболее приемлемым образом, всегда в соответствии с верховенством закона и с фундаментальными ценностями и нормами, они могут существенно повысить устойчивость вузов. Только таким образом вузы могут служить обществу в целом.

Как вузы могут формировать предприимчивое поведение

Только в Европейском союзе (ЕС) в настоящее время 96,6 млн человек подвержены риску бедности и социальной изоляции [12]. Решение этой проблемы, связанной с социальной ситуацией, требует новаторских подходов, и было высказано предположение, что инклюзивное предпринимательство может быть частью решения. Тем не менее многие непредставленные группы (с точки зрения предпринимательской активности) сталкиваются со значительными препятствиями на пути к предпринимательству.

Это исследование показывает, как вузы могут использовать свои междисциплинарные знания и опыт в партнерстве с правительством, промышленностью и гражданским обществом для решения экономических и социальных проблем в недостаточно представленных сообществах путем повышения уровня предприимчивого поведения. Новые исследования в литературе продемонстрировали, как некоторые вузы предоставляют индивидуальную и целостную поддержку предприятий непредставленным группам в своих сообществах. Однако такие инициативы не распространены, и мало исследований о том, как другие вузы могут воспроизвести инициативы. Посредством представления концептуальной модели в этой тематике исследований нужно рассмотреть, как вузы могут выйти за рамки своей формальной образовательной среды и динамично поддерживать развитие предприимчивых компетенций и моделей поведения среди людей в своих местных сообществах.

Выводы выделяют шесть ключевых областей для рассмотрения в таких разработках, включая: (1) преподавание и обучение; (2) ресурсы; (3) инфраструктуру; (4) междисциплинарные подходы; (5) заинтересованные стороны и (6) культуру. Здесь необходимо по-новому посмотреть на эффективное взаимодействие с сообществом вузов и предположить, что вовлечение сообщества вузов посредством обучения предпринимательству является новым способом повышения ценности как для недостаточно представленных сообществ, так и для вузов.

Успех в высшем образовании: создание новой нормы посредством устойчивости

В современной конкурентной среде отрасль ВО испытывает фискальный, организационный и конкурентный стресс. Как преподаватели и руководители высшего образования считают, что роль вузов в обществе очень важна, потому что коллективная работа по своей сути касается обучения студентов, исследований и служения своему университету и обществу в целом.

Тем не менее с сегодняшним конкурентным рынком высшего образования университеты и колледжи требуют от преподавателей, персонала и администраторов «выполнить работу», создавая умственный, физический, эмоциональный и духовный стресс. Необходимо углубиться в исследования об идеях и методах, призванных помочь работникам университетов и колледжей стать устойчивыми перед лицом этого набора экстремальных проблем. В частности, в центре внимания находятся ее четыре аспекта, включая физическую, эмоциональную, духовную и умственную устойчивость [17, 18, 20].

Утверждается, что, инвестируя в собственную устойчивость как лидеров, мы способствуем устойчивости наших коллег, команд и учреждений, чтобы они могли процветать в сегодняшней среде высшего образования, несмотря на ее постоянные проблемы. Предлагается создать новый подход к воспитанию и возвращению новых талантов среди коллектива вуза. Увеличение работы над кадровым составом вуза. Предлагать несколько моделей повышения квалификации для сотрудников вуза, создать специализированный курс для преподавателей, заведующих кафедрой и деканов. Для каждого из них предлагать свою модель обучения, где есть необходимые уровни владения методами управления информацией, составления конкурентоспособных учебных планов, новый подход к воспитанию студентов.

Управление вовлеченностью студентов в высшие учебные заведения

Управления некоторыми аспектами вуза с помощью новых технологий. Мобильное обучение и приложения в сфере образования — это что-то новое и интересное. Однако это не означает, что факультет менее актуален. Напротив, профессорско-преподавательский состав всегда является важным фактором, который необходим для повышения вовлеченности студентов. Беспризорные учащиеся на самом деле более проблематичны, чем учащиеся с низкой успеваемостью. Это происходит из-за того, что невнимательный ученик приносит беспорядок в класс. Таким образом, повышение вовлеченности

студентов сегодня является ключевой задачей в высшем образовании. Исследования показывают, что активное обучение, ценность и ожидания являются важными факторами, определяющими вовлеченность учащихся. На некоторые из этих факторов могут косвенно влиять преподаватели и разработчики курсов. На этом фоне целью является — изучить, можно ли решить проблему снижения вовлеченности студентов в ВО с помощью проекта CURP, то есть приложения CURriculum and Course Planning (CURP). CURP является результатом усилий в Утрехтском университете по разработке курсов и учебных планов с использованием карточек для учета различных видов деятельности, стратегическая цель которого — стимулирование участия студентов в высшем образовании. Результат эксперимента — приоритетный список требований, которые могут быть реализованы с течением времени, структурированный в соответствии с моделью вовлечения студентов.

Вывод

В заключение хотелось бы отметить, что основные результаты обсуждения, представленные в этой статье, указывают на программу предстоящих исследований. Таким образом, подтверждают убежденность в том, что качество образования является функцией устойчивости системы высшего образования, которая включает в себя должное внимание ко всем заинтересованным сторонам, в том числе и преподавателей. Также присутствуют доводы в пользу того, что популярная в некоторых вузах ориентация на студентов и их потребности часто реализуются за счет преподавателей.

Поскольку профессорско-преподавательский состав является основой вуза, чрезмерный акцент на потребностях студентов без должного учета потребностей профессорско-преподавательского состава, включая профессиональное развитие, благополучие и даже уважение, обязательно подорвет устойчивость и конкурентную способность вуза.

Литература

1. Brennan, J. Higher education and social change. / Brennan, J. // Higher Education, 2008. – 56(3). – P. 381–393.
2. Bayani Abbasy, M. Predictable influence of IoT (internet of things) in the higher education. / Bayani Abbasy, M., & Vilchez Quesada, E. // International Journal of Information and Education Technology, 2017. –7(21). – P. 914–920.
3. Delgado-Marquez, B. Being highly internationalized strengthens your reputation: An empirical investigation of top higher education institutions. / Delgado-Marquez, B., Escudero-Torres, M., & Hurtado-Torres, N. // Higher Education, 2013. – 66(5). – P. 619–633.
4. Garrison, R. Blended learning: Uncovering its transformative potential in higher education. / Garrison, R., & Kanuka, H. // The Internet and Higher Education, 2004. –7(2). – P. 95–105.
5. Ginsberg, B. The fall of the faculty: The rise of the all-administrative university and why it matters. Oxford: Oxford University Press, 2011. – 200 p.
6. Grech, A., & Camilleri, A. (2017). Blockchain in education. JRC Science for Policy Report. Luxembourg: Publications office of the European Union.
7. Султаналиева, Р.М. Перспективы сотрудничества в научной сфере университетов Кыргызской Республики и стран ШОС. / Р.М. Султаналиева, Р.Н. Аскарбеков // Материалы международной научно-практической конференции «Перспективы интеграции университетов ШОС и АТР в науке и образовании». –Хабаровский государственный университет экономики и права. – 2017. – С. 85–91.
8. Демиденко, А.П. Использование нечетких множеств в задаче оценки качества образования. / А.П. Демиденко, М.В. Кан, Д.Г. Сундеев // Проблемы автоматизации и управления. – 2012. – №2 (23). – С. 100–109.

9. Исроилов, С.Г. Разработка защищенной системы электронного документооборота на основе блокчейн-технологии. / С.Г. Исроилов, С.Н. Верзунов // Проблемы автоматизации и управления. – 2021. – №2 (41). – С. 61–76.
10. Торобеков, Б.Т. Организация образовательной деятельности вуза на основе информационной системы. / Б.Т. Торобеков, Н.Ж. Саитов // Известия КГТУ им. И. Раззакова. – 2010. – №19. – С.65–70.
11. Торобеков, Б.Т. Развитие информационной системы в управлении вузом. / Б.Т. Торобеков // «Наука, техника и образование». – 2017. – №4(34). – С.34–36.
12. The missing entrepreneurs 2017: Policies for inclusive entrepreneurship. Paris: OECD Publishing. <https://www.european-microfinance.org/publication/missing-entrepreneurs-2017-policies-inclusive-entrepreneurship>
13. Мамедова, Г.А. Современные технологии электронного образования. / Г.А. Мамедова, Ф.Т. Агаев // Открытое образование. – 2017. – №3. – С.73–79. <https://doi.org/10.21686/1818-4243-2017-3-73-79>
14. Online Report Card – Tracking Online Education in the United State. 2015. URL: <https://onlinelearningconsortium.org>
15. Сатунина, А.Е. Электронное обучение: плюсы и минусы / А.Е. Сатунина // Современные проблемы науки и образования. – 2006. – No. 1. – С. 89–90.
16. Harrist, M. New Technology Bridges Oracle, Hadoop, and NOSQL Data Stores. URL: <http://www.oracle.com/us/corporate/features/big-datasql/index.html>
17. Picciano, A.G. The evolution of big data and learning analytics in American higher education / A.G. Picciano // Journal of Asynchronous Learning Networks. 2012. – Vol. 16. – No. 3. –P. 9–20.
18. Yefim Kats. Learning management system technologies and software solutions for online teaching: Tools and application. / Yefim Kats // Ellis University & Rivier college. – 2010. –461 P.
19. Алексеева, А.Н. Системы управления проектами как средство развития навыков управления временем студентов вузов / Н.В. Кравченко, Л.В. Горбатюк, Е.А. Мурзина // Управление в социальных и экономических системах. – 2019. – №28 – С. 91–93.
20. Перминова, О.М. Управление конкурентоспособностью государственного вуза на основе формирования его инновационно-образовательного потенциала / О.М. Перминова // Ижевский государственный технический университет им. М.Т. Калашникова. Ижевск, 2019. – 184с.

УДК 681.3

*Великанова Л.И., Лыченко Н.М., nlychenko@mail.ru**Институт машиноведения и автоматизации НАН КР, Бишкек, Кыргызстан*

МОДЕЛИРОВАНИЕ КРАТКОСРОЧНОГО ПРОГНОЗА ЗАГРЯЗНЕННОСТИ ВОЗДУХА ТВЕРДЫМИ ЧАСТИЦАМИ НА ОСНОВЕ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ С УЧЕТОМ ФАКТОРА ЗАГРЯЗНЕНИЯ

Представлены результаты моделирования краткосрочного прогноза концентраций твердых частиц PM_{2.5} и индекса качества воздуха г. Бишкек с учетом метеорологических факторов и загрязняющего фактора (количества тонн угля, ежедневно сжигаемого на городской ТЭЦ). Модели прогноза построены на основе обобщенно-регрессионной нейронной сети и искусственной нейронной сети прямого распространения с использованием алгоритма обратного распространения ошибки на различных обучающих выборках. Приведен сравнительный анализ моделей. Показано, что учет фактора загрязнения позволил снизить ошибку прогнозирования на 2-7 %.

Ключевые слова: искусственные нейронные сети GRNN и FFBN, обучающая выборка, концентрация твердых частиц PM_{2.5}, индекс качества воздуха, метеофакторы, загрязняющий фактор, ошибка прогнозирования.

Введение

Проблема загрязнения атмосферного воздуха г. Бишкек по-прежнему остается актуальной. Также актуальными остаются исследования процессов загрязнения и связанных с ними причинных факторов. Одними из важных факторов, оказывающих влияние на экологическое состояние атмосферного воздуха, являются характерные для соответствующей местности метеофакторы. В литературе опубликовано большое количество исследований на эту тему, например [1-4]. В [5] на основе регрессионного анализа исследованы взаимосвязи между загрязненностью атмосферного воздуха г. Бишкек и сопутствующими ей климатическими условиями в различные периоды года, выявлены положительные и отрицательные корреляции. Учитывая это, а также имея доступ к открытой информации о метеопараметрах [6], концентрациях твердых частиц PM_{2.5} и индексе качества воздуха AQI [7], были разработаны различные модели прогноза уровня загрязненности воздуха на основе искусственных нейронных сетей (ИНС). В [8] обоснована методология краткосрочного прогноза концентрации PM_{2.5} в зависимости от значений PM_{2.5}, измеренных в сроки, предшествующие сроку прогноза и от метеопараметров, измеряемых каждые 3 часа: T_d – температура точки росы; T – температура воздуха; RH – относительная влажность воздуха; Ws – скорость ветра в м/с; Pr – интенсивность осадков (1- ливень или сильный дождь/снег, 0.75- дождь/снег или слабый ливневый дождь, 0.5- слабый дождь/снег, 0.25 – морось).

Основная особенность методологии заключается в декомпозиции значений измеряемых параметров по сезонам (зима-отопительный/ лето-неотопительный) и по срокам измерений: 0 часов, 3 часа, 6 часов, 9 часов, 12 часов, 15 часов, 18 часов 21 час с последующим построением моделей ИНС для каждого срока измерений. При этом для каждой модели были определены наиболее эффективные, с позиции средних ошибок прогноза, варианты входных векторов сети, учитывающих различную историю метеопараметров (на 3 часа, 6 часов и т.д. ранее). Как показано в [8], идея декомпозиции данных концентраций PM_{2.5} и метеопараметров по срокам измерения оправдала себя, поскольку позволила снизить ошибку прогноза.

Благодаря информации, предоставленной ОАО «Электрические станции» КР, удалось выявить существенные корреляции между уровнем загрязнения атмосферного воздуха города и количеством тонн угля, ежедневно потребляемого ТЭЦ г. Бишкек. Как показано в [9], учет этого фактора загрязнения, как одного из источников загрязнения атмосферного воздуха города, позволил повысить точность прогнозирования класса AQI

из четырех интегрированных классов ($AQI \leq 50$ / $50 < AQI \leq 100$ / $100 < AQI \leq 150$ / $AQI > 150$) до 80%.

В настоящей статье представлено развитие методологии для краткосрочного прогноза PM2.5 и AQI с учетом новых метеоданных, а также данных о ежедневно потребляемом ТЭЦ г. Бишкек количестве тонн угля и его типе (по месту добычи). Используются модели, построенные на основе двух типов ИНС: на основе нейронной сети с прямым распространением сигнала и обратным распространением ошибки FFNN [10] и на основе обобщенно-регрессионной нейронной сети GRNN [11].

Анализ данных загрязнения атмосферного воздуха г. Бишкек

На рисунке 1 представлены временные ряды изменения содержания PM2.5 в атмосферном воздухе и индекса качества воздуха (AQI) г. Бишкек за период с 09.02.2019 по 30.09.2022 с интервалом в 3 часа [7]. На них так же, как и в предыдущих работах авторов (например, в [8, 9]), отмечается явная сезонная составляющая, связанная с началом/ концом отопительных сезонов в городе.

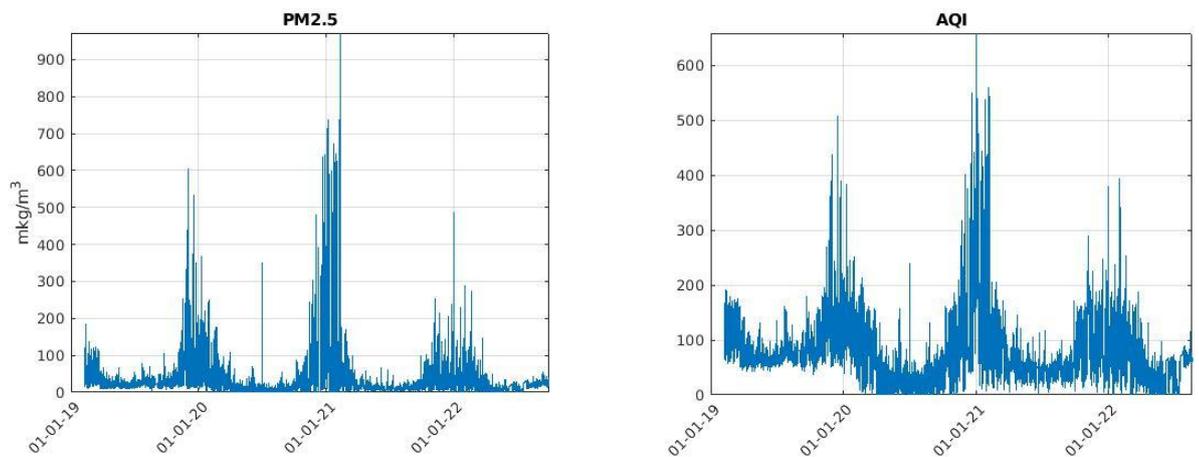


Рисунок 1 – Значения концентраций PM2.5 и индекса качества воздуха AQI в воздухе г. Бишкек за период 09.02.2019 –30.09.2022.

Полученные автокорреляционная и частичная автокорреляционная функции так же, как и в [9], указывают на нестационарность представленных временных рядов и на их периодичность в 24 часа, что обуславливается повторением суточной истории в процессах загрязнения: в утренние часы, как правило, уровень загрязнения ниже, чем в вечерние, что, в основном, объясняется физико-географическими и климатическими условиями города.

В таблице 1 представлены статистические характеристики временных рядов PM2.5 и AQI за периоды, соответствующие началу и концу отопительных сезонов в городе: среднее (математическое ожидание) *mean* и стандартное (среднеквадратическое) отклонение σ .

Таблица 1 – Средние значения и стандартные отклонения концентраций PM2.5 и AQI

Период наблюдений	PM2.5		AQI	
	<i>mean</i> , mkg/m ³	σ , mkg/m ³	<i>Mean</i>	Σ
09.02.2019-23.03.2019	44.0	23.8	115.1	32.9
24.03.2019-31.10.2019	22.3	10.4	70.0	21.2
01.11.2019-31.03.2020	40.7	122.0	102.2	128.9
01.04.2020-09.10.2020	6.9	8.4	26.0	23.1
10.10.2020-31.03.2021	76.1	115.2	126.9	97.8
01.04.2021-07.10.2021	12.7	9.7	47.2	22.3
08.10.2021-03.04.2022	39.2	40.5	96.0	52.4
04.04.2022-30.09.2022	12.4	10.4	43.6	30.0

Стоит отметить, что средние значения зимних (отопительных) периодов 2019, 2019-2020, 2021-2022 годов приблизительно одинаковые, однако среднеквадратические отклонения в эти периоды сильно отличаются. Самый «грязный» за всю историю наблюдений период – отопительный период 2020-2021 годов. Характеристики летних периодов 2021, 2022 годов очень близкие, чего нельзя сказать о соответствующих периодах 2019 и 2020 годов. Различия в статистических оценках загрязненности воздуха в одинаковые периоды объясняются, прежде всего – различиями в характеристиках факторов загрязнения и – в меньшей степени – различиями в характеристиках метеофакторов. Так, например, снижение загрязненности воздуха весной и летом 2020 года, как отмечено в [9], связано в большой степени со снижением производственной и транспортной нагрузки в городе в связи с объявленным карантином по случаю пандемии. Также в [9] показано, что отопительный период 2020-2021 годов характеризуется более низкими среднесуточными температурами воздуха и дополнительными факторами, имеющими отношение к источникам загрязнения.

Таким образом, временные ряды наблюдений $PM_{2.5}$ и AQI имеют выраженный нестационарный характер, что соответствует природе процессов загрязнения атмосферного воздуха, являющихся следствием множества явных и неявных факторов. Поэтому построение прогнозов таких процессов – нетривиальная задача.

Методология краткосрочного прогноза концентрации $PM_{2.5}$ и индекса качества воздуха AQI с учетом фактора загрязнения

Один из источников загрязнения атмосферного воздуха г. Бишкек – выбросы городской ТЭЦ, которая в различные периоды работы использует разное количество тонн угля различных типов, определяемых местом добычи. В основном это: Кара-Кече, Шубаркуль (до 16.04.2021г.) и Кара-Жыра (с 19.07.2021г.). То есть, рассматриваемый фактор загрязнения – существенно неоднородный.

Корреляционный анализ среднесуточных концентраций загрязнения воздуха частицами $PM_{2.5}$ и количества тонн угля (*Coal*), ежедневно потребляемого ТЭЦ г. Бишкек, проведенный на выборке 09.02.2019 – 30.09.2022, показал, что так же, как и для выборки 09.02.2019 – 31.03.2021 [9], коэффициент корреляции достаточно высок: $R(Coal/PM_{2.5})=0.49$.

В таблице 2 представлены коэффициенты корреляций $R(Coal/PM_{2.5})$ среднесуточных концентраций загрязнения воздуха частицами $PM_{2.5}$ ($PM_{2.5}$) и количества тонн угля, ежедневно потребляемого городской ТЭЦ (общего угля – *Coal* и по типам (Кара-Кече – $Coal_1$, Кара-Жыра – $Coal_2$, Шубаркуль – $Coal_3$), рассчитанные для различных периодов наблюдений. Анализ коэффициентов корреляции подтверждает, что воздействие фактора загрязнения, связанного с выбросами ТЭЦ в воздух города твердых частиц $PM_{2.5}$, имеет нестационарный характер, обусловленный неоднородностью типа используемого угля. Установленные корреляции могут служить дополнительной исходной информацией для формирования более эффективных моделей прогнозирования загрязнения воздуха на основе ИНС.

Таблица 2 – Коэффициенты корреляций $R(Coal/PM_{2.5})$

Выборка наблюдений	Тип угля			Общий уголь <i>Coal</i>
	<i>Кара-Кече</i> $Coal_1$	<i>Кара-Жыра</i> $Coal_2$	<i>Шубаркуль</i> $Coal_3$	
09.02.2019 – 31.03.2021	0.09	-	0.55	0.56
09.02.2019 – 31.12.2021	0.04	0.54 (с 19.07.2021)	0.55 (до 16.04.2021)	0.53
09.02.2019 – 30.09.2022	0.04	0.52 (с 19.07.2021)	0.55 (до 16.04.2021)	0.49

В соответствии с предложенной в [8] методологией проведена декомпозиция архивных данных измерения $PM_{2.5}$ и AQI [7] и данных метеослужбы RP5 [6] по сезонным периодам и по срокам измерений: $i=0$ часов, 3 часа, 6 часов, 9 часов, 12 часов, 15 часов, 18 часов, 21 час. Для каждого срока и обоих периодов (Зима/Лето) на базе ИНС с

прямым распространением сигнала и обратным распространением ошибки (функция *newff* в среде программирования MatLab) и обобщенно-регрессионной ИНС (функция *newgrnn*) были построены модели для прогноза значений PM2.5 и AQI для двух вариантов набора входных параметров: с использованием данных по расходу сжигаемого угля ТЭЦ г.Бишкека и без использования этих данных. Кроме того, проверялось влияние выбора в качестве входных векторов метеопараметров в предшествующие сроки измерения *i-1, i-2, ...* с наиболее выраженными коэффициентами корреляций с измеренными в срок *i* концентрациями PM2.5 и AQI на точность прогноза [5].

В общей сложности было исследовано по 6 вариантов моделирования PM2.5 и AQI на различных обучающих и тестовых множествах летнего и зимнего периодов и выявлено влияние различных факторов в сроки, соответствующие представленным в таблицах 3-5, на точность прогноза Pm2.5 и AQI в срок *i*.

Число сигналов в таблицах 3 – 5 означает размерность входных векторов при обучении и моделировании ИНС. В данных моделях значения *T* и *RH* усредняются.

Таблица 3 – Контролируемые параметры без учета угля

Pm2.5	Ws	PR	T	RH	Число сигналов
i-1	i-1 i-2	i-1 i-2	i-1 i-2	i-1 i-2	7

Таблица 4 – Контролируемые параметры с учетом общего угля (*Coal*)

Pm2.5	Ws	PR	T	RH	Coal	Число сигналов
i-1	i-1 i-2	i-1 i-2	i-1 i-2	i-1 i-2	i-1	8

Таблица 5 – Контролируемые параметры по срокам прогноза с учетом угля и его типа

Pm2.5	Ws	PR	T	RH	Coal ₁	Coal ₂	Coal ₃	Coal	Число сигналов
i-1	i-1 i-2	i-1 i-2	i-1 i-2	i-1 i-2	i-1	i-1	i-1	i-1	11

Для оценки адекватности моделей и точности прогнозов, построенных на их основе, использованы следующие оценки [12]:

1) Средняя абсолютная процентная ошибка (Mean Absolute Percentage Error, MAPE):

$$MAPE = \frac{1}{N} \sum_{t=1}^N \frac{|e_t|}{y_t} 100\%,$$

где $e_t = (y_t - y_{ft})$ – ошибка прогноза или остатки модели,

y_t – наблюдаемое (измеренное) значение в срок t ,

y_{ft} – значение в срок t , рассчитанное по модели,

n – длина выборки наблюдений.

Это коэффициент, не имеющий размерности, с очень простой интерпретацией: ошибка составила $n\%$ от фактических значений. Из-за деления на фактические значения этот коэффициент чувствителен к масштабу переменных (при y_t , близких к нулю, MAPE стремительно возрастает) и не может быть рассчитан, когда $y_t=0$.

2) Средняя абсолютная ошибка MAE, деленная на Среднее (MAE/Mean Ratio) – альтернатива MAPE, преодолевающая ее проблемы:

$$MAEMR = \frac{1}{N} \sum_{t=1}^N |e_t| / Mean(y_t),$$

где $Mean(y_t)$ – среднее значение y_t .

Это отношение изменяет масштаб ошибки, чтобы сделать ее сопоставимой по временным рядам различных масштабов.

Результаты моделирования

В качестве примеров результатов моделирования краткосрочного прогноза в таблицах 6 – 7 приведены средние ошибки прогноза AQI для всех сроков прогноза за 17 суток тестовых периодов. В таблице 6 обучающее множество - интегрированный период: (01.04.2019– 31.07.2019) +(01.04.2021 –30.09.2021) +(04.04.2022–30.09.2022); тестовое множество - период 02.08-18.08.2022.

В таблице 7 обучающее множество – интегрированный период: (09.02.2019 – 23.03.2019) + (01.11.2019 – 31.03.2020) + (01.11.2020 – 09.12.2020) + (01.01.2021 – 31.03.2021) +(08.10.2021 – 03.04.2022); тестовое множество: период (15.12.2020 – 31.12.2020). При этом контролируемые параметры выбирались согласно наиболее выраженным коэффициентам корреляций измеренных в срок i концентраций AQI с измеренными в сроки $i-1, i-2, \dots$ метеопараметрами (Таблица 3) и учитывался расход угля ТЭЦ и его тип.

В таблице 8 приведены соответствующие таблице 7 графики измеренных и прогнозных значений AQI для зимнего периода (15.12.2020 – 31.12.2020), полученные при моделировании с использованием ИНС *newff*. В приведенных графиках используются следующие обозначения: Ряд 1 – Измеренные значения AQI; Ряд 2 – Прогнозируемые значения AQI. Как следует из анализа графиков прогнозных значений AQI в отопительный период (таблица 8), формы кривых прогнозных значений качественно соответствуют формам кривых измеренных значений.

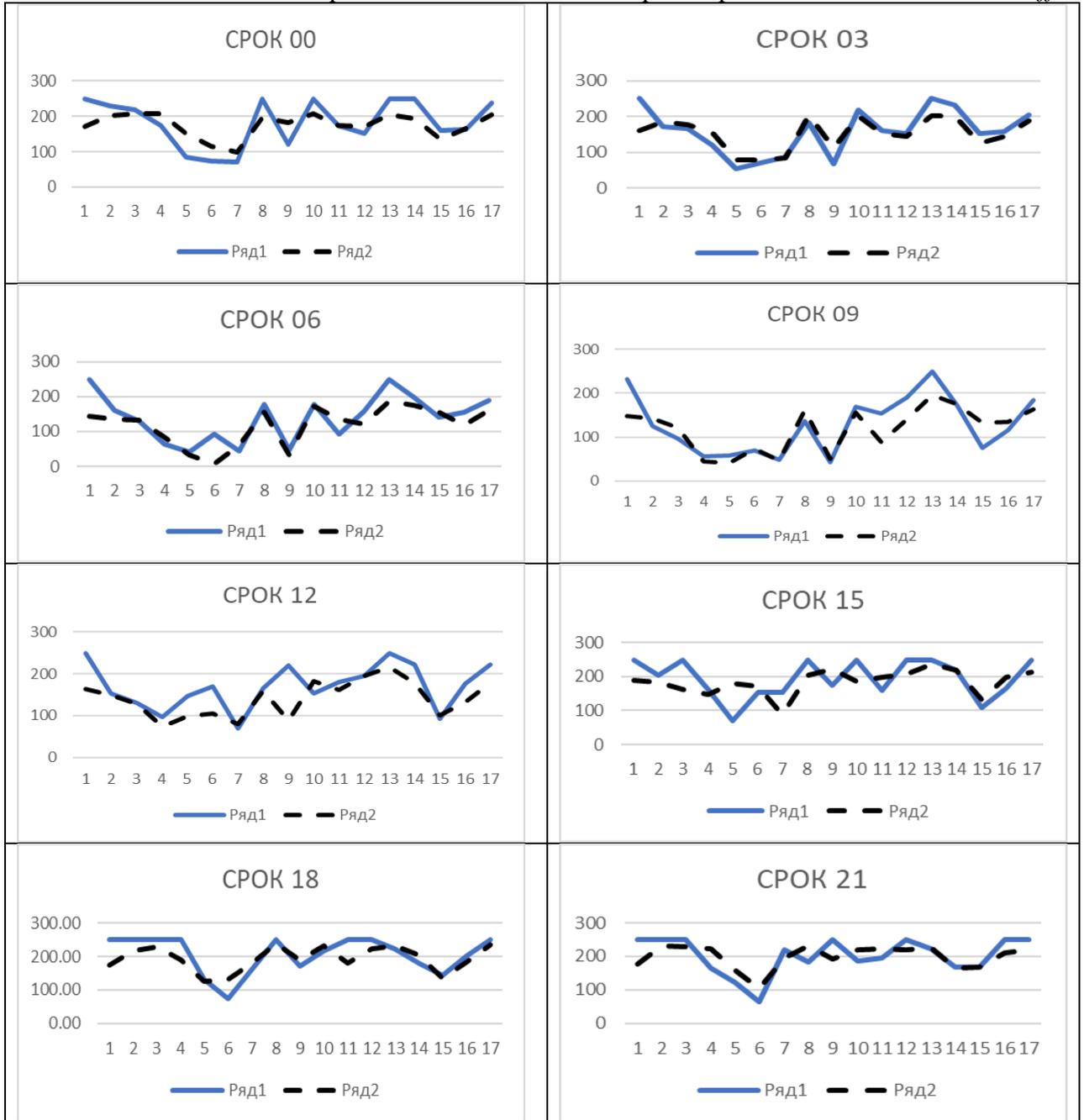
Таблица 6 – Средние ошибки моделированных значений AQI по срокам прогноза в летний период при использовании ИНС *newff*.

Сроки прогноза, час	Без учета расхода угля		С учетом общего расхода угля		С учетом общего расхода угля и его типов	
	MAPE	MAEMR	MAPE	MAEMR	MAPE	MAEMR
00	14,75	16,89	35,60	37,38	17,78	16,85
03	16,89	17,91	34,69	35,03	6,63	6,74
06	6,95	7,10	47,63	47,87	8,77	8,83
09	17,50	18,19	34,13	34,59	7,59	7,56
12	9,68	10,03	38,63	39,23	15,69	14,91
15	10,11	10,76	48,35	48,73	14,17	13,65
18	25,64	26,04	55,16	55,39	15,75	15,43
21	7,49	7,74	31,97	32,61	20,72	19,97
<i>Mean</i>	13,63	14,33	40,77	41,35	13,39	12,99

Таблица 7 – Средние ошибки моделированных значений AQI по срокам прогноза в зимний период при использовании ИНС *newff*.

Сроки прогноза, час	Без учета расхода угля		С учетом общего расхода угля		С учетом общего расхода угля и его типов	
	MAPE	MAEMR	MAPE	MAEMR	MAPE	MAEMR
00	22,97	16,99	28,84	20,01	23,90	18,94
03	20,78	18,07	20,99	18,46	18,20	15,73
06	29,04	22,37	28,07	22,43	25,12	21,70
09	38,66	34,81	26,55	26,62	21,18	21,46
12	25,33	25,79	24,07	23,41	21,92	23,76
15	25,33	22,32	27,66	21,59	27,53	21,50
18	17,74	16,44	21,04	17,24	16,84	15,51
21	16,22	15,76	24,18	18,86	18,56	15,78
<i>Mean</i>	24,51	21,57	25,18	21,08	21,66	19,30

Таблица 8 – Графики измеренных и моделированных значений AQI по срокам прогноза в отопительный период при использовании ИНС *newff*.



Обсуждение и выводы

Из таблицы 6 видно, что в летний период при моделировании значений AQI учет общей массы сжигаемого ТЭЦ угля существенно увеличивает ошибки прогноза, однако учет типов угля по месту добычи дает некоторое (до 2%) уменьшение ошибок. В зимний период (таблица 7) учет типов угля также повышает точность прогноза на 2%.

В таблице 9 представлены средние значения ошибок прогноза PM2.5 и AQI от средних значений ошибок по каждому сроку измерений за тестовые периоды для неотапительных (летних) и отопительных (зимних) сезонов: август 2019 г. и декабрь 2020 г., рассчитанные ИНС *newff* и ИНС *newgrnn* на обучающих выборках 2019 – 2021 гг. (01.04.2019– 31.07.2019) +(01.04.2021 –30.09.2021) и (09.02.2019 – 23.03.2019) + (01.11.2019 – 31.03.2020) + (01.01.2021 – 31.03.2021) +(08.10.21-31.12.21) соответственно.

Таблица 9. Средние значения ошибок прогноза PM2.5 и AQI при использовании ИНС *newff* и *newgrnn*.

Лето 2019, 2021								
Модель ИНС	Newff				newgrnn			
	AQI		PM2.5		AQI		PM2.5	
	MAPE	MAEMR	MAPE	MAEMR	MAPE	MAEMR	MAPE	MAEMR
Без учета угля	22,15	25,21	35,39	36,69	13,12	13,67	28,41	28,91
С учетом общего угля	28,28	28,62	24,09	26,16	10,80	11,28	25,82	26,18
С учетом типов угля	17,82	18,27	36,58	36,95	16,04	17,17	33,01	34,13
Зима 2019-2021								
Модель ИНС	Newff				newgrnn			
	AQI		PM2.5		AQI		PM2.5	
	MAPE	MAEMR	MAPE	MAEMR	MAPE	MAEMR	MAPE	MAEMR
Без учета угля	23,95	24,20	72,92	52,66	25,12	25,46	60,97	42,02
С учетом общего угля	26,54	26,20	80,54	47,02	27,77	27,48	69,02	53,20
С учетом типов угля	23,94	20,82	63,10	41,71	30,64	25,87	130,20	55,38

Сравнивая результаты моделирования AQI и PM2.5 (таблица 9), приходим к выводу, что обе нейросети лучше справляются с задачей прогноза AQI, чем прогноза PM2.5. Из этой же таблицы видно, что в неотапительный период ошибки прогноза близки для обоих типов примененных ИНС, однако для отопительного периода ошибки прогноза как AQI, так и PM2.5, при применении ИНС *newff* существенно меньше. В качестве примера на рисунке 2 приведены графики значений AQI, измеренных в срок 03 часа, с 15 по 31 декабря 2020 г. (Ряд 1) и их прогнозные значения на 3 часа вперед, рассчитанные по своим оптимальным моделям на основе ИНС FFBNN (Ряд 2) и ИНС GRNN (Ряд 3), обучающая выборка – соответствующая таблице 7. Как видно из этих графиков, ИНС FFBNN лучше справляется с поставленной задачей, чем ИНС GRNN.

Сравнение таблиц 7 и 9 показывает, что по мере увеличения примеров для обучающего множества (в таблице 7 добавлены данные 2022 г.) ошибки прогноза уменьшаются. Так при учете типов угля в отопительный период при обучении ИНС *newff* на данных 2019 – 2021 гг. $MAPE=23.94$, $MAEMR=20.82$ (таблица 9). При добавлении данных за период январь – март 2022 г. $MAPE=21.66$, $MAEMR=19.30$ (таблица 7).

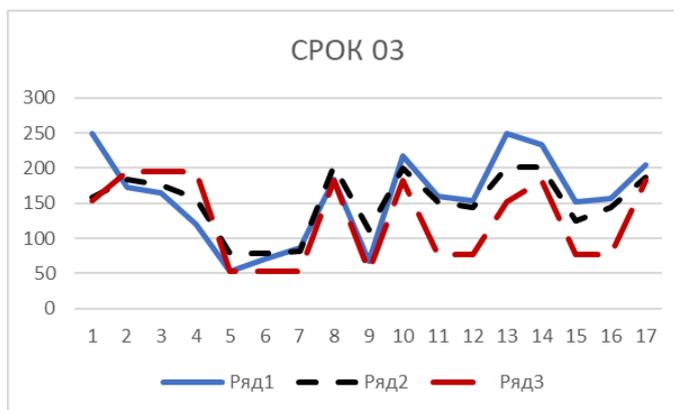


Рисунок 2 – Сравнение краткосрочного прогноза AQI ИНС двух типов с реальными данными в срок наблюдений 03 часа за период 15.12.2020-31.12.2020.

Анализ ошибок прогноза PM_{2.5} и AQI позволяет сделать следующие выводы.

1. Прогноз AQI в подавляющем большинстве случаев дает существенно меньшую ошибку по сравнению с прогнозом PM_{2.5} как в отопительный, так и в неоперительный периоды при использовании обоих типов нейросетей.

2. ИНС *newgrnn* дает меньшую ошибку в неоперительный период, *newff* - в отопительный. Поскольку прогноз загрязнения воздуха в отопительный период представляется более важным, использование ИНС *newff* предпочтительнее.

3. Оба типа используемых ИНС резко реагируют на выбор обучающего и тестового множеств. Наименьшая ошибка прогноза была получена при выборе обучающего и тестового множества в течение одного года. Добавив в обучающее множество новые данные, ошибка прогноза при использовании ИНС *newff* увеличилась, но по мере дальнейшего добавления данных - уменьшалась.

4. Введение в параметры обучения массы ежесуточно сжигаемого ТЭЦ угля и его типов позволило уменьшить ошибку прогноза AQI при использовании ИНС *newff* на (2-7)%.

Заключение

Таким образом, в работе представлено развитие методологии для краткосрочного прогноза PM_{2.5} и AQI с учетом новых метеоданных, а также данных о ежедневно потребляемом ТЭЦ г. Бишкек количестве тонн угля и его типе (по месту добычи). Используются модели, построенные на основе двух типов ИНС: на основе нейронной сети с прямым распространением сигнала и обратным распространением ошибки FFNN и на основе обобщенно-регрессионной нейронной сети GRNN. Показано, что точность прогнозирования зависит от типов используемых ИНС, а также - в большой степени - от выбора обучающих и тестовых множеств. Это подтверждает нестационарность процессов загрязнения атмосферного воздуха, обусловленных наличием случайных и неучтенных факторов. Учет этих факторов - предмет дальнейших исследований.

Благодарности

Авторы выражают благодарность Отделу по связям с общественностью ОАО «Электрические станции» КР за предоставленную информацию о количестве тонн ежедневно сжигаемого на ТЭЦ г. Бишкек угля и его типу по месту добычи за период с 9 февраля 2019 года по 30 сентября 2022 года.

Литература

1. Burhan Baran. Prediction of Air Quality Index by Extreme Learning Machines//International Artificial Intelligence and Data Processing Symposium, IDAP 2019. DOI:10.1109/IDAP.2019.8875910

2. Soubhik Mahanta, T. Ramakrishnudu; Rajat Raj Jha; Niraj Tailor. Urban Air Quality Prediction Using Regression Analysis//IEEE Region 10 Conference, TENCN 2019. DOI: 10.1109/TENCN.2019.8929517
3. Puttakul Sakul-Ung; Pitiporn Ruchanawet; Nataporn Thammabunwarit; Amornvit Vatcharaphrueksadee; Chatchawan Triperm. PM2.5 Prediction based Weather Forecast Information and Missingness Challenges: A Case Study Industrial and Metropolis Areas//Research, Invention, and Innovation Congress, RI2C 2019. DOI: 10.1109/RI2C48728.2019.8999941
4. Anikender Kumar, P.Goyal. Forecasting of daily air quality index in Delhi // Science of The Total Environment Volume 409, Issue 24, 2011, Pages 5517-5523.
5. Лыченко, Н. М. Регрессионный анализ метеорологических факторов и концентраций частиц PM2.5 в атмосферном воздухе г.Бишкек / Н. М. Лыченко // Проблемы автоматизации и управления. – 2019. – № 2(37). – С. 5-15. – DOI 10.5281/zenodo.3594649. – EDN QGNVJK.
6. Сайт «Расписание погоды rp5.ru» Архив погоды в Бишкеке https://rp5.ru/%D0%90%D1%80%D1%85%D0%B8%D0%B2_%D0%BF%D0%BE%D0%B3%D0%BE%D0%B4%D1%8B_%D0%B2_%D0%91%D0%B8%D1%88%D0%BA%D0%B5%D0%BA%D0%B5 (дата обращения: 05.10.2022)
7. AirNow Department of State // https://airnow.gov/index.cfm?action=airnow.global_summary#U.S._Department_of_State_Bishkek, (дата обращения: 05.10.2022).
8. Великанова, Л. И. Мультирегрессионные и обобщенно-регрессионные нейросетевые модели краткосрочного прогноза загрязнения PM2.5 в г. Бишкек с учетом метеорологических параметров / Л. И. Великанова, Н. М. Лыченко // Проблемы автоматизации и управления. – 2019. – № 2(37). – С. 16-25. – DOI 10.5281/zenodo.3594777. – EDN MNNDYU.
9. Лыченко, Н. М. Модификация классификатора индекса качества воздуха Г. Бишкек с учетом фактора загрязнения / Н. М. Лыченко, А. В. Сороковая // Проблемы автоматизации и управления. – 2021. – № 3(42). – С. 101-110. – EDN BXWCYS.
10. Donald F.Specht A general regression neural network //IEEE Transactions on neural networks. — November 1991. — Vol. 2, № 6. 568–576. doi:10.1109/72.97934
11. Widi Aribowo. Optimizing Feed Forward Backpropagation Neural Network Based on Teaching-Learning-Based Optimization Algorithm for Long-Term Electricity Forecasting // International Journal of Intelligent Engineering and Systems, Vol.15, No.1, 2022. DOI: 10.22266/ijies2022.0228.02
12. Современное прогнозирование. URL: <https://forecasting.svetunkov.ru/etextbook/> (дата обращения: 30.09.2021).

УДК 004

С.Н. Верзунов, verzunov@hotmail.com

Институт машиноведения и автоматизации НАН КР

СИСТЕМА ИСКУССТВЕННОГО ИНТЕЛЛЕКТА ДЛЯ ДИАГНОСТИКИ COVID-19 ПО КТ-СНИМКАМ

Пандемия SARS-CoV-2 по-прежнему остается актуальной для Кыргызстана. Поэтому в статье разработана система искусственного интеллекта на основе глубоких сверточных нейронных сетей для диагностики COVID-19 на основе анализа КТ-снимков. Для сегментации КТ-снимки разбиваются на двумерные изображения, которые обрабатываются с помощью двух модифицированных сетей U-Net. Одна сеть определяет поврежденные области легких на слое КТ-снимка, а другая – область снимка, соответствующую легким. Затем находится степень повреждения легких как отношение площади поврежденной области легких к площади легких на срезе. Окончательная степень повреждения легких определяется как среднее значение степени повреждения легких по всем срезам. Разработанная система искусственного интеллекта реализует возможность одновременной постановки диагноза и сегментации КТ-снимков с целью определения степени повреждения легких с большей точностью, чем получено ранее в других работах.

Ключевые слова: повреждение легких, глубокая нейронная сеть, искусственный интеллект, пакетная нормализация, дополнение данных

Особенности диагностики COVID-19 по КТ-снимкам. Как известно [1], COVID-19 входит в число крупнейших глобальных кризисов в области общественного здравоохранения. Из-за последствий пандемии за последние годы в мире было зафиксировано чрезвычайно большое количество смертей. Пандемия SARS-CoV-2 по-прежнему остается актуальной для Кыргызстана. Последние недели середины 2022 года в стране выявляли 100–200 новых случаев заражения, а общее количество зарегистрированных больных COVID-19 с начала года на конец августа достигло почти 21 тысячи. При этом более 65% всех зарегистрированных на тот момент случаев заболевания COVID-19 пришлось на январь 2022 года (выявлено 13 763 случая коронавируса). Это сказалось на том, что последующие три месяца – февраль, март и апрель – показатель заболеваемости вирусом в несколько раз превышал аналогичный период 2021 года [1].

Циклическое чередование подъёмов и спадов характерно для респираторных вирусов. В Кыргызстане очередная большая волна в 2022 году пришлась на июнь-июль (рис. 1). В июле она достигала пика и затем в августе пошла на спад. Это связано с уровнем коллективного иммунитета. Со временем, как показывает практика, коллективный иммунитет имеет тенденции к снижению, кроме того, вирус мутирует, что приводит к распространению новых штаммов и повторным заболеваниям [2].

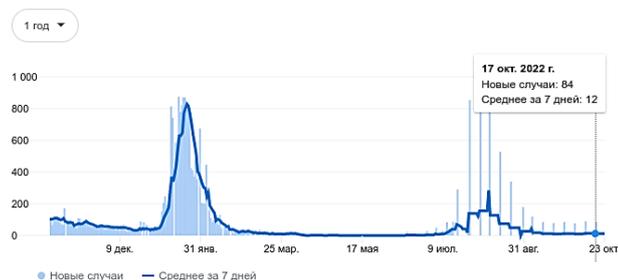


Рисунок 1 – Заболеваемость населения Кыргызстана коронавирусной инфекцией в 2022 году

Таким образом, говорить об окончании пандемии пока не приходится. На конец октября фиксируется около 84 случаев в неделю [3]. Очередная волна COVID-19, по прогнозам экспертов, ожидается глубокой осенью – в ноябре–декабре и захватит февраль будущего года. В ближайшие пять лет эти волны будут повторяться, как это происходит с

гриппом, пока вирус либо совсем не ослабнет, либо популяция человечества не приспособится к нему.

Самый главный фактор заключается в том, что новые штаммы COVID-19 перестали поражать легкие. То есть, к счастью, они не приводят к быстрому летальному исходу, не вызывают значительного поражения легких. Вирус останавливается в верхних дыхательных путях, не вызывает былой летальности и не приводит к прежней нагрузке на систему здравоохранения. Кроме того, в Кыргызстан завезено большинство из существующих в мире штаммов коронавируса. Теоретически вирус мог мутировать и в Кыргызстане, но это маловероятно. Как правило, мутация вируса происходит в странах с высокой плотностью населения, однако мутировавший вирус может быть создан искусственно [2]. Недавно в прессе появились сообщения, что исследования Бостонского университета, связанные с тестированием, созданной в лаборатории гибридной версии вируса SARS-CoV-2, которая имеет летальность в 80%, вызвали опасения у общественности.

Конечно, нет никаких доказательств того, что работа, проводившаяся при соблюдении мер предосторожности, была сделана ненадлежащим или небезопасным образом. Более того, она была одобрена внутренним комитетом по проверке биобезопасности и комиссией по здравоохранению Бостона, сообщили в университете, где и была проведена работа. Однако выяснилось, что исследовательская группа не согласовала свою работу с Национальным институтом аллергии и инфекционных заболеваний США, который был одним из спонсоров проекта. В отчетах о проделанной работе группа ученых не давала понять, что она проводит эксперименты, которые могут быть связаны с усилением патогена. Ранее ученые Бостонского университета рассказали о проведенном ими исследовании, в ходе которого был создан гибридный вирус, в котором белок шипа версии SARS-2 «Омикрон» был слит с вирусом штамма Ухань, оригинальной версии, появившейся в Китае в 2020 году. Испытания показали, что новый штамм оказался более смертоносным для одного из видов лабораторных мышей, чем омикрон-штамм, убив 80% зараженных мышей. Поэтому нельзя исключать в будущем возникновения более заразного и более опасного штамма COVID-19 [4].

Таким образом, вариант SARS-COV-2 «Омикрон», наиболее распространённый в настоящее время, связан с меньшим количеством и менее серьезными изменениями по результатам компьютерной томографии (КТ) грудной клетки по сравнению с прежним вариантом «Дельта». Пациенты с «Омикроном» имели большую частоту повреждения бронхиальной стенки, но меньшую клиническую тяжесть и лучшие результаты при госпитализации, чем пациенты с вариантом «Дельта» [5].

В настоящее время многие независимые клинические исследования показали, что выявление поражений COVID-19 на КТ-изображениях является важным шагом в клинической оценке лечения [6] и может быть полезно в текущем контексте из-за возможности таким образом выявить штамм, который вызвал заболевание, тяжесть и стадию течения заболевания. Стадию можно определить, оценивая динамику повреждений со временем.

Однако поиск точного метода сегментации поражений COVID-19 по КТ-снимкам является очень сложной задачей из-за больших различий в размере, форме и положения поражений, неопределенности границ, а также сложной структуры поврежденных областей [7]. В прошлых работах [8, 9] нами была разработана система искусственного интеллекта для диагностики коронавирусной инфекции на основе бинарной (COVID-19, норма) и многоклассовой классификации изображений грудной клетки по 6 классам: норма, вирусная пневмония, бактериальная пневмония, COVID-19, туберкулез, неуточненное заболевание легких на основе анализа рентгеновских снимков. Были разработаны и протестированы архитектуры на базе предварительно обученных сверточных нейронных сетей для дифференциальной диагностики коронавирусной инфекции и различных видов пневмонии. Они могут помочь врачам повысить эффективность и точность диагностики и даже выявить очаги инфекции, не обнаруженные непосредственно, что может иметь серьезные последствия для мониторинга и лечения заболеваний, вызванных COVID.

Рассмотренные нами нейронные сети показывают высокую точность классификации изображений рентгеновских снимков грудной клетки. В работе [9] нами было изучено большое количество предварительно обученных сверточных нейронных сетей (CNN) для диагностики COVID-19: AlexNet, VGG, GoogLeNet, DenseNet, XceptionNet, MobileNet, SqueezeNet, Inception-ResNet, NasNetmobile, ShuffleNet, EfficientNet и ResNet-50, широко используемых для классификации изображений. В работе [8] для классификации заболеваний легких хороший результат показала предварительно обученная сверточная нейронная сеть EfficientNetB3. Она содержит 29 сверточных и один полносвязный слой. EfficientNet использует технику, называемую составным коэффициентом, для простого, но эффективного масштабирования моделей. Вместо случайного увеличения ширины, глубины или разрешения составное масштабирование равномерно масштабирует каждое измерение с определенным фиксированным набором коэффициентов масштабирования. Используя метод масштабирования и AutoML, авторы EfficientNet разработали еще 6 моделей различной размерности, которые превзошли по точности большинство современных сверточных нейронных сетей и с гораздо большей эффективностью [10]. Однако разработанные нами ранее системы пригодны только для обработки рентгеновских, но не КТ-снимков.

Достаточно недавно созданы специфичные архитектуры сверточных сетей для сегментации медицинских изображений [5]. Они показали высокий результат в задачах сегментации медицинских изображений благодаря высокой мощности представления, свойствам совместного использования фильтров и быстрого объединения результатов работы параллельных ветвей. Тем не менее системы искусственного интеллекта для сегментации поражений COVID-19 с использованием таких нейронных сетей все еще относительно мало исследованы и обладают невысокой точностью.

Однако архитектуры глубоких нейронных сетей семейства U-Net представляют собой быстро развивающийся и популярный класс технологий сегментации, которые могут быть применены к КТ-изображениям. Кроме того, другие разновидности, такие как VB-Net, FCN и SegNet, также имеющие хорошие перспективы, входят сейчас в состав фреймворков с открытым исходным кодом, например, MONAI, который является бесплатным и поэтому вполне доступным для практического применения.

В работе [11] предлагается метод, основанный на сочетании глубокой нейронной сети VGG16 и ансамблевого обучения, с целью повышения производительности глубоких нейронных сетей для решения задачи постановки диагноза по данным компьютерной томографии. Экспериментальные результаты показывают, что глубокая нейронная сеть VGG16 в сочетании с интегрированным подходом к обучению может значительно улучшить эффективность классификации. Метод, предложенный в этой статье, имеет следующие недостатки, т.к. рассматривается только предварительно обученная сверточная основа VGG, а некоторые новые сетевые методы не используются. В предлагаемом методе усиление контраста используется для улучшения качества исходных изображений.

Затем предварительно обученная модель DenseNet-201 дообучается с использованием технологии трансферного обучения. Для извлечения признаков использовались два полносвязных слоя и слой пуллинга, а извлеченные признаки затем оптимизировались с помощью специального алгоритма для выбора наиболее оптимальных функций обучения. Использование только выбранных признаков важно для повышения точности подхода, однако оно напрямую не влияет на вычислительные затраты метода. Поэтому в предложенном в этой работе методе использовался новый параллельный метод с высоким индексом для слияния двух оптимальных векторов; а затем результат передавался на машину экстремального обучения для окончательной классификации.

Как известно, семантическая сегментация изображений – это задача разделения частей изображения на подгруппы пикселей, принадлежащих к соответствующим объектам, с его классификацией. В отличие от задач локализации и детектирования объектов на изображении, задача семантической сегментации является более сложной. Это обусловлено необходимостью не только верно определить принадлежность объекта на изображении к

определенному классу, но и максимально четко определить границы и структуру объекта на изображении [11]. Сети U-Net – это метод, изначально предложенный для сегментации медицинских изображений. Это одна из первых моделей сегментации на основе технологий глубокого обучения, и архитектура U-Net также используется во многих вариантах генеративно-состязательных сетей [12].

Основная идея сетей U-Net заключается в том, чтобы, во-первых, обеспечить возможность обучения на частично размеченном наборе данных и прогнозировать верную разметку других неразмеченных областей этого же набора данных. Во-вторых, обучаясь на нескольких частично размеченных наборах данных, обобщить их затем на новые данные. На рисунке 2 показана архитектура сети U-Net, в которой можно выделить отдельную ветвь для кодирования и ветвь декодирования для каждого из 4 уровней разрешения. В ветви кодирования каждый слой содержит две $3 \times 3 \times 3$ свертки (за обеими следует слой ReLu) и есть объединяющий слой пуллинга, т.е. выбора максимального элемента из соседних размером $2 \times 2 \times 2$ с шагом 2 в каждом направлении позади них. В ветви декодирования каждый слой содержит $2 \times 2 \times 2$ слой деконволюции с размером шага 2, за которыми следуют два сверточных слоя с ядром $3 \times 3 \times 3$ и слой ReLu.

По более короткой ветви слои с одинаковым разрешением в ветви кодирования передаются в ветвь декодирования, предоставляя ему оригинальные признаки с высокого разрешения изображения.

Последний слой представляет собой сверточный слой $1 \times 1 \times 1$, который уменьшает количество выходных каналов так, что конечное количество выходных каналов равно количеству категорий меток [13].

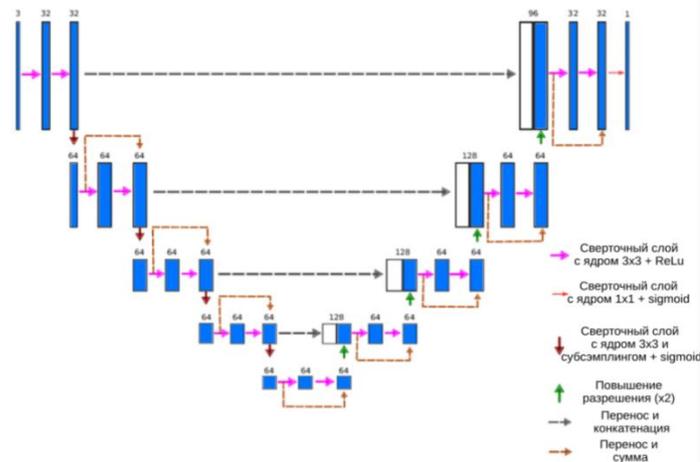


Рисунок 2 – Архитектура сети U-Net

В работе [12] предложена система автоматической сегментации КТ-снимков пациентов, больных COVID-19, на основе сети SInfNet позволяющая получать количественные характеристики степени этих повреждений, выступая таким образом в качестве инструмента поддержки принятия клинических решений. Метод включает пять этапов: сегментация легких, сегментация и извлечение легочных сосудов, сегментация повреждений, классификация заболевания и количественная оценка степени повреждения легких. Однако точность сегментации с помощью этой системы оказалась сравнительно не велика. Сеть на основе архитектуры U-Net, описанная в работе [12], показала лучшие результаты

Таким образом, различные виды пневмонии могут вызывать поражение легких, приводящее к абсолютно различным последствиям. По сравнению с обычной, пневмония, вызванная COVID-19 и его вариантами, имеет более длительное течение, сравнительно более высокую скорость передачи и смертность и может даже вызвать необратимое повреждение легких с соответствующими последствиями. Поэтому крайне важно различать пневмонию,

вызванную COVID-19 и другими патогенами, по особенностям повреждения легких, так как это имеет определяющее значение для стратегии лечения и прогноза пациентов.

То есть для более точной постановки диагноза система искусственного интеллекта может использовать КТ-снимки для одновременной классификации с целью точной постановки диагноза и сегментации изображений для определения поврежденных участков легких, что позволит более эффективно использовать такую систему в медицинских целях. Исходя из анализа литературных источников, в настоящее время известно очень немного систем [12], позволяющих решить сразу две эти задачи. Связано это с тем, что точность результатов очень сильно зависит от полноты наборов данных, размеров изображений и качества маркировки данных. В связи с этим основная задача настоящего исследования состоит в том, чтобы реализовать способ одновременной постановки диагноза и сегментации КТ-снимков с целью определения степени повреждения легких с более высокой точностью, чем ранее, и достаточной для принятия клинических решений.

Система искусственного интеллекта (СИИ) для диагностики COVID-19 по КТ-снимкам. Новизна предлагаемого нами решения заключается в одновременной диагностике COVID-19 и сегментации пораженной, а также всей области легких по срезам КТ-снимка с целью определения тяжести заболевания. Как показано на рисунке 3, предлагаемая система может обеспечить выполнение всех этих функций.

Во-первых, для диагностики заболеваний легких исходное изображение должно быть предварительно обработано, так как различные томографы будут давать КТ-снимки, значительно отличающиеся друг от друга, а весь набор данных расширен с помощью методов дополнения данных, включая масштабирование, обрезку, отражение, вращение, перенос, аффинные преобразования и, возможно, расширение с помощью изменения яркости и контраста.

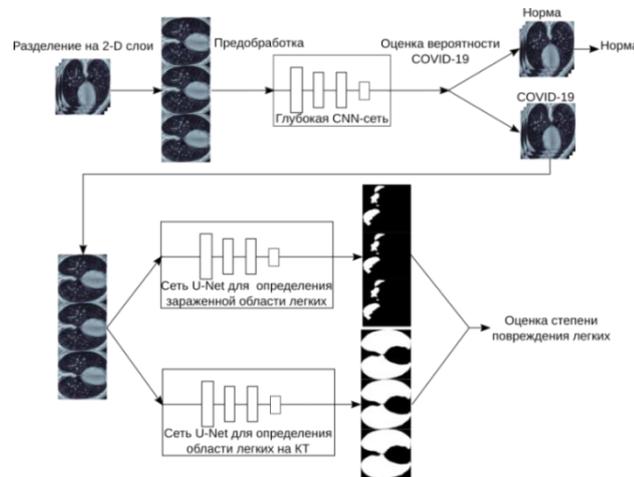


Рисунок 3 – Схема работы предлагаемой СИИ для диагностики COVID-19 по КТ-снимкам

Как это принято, расширение данных должно использоваться только для обучающего набора данных, а не проверочного набора для более точной оценки качества работы сети. Затем эти данные передаются в глубокую сверточную сеть для обучения до достижения максимальной точности классификации, как это описано нами ранее в работе [9]. КТ-снимок грудной клетки в результате обработки первой сетью может попасть в одну из 2 категорий: «COVID-19» и «норма». После этого КТ-снимки, относящиеся к классу «COVID-19», используются в качестве входных данных для сетей сегментации снимков.

Для сегментации КТ-снимки разбиваются на двухмерные изображения, которые обрабатываются с помощью двух модифицированных сетей U-Net. Одна сеть определяет поврежденные области легких на слое КТ-снимка, а другая – область снимка, соответствующую легким. Затем находится степень повреждения легких как отношение площади поврежденной области легких к площади легких на каждом КТ-срезе.

Окончательная степень повреждения легких определяется как среднее значение степени повреждения легких по всем срезам.

Предварительная обработка КТ-снимков. Набор данных «COVID-19 CT scans» (рис. 4) [15] использовался в настоящем исследовании как для классификации по 2 классам (норма и COVID-19), так и для сегментации по 2 меткам (область легких и зараженная область). Всего этот набор включает 2112 КТ-слоев, принадлежащих 10 пациентам, из них на 1615 слоях есть повреждения, вызванные COVID-19.

Для проведения вычислительных экспериментов по обучению нейронных сетей использовался суперкомпьютер, предоставленный в рамках договора о сотрудничестве между ИМА НАН КР и КРСУ им. Б.Н. Ельцина. Основные характеристики используемого программного и аппаратного обеспечения приведены в работе [8].

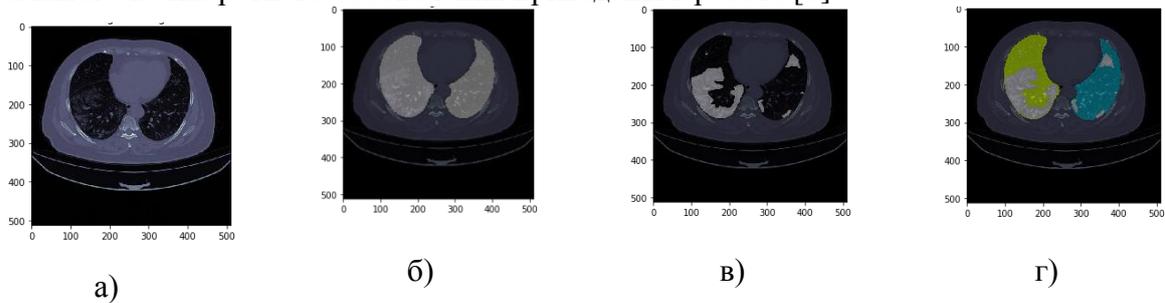


Рисунок 2.28 – Набор данных «COVID-19 CTscans»: а) слой КТ-снимка, б) область легких, в) поврежденная область, область легких и поврежденная область

Предварительная обработка цифровых данных КТ выполняется в два этапа. На первом этапе выполняется нормализация размера входных данных и коррекция интенсивности изображений. Это необходимо для того, чтобы упростить обучение нейронных сетей с учетом того, что разное оборудование может иметь различные настройки и давать отличающиеся по этим параметрам КТ-срезы. Разрешение КТ-снимков варьируется от 401×630 до 630×630 точек, что требует больших вычислительных затрат для обучения нейронных сетей. Чтобы преодолеть это ограничение, необходимо масштабировать КТ-изображения. Для этого необходимо обрезать центральную область, содержащую изображения легких, а затем применить метод билинейной интерполяции, чтобы изменить размер всех изображений до 256×256 пикселей. Далее применялся метод преобразования, основанный на адаптивном алгоритме выравнивания гистограмм [16] для нормализации статистических характеристик изображений (рис. 5).

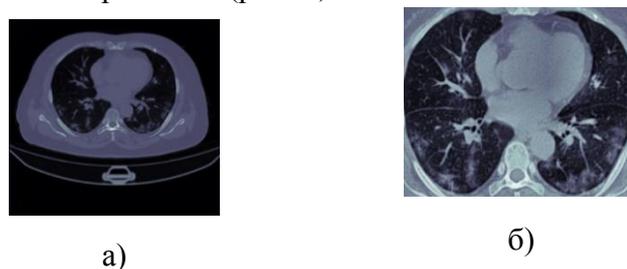


Рисунок 5 – Результаты предварительной обработки изображения: а) исходное изображение; б) предварительно обработанное изображение

На втором этапе для повышения надежности СИИ к различным помехам во входных данных применялось дополнение (рис. 6), включая масштабирование на 80 – 120% от исходного размера, вертикальное и горизонтальное отражение, вращение в диапазоне – от -40° до 40° , перенос на 20% по обеим осям, а также искажение с перспективой в диапазоне от -16° до 16° .

Как уже было сказано выше, расширение данных используется только для обучающего набора, но не для тестового набора данных.



Рисунок 6 – Результаты дополнения данных: а) дополненные КТ-снимки; б) дополненные маски зараженных областей; в) дополненные маски областей легких

Цель этапа предварительной обработки – повышение разнообразия обучающих данных ради расширения возможностей обобщения обученной модели, что особенно полезно при небольшом размере набора данных [17]. Всего было дополнительно сгенерировано 100 слоев вместе с соответствующими им масками зараженных областей и областей легких.

Глубокая нейронная сеть для диагностики COVID-19 легких по КТ-снимкам. Разработанная архитектура глубокой нейронной сети для диагностики COVID-19, по данным КТ, показана на рисунке 7.

Архитектура глубокой нейронной сети организована, как и обычно, в виде стека чередующихся слоев Conv2D (с функцией активации ReLu) и слоев пуллинга MaxPooling2D. Однако особенностью предложенной архитектуры является использование слоев пакетной нормализации (BatchNormalization), нормализующих входные данные. Пакетная нормализация реализует преобразование, которое поддерживает среднее выхода слоя близким к 0, а стандартное отклонение – близким к 1. Важно отметить, что пакетная нормализация работает по-разному во время обучения и во время работы сети.

Во время обучения слой нормализует свои выходные данные, используя среднее значение и стандартное отклонение текущего пакета входных данных. То есть для каждого нормируемого канала этот слой возвращает:

$$C = \gamma \frac{B - M(B)}{\sqrt{D(B) + \varepsilon}} + \beta,$$

где C – выходной тензор, B – пакет входных данных, ε – небольшая константа, γ – коэффициент масштабирования, β – коэффициент смещения (M – среднее и D – дисперсия).

Во время работы сети слой нормализует свои выходные данные, используя скользящее среднее от среднего значения и стандартного отклонения пакетов, которые он запомнил во время обучения. То есть возвращается:

$$C = \gamma \frac{B - m}{\sqrt{d + \varepsilon}} + \beta.$$

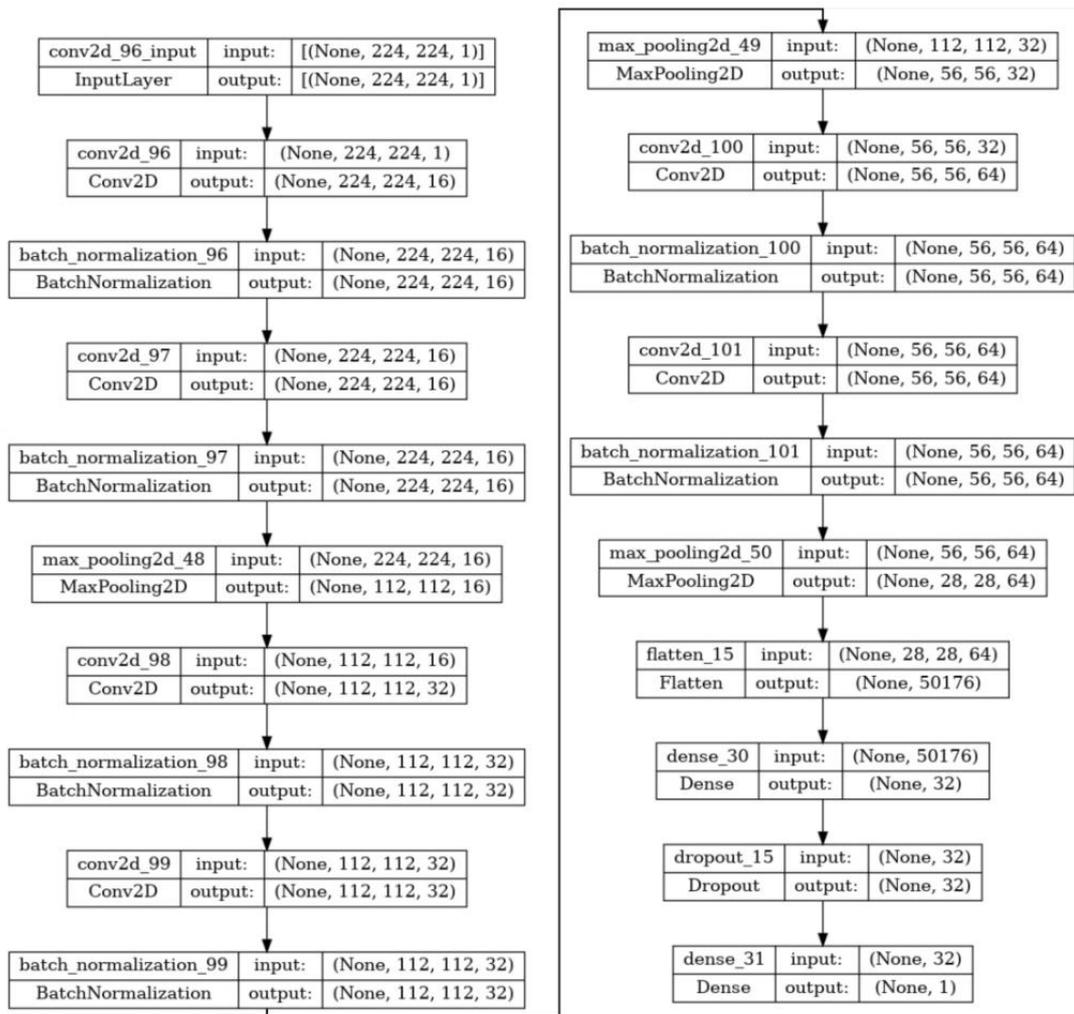


Рисунок 7 – Архитектура глубокой нейронной сети для классификации КТ-слоев

Параметры m и d обновляются каждый раз, когда слой вычисляется в процессе обучения:

$$m = m * \mu + M(\mathbf{B}) * (1 - \mu),$$

$$d = d * \mu + D(\mathbf{B}) * (1 - \mu),$$

где μ – параметр момента.

Таким образом, слой может нормализовать свои входные данные во время работы только после обучения на данных, которые имеют такие же статистические характеристики.

Необходимость использования слоев BatchNormalization в архитектуре глубокой нейронной сети объясняется ее чувствительностью к исходным начальным случайным весам, входным данным и конкретному алгоритму обучения. Она выражается в том, что статистическое распределение входных данных по слоям ближе к концу сети может меняться после каждого пакета при обновлении весов. Это может привести к тому, что алгоритм обучения всегда будет пытаться достичь динамически изменяющуюся цель. Как известно, такая особенность характерна для алгоритма обратного распространения ошибки, потому что веса сети обновляются слой за слоем в обратном порядке от конца сети к ее началу с использованием оценки ошибки, которая допускает, что веса в слоях, предшествующих текущему слою, являются фиксированными. Репараметризация с помощью пакетной нормализации существенно снижает проблему применения обновлений при большом количестве слоев глубокой нейронной сети.

На рисунке 8 показан график, отражающий процесс обучения сети. В качестве функции потерь использовалась бинарная кроссэнтропия:

$$c(\mathbf{Y}, \hat{\mathbf{Y}}) = \sum_i^N Y_i \ln(\hat{Y}_i) + (1 - Y_i) \ln(1 - \hat{Y}_i), \quad (2.4)$$

где \mathbf{Y} – тензор, состоящий из 1 и 0, описывающий поврежденные и здоровые участки легких, соответственно, $\hat{\mathbf{Y}}$ – выход сети, N – произведение количества строк и столбцов матрицы \mathbf{Y} . А в качестве оптимизатора использовался Adam с постоянным параметром скорости обучения $\text{lr}=5 \cdot 10^{-4}$.

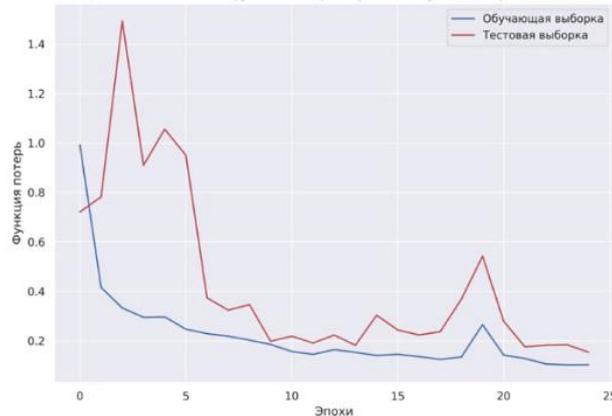


Рисунок 8 – Динамика изменения функции потерь в процессе обучения нейронной сети для постановки диагноза по КТ-снимкам

При этом при обучении на вход сети подавались 2D-слои трехмерного КТ-снимка. Это сделано для экономии вычислительных ресурсов и для обеспечения большего разнообразия данных с целью уменьшения переобучения сети для диагностики COVID-19. Матрица ошибок, в которой указано количество правильных и неправильных результатов, полученных с помощью нейронной сети при двух разных значениях дискриминационного порога θ , показана на рисунке 9.

Как известно, дискриминационным порогом называют значение, принимаемое дискриминирующей функцией в задачах бинарной классификации, которое позволяет разделить классы.

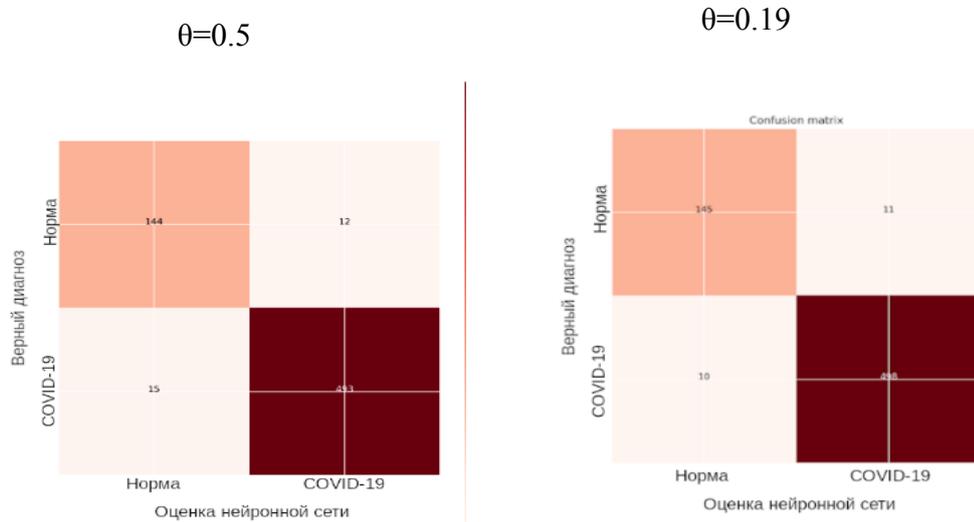


Рисунок 9 – Матрица ошибок при двух различных значениях дискриминационного порога θ

Для общей оценки качества работы обученной нейронной сети использовались также такие характеристики, как верность, точность, оценка f1 и площадь под ROC-кривой (ReceiverOperatingCharacteristic, рабочая характеристика приемника), т.н. AUC

(AreaunderROC curve). Подробно определения этих характеристик были даны в работе [9], и, кроме того, использовался такой, как полнота:

$$R_c = \frac{T_p}{T_p + F_n},$$

где T_p – количество КТ-слоев, содержащих повреждения от COVID-19, правильно идентифицированных как содержащие; F_n – количество слоев, содержащих повреждения от COVID-19, ошибочно идентифицированных как несодержащие [15]. Ниже, в таблице 2.5, показаны количественные оценки качества работы сети.

Таблица 2.5 – Количественные оценки качества работы сети для диагностики COVID-19

	$\theta=0.5$	$\theta=0.19$
Точность A_c	0.9593	0.9683
Верность P_r	0.9762	0.9784
Полнота, R_c	0.9704	0.9803
Оценка, F1	0.9733	0.9793
AUC	1	0.99

Распределение классификаций, выполненных обученной нейронной сетью по каждому классу, показано на рисунке 2.34. А на рисунке 2.35 изображена ROC-кривая при двух различных значениях дискриминационного порога θ . Значение дискриминационного порога $\theta=0.15$ было подобрано нами методом половинного деления исходя из максимального значения точности.

По сравнению с работой [12], где используется аналогичный набор данных, таким образом удалось повысить точность классификации на 2 %, т.е. 94.52% против 96.83%. Улучшение на 9% можно объяснить добавленными слоями нормализации и дополнением данных.

Глубокие нейронные сети для определения степени повреждения легких по КТ-снимкам. Задача определения степени повреждения легких может быть сформулирована как задача семантической сегментации слоев КТ-снимка по двум классам – поврежденная и здоровая область. Кроме того, необходимо также одновременно решить еще одну задачу – сегментировать КТ-слои на области, принадлежащие легким, и не принадлежащие им.

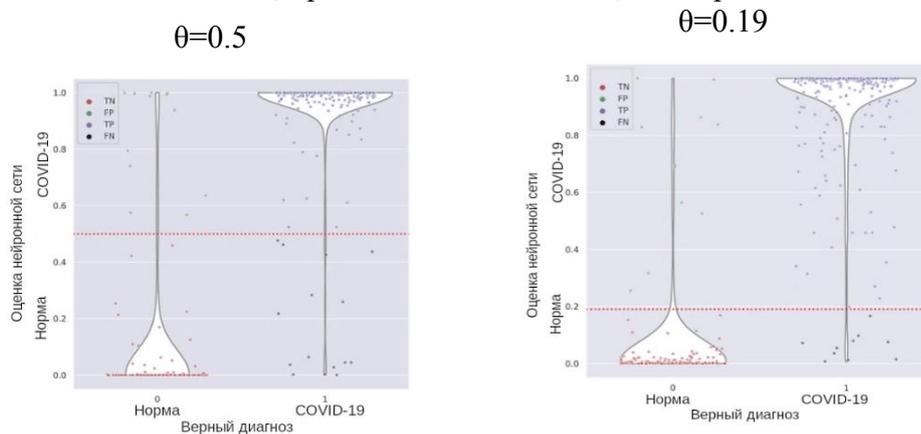


Рисунок 10 – Характеристики обученной нейронной сети при двух различных значениях дискриминационного порога θ

Таким образом, для решения задачи необходимо обучить две глубокие нейронные сети – отдельно для сегментации поврежденной области и области легких. При обучении на вход сетей подавались 2D-слои трехмерного КТ-снимка. Это сделано для экономии вычислительных ресурсов и для обеспечения большего разнообразия данных с целью уменьшения переобучения сети для определения поврежденной COVID-19 области, так как при этом из выборок слоев использовалось только относительно небольшое количество

двухмерных слоев, на которых повреждения присутствуют, что позволило улучшить обобщающую способность глубокой нейронной сети.

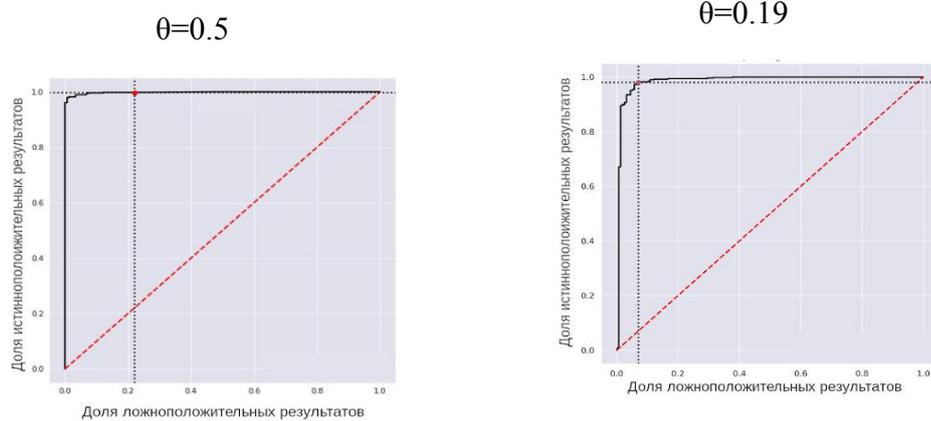


Рисунок 11 – Кривая ROC при двух различных значениях дискриминационного порога θ

В таблице 2.6 показана архитектура глубоких нейронных сетей для определения степени повреждения легких. Она одинакова для обеих нейронных сетей.

Отличием предложенной архитектуры от известной архитектуры сети U-Net является использование нескольких слоев пакетной нормализации (BatchNormalization), нормирующих свои входные данные. Пакетная нормализация, как уже было сказано выше, реализует преобразование, которое поддерживает среднее выхода слоя близким к 0, а стандартное отклонение вывода близким к 1.

В качестве функции потерь нами предлагается использовать метрику:

$$loss(\mathbf{y}, \hat{\mathbf{y}}) = \frac{1}{2} c(\mathbf{y}, \hat{\mathbf{y}}) + \frac{1}{2} (1 - s(\mathbf{y}, \hat{\mathbf{y}})), \quad (2.5)$$

где \mathbf{y} – матрица, состоящая из 1 и 0, описывающая поврежденные и здоровые участки легких, соответственно, $\hat{\mathbf{y}}$ – выход сети, c – бинарная кроссэнтропия, описываемая формулой (1), а s – метрика Сёрнсена, выбранная целевой функцией:

$$s(\mathbf{y}, \hat{\mathbf{y}}) = \frac{2 \sum_i^N (y_i \hat{y}_i)}{\sum_i^N y_i + \sum_i^N \hat{y}_i + 1}. \quad (2.6)$$

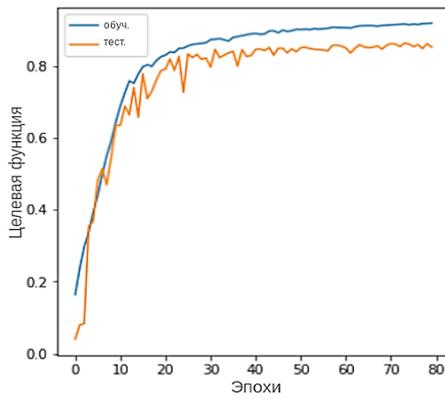
Таблица 2.6 – Архитектура глубокой нейронной сети для сегментации КТ-снимков

Layer (type)	OutputShapeParam #	Connectedto
input_3 (InputLayer)	[(None, 224, 224, 1)]	0
conv2d_38 (Conv2D)	(None, 224, 224, 32)	320
conv2d_39 (Conv2D)	(None, 224, 224, 32)	9248
batch_normalization_16 (BatchNormalization)	(None, 224, 224, 32)	128
max_pooling2d_8 (MaxPooling2D)	(None, 112, 112, 32)	0
dropout_8 (Dropout)	(None, 112, 112, 32)	0
conv2d_40 (Conv2D)	(None, 112, 112, 64)	18496
conv2d_41 (Conv2D)	(None, 112, 112, 64)	36928
batch_normalization_17 (BatchNormalization)	(None, 112, 112, 64)	256
max_pooling2d_9 (MaxPooling2D)	(None, 56, 56, 64)	0
dropout_9 (Dropout)	(None, 56, 56, 64)	0
conv2d_42 (Conv2D)	(None, 56, 56, 128)	73856
conv2d_43 (Conv2D)	(None, 56, 56, 128)	147584
batch_normalization_18 (BatchNormalization)	(None, 56, 56, 128)	512
max_pooling2d_10 (MaxPooling2D)	(None, 28, 28, 128)	0
dropout_10 (Dropout)	(None, 28, 28, 128)	0
conv2d_44 (Conv2D)	(None, 28, 28, 256)	295168

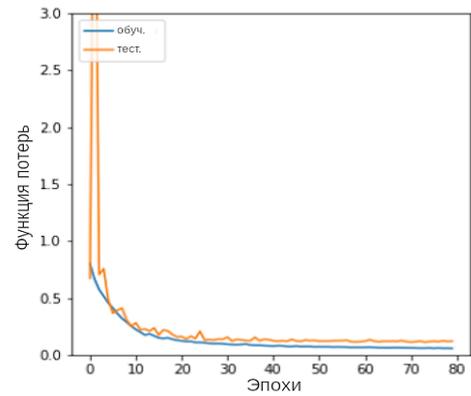
conv2d_45 (Conv2D)	(None, 28, 28, 256)	590080	['conv2d_44[0][0]']
batch_normalization_19 (Batch Normalization)	(None, 28, 28, 256)	1024	['conv2d_45[0][0]']
max_pooling2d_11 (MaxPooling2D)	(None, 14, 14, 256)	0	
['batch_normalization_19[0][0]']			
dropout_11 (Dropout)	(None, 14, 14, 256)	0	['max_pooling2d_11[0][0]']
conv2d_46 (Conv2D)	(None, 14, 14, 512)	1180160	['dropout_11[0][0]']
conv2d_47 (Conv2D)	(None, 14, 14, 512)	2359808	['conv2d_46[0][0]']
conv2d_transpose_8 (Conv2DTranspose)	(None, 28, 28, 256)	524544	['conv2d_47[0][0]']
concatenate_8 (Concatenate)	(None, 28, 28, 512)	0	['conv2d_transpose_8[0][0]', 'batch_normalization_19[0][0]']
batch_normalization_20 (Batch Normalization)	(None, 28, 28, 512)	2048	['concatenate_8[0][0]']
conv2d_48 (Conv2D)	(None, 28, 28, 256)	1179904	['batch_normalization_20[0][0]']
conv2d_49 (Conv2D)	(None, 28, 28, 256)	590080	['conv2d_48[0][0]']
conv2d_transpose_9 (Conv2DTranspose)	(None, 56, 56, 128)	131200	['conv2d_49[0][0]']
concatenate_9 (Concatenate)	(None, 56, 56, 256)	0	['conv2d_transpose_9[0][0]', 'batch_normalization_18[0][0]']
batch_normalization_21 (Batch Normalization)	(None, 56, 56, 256)	1024	['concatenate_9[0][0]']
conv2d_50 (Conv2D)	(None, 56, 56, 128)	295040	['batch_normalization_21[0][0]']
conv2d_51 (Conv2D)	(None, 56, 56, 128)	147584	['conv2d_50[0][0]']
conv2d_transpose_10 (Conv2DTranspose)	(None, 112, 112, 64)	32832	['conv2d_51[0][0]']
concatenate_10 (Concatenate)	(None, 112, 112, 128)	0	['conv2d_transpose_10[0][0]', 'batch_normalization_17[0][0]']
batch_normalization_22 (Batch Normalization)	(None, 112, 112, 128)	512	['concatenate_10[0][0]']
conv2d_52 (Conv2D)	(None, 112, 112, 64)	73792	['batch_normalization_22[0][0]']
conv2d_53 (Conv2D)	(None, 112, 112, 64)	36928	['conv2d_52[0][0]']
conv2d_transpose_11 (Conv2DTranspose)	(None, 224, 224, 32)	8224	['conv2d_53[0][0]']
concatenate_11 (Concatenate)	(None, 224, 224, 64)	0	['conv2d_transpose_11[0][0]', 'batch_normalization_16[0][0]']
batch_normalization_23 (Batch Normalization)	(None, 224, 224, 64)	256	['concatenate_11[0][0]']
conv2d_54 (Conv2D)	(None, 224, 224, 32)	18464	
['batch_normalization_23[0][0]']			
conv2d_55 (Conv2D)	(None, 224, 224, 32)	9248	['conv2d_54[0][0]']
conv2d_56 (Conv2D)	(None, 224, 224, 1)	33	['conv2d_55[0][0]']

На рисунке 12 показана динамика изменения метрики (2.6) и функции потерь (2.5) сети для определения зараженной области легких. При этом использовался оптимизатор Adam с размером пакета, равным 32 на протяжении 80 эпох со скоростью обучения 0.0005 и параметром импульса 0.95.

Тот же набор данных использовался в примере, приведенном в работе [66]. Точность описанной в этой работе глубокой нейронной сети ниже нашей по показателю $s=0.87$, который составляет там всего 0.78. Улучшение на 9% можно объяснить добавленными слоями нормализации и дополнением данных, а также выбранной функцией потерь.



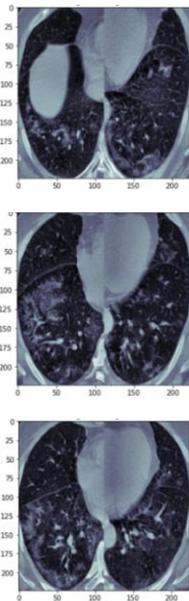
а)



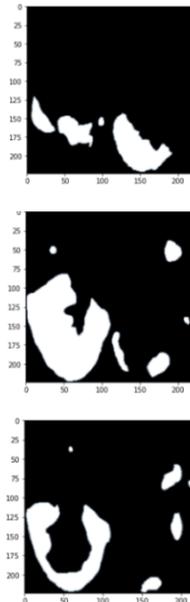
б)

Рисунок 13 – Динамика изменения: а) метрики Сёрнсена и б) функции потерь в процессе обучения нейронной сети для оценки поврежденной области

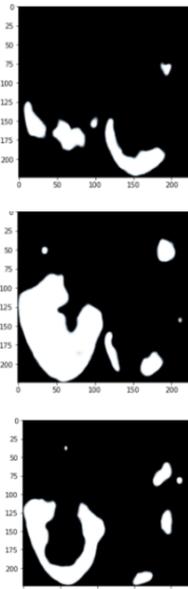
Примеры определения поврежденной области легких с помощью обученной нейронной сети показаны на рисунке 14.



а)



б)



в)

Рисунок 14 – Примеры КТ-срезов и поврежденных участков легких: а – исходное изображение, б – сегментация, выполненная врачами, в – сегментация, выполненная нейронной сетью

На рисунке 15 показана динамика изменения метрики (2.6) и функции потерь (2.5) сети для определения зараженной области легких. При этом использовался оптимизатор Adam с размером пакета, равным 32 на протяжении 40 эпох со скоростью обучения = 0.0005 и параметром импульса = 0.95.

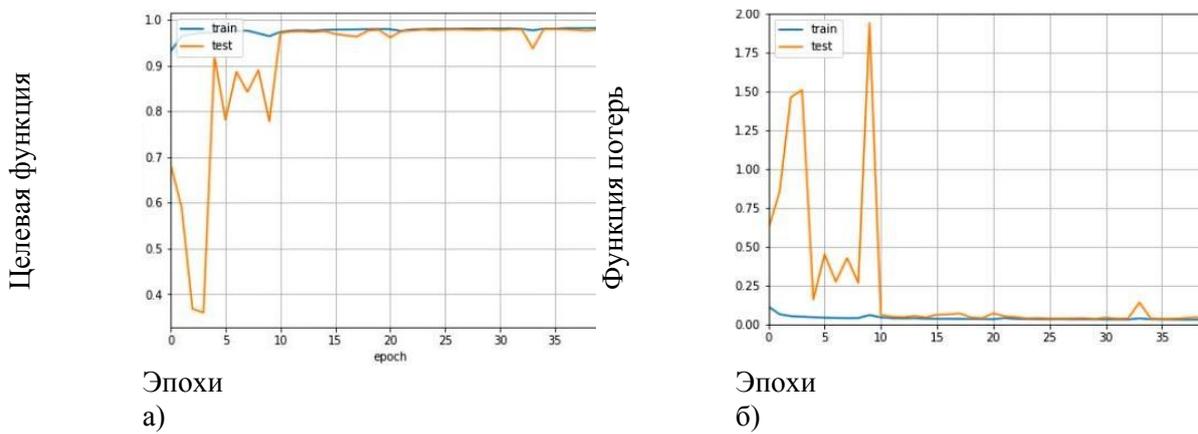


Рисунок 15 – Динамика изменения: а) метрики Сёренсена и б) функции потерь в процессе обучения нейронной сети для оценки области легких

Примеры сегментации КТ-слоев с обученной нейронной сетью с целью выделения области легких, где а – исходное изображение, б – сегментация, выполненная врачами, в – сегментация, выполненная обученной нейронной сетью, показаны на рисунке 16.

Таким образом, одна сеть определяет поврежденные области легких на слое КТ-снимка, а другая – область снимка, соответствующая легким.

Визуально маски зараженных областей и маски областей легких, определенные врачами и построенные нейронной сетью на рис 2.37, 2.39, практически совпадают.

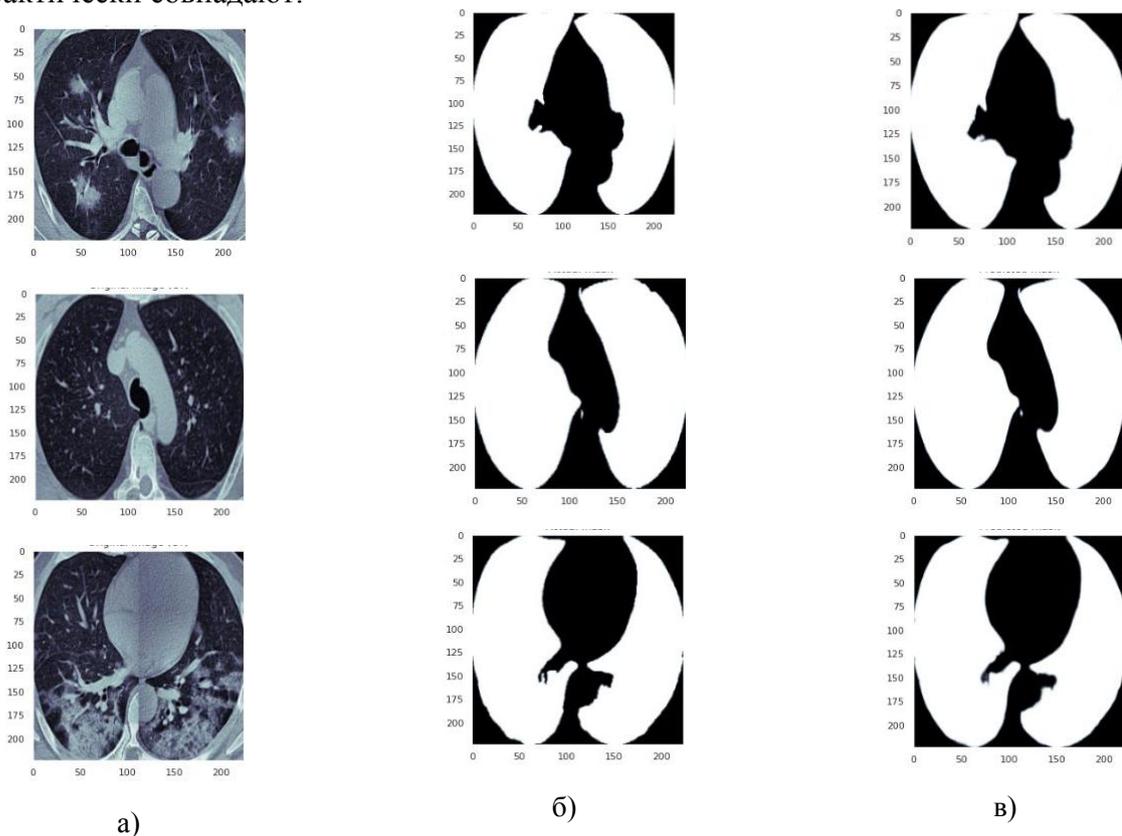


Рисунок 16 – Примеры КТ-снимков, областей легких, определенных врачами и обученной нейронной сетью

Такой результат достигается за счет снижения чувствительности нейронной сети к помехам во входных данных, возникающих из-за использования различного оборудования для получения КТ-снимков за счет слоев нормализации и дополнения данных. Затем

находится степень повреждения легких как отношение площади поврежденной области легких к площади легких на срезе. Окончательная степень повреждения легких определяется как среднее значение степени повреждения легких по всем срезам.

Выводы. Пандемия коронавирусной инфекции по-прежнему остается актуальной для Кыргызстана. Объясняется это циклическим чередованием подъёмов и спадов заболеваемости, характерных для COVID-19 и других респираторных вирусов. Кроме того, искусственно, с помощью методов геномной инженерии недавно была разработана гибридная версия вируса SARS-CoV-2, которая имеет летальность в 80%, что вызывает большие опасения у общественности. Поэтому нельзя исключать в будущем возникновения более заразного и более опасного штамма COVID-19.

В настоящее время многие независимые клинические исследования показали, что сегментация поражений COVID-19 на КТ-изображениях легких является важным шагом в клинической оценке лечения и может быть полезной в этом контексте из-за возможности таким образом выявить штамм, который вызвал заболевание (исходя из того, что разные штаммы в разной степени повреждают легкие), тяжесть и, возможно, стадию течения заболевания. Однако поиск точного метода сегментации поражений COVID-19 по КТ-снимкам является очень сложной задачей из-за больших различий в размере, форме и положении поражений, неопределенности границ, а также сложной структуры поврежденных областей.

Для получения более точных результатов система искусственного интеллекта должна использовать КТ-снимки для одновременной классификации с целью постановки диагноза и сегментацию изображений для определения поврежденных участков легких, что позволит более эффективно использовать такую систему в медицинских целях. Исходя из анализа литературных источников, в настоящее время известно очень немного систем, позволяющих решить сразу две эти задачи. Связано это с тем, что точность результатов очень сильно зависит от полноты наборов данных, размеров изображений и качества маркировки данных.

Поэтому была разработана система искусственного интеллекта на основе глубоких сверточных нейронных сетей для диагностики COVID-19 на основе анализа КТ-снимков. Для сегментации КТ-снимки разбиваются на двумерные изображения, которые обрабатываются с помощью двух модифицированных сетей U-Net. Одна сеть определяет поврежденные области легких на слое КТ-снимка, а другая – область снимка, соответствующую легким. Затем находится степень повреждения легких как отношение площади поврежденной области легких к площади легких на срезе. Окончательная степень повреждения легких определяется как среднее значение степени повреждения легких по всем срезам. По сравнению с аналогичными работами удалось повысить точность классификации на 2%, а также точность сегментации зараженной области на 9%. Повышение точности достигается за счет снижения чувствительности нейронной сети к помехам во входных данных, возникающих из-за использования различного оборудования для получения КТ-снимков. Для этого используется дополнение данных, включая масштабирование, вертикальное и горизонтальное отражение, вращение изображения, перенос, а также искажение перспективой. Кроме того, с целью уменьшения чувствительностью к исходным начальным значениям весов сети, помехам во входных данных и конкретному алгоритму обучения в архитектуру глубоких нейронных сетей добавлены слои пакетной нормализации. Таким образом, разработанная система искусственного интеллекта реализует возможность одновременной постановки диагноза и сегментации КТ-снимков с целью определения степени повреждения легких с большей точностью, чем получено ранее в других работах.

Литература

1. Макиев, В. Г. Использование искусственного интеллекта для диагностики COVID-19 по данным компьютерной томографии / В. Г. Макиев, Г. Г. Макиев // Актуальные научные исследования в современном мире. – 2020. – № 12-2(68). – С. 73–81. – EDN JEJBY.

2. <https://cabar.asia/ru/lokdauna-iz-za-covid-19-ne-budet-dlya-kyrgyzstana-eto-vopros-vyzhivaniya> (дата обращения: 17.10.2022)
3. <https://github.com/CSSEGISandData/COVID-19> (дата обращения: 25.10.2022)
4. https://www.ferra.ru/news/techlife/letalnost-80-sozdannyi-v-laboratorii-shtamm-koronavirusa-vyzval-opaseniya-u-pravitelstva-19-10-2022.htm?utm_source=yxnews&utm_medium=mobile (дата обращения: 23.10.2022)
5. <https://translate.google.com/website?sl=en&tl=ru&hl=ru&client=webapp&u=https://doi.org/10.1148/radiol.220533> (дата обращения: 07.09.2022)
6. Мареев В.Ю. Как оценивать результаты лечения больных с новой коронавирусной инфекцией (COVID-19)? Шкала оценки клинического состояния (ШОКС–КОВИД). / В.Ю.Мареев, Ю.Л.Беграмбекова, Ю.В. Мареев // Кардиология. – 2020 – 60(11) – С. 35–41. <https://doi.org/10.18087/cardio.2020.11.n1439>
7. <https://arxiv.org/pdf/1505.04597.pdf> (дата обращения: 07.09.2022)
8. Верзунов, С. Н. Диагностика различных видов пневмонии по рентгеновским снимкам грудной клетки / С. Н. Верзунов // Проблемы автоматизации и управления. – 2022. – № 1(43). – С. 84–94. – EDN ККСВК.
9. Верзунов, С. Н. Сравнение глубоких нейронных сетей на основе различных предварительно обученных CNN для диагностики COVID-19 по рентгеновским снимкам / С. Н. Верзунов, Х. А. Раимжанов // Проблемы автоматизации и управления. – 2021. – № 1(40). – С. 12-25. EDN: RVDXSO
10. Classification of Positive COVID-19 CT Scans Using Deep Learning. Khan, Muhammad Attique, Hussain, Nazar, Majid, Abdul, Alhaisoni, Majed, Syed Ahmad Chan, Bukhari, Kadry, Seifedine, Nam, Yunyoung, Yu-Dong, Zhang. // Computers, Materials, & Continua ; 66(3):2923-2938, 2021.
11. <https://core.ac.uk/download/pdf/53095817.pdf> (17.10.2022)
12. <https://arxiv.org/pdf/2208.02910.pdf> (дата обращения: 17.10.2022)
13. Ö. Çiçek, A. Abdulkadir, S. S. Lienkamp, T. Brox, and O. Ronneberger, “3D U-net: Learning dense volumetric segmentation from sparse annotation,” Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics), vol. 9901 LNCS, pp. 424–432, 2016, doi: 10.1007/978-3-319-46723-8_49/TABLES/3.
14. Fung, D.L.X., Liu, Q., Zammit, J. et al. Self-supervised deep learning model for COVID-19 lung CT image segmentation highlighting putative causal relationship among age, underlying disease and COVID-19. J TranslMed 19, 318 (2021). <https://doi.org/10.1186/s12967-021-02992-2>
15. <https://www.kaggle.com/datasets/andrewmvd/covid19-ct-scans> (11.05.2022).
16. Tang J, Mat Isa N. Adaptive image enhancement based on bi-histogram equalization with a clipping limit//Comput. Electr. Eng. 2014; 40:86 –103. doi:10.1016/j.compeleceng.2014.05.017
17. Франсуа Ш. JavaScript для глубокого обучения: TensorFlow.js./ Ф. Шолле, Э. Нильсон, С. Байлесчи, Шэнкунг Цэй // СПб.:Питер, 2021. — 576 с.

УДК 004.93

Б.Р. Сабитов¹, *sabitov.baratbek@gmail.com***Н.С. Сейткажиева²**, *naz.salbarovna@gmail.com***А.Дж. Картанова³**, *a.kartanova@mail.ru*¹*Кыргызский национальный университет им. Ж.Баласагына,*²*Кыргызский государственный университет им. И.Арабаева,*³*Кыргызский государственный технический университет им.И.Раззакова*

МЕТОДЫ КОМПЬЮТЕРНОГО ЗРЕНИЯ В ЗАДАЧАХ ПРОГНОЗИРОВАНИЯ БОЛЕЗНЕЙ РАСТЕНИЙ С ИСПОЛЬЗОВАНИЕМ ТРАНСФЕРНОГО ОБУЧЕНИЯ

В статье решается задача прогнозирования и разработки информационной системы распознавания болезней томатов путем построения модели нейронной сети на основе архитектуры MobileNetV2 с применением трансферного обучения. Преимуществом использования предварительно обученной нейронной сети MobileNetV2 является низкая потребность в вычислительных ресурсах и то, что она идеально подходит для развертывания на мобильных устройствах и на компьютерах без графического процессора. В результате исследований было подтверждено, что по фотографиям больных листьев томатов можно отличать больные помидоры от здоровых, диагностировать и прогнозировать различные виды заболеваний, построена архитектура нейронной сети для диагностики заболеваний томатов. Разработана модель для прогнозирования девяти видов болезней томатов. Получены результаты идентификации болезней растения и прогноза с точностью до 92,6% и более.

Ключевые слова: прогноз, модель, нейронная сеть, изображение, трансферное обучение, компьютерное зрение, глубокое обучение, оптимизация параметров, классификация заболеваний, болезни листьев растений.

Введение

Фермеры, выращивающие томаты, ежегодно несут убытки из-за различных болезней растения. Своевременная диагностика заболеваемости томатов на ранней стадии может предотвратить неурожай и экономические потери.

Рассмотрим задачу идентификации и прогнозирования болезней томатов. Существуют девять наиболее распространенных заболеваний томатов: Tomato_Bacterial_spot, Tomato_Early_blight, Tomato_Late_blight, Tomato_Leaf_Mold, Tomato_Septoria_leaf_spot, Tomato_Spider_mites_Two_spotted_spider_mite, Tomato__Target_Spot, Tomato__Tomato_YellowLeaf__Curl_Virus, Tomato__Tomato_mosaic_virus.

Постановка задачи состоит в том, чтобы построить модель глубокого обучения, которую можно развернуть как веб-систему или как мобильное приложение, с помощью которого фермер, лишь фотографируя лист помидора, может получить оценку о состоянии растения, а в случае болезни получить диагноз о болезни растения с дальнейшим прогнозом заболеваемости.

Обзор научных публикаций

В работе [1] была предложена методика классификации болезней листьев томата. В предлагаемом методе использовалась предварительно обученная модель AlexNet[2] в качестве экстрактора признаков и CNN для классификации заболеваний. При этом модель достигала точности 76,1%. В другом исследовании [3] была принята модель CNN для классификации болезней листьев томатов на основе трансферного обучения с точной настроенной моделью MobileNetV2. Результат показал, что точная настройка модели MobileNetV2 может идентифицировать заболевание с точностью более 90%. В работе [4] авторы предложили пользовательскую модель CNN для классификации болезней томатов со средней точностью 91,2%. В статье [5] была предложена улучшенная модель CNN на основе модуля Inception и расширенной свертки для выявления 14 различных болезней сельскохозяйственных культур. Предложенная модель CNN достигала точности 99,37% с использованием набора данных PlantVillage. В работе [6] авторы предложили

облегченную модель CNN с восемью скрытыми слоями для идентификации изображений болезней растений томата из набора данных PlantVillage и получили точность 98,4%. Результаты, представленные в исследовании [7], указывают на эффективность предварительно обученных CNN DenseNet169 и ResNet152V2 для решения задач прогнозирования, что может помочь при создании более сложных и совершенных CNN.

Методы исследования

В целях увеличения и улучшения набора данных PlantVillage применяли методы библиотеки TensorFlow и использовали различные технологии предварительной обработки данных, добавляя случайные преобразования изображений. Также провели оптимизацию предварительной выборки с использованием трансферного обучения. Создали базовую модель из предварительно обученной нейронной сети с использованием в качестве средства извлечения признаков, добавляя слой классификатора поверх базовой модели. Заморозили базовые слои для получения начальной точности на 80% и более.

Одним из способов еще большего повышения производительности является обучение (или «точная настройка») весов верхних слоев предварительно обученной модели наряду с обучением добавленного классификатора. В процессе обучения веса должны быть настроены на функции с общих карт объектов, связанных с конкретным набором данных. Поэтому проводили тонкую настройку всей модели, чтобы адаптировать эти специализированные функции для работы с новым набором данных. После оптимизации производительности модели с помощью оптимизатора Adam была достигнута точность проверки, более 92,02% и потеря проверки 0,0315 за 11 эпох, что указывает на то, что наша модель хорошо работает с новыми данными. Приведем соответствующий код:

```
from tensorflow.compat.v1 import ConfigProto
from tensorflow.compat.v1 import InteractiveSession

config = ConfigProto()
config.gpu_options.per_process_gpu_memory_fraction = 0.5
config.gpu_options.allow_growth = True
session = InteractiveSession(config=config)
```

Базу данных, состоящую из входных изображений, имеющих размер 256x256, приводили к масштабу 224x224 для дальнейшего использования в MobileNetV2.

IMAGE_SIZE = 224, BATCH_SIZE = 16

Для загрузки изображений из каталога базы данных в набор данных использовали тензорный поток TensorFlowtf, загрузку выполняли с помощью API-интерфейса. Эта функция загружает изображения из указанного нами каталога в рабочую среду и конвертирует изображения в тензоры. При помощи модуля tf для каждого RGB изображения загружаются также их метки. Приведем соответствующий код:

```
dataset = tf.keras.utils.image_dataset_from_directory(
    "D:/Dataset_Tomato/",
    shuffle = True,
    image_size = (IMAGE_SIZE, IMAGE_SIZE),
    batch_size = BATCH_SIZE,
)
```

Found 18160 files belonging to 10 classes.

Таким образом, база данных состояла из 10 классов, содержащих 18160 изображений. Рассмотрим метки нашего набора данных. Для каждого класса – типа болезни, имеются метки классов:

```
class_names = dataset.class_names
class_names
```

```
['Tomato__Bacterial_spot',
'Tomato__Early_blight',
'Tomato__Late_blight',
'Tomato__Leaf_Mold',
'Tomato__Septoria_leaf_spot',
'Tomato__Spider_mites_Two_spotted_spider_mite',
'Tomato__Target_Spot',
'Tomato__Tomato_Yellow_Leaf_Curl_Virus',
'Tomato__Tomato_mosaic_virus',
'Tomato__healthy']
```

База данных содержала девять классов изображений больных листьев и один класс изображений фиксировали как здоровые растения. Увеличивая размеры изображений, визуализировали все изображения с заголовками болезней. При визуализации использовали функции `imshow()`, `plt.imshow()` библиотеки `Matplotlib`, которые преобразуют изображения в трехмерный массив `numpy`. Каждому классу изображений растения присвоили заголовок. Результат применения этих функций показан ниже:

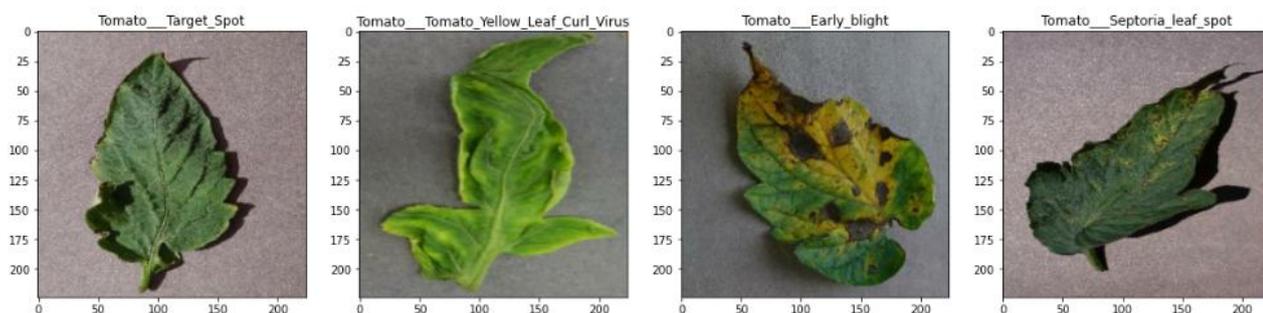


Рисунок 1 – Присвоение меток к классам болезней томатов

Вычисляли количество изображений в каждом классе. Это программная реализация распределения меток классов.

```
y = np.concatenate([y for x, y in dataset], axis=0)
print(np.bincount(y))#np.bincount(array) подсчитывает появление каждого элемента в массиве.
```

```
[2127 1000 1909 952 1771 1676 1404 5357 373 1591]
```

Распределение количества изображений листьев томатов по каждому классу болезней показано на рисунке 2. На рисунке 3 приведена количественная шкала и тепловая карта болезней томатов.

Из рисунка 2 видно, что класс `label7('Tomato__Tomato_YellowLeaf__Curl_Virus')` имеет наибольшее количество меток (5357), а класс `label8('Tomato__Tomato_mosaic_virus')` имеет наименьшее количество меток (373). Сохраняя пропорции, разделили набор данных на наборы `train`, `val`, `test` в соотношении 7:2:1.

Tomato__Bacterial_spot	2127
Tomato__Early_blight	1000
Tomato__healthy	1591
Tomato__Late_blight	1909
Tomato__Leaf_Mold	952
Tomato__Septoria_leaf_spot	1771
Tomato__Spider_mites_Two_spotted_spider_mite	1676
Tomato__Target_Spot	1404
Tomato__Tomato_mosaic_virus	373
Tomato__Tomato_Yellow_Leaf_Curl_Virus	5357

```
print(f"Total number of images: {sum(image_counts.values())}")
```

Total number of images: 18160

Рисунок 2 – Количественные показатели классов

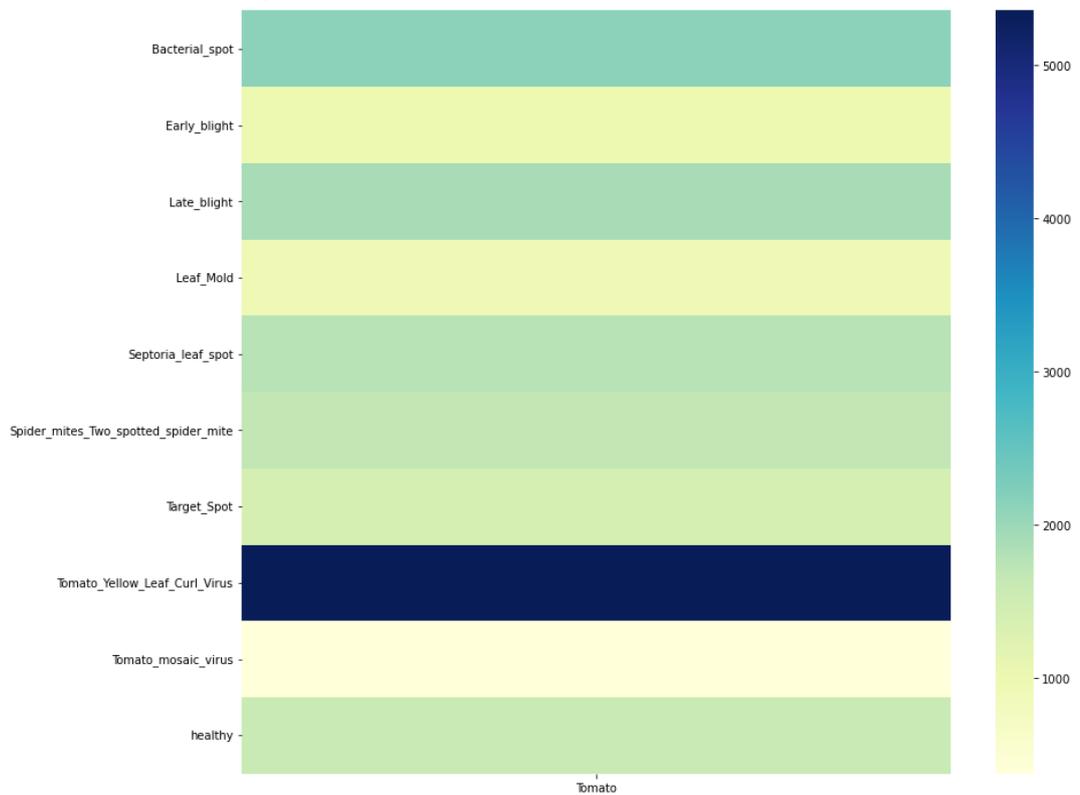


Рисунок 3 – Количественная шкала и тепловая карта болезней томатов

Разбили изображения болезней листьев томатов по классам заболеваемости, как показано на рисунке 4.



Рисунок 4 – Разбиение изображений болезней томатов по типу болезней

В случае отсутствия большого набора данных изображений хорошей практикой является искусственное введение разнообразия выборок путем применения к обучающим изображениям случайных, но реалистичных преобразований, таких как вращение и горизонтальное отражение. Это помогает представить модель в различных аспектах обучающих данных и уменьшить переобучение моделей.

Например, имеется один образец обучающих данных, тогда мы создавали из него четыре образца, используя горизонтальное отражение, вращение, масштабирование, контрастирование и далее использовали все эти пять полученных изображений для обучения:

```
data_augment = tf.keras.Sequential(
    [layers.experimental.preprocessing.RandomFlip("horizontal_and_vertical"),
     layers.experimental.preprocessing.RandomRotation(0.2)])
```

До сих пор предварительно обрабатывали данные – значения пикселей, которые находились в диапазоне [0,255]. Но данные MobieNetV2 нормированы и находятся в диапазоне [-1,1]. Поэтому изменяли масштаб значений пикселей так, чтобы они входили в диапазон [-1,1]. Для этого использовали функцию preprocess_input():

```
preprocess_input = tf.keras.applications.mobilenet_v2.preprocess_input
```

Создали базовую модель из предварительно обученной модели MobileNet. Самый последний слой этой модели содержит классификационный слой, который выполняет разделения изображений на классы. Этот слой исключали, а оставшиеся слои использовали для создания модели. Последний слой перед операцией выравнивания содержит слой узкого места. Функции слоя узкого места сохраняют большую общность по сравнению с конечным/верхним слоем. Сначала создали экземпляр модели MobileNetV2 с предварительно загруженными весами, которые обучались на множестве ImageNet. Если указать аргумент include_top=False, тогда загружается сеть, которая не включает слои классификации сверху, что идеально подходит для извлечения признаков. Этот экстрактор – элемент, функции преобразует каждое изображение 224x224x3 в блок функций 7x7x1280.

Извлечение признаков

На этом шаге заморозили базовую модель свертки, созданную на предыдущем шаге, и использовали ее в качестве составляющих признаков. Кроме того, добавили классификатор поверх него и обучили классификатор верхнего уровня. Установив

layer.trainable = False, мы предотвратили обновление весов в данном слое во время обучения. MobileNet V2 имеет много слоев, поэтому установка флага обучаемости всей модели в значение False заморозит их все. В архитектуре модели также содержится много слоев tf.keras.layers.BatchNormalization. Эти слои являются особыми случаями, и в контексте точной настройки следует соблюдать меры предосторожности. Если для них установить флаг layers.trainable = False, слой BatchNormalization преобразуется в режим вывода и не будет обновлять свою статистику среднего значения и дисперсии. Когда размораживали модель, содержащую слои BatchNormalization, для тонкой настройки, оставили слои BatchNormalization в режиме вывода, передав training = False в случае вызова базовой модели. В противном случае обновления, примененные к необучаемым весам, разрушат то, чему научилась модель. Таким образом, архитектура нейронной сети имеет следующий вид:

```
base_model.trainable = False

base_model.summary()

Model: "mobilenetv2_1.00_224"
-----
Layer (type)                Output Shape              Param #   Connected to
-----
input_1 (InputLayer)        [(None, 224, 224, 3
                           )]
Conv1 (Conv2D)              (None, 112, 112, 32)    864      ['input_1[0][0]']
bn_Conv1 (BatchNormalization) (None, 112, 112, 32)    128      ['Conv1[0][0]']
Conv1_relu (ReLU)          (None, 112, 112, 32)    0        ['bn_Conv1[0][0]']
expanded_conv_depthwise (Depth (None, 112, 112, 32)    288      ['Conv1_relu[0][0]']
wiseConv2D)
block_16_depthwise_relu (ReLU) (None, 7, 7, 960)      0        ['block_16_depthwise_BN[0][0]']
block_16_project (Conv2D)   (None, 7, 7, 320)      307200   ['block_16_depthwise_relu[0][0]']
block_16_project_BN (BatchNorm (None, 7, 7, 320)      1280     ['block_16_project[0][0]']
alization)
Conv_1 (Conv2D)            (None, 7, 7, 1280)     409600   ['block_16_project_BN[0][0]']
Conv_1_bn (BatchNormalization) (None, 7, 7, 1280)     5120     ['Conv_1[0][0]']
out_relu (ReLU)           (None, 7, 7, 1280)     0        ['Conv_1_bn[0][0]']
-----
Total params: 2,257,984
Trainable params: 0
Non-trainable params: 2,257,984
```

Чтобы сгенерировать прогнозы из блока функций, усредняли пространственные местоположения 7x7, использовали слой tf.keras.layers.GlobalAveragePooling2D для преобразования функций в один вектор из 1280 элементов для каждого изображения.

```
global_average_layer = tf.keras.layers.GlobalAveragePooling2D()
feature_batch_average = global_average_layer(feature_batch)
print(feature_batch_average.shape)

(16, 1280)
```

Для генерации прогнозов добавили на слой softmax.

```
prediction_layer = tf.keras.layers.Dense(10, activation="softmax")
```

Создавали модель и производили масштабирование от -1 до 1, применяя увеличение данных и далее отправив их в базовую модель, а затем использовали глобальный средний слой и слой прогнозирования. Ниже приводится листинг кода:

```
inputs = tf.keras.Input(shape=(224, 224, 3))
x = preprocess_input(inputs)
x = base_model(x, training=False)
x = global_average_layer(x)
x = tf.keras.layers.Dropout(0.2)(x)
outputs = prediction_layer(x)
model = tf.keras.Model(inputs, outputs)
```

Вводили параметры модели и компилировали ее, листинг кода:

```
base_learning_rate = 0.0001
model.compile(optimizer=tf.keras.optimizers.Adam(learning_rate=base_learning_rate),
              loss=tf.keras.losses.SparseCategoricalCrossentropy(from_logits=False),
              metrics=['accuracy'])
```

Затем компилировали модель. Результат процесса обучения с 10 эпохами приводится ниже:

```
history = model.fit(train, epochs=initial_epochs, validation_data=val, batch_size = 16, verbose =1)

Epoch 1/10
794/794 [=====] - 195s 242ms/step - loss: 1.5608 - accuracy: 0.4877 - val_loss: 1.0930 - val_accuracy:
0.6814
Epoch 10/10
794/794 [=====] - 205s 258ms/step - loss: 0.4608 - accuracy: 0.8461 - val_loss: 0.4574 - val_accuracy:
0.8579
```

Оценка в метрике проверки лучше, чем в метрике обучения, основной фактор заключается в том, что такие слои, как `tf.keras.layers.BatchNormalization` и `tf.keras.layers.Dropout`, влияют на точность во время обучения. Они отключаются при расчете потерь проверки. В меньшей степени это также связано с тем, что метрики обучения сообщают среднее значение для эпохи, в то время как метрики проверки оцениваются после эпохи, поэтому метрики проверки видят модель, которая обучалась немного дольше.

Тонкая настройка модели

В эксперименте по извлечению признаков проводили обучение только несколько слоев поверх базовой модели `MobileNetV2`. Веса предварительно обученной сети не обновлялись во время обучения. Одним из способов еще большего повышения производительности является обучение (или «точная настройка») весов верхних слоев предварительно обученной модели наряду с обучением добавленного нами классификатора. В процессе обучения веса с общих наборов объектов должны быть настроены на функции, связанные с конкретным набором данных. Кроме того, необходимо настроить небольшое количество верхних слоев, а не всю модель `MobileNetV2`. В большинстве сверточных сетей чем выше уровень, тем более он специализирован. Первые несколько слоев изучают очень простые и общие функции, которые применимы практически ко всем типам изображений. По мере того, как мы поднимаемся выше, функции становятся все более специфичными для набора данных, на котором была обучена модель.

Цель тонкой настройки — адаптировать эти специализированные функции для работы с новым набором данных, а не перезаписывать общее обучение. Положив `base_model.trainable = True`, разморозили верхние слои модели. Далее продолжили обучение модели. Ранее обучение модели проводили на 10 эпохах. Добавили еще одну

эпоху и увидели, как увеличение шага эпохи повысила точность модели. Ожидали, что точность повысится на несколько процентных пунктов, как показано в окне результата:

```

fine_tune_epochs = 1
total_epochs = initial_epochs + fine_tune_epochs

history_fine = model.fit(train, epochs=total_epochs, initial_epoch=history.epoch[-1], validation_data=val, batch_size = 16, verbose
Epoch 10/11
794/794 [=====] - 280s 349ms/step - loss: 0.4538 - accuracy: 0.8426 - val_loss: 0.2551 - val_accuracy:
0.9157
Epoch 11/11
794/794 [=====] - 279s 351ms/step - loss: 0.2448 - accuracy: 0.9164 - val_loss: 0.2173 - val_accuracy:
0.9268
    
```

Рассмотрим графики обучения и валидации точности модели/обучения и проверки ошибки модели при тонкой настройке последних нескольких слоев базовой модели MobileNetV2 и обучении классификатора поверх нее, см. рисунки 5 – 6. Потери при проверке намного выше, чем потери при обучении, поэтому получено некоторое переоснащение. В данном случае также получено некоторое переобучение модели, поскольку новый обучающий набор относительно мал и похож на исходные наборы данных MobileNetV2.

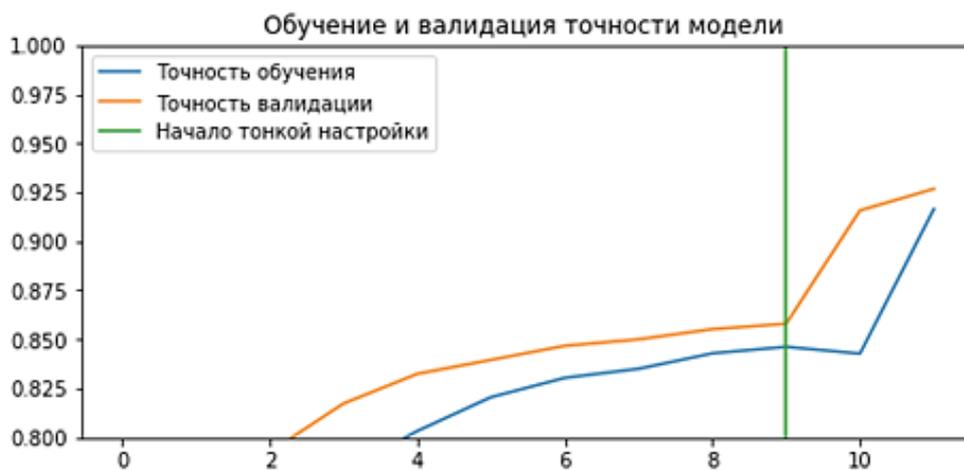


Рисунок 5 – Валидация точности модели на данных обучения

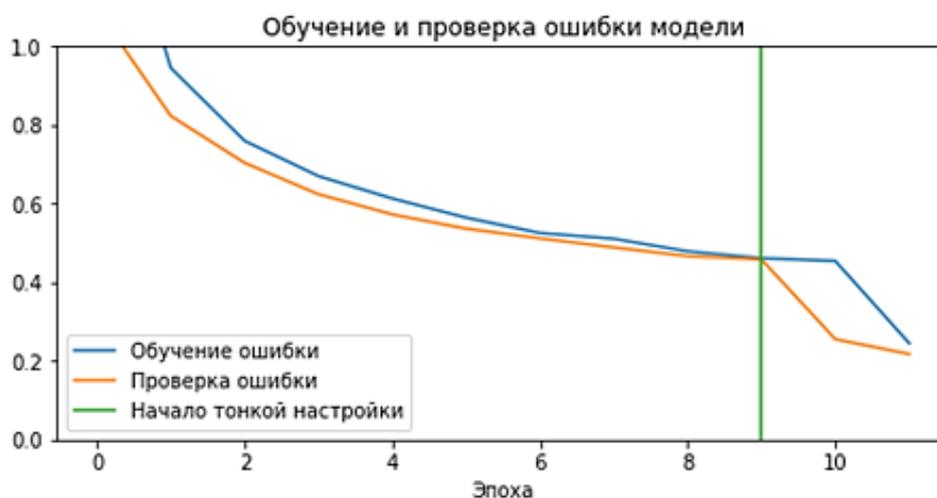


Рисунок 6 – Проверка ошибки модели на данных обучения

В целях проверки производительности модели протестировали первое изображение на прогноз, см. рисунок 7.

Вывод первого изображения

Фактическая метка первого изображения: Tomato__Septoria_leaf_spot

Метка прогноза Tomato__Septoria_leaf_spot

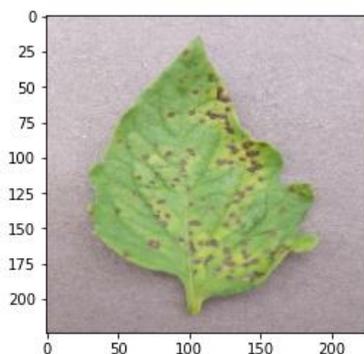


Рисунок 7 – Тестирование производительности модели для первого изображения

<p>Actual: Tomato__Septoria_leaf_spot, Прогноз: Tomato__Septoria_leaf_spot Точность модели: 91.57%</p>	<p>Actual: Tomato__Spider_mites_Two_spotted_spider_mite, Прогноз: Tomato__Spider_mites_Two_spotted_spider_mite. Точность модели: 90.96%</p>	<p>Actual: Tomato__Late_blight, Прогноз: Tomato__Late_blight. Точность модели: 94.96%</p>
		
<p>Actual: Tomato__Tomato_Yellow_Leaf_Curl_Virus, Прогноз: Tomato__Tomato_Yellow_Leaf_Curl_Virus. Точность модели: 100.0%</p>	<p>Actual: Tomato__Late_blight, Прогноз: Tomato__Bacterial_spot. Точность модели: 49.3%</p>	<p>Actual: Tomato__Leaf_Mold, Прогноз: Tomato__Leaf_Mold. Точность модели: 94.51%</p>
		
<p>Actual: Tomato__Spider_mites_Two_spotted_spider_mite, Прогноз: Tomato__Spider_mites_Two_spotted_spider_mite. Точность модели: 98.79%</p>	<p>Actual: Tomato__healthy, Прогноз: Tomato__healthy. Точность модели: 95.79%</p>	<p>Actual: Tomato__Tomato_Yellow_Leaf_Curl_Virus, Прогноз: Tomato__Tomato_Yellow_Leaf_Curl_Virus. Точность модели: 99.58%</p>
		

Рисунок 8 – Тестирование производительности и точности модели по классам заболеваемости томатов

Проводили несколько проверочных тестов на прогноз болезни томатов по классам заболеваемости, в результате получен ожидаемый прогноз по разработанной модели. Основные результаты вычислительных экспериментов по прогнозированию болезней томатов в процентном соотношении показаны на рисунке 8.

Очевидно, что разработанная модель позволяет прогнозировать различные болезни томатов с достаточной и надежной точностью. Процент прогноза в некоторых случаях достигает 99% и выше. Лишь в отдельных случаях видно, что модель ошибается. Построенная модель обладает обобщающим свойством, это показывают полученные прогнозы с большой точностью на тестовых данных, не участвующих при обучении и проверке модели. В целом применение трансферного обучения для прогнозирования болезней томатов является удобным инструментом.

Заключение

В настоящей работе представлены визуализация и предварительный анализ данных больных и здоровых листьев томатов. Объем базы данных состоит из 18160 изображений. Для создания модели использовано трансферное обучение с применением предварительно обученной модели MobileNet V2, которая обучалась на больших данных. Для конкретной задачи сельского хозяйства создан экземпляр модели с загруженными весами MobileNet V2, который обучался на множестве большого набора изображений ImageNet и применен к задаче построения нейронной сети для построения модели глубокого обучения с тонкой настройкой. Построена модель для прогнозирования 9 видов болезней томатов. Получены результаты прогноза и идентификации болезней растения с точностью до 92.6% и более.

Литература

1. Batool, A., Hyder, S.B., Rahim, A., Waheed, N., Asghar, M.A.: Fawad: Classification and identification of tomato leaf disease using deep neural network. In: 2020 Int. Conf. Eng. Emerg. Technol. ICEET 2020 (2020)Google Scholar
2. Krizhevsky, A., Sutskever, I., Hinton, G.E.: ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems. (2012)Google Scholar
3. Zaki, S.Z.M., Zulkifley, M.A., MohdStofa, M., Kamari, N.A.M., Mohamed, N.A.: Classification of tomato leaf diseases using mobilenet v2. IAES Int. J. Artif. Intell. (2020)Google Scholar
4. Agarwal, M., Singh, A., Arjaria, S., Sinha, A., Gupta, S.: ToLeD: Tomato leaf disease detection using convolution neural network. In: Procedia Computer Science. (2020)CrossrefGoogleScholar
5. Wang, L., Sun, J., Wu, X., Shen, J., Lu, B., Tan, W.: Identification of crop diseases using improved convolutional neural networks. IET Comput. Vis. 14(7), 538– 545 (2020)Wiley Online Library Web of Science@Google Scholar
6. Agarwal, M., Gupta, S.K., Biswas, K.K.: Development of Efficient CNN model for Tomato crop disease identification. Sustain. Comput. Informatics Syst. 28, 100407 (2020)CrossrefWeb of Science@Google Scholar
7. Верзунов, С. Н., Раимжанов, Х. А. Сравнение глубоких нейронных сетей на основе различных предварительно обученных CNN для диагностики COVID-19 по рентгеновским снимкам / С. Н. Верзунов, Х. А. Раимжанов // Проблемы автоматизации и управления. – 2021. – № 1(40). – С. 12–25.

УДК 004

*Сабитов Б.Р.¹, sabitov.baratbek@gmail.com**Сейтказиева Н.С.², naz.salbarovna@gmail.com**Картанова А.Дж.³, a.kartanova@mail.ru*¹*Кыргызский национальный университет им. Ж.Баласагына,*²*Кыргызский государственный университет им. И.Арабаева,*³*Кыргызский государственный технический университет им.И.Раззакова*

ИДЕНТИФИКАЦИЯ БОЛЕЗНЕЙ ТОМАТОВ НА ОСНОВЕ МНОГОКЛАССОВОЙ КЛАССИФИКАЦИИ

Томаты – одна из самых потребляемых культур в мире. Томаты различаются по качеству в зависимости от того, как они выращиваются и удобряются, таким образом, качественная урожайность играет ключевую роль и является конечной целью любого фермера. Но томаты, как и все растения, имеют множество болезней, которые влияют на их урожайность. Болезнь листьев является основным фактором, влияющим на количество и качество урожая. Поэтому крайне важны ранняя диагностика и классификация болезней томатов, которые уменьшат заболеваемость и повысят урожайность томатов. В статье предлагается решение актуальной проблемы точного выявления болезней томатов на ранних стадиях для дальнейшего информирования фермеров. Развитие новых технологий, алгоритмов и методов машинного и глубокого обучения позволяет автоматизировать решение подобных задач. Одним из мощных и современных подходов являются сверточные нейронные сети(CNN) для эффективного определения и классификации болезней томатов. В вычислительном эксперименте использовался набор данных среды Google Colab, содержащий 16484 изображения листьев томатов, выделены девять классов больных листьев, пораженных различными заболеваниями, и один класс здоровых листьев. Для классификации болезней томатов построены нейронные сети. Точность модели, обученной на протяжении 30 эпох, дала результат в 91,26%, а на тестовых данных точность достигала до 80%.

Ключевые слова: CNN, машинное обучение, компьютерное зрение, эксперимент, классификация изображений.

Введение

Роль растений в жизни и хозяйственной деятельности человека известна. Благодаря процессу фотосинтеза происходит образование кислорода, который жизненно необходим для всех обитателей земной поверхности. Растения – богатейший источник витаминов и минералов. Помидор – богатое питательными веществами растение, употребляемое в пищу, овощная культура, широко культивируемая не только во всех регионах Кыргызстана, а также в мире в целом. Помидор, вносящий значительный вклад в сокращение бедности, рассматривается как источник дохода для многих фермерских хозяйств и домохозяйств.

Помидоры являются одной из самых богатых питательными веществами культур на планете, и их выращивание и производство оказывают значительное влияние на экономику сельского хозяйства. Помидоры отличаются высокими питательными, вкусовыми и диетическими качествами, содержащимися в томатах, холин понижает содержание холестерина в крови, предупреждает жировое перерождение печени, повышает иммунные свойства организма, способствует образованию гемоглобина, а также обладает противовоспалительным действием[1]. Спрос на помидоры также увеличивается в результате их широкого использования. По статистике, мелкие фермеры производят более 80% сельскохозяйственной продукции [2], из-за болезней и вредителей погибает около 50% их урожая. Болезни и паразитические насекомые являются ключевыми факторами, влияющими на рост томатов, что делает необходимым исследование в диагностике болезней полевых культур.

Ручная идентификация вредителей и патогенов неэффективна и дорога. Поэтому необходимо предоставить фермерам автоматизированные решения обработки изображений на основе искусственного интеллекта (ИИ). Изображения используются и принимаются в качестве источника информации для выявления заболеваний в приложениях компьютерного зрения на основе распознавания образов. Благодаря

наличию соответствующих программных пакетов изображения обрабатывают с помощью технологии интеллектуальной идентификации изображений, которая повышает эффективность распознавания, снижает затраты и повышает точность.

Ранняя и точная диагностика помогает снизить риск экологического ущерба. Без систематического выявления заболеваний страдает качество и количество продукции. Это оказывает дальнейшее пагубное влияние на экономику страны [1]. Пестициды, широко применяемые в последнее время, негативно влияют на сельскохозяйственную экосистему. В результате продуктивность растений может быть снижена на 50% из-за стресса или болезней. Поэтому необходимы быстрые и эффективные методы классификации и обнаружения болезней, которые могут помочь агроэкосистеме.

Передовые технологии обнаружения болезней, рассматриваемые в данной работе, это такие, как распознавание образов и нейронные сети, позволяют разработать автоматизированные системы, способные обнаруживать болезни растений, в частности помидоров, на ранних стадиях. Первый шаг в обнаружении болезни – это осмотр растения, а затем на основе предыдущего опыта определение того, с чем работать, это следующий шаг. Этому методу не хватает научной последовательности, потому что опыт фермеров разный, в результате чего процесс менее надежен. Существует вероятность того, что фермеры неправильно классифицируют заболевание, и неправильное лечение нанесет вред растению. Точно так же выезд специалистов в области предметной области стоит дорого. Существует потребность в разработке автоматизированных методов обнаружения и классификации болезней растения на основе изображений, которые могут взять на себя роль эксперта в предметной области.

Согласно прошлым исследованиям, 80–90% болезней растений проявляется на листьях. Преобладают такие болезни овощей, как бактериоз, поздняя мучнистая роса, пятнистость листьев, томатная мозаика, желтая курчавость. Они серьезно влияют на рост растений, что приводит к снижению качества и количества продукции. Отслеживание фермерами и распознавание различных форм болезни с зараженными растениями занимает много времени. Оценка фермерами типа болезни растений может быть ошибочной. Это решение может привести к недостаточным и контрпродуктивным мерам защиты, а раннее обнаружение может снизить затраты на обработку, уменьшить воздействие химических веществ на окружающую среду и свести к минимуму риск потерь.

Обзор научных публикаций и их анализ

Различные исследователи использовали передовые технологии, такие как машинное обучение и различные архитектуры нейронных сетей InceptionV3, сеть VGG16 и SqueezeNet, для создания автоматизированных систем обнаружения заболеваний. В них используются высокоточные методы выявления болезней растения томаты по его листьям. Кроме того, исследователи предложили множество решений, основанных на глубоком обучении, для выявления и классификации заболеваний растения томаты, как описано ниже в [3, 4,5,6,7].

В работе [3,4] предложена предварительно обученная сетевая модель для обнаружения и классификации болезней томатов с точностью до 94–95%. Для диагностики болезней томатов использовались модели AlexNet и VGG19. Построенная модель при этом показывает точность 97,49% [8]. Трансферное обучение и модель CNN используются для точного выявления болезней и могут достигать точности 95% [9]. Нейронная сеть для определения и классификации состояния листьев томатов с использованием трансферного обучения в качестве механизма глубокого обучения на основе AlexNet достигла точности 95,75% [10,11]. Архитектура нейронной сети ReZnet-50 была разработана для выявления болезней листьев томатов с малым числом изображений с названиями «пятнистость», «фитофтороз» и «желтая курчавость листьев». Функция Network Activation для сравнения была изменена на Leaky-ReLU, а размер ядра был обновлен до 11×11 для первого слоя

свертки. Модель предсказывает класс заболеваний с точностью 98,30% и точностью 98,0% после нескольких повторений [12]. Упрощенная восьмиуровневая модель CNN была предложена для обнаружения и классификации болезней листьев томатов [13]. Нейронные сети на основе предварительно обученных моделей CNN EfficientNetB3 и ResNet152 на протяжении 5 эпох до достижения максимальной точности показали наилучшие результаты (83%) по сравнению с другими 28 моделями.[14]

Сверточные нейронные сети (CNN) достигли больших успехов при обработке изображений с использованием методов глубокого обучения. Существует ряд локальных приложений и веб-систем для обнаружения болезней сельскохозяйственных культур, которые появились в последнее время. Эти модели можно использовать в разработке приложений для консультаций с экспертами или для скрининга. Эти инструменты могут помочь улучшить устойчивые методы ведения сельского хозяйства и обеспечить продовольственную безопасность. Имеется ряд работ, в которых использовались сверточные нейронные сети CNN для выявления болезней растений и дана оценка точности этих моделей. В качестве базы данных больных и здоровых растений использован набор данных PlantVillage. Авторы недооценивали производительность топологий нейронных сетей с использованием типичных показателей производительности, таких как F1-оценка, полнота, точность и т.д. Вместо этого они оценивали точность модели и время вывода. В настоящей работе предлагается построение новой модели глубокой нейронной сети.

Материалы и методы

В эксперименте для достижения результатов использовали передовые методологии, модели и наборы данных. Набор данных состоял из девяти уникальных классов заболеваний томатов и одного класса здоровых листьев. Использовали общедоступные данные и названия болезней для идентификации классов изучаемого набора данных.

Образцы изображений больных листьев с патогеном: (a) бактериальная пятнистость (*Xanthomonas vesicatoria*), (b) ранняя гниль (*Alternaria solani*), (c) фитофтороз (*Phytophthora infestans*), (d) листовая плесень (*Cladosporium fulvum*), (e) септориоз_листовая пятнистость (грибок *Septoria lycopersici*), (f) паутиные клещи (*Spider_Mites*, флоридана), (g) мишенеобразная пятнистость(*Target_Spot*, грибок *Corynespora*), (h) вирус мозаики томата (*Tomato_Mosaic_Virus*, тобамовирус), (i) вирус желтого скручивания листьев томата (*Tomato_Yellow_Leaf_Curl_Virus*, род *Begomovirus*), (j) здоровые листья (*Healthy_Leaf*).

Для отображения нескольких выходов использовали пакет InteractiveShell.

```
from IPython.core.interactiveshell import InteractiveShell
```

```
InteractiveShell.ast_node_interactivity = "all"
```

В локальный диск загружали все данные о болезнях томатов со следующими метками:

```
['Tomato__Bacterial_spot',
 'Tomato__Early_blight',
 'Tomato__healthy',
 'Tomato__Late_blight',
 'Tomato__Leaf_Mold',
 'Tomato__Septoria_leaf_spot',
 'Tomato__Spider_mites_Two_spotted_spider_mite',
 'Tomato__Target_Spot',
 'Tomato__Tomato_mosaic_virus',
 'Tomato__Tomato_Yellow_Leaf_Curl_Virus']
```

Изучали по отдельности зараженные (больные) и здоровые листья. Общее количество изображений незараженных(здоровых) листьев томатов в нашем случае составляет 1591.

Предварительная обработка данных изображений

Проводили просмотр случайно выбранного изображения здорового листа томата, рисунок 1:

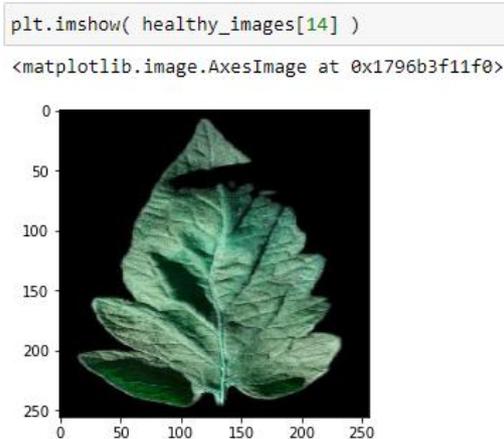


Рисунок 1 – Образец изображения здорового листа томата

Нейронные сети обрабатывают только формы, записанные в виде тензора, поэтому все изображения преобразовывали в тензор. Все изображения с размерами 256x256 пикселей конвертировали в массив numpy как трехмерный тензор, например (1591, 256, 256, 3). У нас всего 1591 запись здоровых листьев томатов в виде тензора 3-го порядка. Выбирали случайные изображения больных листьев томата, как показано на рисунке 2.

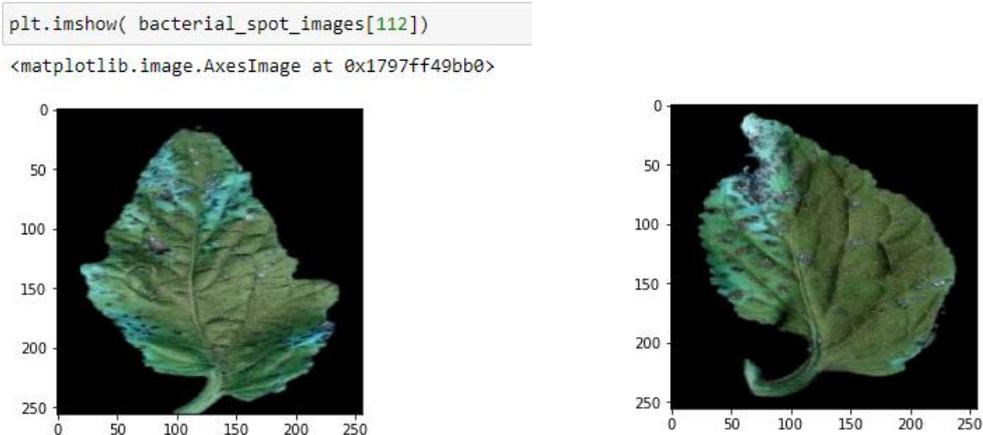


Рисунок 2 – Образцы изображений больных листьев томатов

Обычно изображения имеют три измерения: высоту, ширину и цвет. Пакет со 128-ю черно-белыми изображениями, имеющий размер 256x256, сохранили в тензоре по форме (128,256,256,1), а пакет со 128-ю цветными изображениями – в тензоре по форме (128,256,256,3). Например, количество изображений больных листьев томата для конкретного класса болезни Bacterial_spot_images равно 2127 сохранили в тензоре по форме (2127,256,256,3). Изображения в нашем случае цветные.

Используя библиотеку numpy, все изображения преобразовали в тензор(16484, 256, 256, 3) в виде сочетания изображений здоровых и нездоровых листьев.

Таким образом, получили трехмерный тензор всех изображений с 16484 элементами. Рассматривали отдельно метки изображений в виде сочетания болезней:

```
y_labels= [0] *len(healthy_images) + [1] *len(bacterial_spot_images) + \
```

```
[2] * len(early_blight_images) + [3] * len(late_blight_images) + \
[4] * len(leaf_mold_images) + [5] * len(septoria_leaf_spot_images) + \
[6] * len(yellow_leaf_curl_virus_images) + [7] * len(target_spot_images) + \
[8] * len(mosaic_virus_images)
```

Общее количество меток: len(y_labels) равно 16484.

Все метки изображений конвертировали в массив numpy: y_labels_np = np.array(y_labels).

Например, отдельно взятая метка – это y_labels_np = array([0, 0, 0, ..., 8, 8, 8]).

Затем конвертировали целые числа в одно горячее кодирование. В результате получили: (16484, 9)

```
array([[1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       [1., 0., 0., ..., 0., 0., 0.],
       ...,
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.],
       [0., 0., 0., ..., 0., 0., 1.]], dtype=float32).
```

Далее разделяли данные на обучающие и тестирование. На рисунке 3 показан результат случайного просмотра.

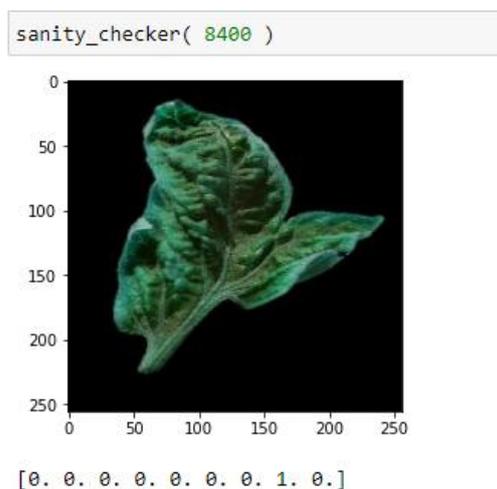


Рисунок 3 – Образцы изображений больных листьев томатов после кодировки. Таким образом, подготовили исходные данные к построению нейронной сети.

Построение модели для всех категорий болезней томатов

Проводили вычислительные эксперименты по обучению нейронной сети, для создания модели использовали следующие библиотеки: Keras, TensorFlow, Sequential, Model, Conv2D, MaxPooling2D, Flatten, Dense, Activation, Dropout, Input, LeakyReLU.

Предварительно нормализовали в данные изображений. Для этого использовали готовые обученные сети с помощью трансферного обучения со следующей архитектурой:

**** Архитектура ****

- Conv -> Maxpool -> Conv-> Maxpool -> Maxpool -> Conv -> Maxpool -> Softmax

Параметры нейронной сети подбирали следующим образом:

**** Оптимизатор ****

- Адам
- Размер партии = 256

- Эпоха = 20

Компилировали модель
`model.summary()`
 Model: "sequential"

Таким образом, архитектура нейронной сети имеет следующий вид, как показано в таблице 1.

Таблица 1 – Архитектура нейронной сети на основе CNN

Layer (type)	Output Shape	Param #
conv2d (Conv2D)	(None, 256, 256, 16)	448
leaky_re_lu (LeakyReLU)	(None, 256, 256, 16)	0
max_pooling2d (MaxPooling2D)	(None, 85, 85, 16)	0
conv2d_1 (Conv2D)	(None, 85, 85, 32)	4640
leaky_re_lu_1 (LeakyReLU)	(None, 85, 85, 32)	0
max_pooling2d_1 (MaxPooling 2D)	(None, 28, 28, 32)	0
conv2d_2 (Conv2D)	(None, 28, 28, 64)	18496
leaky_re_lu_2 (LeakyReLU)	(None, 28, 28, 64)	0
max_pooling2d_2 (MaxPooling2D)	(None, 9, 9, 64)	0
flatten (Flatten)	(None, 5184)	0
dense (Dense)	(None, 256)	1327360
leaky_re_lu_3 (LeakyReLU)	(None, 256)	0
dense_1 (Dense)	(None, 64)	16448
leaky_re_lu_4 (LeakyReLU)	(None, 64)	0
dense_2 (Dense)	(None, 9)	585
activation (Activation)	(None, 9)	0
Total params: 1,367,977		
Trainable params: 1,367,977		
Non-trainable params: 0		

Всего в архитектуре данной модели 1.367.977 параметров. В данном случае, когда сеть небольшая, это число является достаточно большим. Обучали модель с введением оптимизатора Adam с критерием ошибки `categorical_crossentropy`. Точность модели оценивали в метрике `accuracy`. Обучали нашу модель на небольшом количестве эпох, равном 20. Причем данные нормированы и находятся на отрезке [0,1]. На входе в нейронную сеть подавали по 128 изображений, при этом количество нейронов на скрытом слое было также 128. Подготовили тестовые данные для проверки модели.

Отметим, что наша мультиклассовая (многоклассовая) задача на выходе из нейронной сети выдает 9 (девять) разных чисел, каждое из которых отвечает за соответствующую болезнь листьев томата. Результаты обучения нейронной сети следующие:

```
model.compile(optimizer='adam',loss='categorical_crossentropy',metrics=['accuracy'])
```

```
BATCH_SIZE = 128
```

```
EPOCHS = 20
```

```
history = model.fit(
```

```
X_train_norm,
```

```
y_train,# подготовленные данные
```

```
batch_size=BATCH_SIZE,
```

```
epochs=EPOCHS,
```

```
validation_data=(X_test_norm, y_test),
```

```
verbose=1)
```

```
Epoch 1/20
```

71/71 [=====] - 105s 1s/step - loss: 1.5537 - accuracy: 0.4626 - val_loss: 1.1621 - val_accuracy: 0.6050

Epoch 20/20

71/71 [=====] - 82s 1s/step - loss: 0.0267 - accuracy: 0.9911 - val_loss: 1.0744 - val_accuracy: 0.7790

На рисунке 4 показаны результаты обучения модели. Точность и ошибка модели на данных обучения и валидации.

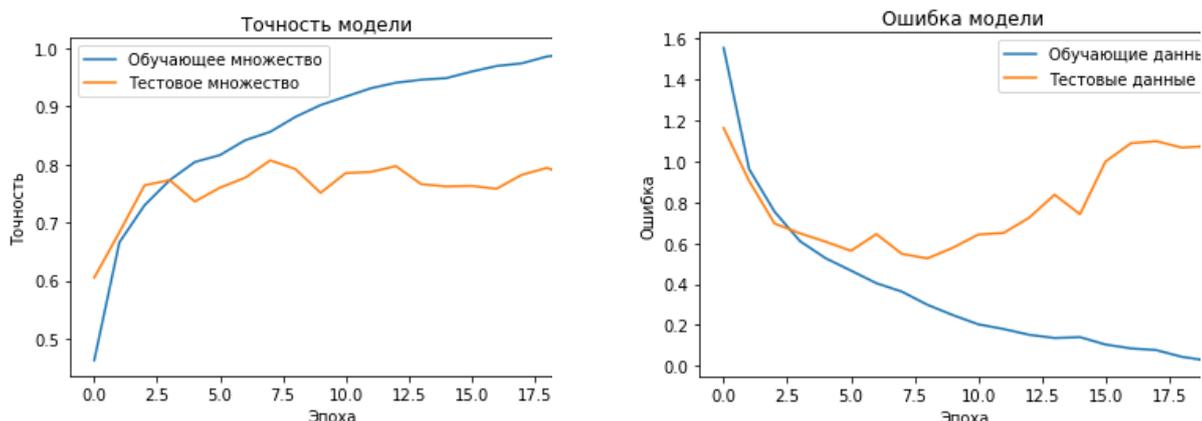


Рисунок 4 – Результаты обучения модели. Точность и ошибка модели на данных обучения и валидации

В данном случае видно, что модель сильно переобучена. На обучающей выборке модель ведет себя очень хорошо. Точность модели при этом достигает более 99,98%, а на тестовой выборке она сильно падает, достигая только предела 78%. Это означает, что построенная нейронная сеть не способна к обобщению.

Метод регуляризации переобученных моделей

При построении многие модели на тестовых данных теряют обобщающие свойства. Точность модели при этом, как видно, сильно теряется. Таким образом, изменили предыдущую построенную модель. Было очевидно, что предыдущая модель переобучена. Создали новую архитектуру модели с учетом дезактивации некоторых слоев, т.е. с добавлением известной функции Dropout. Каждый слой в модели будет либо сверточным с функцией Conv2D, либо MaxPooling2D, причем каждый слой имеет выпрямляемую нелинейную функцию LeakyReLU с добавлением слоя с функцией Dropout. Архитектура модели в этом случае следующая:

```

model.compile(optimizer='adam',loss = 'categorical_crossentropy',metrics =
['accuracy'])
BATCH_SIZE = 128
EPOCHS = 30
history = model.fit(
X_train_norm,
y_train,# prepared data
batch_size=BATCH_SIZE,
epochs=EPOCHS,
validation_data=(X_test_norm, y_test),
verbose=1)
Epoch 1/30
    
```

71/71 [=====] - 99s 1s/step - loss: 1.8363 - accuracy: 0.3617 - val_loss: 1.3722 - val_accuracy: 0.5760

Epoch 30/30

71/71 [=====] - 82s 1s/step - loss: 0.2134 - accuracy: 0.9126 - val_loss: 0.6630 - val_accuracy: 0.8020

На рисунке 5 показаны точность и ошибка улучшенной модели на данных обучения и валидации, полученной с использованием прореживания нейронной сети.

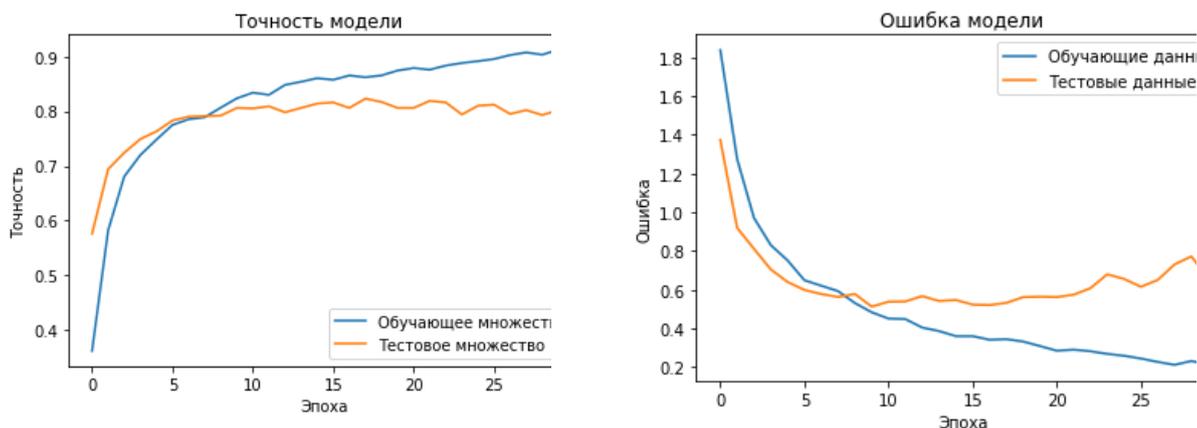


Рисунок 5 – Улучшенная модель. Точность и ошибка модели на данных обучения и валидации

Улучшенная модель показала лучшие результаты и может практически реализовываться. Сохранили модель следующей строкой кода для дальнейшего использования.

```
import pickle
pickle.dump( history.history, open('C:/Users/Admi/Desktop/history_tomato.pkl',
'wb') )
```

Выводы

В данной работе на основе применения различных архитектур нейронных сетей решена задача сельскохозяйственного назначения, в частности, задача распознавания болезней листьев томатов. Для построения модели нейронной сети было использовано всего 16484 изображения больных и здоровых листьев томатов, из них 1591 здоровых растений. Данные обучения и валидации (проверки) не имеют пересечений. Построены модели для прогнозирования 10 классов болезней томатов. Для классификации болезней томатов была построена нейронная сеть. Точность этой модели, обученной на протяжении 30 эпох, дала результат в 91,26%, а на тестовых данных точность достигала до 80%. Последнее связано с количеством изображений, так как для получения более точных моделей методы глубокого обучения требуют большого количества данных. В связи с отсутствием больших данных для рассматриваемой задачи необходимо использовать трансферное обучение и модель, обученную на больших данных.

Литература

1. Schreinemachers P., Simmons E.B., Wopereis M.C. Tapping the economic and nutritional power of vegetables. *Glob. Food Secur.* 2018;16:36–45. doi: 10.1016/j.gfs.2017.09.005. [CrossRef] [Google Scholar]
2. Stilwell M. The global tomato online news processing in 2018. [(accessed on 15 November 2021)]. Available online: <https://www.tomatonews.com/>
3. Hasan M., Tanawala B., Patel K.J. Deep learning precision farming: Tomato leaf disease detection by transfer learning; In *Proceeding of the 2nd International*

- Conference on Advanced Computing and Software Engineering (ICACSE); Sultanpur, India. 8–9 February 2019. [Google Scholar]
4. Adhikari S., Shrestha B., Baiju B., Kumar S. Tomato plant diseases detection system using image processing; Proceedings of the 1st KEC Conference on Engineering and Technology; Laliitpur, Nepal. 27 September 2018; pp. 81–86. [Google Scholar]
 5. Sabrol H., Satish K. Tomato plant disease classification in digital images using classification tree; Proceedings of the International Conference on Communication and Signal Processing (ICCSP); Melmaruvathur, India. 6–8 April 2016; pp. 1242–1246. [Google Scholar]
 6. Salih T.A. Deep Learning Convolution Neural Network to Detect and Classify Tomato Plant Leaf Diseases. Open Access Libr. J.2020;7:12. doi: 10.4236/oalib.1106296. [CrossRef] [Google Scholar]
 7. Ishak S., Rahiman M.H., Kanafiah S.N., Saad H. Leaf disease classification using artificial neural network. J. Teknol.2015;77:109–114. doi: 10.11113/jt.v77.6463. [CrossRef] [Google Scholar]
 8. Rangarajan A.K., Purushothaman R., Ramesh A. Tomato crop disease classification using pre-trained deep learning algorithm. Procedia Comput. Sci. 2018;133:1040–1047. doi: 10.1016/j.procs.2018.07.070. [CrossRef] [Google Scholar]
 9. Coulibaly S., Kamsu-Foguem B., Kamissoko D., Traore D. Deep neural networks with transfer learning in millet crop images. Comput. Ind. 2019;108:115–120. doi: 10.1016/j.compind.2019.02.003. [CrossRef] [Google Scholar]
 10. Sangeetha R., Rani M. Tomato Leaf Disease Prediction Using Transfer Learning; Proceedings of the International Advanced Computing Conference 2020; Panaji, India. 5–6 December 2020. [Google Scholar]
 11. Mortazi A., Bagci U. Automatically designing CNN architectures for medical image segmentation; Proceedings of the International Workshop on Machine Learning in Medical Imaging; Granada, Spain. 16 September 2018; pp. 98–106. [Google Scholar]
 12. Jiang D., Li F., Yang Y., Yu S. A tomato leaf diseases classification method based on deep learning; Proceedings of the Chinese Control and Decision Conference (CCDC); Hefei, China. 22–24 August 2020; pp. 1446–1450. [Google Scholar]
 13. Agarwal M., Gupta S.K., Biswas K.K. Development of Efficient CNN model for Tomato crop disease identification. Sustain. Comput. Inform. Syst. 2020;28:100407–100421. doi: 10.1016/j.suscom.2020.100407. [CrossRef] [Google Scholar]
 14. Верзунов, С. Н. Диагностика различных видов пневмонии по рентгеновским снимкам грудной клетки / С. Н. Верзунов // Проблемы автоматизации и управления. – 2022. – № 1(43). – С. 84–94.

УДК.:303.447.3:528.8.04-022.322:528.931.3

Каримова Гульмира Токтомуратовна, k.gulpeace@kstu.kg*Кыргызский государственный технический университет им. И. Раззакова,
г. Бишкек, Кыргызская Республика***ФИЛЬТРАЦИЯ И АНАЛИЗ СНИМКОВ ВЕГЕТАЦИОННОГО ИНДЕКСА**

В этой статье из многоспектрального изображения получен вегетационный индекс. Разработанная система помогает создать и фильтровать от шумов полученный нормализованный вегетационный индекс. Для тестового прогона взят снимок Landsat 5. Охватываемый район включает растительность, холмистые местности, водоемы, открытую местность, заросли кустарника, сельскохозяйственные угодья. Земельные ресурсы легко интерпретируются путем вычисления их нормализованного разностного индекса растительности для классификации почвенного покрова. Данные дистанционного зондирования с изображения Landsat TM вместе со слоями данных NDVI были использованы для фильтрации и получения снимка. Метод NDVI применяется в соответствии с его характеристикой, подобной растительности, при различных пороговых значениях NDVI, таких как 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4 и 0.5. Результаты моделирования показывают, что NDVI очень полезен для обнаружения поверхностных особенностей видимой области, которые чрезвычайно полезны для принятия решений.

Ключевые слова: архитектура ПО, программное обеспечение, нормализованный вегетационный индекс, Landsat TM, фильтрация

Введение

Многоспектральные снимки дистанционного зондирования Земли очень эффективны для лучшего понимания окружающей среды. Получение информации и извлечения признаков в виде спектральных, пространственно-временных представлений о некоторых объектах, местности или явлений, таких как растительность, классификация почвенного покрова, городская территория, сельскохозяйственные угодья и водные ресурсы, без физического контакта с этими объектами дают большие возможности в изучении этих объектов в условиях лаборатории. [1]

Данные дистанционного зондирования имеют множество областей применения, включая: классификацию растительного покрова, измерение влажности почвы, классификацию типов лесов, измерение содержания жидкой воды в растительности, картографирование снега, классификацию типов морского льда, океанографию. [2] Многоспектральные изображения дистанционного зондирования содержат существенные интегрирующие спектральные и пространственные характеристики объектов.

В этой статье из многоспектрального изображения получен вегетационный индекс.[3] Разработанная система помогает создать и фильтровать от шумов полученный нормализованный вегетационный индекс. Для тестового прогона взят снимок Landsat 5. Охватываемый район включает растительность, холмистые местности, водоемы, открытую местность, заросли кустарника, сельскохозяйственные угодья.

NDVI (Normalized Difference Vegetation Index) – нормализованный относительный индекс растительности – простой количественный показатель количества фотосинтетически активной биомассы (обычно называемый вегетационным индексом). Один из самых распространенных и используемых индексов для решения задач, использующих количественные оценки растительного покрова [4, 5].

Вычисляется по следующей формуле:

$$NDVI = \frac{NIR - RED}{NIR + RED},$$

где:

NIR – отражение в ближней инфракрасной области спектра,

RED – отражение в красной области спектра.

Чаще расчет NDVI употребляется на основе серии разновременных (разносезонных) снимков с заданным временным разрешением, позволяя получать динамическую картину процессов изменения границ и характеристик различных типов растительности (месячные вариации, сезонные вариации, годовые вариации).

Карты NDVI часто используются как один из промежуточных дополнительных слоев для проведения более сложных типов анализа. Результатами которых могут являться карты продуктивности лесов и сельхозземель, карты типов ландшафтов, растительности и природных зон, почвенные, аридные, фитогидрологические и другие эколого-климатические карты. Также на его основе возможно получение численных данных для использования в расчетах оценки и прогнозирования урожайности и продуктивности, биологического разнообразия, степени нарушенности и ущерба от различных естественных и антропогенных бедствий, аварий и т.д.

В целом главным преимуществом NDVI является легкость его получения: для вычисления индекса не требуется никаких дополнительных данных и методик, кроме непосредственно самой космической съемки и знания ее параметров.

Функциональные требования к системе, получения NDVI-снимка

Пользовательские требования:

Открытие NIR- и RED-спектра мультиспектрального снимка

Вывод NDVI по формуле

Функциональные требования:

Обработка изображений

Подсчет NDVI по формуле

Нефункциональные требования:

Язык интерфейса – русский

ОС Windows

Ограничения:

Среда разработки – Visual Studiocommunity 2022

Язык программирования – C#

Преимущества разработанной системы:

- Интуитивно понятный, дружелюбный интерфейс в виде расчетной панели.
- Обрабатывает многозональные снимки различных размеров полос NIR и RED для спутников со средним разрешением.
- Предусмотрены добавление дополнительных функций и использование в режиме реального времени при улучшении нефункциональных требований к аппаратному оснащению.
- При достаточной памяти для хранения и мощности видеокарты быстрая обработка снимков.

В результате анализа предметной области и анализа требований к системе на диаграмме вариантов использования представлен функционал системы (рис.1.)

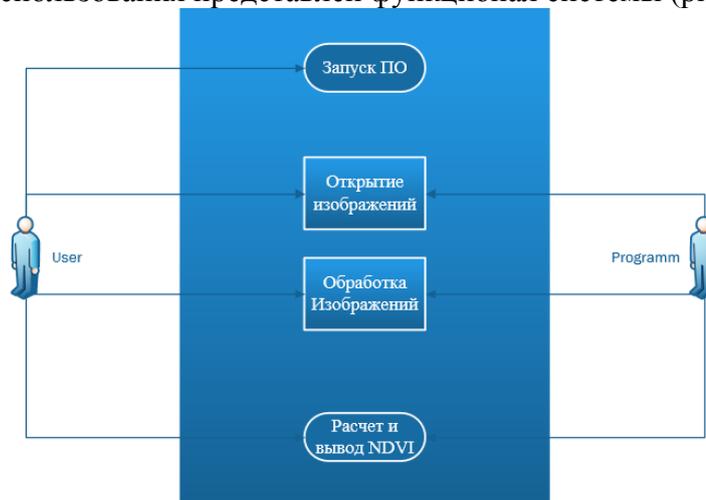


Рисунок 1 – Диаграмма UseCase

На рисунке 2 представлена диаграмма последовательности, которая описывает обобщенную структуру работы системы.

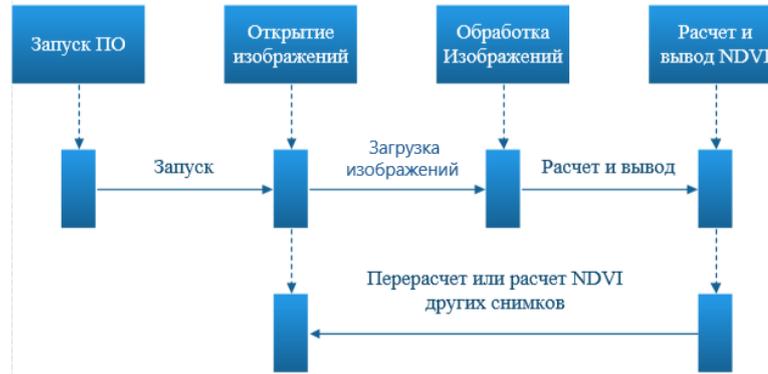


Рисунок 2 – Диаграмма последовательности

Алгоритм программы и программный код

```

Bitmap bm1 = newBitmap(picture_preview1.Image); // присваивание NIR-снимка к классу
"Bitmap" для работы с данным изображением;
Bitmap bm2 = newBitmap(picture_preview2.Image); // присваивание RED-снимка к классу
"Bitmap" для работы с данным изображением;
for (int y = 0; y < bm1.Height; y++) // Идет цикл, рассчитывающий NIR-снимок по оси
ординат (y) / по высоте;
    for (int x = 0; x < bm1.Width; x++) // Идет цикл, рассчитывающий NIR-снимок по оси
абцисс (x) / по ширине;
        {
            if (bm1.GetPixel(y, x).R <= 80 && bm1.GetPixel(y, x).R >= 76) // если RGB от (80; 80; 80)
до (76; 76; 76), то NIR присваивается 0,5; т.к. Red == Green == Blue в черно- белом снимке,
исходя из этого без разницы брать или Red, или Green, или Blue
                x1 = 0.5;
            if (bm1.GetPixel(y, x).R <= 75 && bm1.GetPixel(y, x).R >= 71) // если RGB от (75; 75; 75)
до (71; 71; 71), то NIR присваивается 0,475;
                x2 = 0.475;
            if (bm1.GetPixel(y, x).R <= 70 && bm1.GetPixel(y, x).R >= 66) // если RGB от (70; 70; 70)
до (66; 66; 66), то NIR присваивается 0,45;
                x3 = 0.45;
            if (bm1.GetPixel(y, x).R <= 65 && bm1.GetPixel(y, x).R >= 61) // если RGB от (65; 65; 65)
до (61; 61; 61), то NIR присваивается 0,425;
                x4 = 0.425;
            if (bm1.GetPixel(y, x).R <= 60 && bm1.GetPixel(y, x).R >= 56) // если RGB от (60; 60; 60)
до (56; 56; 56), то NIR присваивается 0,4;
                x5 = 0.4;
        }
    for (int y = 0; y < bm2.Height; y++) // Идет цикл, рассчитывающий RED-снимок по оси
ординат (y) / по высоте
        for (int x = 0; x < bm2.Width; x++) // Идет цикл, рассчитывающий RED-снимок по оси
абцисс (x) / по ширине
            {
                if (bm2.GetPixel(y, x).R <= 34 && bm2.GetPixel(y, x).R >= 30) // если RGB от (34; 34; 34)
до (30; 30; 30), то RED присваивается 0,08;
                    y1 = 0.08;
            }
  
```

```

    if (bm2.GetPixel(y, x).R <= 39 && bm2.GetPixel(y, x).R >= 35) // если RGB от (39; 39; 39)
    до (35; 35; 35), то RED присваивается 0,135;
        y2 = 0.135;
    if (bm2.GetPixel(y, x).R <= 44 && bm2.GetPixel(y, x).R >= 40) // если RGB от (44; 44; 44)
    до (40; 40; 40), то RED присваивается 0,19;
        y3 = 0.19;
    if (bm2.GetPixel(y, x).R <= 49 && bm2.GetPixel(y, x).R >= 45) // если RGB от (49; 49; 49)
    до (45; 45; 45), то RED присваивается 0,245;
        y4 = 0.245;
    if (bm2.GetPixel(y, x).R <= 54 && bm2.GetPixel(y, x).R >= 50) // если RGB от (54; 54; 54)
    до (50; 50; 50), то RED присваивается 0,3;
        y5 = 0.3;
    }
    sum1 = (x1 - y1) / (x1 + y1); // NDVI для свежих лугов / пастбищ;
    sum2 = (x2 - y2) / (x2 + y2); // NDVI для относительно свежих лугов / пастбищ;
    sum3 = (x3 - y3) / (x3 + y3); // NDVI для менее свежих лугов / пастбищ;
    sum4 = (x4 - y4) / (x4 + y4); // NDVI для менее сухих лугов / пастбищ;
    sum5 = (x5 - y5) / (x5 + y5); // NDVI для сухих лугов / пастбищ;
    sum = (sum1 + sum2 + sum3 + sum4 + sum5) / 5; // среднее арифметическое NDVI лугов /
пастбищ;
    Bitmap bm1 = new Bitmap(picture_preview1.Image);
    Bitmap bm2 = new Bitmap(picture_preview2.Image);
    for (int y = 0; y < bm1.Height; y++)
    for (int x = 0; x < bm1.Width; x++)
        {
    if (bm1.GetPixel(y, x).R <= 80 && bm1.GetPixel(y, x).R >= 76)
    x1 = 0.5;
    if (bm1.GetPixel(y, x).R <= 75 && bm1.GetPixel(y, x).R >= 71)
    x2 = 0.475;
    if (bm1.GetPixel(y, x).R <= 70 && bm1.GetPixel(y, x).R >= 66)
    x3 = 0.45;
    if (bm1.GetPixel(y, x).R <= 65 && bm1.GetPixel(y, x).R >= 61)
    x4 = 0.425;
    if (bm1.GetPixel(y, x).R <= 60 && bm1.GetPixel(y, x).R >= 56)
    x5 = 0.4;
        }
    for (int y = 0; y < bm2.Height; y++)
    for (int x = 0; x < bm2.Width; x++)
        {
    if (bm2.GetPixel(y, x).R <= 34 && bm2.GetPixel(y, x).R >= 30)
        y1 = 0.8;
    if (bm2.GetPixel(y, x).R <= 39 && bm2.GetPixel(y, x).R >= 35)
        y2 = 0.135;
    if (bm2.GetPixel(y, x).R <= 44 && bm2.GetPixel(y, x).R >= 40)
        y3 = 0.19;
    if (bm2.GetPixel(y, x).R <= 49 && bm2.GetPixel(y, x).R >= 45)
        y4 = 0.245;
    if (bm2.GetPixel(y, x).R <= 54 && bm2.GetPixel(y, x).R >= 50)
        y5 = 0.3;
        }
    sum1 = (x1 - y1) / (x1 + y1);
    sum2 = (x2 - y2) / (x2 + y2);

```

$$\begin{aligned} \text{sum3} &= (x3 - y3) / (x3 + y3); \\ \text{sum4} &= (x4 - y4) / (x4 + y4); \\ \text{sum5} &= (x5 - y5) / (x5 + y5); \\ \text{sum} &= (\text{sum1} + \text{sum2} + \text{sum3} + \text{sum4} + \text{sum5}) / 5; \end{aligned}$$

В результате генерации программного кода было получено следующее ПО (рис.3)

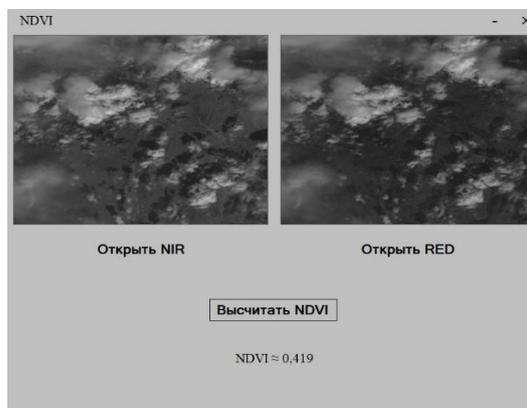


Рисунок 3 – Пользовательский интерфейс

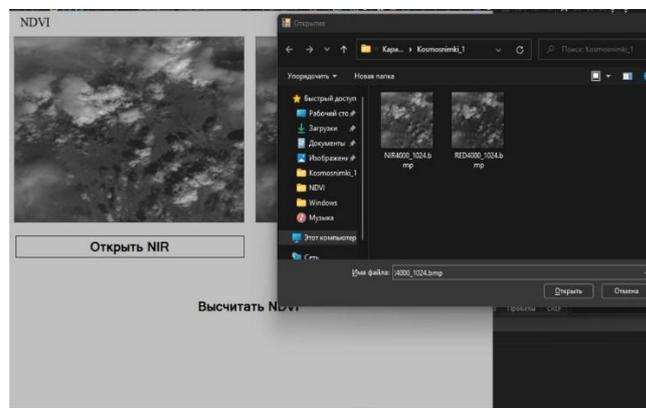


Рисунок 4 – Результат выполнения получения снимка

Выводы

Подготовка и обработка данных NDVI необходимы для анализа обнаружения изменений и являются эффективным способом описания изменений, наблюдаемых в каждой категории землепользования.

За десятилетие наблюдались значительные различия в сельскохозяйственных угодьях, холмистой местности с растительностью и в сухом земледелии.

Спутниковые данные с высоким разрешением позволили бы соответствующим образом улучшить классификацию землепользования. Метод нормализованного разностного индекса растительности с различными пороговыми значениями используется для извлечения признаков.

Литература

1. Воробьева Н.С. Аппроксимация временных рядов NDVI в задаче раннего распознавания видов сельскохозяйственных культур по космическим снимкам / Н.С. Воробьева, А.В. Чернов // Сборник трудов III Международной конференции и молодежной школы «Информационные технологии и нанотехнологии» (ИТНТ – 2017). – Самара: Новая техника. – С. 390–399.
2. Майорова В.И. Контроль состояния сельскохозяйственных полей на основе прогнозирования динамики индекса NDVI по данным космической мультиспектральной и гиперспектральной съёмки / В.И. Майорова, А.М. Банников, Д.А. Гришко, И.С. Жаренов, В.В. Леонов, А.Г. Топорков, А.А. Харлан // Наука и образование: научное издание МГТУ им. Н.Э. Баумана. – 2013. – В. 07. – С. 199 –228.
3. Найдина Т.А. Использование спутниковой информации в моделях биопродуктивности зерновых культур для расчета интенсивности фотосинтеза / Т.А. Найдина // Труды ИПГ. – 2011. – С. 189–194.
4. Токарева О.С. Обработка и интерпретация данных дистанционного зондирования Земли: Учебное пособие / О.С. Токарева. – Томск: Изд-во Томского политехнического университета, 2010. – 148 с.
5. Каримова, Г. Т. Анализ спутниковых данных города Бишкек для инфраструктуры пространственных данных / Г. Т. Каримова // Проблемы автоматизации и управления. – 2022. – № 2(44). – С. 22-31. – EDN PSCHRW.

УДК 004.8: 621.315.23

*Асан уулу Аскам*¹ asanaskat@gmail.com*Бакасова А. Б.*² bakasoaaaina@mail.ru¹*Кыргызский государственный технический университет им. И. Раззакова, Кыргызстан, Бишкек, проспект Ч. Айтматова, 66*²*Институт машиноведения и автоматизации (ИМА) Национальной академии наук Кыргызской Республики. Кыргызстан, Бишкек, проспект, Чуй, 265*

МОДЕЛИРОВАНИЕ ЭЛЕКТРОСТАТИЧЕСКОГО РАЗРЯДА И ИДЕНТИФИКАЦИЯ ЭЛЕКТРОМАГНИТНОГО ПОЛЯ С ИСПОЛЬЗОВАНИЕМ ИСКУССТВЕННЫХ НЕЙРОННЫХ СЕТЕЙ

В статье представлены результаты исследования, связанные с проблемами воздействия электростатического разряда (ЭСР) на современные микропроцессорные устройства релейной защиты и автоматизации и цифровые системы. Оценка электрических и магнитных полей для испытываемого оборудования (ИО) реализована с применением искусственных нейронных сетей (ИНС).

Ключевые слова: электростатический разряд, электромагнитная совместимость, электрическое поле, магнитное поле, искусственные нейронные сети, международный стандарт IEC 61000-4-2, генераторы электростатического разряда.

Введение

Век микропроцессорных устройств релейной защиты и автоматизации принес с собой новые проблемы, связанные с ЭСР, и по мере того, как микропроцессорные устройства становились быстрее и меньше, их чувствительность к ЭСР возрастала. ЭСР является распространенным и разрушительным явлением и может нарушить работу микропроцессорной системы, изменив ее характеристики в худшую сторону, что приведет к аварийному режиму работы или отказу оборудования. А также начальный пик очень быстрых импульсов тока и поля приведет к серьезным проблемам, вплоть до повреждения высокоскоростных цифровых систем. Многие исследователи провели обширные исследования форм токов электростатического разряда, и было показано, что амплитуды и время нарастания меняются в зависимости от зарядного напряжения, скорости приближения, типов электродов и влажности [4-9].

Явления ЭСР в промышленных средах происходят часто, и когда проблемы действительно возникают, решения применяются случайным образом до тех пор, пока проблемы не исчезнут. Разработчики часто не уверены, действительно ли проблема решена, потому что не понимают, как выполнять точные проверочные тесты [1, 2].

Защита от ЭСР стало самостоятельной областью электромагнитной совместимости благодаря многочисленным исследовательским направлениям и аспектам [1]. Одним из таких аспектов является исследование противодействий или невосприимчивость против ЭСР. Невосприимчивость электрического и электронного оборудования к ЭСР проверяется в соответствии со стандартами МЭК 61000-4-2, который описывает процедуру испытания электрического и электронного оборудования на электростатические разряды и определяет форму разрядного тока, который должен создавать генератор электростатического разряда [2].

Проблема устойчивости полупроводниковых устройств к воздействию ЭСР заключается в том, что по сравнению с другими видами перенапряжения она представляет собой чрезвычайно быстрый переходный процесс в виде импульса сложной формы.

В связи с ростом чувствительности полупроводниковых устройств к электростатическому перенапряжению усовершенствование существующих методов испытаний, оценки и защиты от воздействия ЭСР является актуальной задачей.

Целью данной работы является моделирование ЭСР ИНС, которая могла бы точно оценивать электрические и магнитные поля, излучаемые генераторами ЭСР.

Моделирование электростатического разряда (ЭСР)

Как известно, способ тестирования электронных средств на устойчивость к воздействию ЭСР заключается в формировании тестирующих импульсов при помощи генераторов ЭСР. Упрощенная схема генератора ЭСР по стандарту состоит из зарядного резистора R_c (50-100 МОм), накопительного конденсатора C_s (150 пФ \pm 10 %), распределенной емкости C_d (в большинстве случаев опускается), разрядного резистора R_d (330 Ом \pm 10%) и ИО (испытываемое оборудование) [рис. 1].

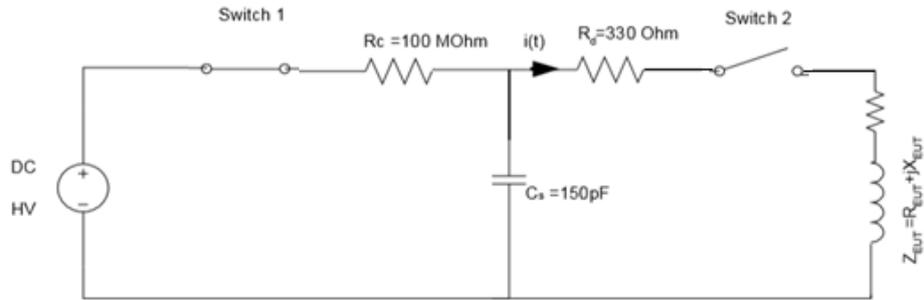


Рисунок 1 – Упрощенная схема генератора ЭСР во время заряда

Когда первый ключ замкнут, второй размыкается, чтобы конденсатор заряжался. После заряда конденсатора первый ключ размыкается, а второй замыкается, и поэтому на ИО возникает электростатический разряд. Эти два переключателя известны как реле, и они являются единственным известным на сегодняшний день пусковым устройством, способным производить воспроизводимые и быстро нарастающие разрядные токи.

Импульс, который должен создавать генератор ЭСР, показан на рисунке 2. Этот импульс разделен на две части: первый пик, называемый «начальным пиком», вызванный разрядом, и второй пик, вызванный повторным разрядом. Время нарастания начального пика составляет от 0,7 нс до 1 нс, а его амплитуда зависит от зарядного напряжения симулятора электростатического разряда.

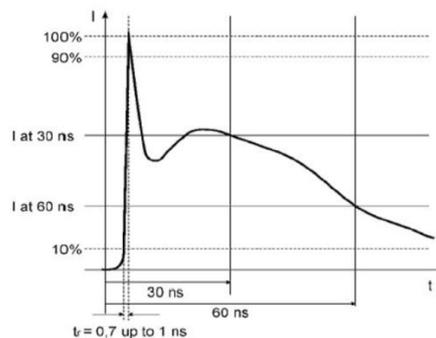


Рисунок 2 – Типичная форма выходного тока генератора ЭСР

Для моделирования генератора ЭСР стандарта 61000-4-2 (европейского) выбрана RC-цепь при напряжении постоянного тока 2 кВ, так как моделирование цепи RLC не может быть выполнено, потому что в Matlab моделирование нелинейных элементов было выполнено с источниками тока. Следовательно, индуктивность не может быть включена последовательно с источником тока.

Выберем начальные условия для схемы ЭСР (рис.1), для ИО ($L_{ИО}=0$, $I(0)=0$), начальное напряжение конденсатора ($U_C(0)=V_0$, (V_0)) и ключи 1 и 2 открываются и закрываются соответственно. Однородное дифференциальное уравнение, составленное по закону Кирхгоффа для первого контура при нулевых начальных условиях:

$$U_R + U_C = 0 \Rightarrow R_C C \frac{dU_C(t)}{dt} + U_C(t) = 0 \quad (1)$$

Решая (1), получим напряжение на конденсаторе:

$$\left. \begin{aligned} U_C(t) &= K \cdot e^{-\frac{t}{RC}} \\ U_C(0) &= V_0 \end{aligned} \right\} \Rightarrow U_C(t) = V_0 \cdot e^{-\frac{t}{RC}} \\ t \geq 0$$

Следовательно, ток можно получить как:

$$i(t) = \frac{V_0}{R_C} \cdot e^{-\frac{t}{\tau}}, t \geq 0^+ \quad (2)$$

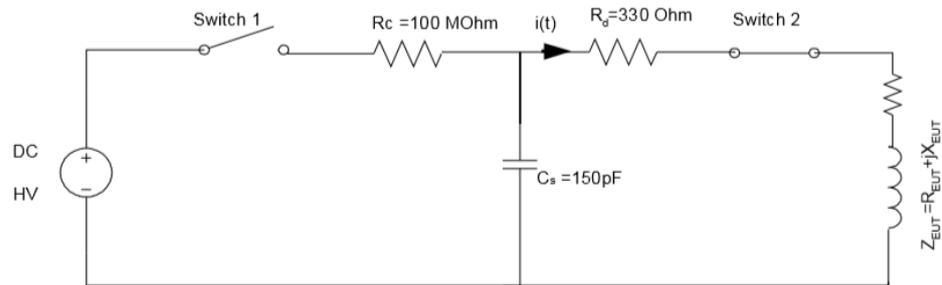


Рисунок 3 – Схема ЭСР во время разряда с учетом комплексного электрического сопротивления ИО

С учетом ИО ($Z_{ИО} = R_{ИО} + jX_{ИО}$) начальными условиями схемы, как показано на рисунке 3, также являются $I(0)=0$ и $U_C(0)=V_0$.

Решая дифференциальное уравнение цепи, составленное для второго контура (рис. 3), где напряжение конденсатора:

$$\left. \begin{aligned} U_C(t) + L \frac{di(t)}{dt} + R \cdot i(t) &= 0 \\ i(t) &= C \frac{dU_C(t)}{dt} \\ i(0) &= 0, U_C(0) = V_0 \end{aligned} \right\} \Rightarrow LC \frac{d^2i}{dt^2} + RC \frac{di}{dt} + i(t) = 0 \quad (3)$$

где $R=R_d+R_{ИО}$.

Ток $i(t)$ для всей цепи определяется уравнением (4):

$$i(t) = \begin{cases} C \cdot \frac{\lambda_1 \cdot \lambda_2}{\lambda_1 - \lambda_2} \cdot V_0 \cdot (e^{\lambda_2 t} - e^{\lambda_1 t}), & \text{если } R > 2\sqrt{\frac{L}{C}} \\ \frac{C \cdot R}{2 \cdot L} \cdot V_0 \cdot \left(\frac{R}{2 \cdot L} \cdot t - 2 \right) \cdot e^{-\frac{R}{2L}t}, & \text{если } R = 2\sqrt{\frac{L}{C}} \\ C \cdot V_0 \cdot e^{-\alpha t} \left(-\frac{\alpha^2}{\beta} \cdot \sin(\beta t) - \beta \cdot \cos(\beta t) \right), & \text{если } R < 2\sqrt{\frac{L}{C}} \end{cases} \quad (4)$$

где $\lambda_{1,2} = -\alpha \pm \beta = -\frac{R}{2L} \pm \sqrt{\left(\frac{R}{2L}\right)^2 - \frac{1}{LC}}$ – вещественные корни, где $\alpha = \frac{R}{2L}$ – константа демпфирования, коэффициент $\omega_0 = 2\pi f_0$ – резонансная частота, последовательно

включенного RLC-контура, и через него определяем $\beta = \sqrt{\alpha^2 - \omega_0^2} = \sqrt{\left(\frac{R}{2L}\right)^2 - \frac{1}{LC}}$ – коэффициент затухающего колебания.

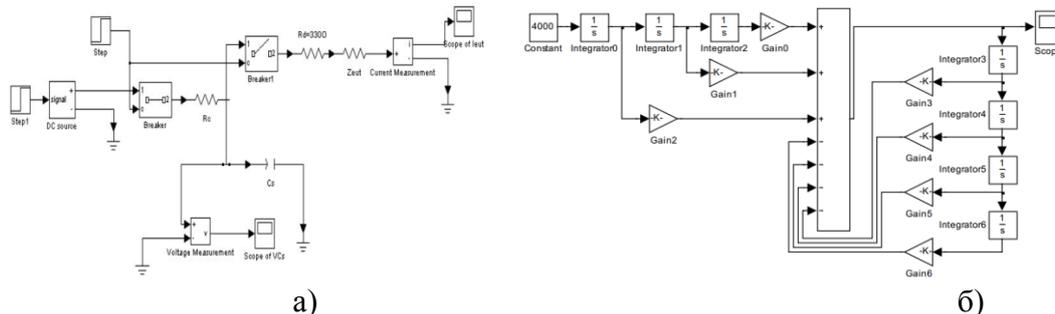


Рисунок 4 – Схема генератора ЭСР, разработанная в Matlab: а) для генератора ЭСР NSG-433, б) для генератора ЭСР NSG-438

Результаты моделирования

На рисунке 5(а) и (б) показаны репрезентативные формы волн Н-поля по отношению к разрядному току в течение первых 200 нс соответственно, когда датчик Н-поля расположен на расстоянии 10 см (точка Y1) от точки разряда для двух генераторов ЭСР. На рисунке 6(а), (б) и рисунке 7(а), (б) показаны пиковые Н-поле и Е-поле соответственно при зарядном напряжении ± 2 кВ для модели генераторов ЭСР NSG-433 и NSG-438. Очевидно, что каждый генератор создает разное электромагнитное поле.

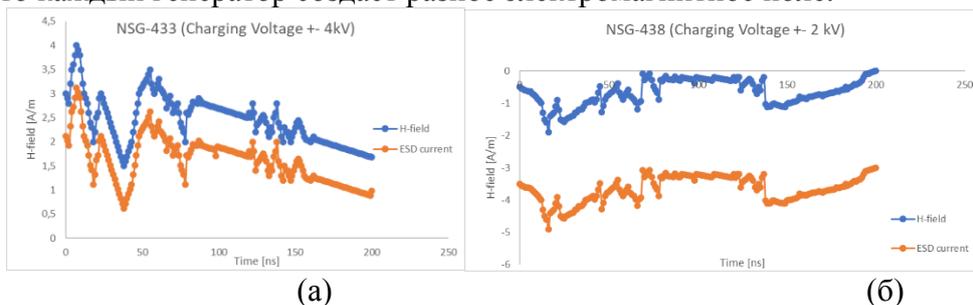


Рисунок 5 – Ток ЭСР и Н-поле для двух генераторов ЭСР в 10 см от точки разряда

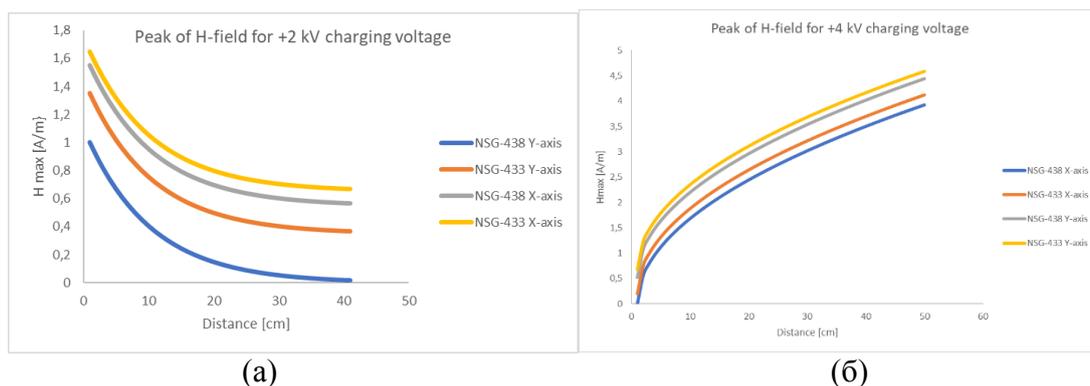


Рисунок 6 – Пик Н-поля для различных расстояний в двух перпендикулярных направлениях от точки разряда с использованием модели генераторов электростатического разряда NSG-433 и NSG-438 и для двух разных зарядных напряжений

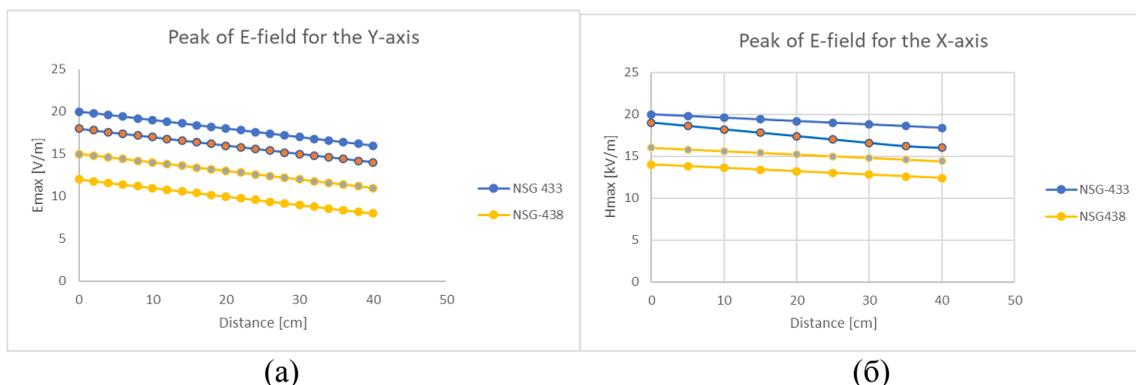


Рисунок 7 – Пик электрического поля для различных расстояний от точки разряда по осям X и Y с использованием модели генераторов ЭСР NSG-433 и NSG-438

Модель ИНС для идентификации электрических и магнитных полей

В работе [11] описана реализация трехслойной ИНС для генераторов ЭСР.

В данной работе рассмотрим реализацию ИНС через типичную модель нейронной сети, известную как традиционная модель многослойного персептрона (MLP). Для обучения сети необходимо выбрать соответствующее количество репрезентативных примеров соответствующего явления, чтобы сеть могла изучить основные характеристики проблемы [10]. Обучение обратному распространению может привести к локальному, а не глобальному минимуму. Характерная архитектура MLP традиционно определяется путем перебора разных вариаций количества скрытых слоев, числа узлов в скрытом слое и т.д.

В отличие от [11] в нашем случае с целью уменьшения ошибок между измеренными параметрами модели и выходными данными ИНС взяли 10000 исходных данных для обучающей выборки.

После окончания механизма обучения и установки весов и смещений для каждого нейрона в ИНС далее идет проверка результатов обучения путем отслеживания работы сети в других ситуациях.

Реализация искусственных нейронных сетей

В качестве входных данных нейронных сетей рассматриваются параметры, играющие очень важную роль при оценке электрического и магнитного полей, излучаемых ЭСР, а в качестве выходных параметров – пиковые значения электрического и магнитного поля, представленные в таблице 1.

Таблица 1 – Входные и выходные данные ИНС

Входные переменные	Выходные переменные
Генератор ЭСР (G)	Пиковое значение эл. поля ($E_{\text{макс}}$)
Зарядное напряжение (U)	Пиковое значение магнитного поля ($H_{\text{макс}}$)
Время нарастания (t_r)	
Максимальный разрядный ток ($I_{\text{макс}}$)	
Ток при 30 нс (I_{30})	
Ток при 60 нс (I_{60})	
Расстояние (ρ)	
Направление (D)	

Для моделей генераторов NSG-433 и NSG-438 (рис. 4.) выполнено множество измерений, представляющих собой комбинации различных параметров, где в каждом отдельном измерении параметры могут принимать различные значения:

- а) тип генератора ЭСР (NSG-433 или NSG-438);
- б) зарядное напряжение генератора (± 2 кВ, ± 4 кВ);
- в) два перпендикулярных направления;
- г) параметры формы тока (время нарастания, максимальный разрядный ток, ток при 30 нс и ток при 60 нс).

Таким образом, эти измерения представляют собой комбинации всех вышеперечисленных различных параметров. В данной работе были взяты такие же параметры, как и в работе [11] для сравнения, какая модель ИНС лучше определяет пиковые значения E- и H-поля.

Предложенные модели ИНС были обучены с помощью MATLAB Neural Network Toolbox [7]. Для обучения и тестирования искусственной нейронной сети использовалось тысяча семьсот шестьдесят значений каждого входа и выхода. В каждой обучающей итерации (эпохе) из обучающей выборки удалялось 20 % случайных данных, и для этих данных вычислялась ошибка вариации.

Оценочные значения E-полей и H-полей сравнивались со значениями, полученными из ситуаций, встречавшихся во время обучения, т.е. тысяча семьсот шестьдесят значений, и других, которые не встречались (табл. 2).

Таблица 2

№	Структура	Эпохи	Правило обучения	Передаточная функция
1	8/19/8/2	8348	Лев.-Марк.	Лог-сиг
2	8/13/11/2	9629	Град. спуска	Лог-сиг
3	8/22/24/2	10000	Лев.-Марк.	Гип-кас.
4	8/11/26/4/2	9711	Град. спуска	Гип-кас.
5	8/7/8/10/2	9509	Град. спуска	Лог-сиг
6	8/26/21/2	10000	Квазиныйтон.	Лог-сиг
7	8/10/11/14/2	8612	Град. спуска	Лог-сиг
8	8/14/23/2	9890	Лев.-Марк.	Гип-кас.
9	8/9/8/5/2	9935	Квазиныйтон.	Жестк. пред.
10	8/14/8/2	10000	Град. спуска	Лог-сиг
11	8/19/25/2	9881	Лев.-Марк.	Гип-кас.
12	8/8/8/15/2	10000	Лев.-Марк.	Жестк. пред.
13	8/10/13/2	8427	Град. спуска	Лог-сиг
14	8/29/23/2	10000	Квазиныйтон.	Лог-сиг
15	8/12/9/20/2	10000	Град. спуска	Гип – кас.

После обширного моделирования со всеми возможными комбинациями и построения нескольких моделей было обнаружено пятнадцать лучших архитектур ИНС, которые представлены в таблице 2, которые почти в равной степени соответствуют критериям (среднеквадратичная ошибка не более 0,5%, или количество эпох достигло максимального значения, где было установлено значение 10000).

Полученные результаты

Обученная ИНС для оценки электрического и магнитного поля, излучаемого электростатическими разрядами, была применена к двадцати различным темам, которые не участвовали в процессах обучения, проверки и тестирования. Полученные результаты ИНС, представленные в таблице 3, сравнивались со значениями электрического и магнитного поля, измеренными в ходе математического моделирования генераторов ЭСР для точно таких же параметров (таблица 3).

Таблица 3 – Измеренные электрические и магнитные поля

№ количество наборов данных	E_{max} (кВ/м)		H_{max} (А/м)	
	Измерено	ИНС	Измерено	ИНС
	14,5	14,19	-1,05	-1,1
	20,09	20,28	-3,68	-3,52
	19,82	19,83	-2,45	-2,56
	17,32	17,69	-1,6	-1,54
	17,12	16,79	-1,2	-1,25
	-28,82	-27,97	4,55	5,09
	-27,43	-26,54	2,5	2,63
	-27,12	-28,1	1,76	1,73

	-25,22	-26,1	1,53	1,49
	-32,91	-34,22	5,17	5,27
	21,58	20,98	-4,33	-3,54
	19,52	20,83	-2,63	-2,57
	16,92	17,51	-1,78	-1,77
	-27,43	-28,86	4,92	5,22
	25,72	26,54	-2,47	-2,59
	21,82	21,88	-1,72	-1,7
	-28,54	-28,68	2,6	2,83
	-27,81	-26,92	1,92	1,81
	-26,84	-26,12	1,55	1,42
	27,46	27,35	-4,76	-4,89

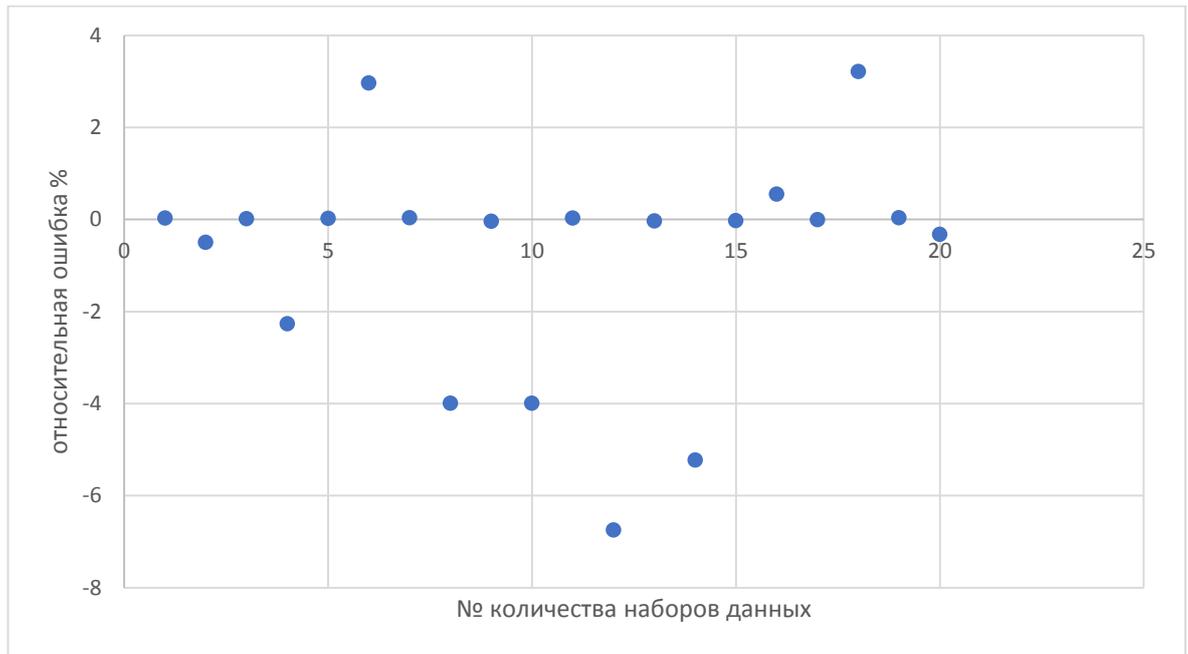


Рисунок 8 – Относительная ошибка между измеренными и рассчитанными значениями E_{\max}

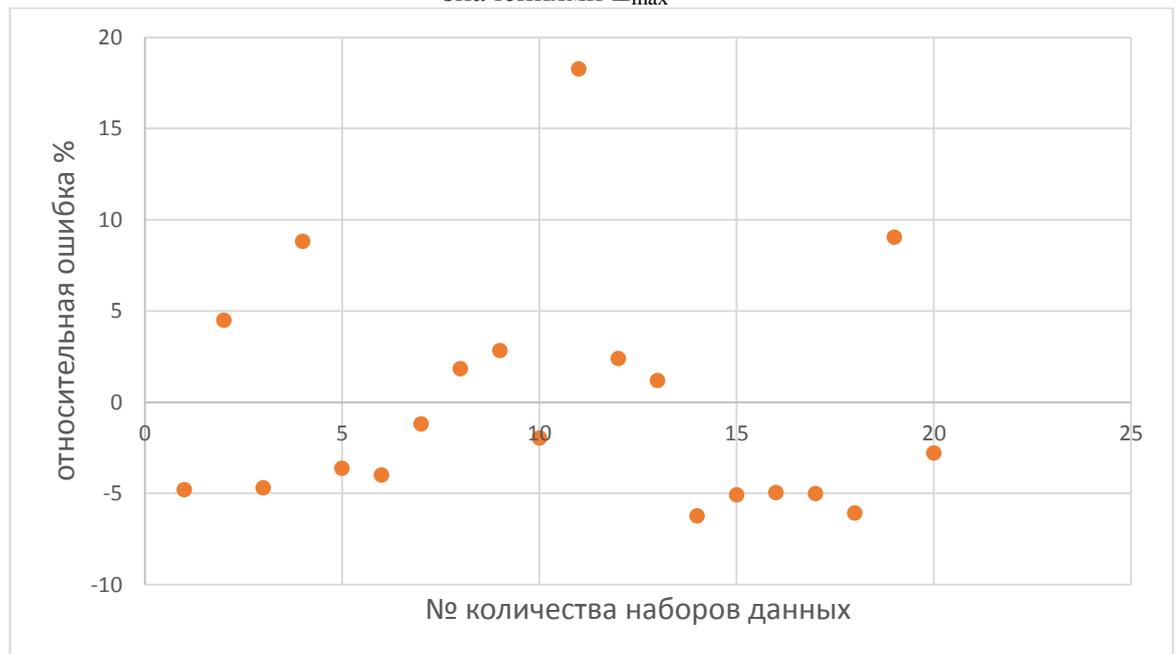


Рисунок 9 – Относительная ошибка между измеренными и рассчитанными значениями H_{\max}

Результаты, полученные с использованием разработанной модели ИНС, очень близки к фактическим значениям, что явно свидетельствует о том, что предложенный метод ИНС хорошо работает и имеет приемлемую точность. Разработанная нейронная сеть, в отличие от описанной архитектуры ИНС в работе [11], работает быстрее, потребляет меньшую память и имеет компактную структуру. На рисунках 8 и 9 представлена процентная ошибка между фактическими измерениями и результатами ИНС.

Выводы

В данной статье рассматривается искусственная нейронная сеть для оценки электрического и магнитного поля, излучаемого электростатическими разрядами. Результаты разработанной модели ИНС доказали свою точность, поскольку они очень близки к фактически измеренным результатам, что явно свидетельствует о том, что предложенный метод ИНС работает хорошо. С помощью предложенного метода ИНС можно легко и точно рассчитать создаваемое электромагнитное поле, излучаемое электростатическими разрядами путем измерения разрядного тока. Это будет чрезвычайно полезно для лабораторий, занимающихся испытаниями ЭСР, поскольку они смогут проводить проверку генераторов ЭСР после пересмотра действующего стандарта, в котором измерения электромагнитного поля будут обязательными.

Литература

1. Boxleitner W., "Electrostatic Discharge and Electronic Equipment: A practical guide for designing to prevent ESD problems", IEEE Press, 1989
2. Dangelmayer G.T., "ESD Program Management", Van Nostrand Reinhold, New York, 1990 "Fundamentals of ESD", articles and information available through the Internet address www.esda.org
3. Gloser K., Gutheil B., Mann M., Weib P., "Modelling of ESD with respect to different kinds of earth coupling", XIIIth International Symposium on High Voltage Engineering, Netherlands 2003
4. Greason D.W., "Methodology to study the resistance of spark discharges", IEEE Transactions on Dielectrics and Electrical Insulation, Vol. 35, No. 2, pp. 359-365, March - April 1999
5. Pommerenke D., "ESD: transient fields, arc simulation and rise time limit", Journal of Electrostatics, Vol. 36, pp. 31-54, October 1995
6. Pommerenke D., Aidam M., "ESD: waveform calculation, field and current of human and simulator ESD", Journal of Electrostatics, Vol. 38, pp. 33-51, October 1996
7. Bonisch S., Pommerenke D., Kalkner W., "Broadband measurement of ESD risetimes to distinguish between different discharge mechanisms", Journal of Electrostatics, Vol. 56, pp. 363–383, October 2002
8. Бакасова, А. Б. Применение нейронных сетей в задачах электромагнитных помех / А. Б. Бакасова, А. Асан Уулу // Проблемы автоматизации и управления. – 2022. – № 1(43). – С. 95-103. – EDN DJPYDH.
9. Верзунов, С. Н. Система искусственного интеллекта для онлайн мониторинга подземных силовых кабельных линий на основе технологий глубокого обучения / С. Н. Верзунов // Проблемы автоматизации и управления. – 2021. – № 3(42). – С. 83-94. – EDN DVCIKE.
10. Асан Уулу, А. Обзор альтернативных подходов к моделированию электромагнитной обстановки на высоковольтной электрической подстанции / А. Асан Уулу, А. Б. Бакасова // Проблемы автоматизации и управления. – 2022. – № 2(44). – С. 4-14. – EDN JZHUPM.
11. Maris, T & Ekonomou, L & Fotis, G.P. & Nakulas, Angelos & Zoulias, Emmanouil. (2007). Electromagnetic field identification using artificial neural networks.

УДК 004

В. Гайдамако*Институт машиноведения и автоматизации НАН КР, Бишкек, Кыргызстан***К ВОПРОСАМ ПЛАНИРОВАНИЯ И БАЛАНСИРОВКИ НАГРУЗКИ В HADOOP MAPREDUCE**

MapReduce является одной из самых известных моделей программирования для обработки больших данных в распределенных и облачных вычислениях. Балансировка нагрузки в MapReduce является одним из факторов, который позволяет повысить производительность приложения за счет решения таких проблем, как обеспечение локальности и устранение перекоса данных. В распространенной реализации Map-Reduce Apache Hadoop равномерное распределение нагрузки по узлам кластера может быть достигнуто как за счет тщательного планирования заданий, так и за счет применения алгоритмов балансировки. В статье рассмотрены некоторые из подходов.

Ключевые слова: Hadoop, MapReduce, планирование заданий, балансировка нагрузки, локальность данных, перекос данных.

Введение

MapReduce является одной из самых известных моделей программирования для обработки больших данных (BigData) в распределенных и облачных вычислениях. Термин BigData устойчиво применяется начиная с 2011 года и означает технологии обработки очень больших и сложных наборов данных, которые трудно или невозможно анализировать с помощью традиционных подходов и приложений. Технология MapReduce была предложена компанией Google для кластерной обработки больших объемов данных и была реализована в разных программных продуктах, включая Apache Hadoop. Балансировка нагрузки (назначение и перемещение нагрузки между имеющимися ресурсами, которое обеспечивает максимальную производительность при минимальных затратах) в MapReduce является одним из факторов, который позволяет повысить производительность приложения за счет решения таких проблем, как обеспечение локальности и устранение перекоса данных. Локальность данных – один из принципов работы с большими данными наряду с горизонтальной масштабируемостью и отказоустойчивостью. Локальность означает, что данные должны обрабатываться по возможности на той же машине, где хранятся, так как расходы на передачу данных иногда могут превышать расходы на их обработку. Перекос данных означает, что некоторые узлы обработки получают большое количество данных, в то время как другие простаивают [1]. Балансировка нагрузки сильно зависит от того, являются ли вычислительные кластеры однородными или гетерогенными [2]. Некоторые авторы подчеркивают, что балансировка нагрузки включает две задачи – выделение ресурсов (resource allocation) и диспетчеризацию, или планирование заданий (tasks scheduling) [3], поэтому наряду с, собственно, алгоритмами балансировки должны рассматриваться и алгоритмы планирования заданий.

Принцип работы MapReduce

Обработка данных производится в три этапа – Map, Shuffle и Reduce (рис. 1). Иногда в отдельную стадию выделяют распределение результатов Map по разделам (partition) – Shuffle. Используется архитектура Master-Slave (Главный-Подчиненный). Входные и выходные данные можно представить в виде [4]:

(input) $\langle k_1, v_1 \rangle \rightarrow \text{map} \rightarrow \langle k_2, v_2 \rangle \rightarrow \text{combine} \rightarrow \langle k_2, v_2 \rangle \rightarrow \text{reduce} \rightarrow \langle k_3, v_3 \rangle$ (output)

На стадии Map происходит предварительная обработка входных данных – главный узел кластера (masternode) получает данные в виде списка значений, данные разбиваются на сплиты (split) и передаются рабочим узлам (workernode), на которых выполняется задаваемая пользователем функция (map) – эти узлы называются мапперы (mapper). Функция (map) применяется к сплиту и выдаёт множество пар ключ–значение « k_2, v_2 », которые записываются во временное хранилище. В примере на рисунке 1 функция (map) подсчитала частоту появления каждого слова в сплите и сформировала пары («слово»,

«частота»). Далее вывод каждого (map) локально обрабатывается функцией (combine) для локальной сортировки по ключам и, возможно, дополнительной агрегации, это нужно для уменьшения объемов передаваемых данных. Далее нужно сгруппировать все полученные от всех mapper пары по значению ключа и передать их рабочим узлам, выполняющим функцию (reduce) – редьюсер (reducer). На стадии Shuffle рабочие узлы сортируют и распределяют данные на основе ключей, ранее созданных функцией (map), по разделам (partition) для каждого редьюсера так, чтобы данные для одного ключа размещались на этом рабочем узле (принцип локальности данных). Пользователь может управлять процессом распределения (partitioning), то есть определять, какой ключ попадет к какому редьюсеру, с помощью собственной функции. На стадии Reduce производится параллельная обработка рабочими узлами всех групп данных по порядку следования ключей – функция (reduce) и агрегация результатов на главном узле. Функция (reduce) вычисляет финальный результат для отдельного раздела, в нашем примере это подсчет суммарной частоты для каждого слова, и передает его главному узлу. Главный узел получает результаты от рабочих узлов и при необходимости производит их обработку. Получившийся после прохождения всех необходимых шагов результат и есть решение задачи. Для обработки данных в соответствии с вычислительной моделью MapReduce должны быть определены функции (map) и (reduce), заданы имена входных и выходных файлов и параметры обработки [1, 4].

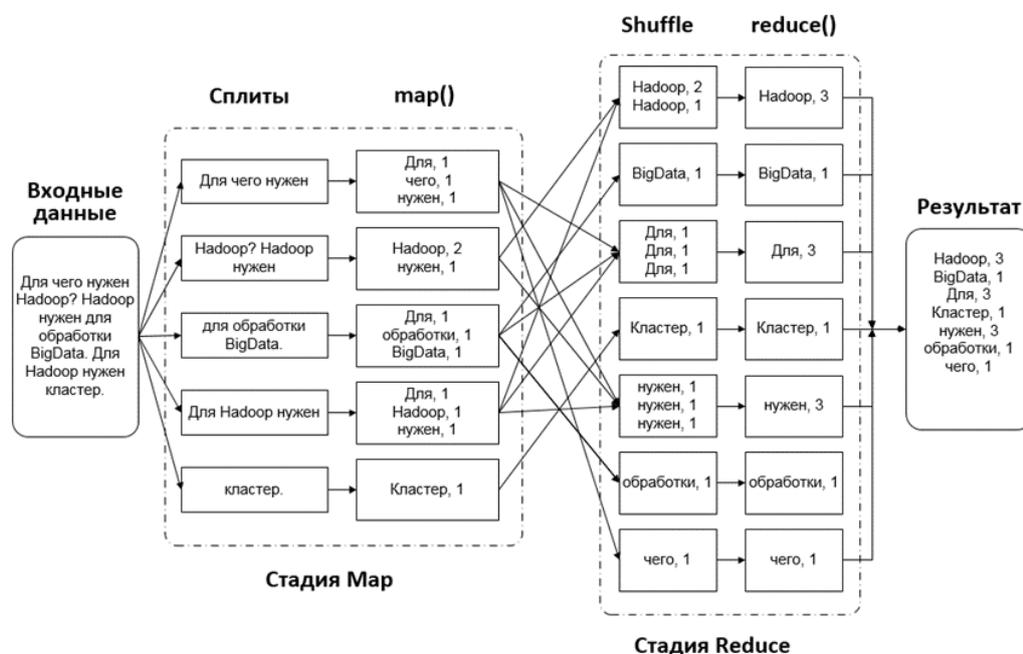


Рисунок 1– Принцип работы MapReduce

Все запуски функций (map) и (reduce) работают независимо и могут работать параллельно на разных машинах кластера. На стадии Shuffle осуществляется параллельная сортировка, которая также может работать на разных машинах кластера. Балансировка нагрузки может применяться на каждой из стадий обработки, и использование эффективных алгоритмов балансировки позволяет существенно повысить производительность MapReduce. Несбалансированность нагрузки приводит к большой разнице во времени выполнения задач на разных узлах, плохому параллелизму и увеличению общего времени выполнения задания. Дисбаланс возникает при разделении входных данных на сплиты, при обработке – у разных узлов разная рабочая нагрузка (перекос вычислений), в размерах разделов на выходе Map – перекося данных.

Существует множество коммерческих и свободных продуктов, использующих эту модель распределенных вычислений. Программный продукт Hadoop от Apache Foundation

[5] включает реализацию MapReduce и представляет набор других инструментов для работы с большими данными.

Hadoop

Hadoop [5]. ApacheHadoop – масштабируемая, гибкая и надежная платформа распределенных вычислений для работы с большими данными в кластере, использующем общедоступное оборудование. Основные компоненты Hadoop (Hadoopcore):

- Hadoopcommon – содержит библиотеки и утилиты, используемые другими модулями.
- HadoopDistributedFileSystem (HDFS) – распределённая файловая система, данные хранятся на компьютерах кластера.
- HadoopYARN – платформа управления ресурсами кластера и планирования задач.
- HadoopMapReduce – реализация модели программирования MapReduce для обработки больших данных.
- HadoopOzone – масштабируемое распределенное хранилище объектов, в отличие от HDFS работает с файлами разных размеров. Входит в состав Hadoop с 2020 г.

В настоящее время Hadoop включает, помимо MapReduce, другие инструменты обработки больших данных: Hive – инструмент SQL-подобных запросов к большим данным (SQL-запросы трансформируются в серию задач MapReduce); Pig – язык программирования для высокоуровневого анализа данных, строка кода может трансформироваться в серию задач MapReduce; Hbase – колоночная база данных, реализующая парадигму BigTable; Cassandra – высокопроизводительная распределенная key-value база данных; ZooKeeper – сервис для распределённого хранения настроек и синхронизации изменений; Mahout – библиотека и движок машинного обучения; ApacheSpark – движок для распределённой обработки данных, HDFS и YARN; ApacheImpala – база данных с открытым кодом, поддерживает SQL-запросы; ApacheSolrSearch – поисковая платформа; ApacheOozie – серверная система планирования рабочих процессов для управления заданиями Hadoop.

Такие компании, как Cloudera, Hortonworks, MapR, публикуют собственные сборки Hadoop, которые стали даже более популярными, чем, собственно, HadoopотApache, так как требуют меньше усилий при установке.

Для организации распределенного хранения и обработки данных используется архитектура Master-Slave. Мастер-узлом (главным узлом) для хранения данных в HDFS является NameNode, подчиненные или рабочие узлы (workers) – DataNode. В MapReduce главный узел – JobTracker (управление заданиями), подчиненный – TaskTracker (управление задачами). На физическом (или виртуальном) мастер-узле кластера может работать по одному экземпляру NameNode и JobTracker, а также DataNode и TaskTracker, на других машинах кластера работают DataNode и TaskTracker. Jobtracker отвечает за полное выполнение отправленного задания, TaskTracker действуют как подчиненные, каждый из которых выполняет свою задачу. В Hadoop два типа задач: задачи Map (SplitиMapping), задачи Reduce (Shuffling и Reducing)[4].

Планирование в Hadoop MapReduce

Hadoop автоматически обеспечивает балансировку нагрузки, распараллеливание, планирование задач и отказоустойчивость [6]. Планирование в Hadoop MapReduce осуществляется на уровне задания (job) и уровне задачи (task), задание состоит из нескольких задач. На уровне из очереди заданий в соответствии с выбранной стратегией планирования выбирается задание целиком, при планировании на уровне задач планируется каждая задача задания. Задачи назначаются в ответ на сообщения о состоянии (heartbreakes), получаемые от рабочих узлов каждые несколько секунд. У каждого рабочего узла есть фиксированное количество слотов map и слотов reduce. На ядро процессора (ЦП) приходится один слот. Слотовая модель упрощает управление

памятью и ЦП на ведомых устройствах. Стратегии планирования определяют, когда и какой машине задача должна быть передана для обработки, то есть они влияют на балансировку нагрузки. По умолчанию Hadoop использует стратегию FIFO, простую очередь («первым поступил — первым выполняется», приоритет задачи определяется временем поступления), но можно создавать подключаемые пользовательские планировщики (диспетчеры) [6]. Разработаны различные алгоритмы планирования для MapReduce, улучшающие балансировку нагрузки.

FIFO-планировщик

Планировщик Hadoop по умолчанию работает с очередью заданий, рассчитан на гомогенный кластер. В этом планировщике каждое задание разделено на отдельные задачи, которым назначается свободный слот для обработки [6, 7]. JobTracker берет первое задание из очереди, оно назначается всему кластеру, следующее задание может выполняться только после завершения текущего, что приводит к увеличению времени ожидания, особенно для коротких задач. Приоритетов нет. Следовательно, в этом планировщике время ожидания задания, особенно для коротких заданий, увеличивается, и никакие задания не могут быть вытеснены. Однако очень часто MapReduce разворачивается в гетерогенной среде, вычислительные ресурсы и ресурсы данных совместно используются несколькими пользователями и приложениями, и использование планирования по умолчанию значительно снижает производительность приложений MapReduce.

Справедливый планировщик

Разработан Facebook. Ресурсы назначаются задачам так, чтобы в среднем каждая задача получала равную долю ресурсов. Задания заносятся в пулы (несколько очередей) и каждому пользователю назначается один пул [6, 7]. Каждому пулу назначается слот, по умолчанию пулам распределяется равное количество слотов, но через файлы конфигурации можно изменять количество выделяемых в зависимости от типа задач. Так как пользователю назначается один пул, даже если он запустит много задач, он получает такую же долю ресурсов кластера, что и все остальные пользователи, справедливость соблюдена. Если есть свободные слоты, они распределяются по имеющимся задачам. Планировщик отслеживает время вычислений для задачи, периодически определяет разницу между полученным и «идеальным» временем. Полученный результат показывает дефицит времени для задачи и определяет ее приоритет, наивысший приоритет будет у задачи с максимальным дефицитом. В этом планировщике может возникнуть проблема залипания слотов (stickyslots) – ситуация, когда задания никогда не покидают выделенных им слотов, что приводит к нарушению принципа локальности данных [7].

Планировщик вычислительной мощности

Разработан Yahoo! для гарантии справедливого распределения ресурсов среди большого числа пользователей кластера [6]. Для этой цели вместо пулов создаются очереди с настраиваемым количеством слотов задач Map и Reduce. Доступные ресурсы (вычислительная мощность) распределяются по очередям в соответствии с приоритетами. Если в одних очередях есть свободные ресурсы, они распределяются по другим очередям. В очереди приоритет заданий определяется исходя из времени поступления задания, класса задания и настроек приоритета для пользователей в соответствии с соглашением об уровне обслуживания (SLA). Когда слот освобождается, планировщик выбирает задание с наибольшим временем ожидания из очереди с наименьшей нагрузкой. Таким образом, планировщик мощности обеспечивает совместное использование кластера для пользователей, а не для заданий, как в случае со справедливым планировщиком.

Планировщик задержки (Delayscheduler). Представляет собой оптимизацию справедливого планировщика, которая устраняет проблемы локальности последнего [6-8]. Когда освобождается слот, нужно выбрать задачу задания из очереди для обработки. Если на узле со свободным слотом нет данных для этой задачи (проблема локальности), эта задача временно откладывается до тех пор, пока не освободится слот в узле с нужными

данными. При этом отслеживается время ожидания задачи (время задержки), и, если оно становится достаточно большим, может быть запланирована нелокальная задача.

Планировщик LATE (Longest Approximate Time to End – максимальное приближительное время до завершения). Планировщик LATE был разработан для улучшения времени отклика задания в Hadoop в гетерогенных средах. Некоторые задачи могут выполняться медленно из-за высокой загрузки ЦП, состояния гонки (недоступности совместно используемых ресурсов), временного замедления из-за фоновых процессов. Эти задачи называются спекулятивными задачами. Планировщик LATE пытается найти медленную задачу и выполнить ее копию на другом узле. Это исполнение называется спекулятивным исполнением (HadoopSpeculative). Если новая копия задачи выполняется быстрее, производительность всей задачи улучшится. Планировщик LATE назначает приоритеты для спекулятивного выполнения задачам с большим временем ожидания, а затем выбирает для их выполнения самые быстрые узлы. Планирование LATE сокращает время отклика Hadoop в гетерогенных средах [9].

Планировщик дедлайнов (Deadline constraints scheduler). Планировщик с ограничением дедлайна (критического срока исполнения) был разработан для ограничений реального времени [6, 9]. Планировщик должен дать пользователю немедленное заключение о возможности выполнения задания с соблюдением дедлайна и в случае положительного ответа приступить к выполнению. Если же задание не может быть принято, пользователь может изменить требования к критическому сроку и повторно отправить. Количество заданий, которые могут быть запущены в кластере при условии, что требования по времени для всех заданий выполняются, должно быть максимальным. Результаты эксперимента показали, что, когда сроки выполнения задания разные, планировщик назначает к выполнению разное количество задач и следит за соблюдением указанного срока [9].

Схема планирования для минимизации пропущенных дедлайнов (критических сроков выполнения) заданий реального времени (например, обработка больших мультимедийных данных). Для выполнения задания используется узел, содержащий реплику требуемых данных, если назначенный узел занят. Если доступные узлы не найдены, а ожидаемое время выполнения задания превышает крайний срок, осуществляется поиск самого несрочного задания, и выполняющаяся задача этого задания временно приостанавливается, а ресурсы передаются заданию с истекающим дедлайном. Результат оценки эффективности показал, что предложенная схема сократила время выполнения и улучшила показатель успешности сроков [10].

Адаптивные планировщики выделяют ресурсы для конкретной задачи на основании информации о производительности узлов кластера и определенных пользователем целей для конкретной задачи.

Адаптивный планировщик ATAS (Adaptive Task Allocation Scheduler) для гетерогенных облаков

Усовершенствование исходного спекулятивного метода выполнения Hadoop и планировщика LATE. В ATAS используются более точные методы для определения времени отклика и задач резервного копирования (создания реплик), влияющих на систему, что, как ожидается, повысит коэффициент успешности задач резервного копирования и таким образом эффективно повысит способность системы к реагированию. Эксперименты по моделированию показали, что предложенная схема ATAS может эффективно повысить производительность обработки MapReduce [11].

Метод адаптивного планирования для повышения эффективности и производительности при использовании в гетерогенной среде

В этой модели планировщику сообщается информация о ресурсах кластера и требованиях к заданиям и задается алгоритм классификации. Этот алгоритм классифицирует задания на две категории: исполняемые и неисполняемые. Затем исполняемые задания назначаются соответствующим узлам для успешного выполнения

без сбоев, что увеличивает время выполнения задания. Этот планировщик решает проблему голодания, то есть долгого ожидания для небольших заданий, которая не решается в планировщиках, описанных ранее, а также проблему зависших узлов (stickynode) в справедливом планировщике и несоответствие между ресурсами и заданием. Адаптивный планировщик повышает производительность модели MapReduce в гетерогенной среде, минимизируя накладные расходы главного узла и сетевой трафик[12].

ELRAS – эффективное планирование с учетом локальности и реплик (EfficientlocalityandReplica-AwareScheduling), интегрированных со схемой автономной репликации (AutonomousReplicationScheme –ARS) для обеспечения локальности данных и последовательной работы в гетерогенной среде. ARS самостоятельно принимает решение о репликации объекта данных, учитывая его популярность, и удаляет реплику, если она простаивает. Результаты подтвердили эффективность алгоритма для гетерогенных кластеров и рабочих нагрузок[13].

Балансировка нагрузки MapReduce

В стандартном HadoopMapReduce каждый файл данных в файловой системе HDFS разделен на блоки фиксированного размера, для каждого блока в кластере создаются три реплики на трех разных узлах данных (DataNodes), так, чтобы на одном узле не было двух одинаковых копий и не было одинаковых копий на узлах данных, размещенных в одной стойке (rack). Конечно, при условии, что стоек достаточно (RackAwareness[5]). Это уменьшает вероятность того, что две копии данных одновременно могут стать недоступными из-за выхода из строя узла или стойки. Однако при размещении реплик на узлах данных текущая нагрузка на узел не принимается во внимание. Встроенный балансировщик выполняется многократно, перемещая блоки данных из перегруженных узлов данных в недогруженные (динамическая маршрутизация данных [14]). То есть постоянно происходит балансировка несбалансированного кластера, а лучше было бы поддерживать максимально сбалансированный кластер с самого начала, чтобы уменьшить миграцию нагрузки, потребляющую большое количество системных ресурсов. Поэтому разработка усовершенствованных методов балансировки нагрузки для Hadoop ведется постоянно, некоторые из этих методов описываются ниже.

Cogset унифицированный движок для балансировки нагрузки через статическую маршрутизацию данных[14]. Авторы показали, что слабая связь между HDFS и механизмом MapReduce является причиной плохой локализации данных и требует большого количества перемещений блоков данных, поэтому был предложен подход со статической маршрутизацией. Статическая маршрутизация снижает потребность в хранении временных копий промежуточных данных, но требует более тесной связи между компонентами для хранения и обработки. Чтение и запись меньшего количества временных данных может повысить производительность, а тесная связь между хранением и обработкой может быть использована для повышения локальности данных. Cogset тесно объединяет файловую систему и вычисления в распределенную систему хранения, которая поддерживает параллельную обработку данных на узлах хранения. Cogset состоит из двух этапов: на этапе разделения и репликации хранилище данных распределяется по кластеру, на этапе траверса (обхода) осуществляется доступ к данным. Производительность Cogset оценивалась в сравнении с широко используемым традиционным движком MapReduce на ранее установленном тесте. Результаты тестирования подтвердили более высокую производительность движка на основе статической маршрутизации. В ходе работы авторами была получена система со значительно более высокой производительностью, чем у Hadoop, особенно при работе с небольшими и умеренными объемами данных, однако этот подход не является полностью масштабируемым[9, 14]

Tarazu – набор оптимизаций MapReduce для решения проблемы низкой производительности MapReduce в гетерогенных кластерах. Авторы считают, что низкая производительность MapReduce связана с двумя факторами: чрезмерным и

неравномерным сетевым трафиком, а также с неоднородностью кластера. Tarazu включает: (1) балансировку нагрузки с учетом коммуникаций для стадии Map (Communication-AwareLoadBalancingofMapcomputation–CALB), (2) планирование заданий стадии Map с учетом коммуникаций (Communication-AwareSchedulingofMapcomputation–CAS) во избежание всплесков сетевого трафика и (3) прогнозирующей балансировки нагрузки Reduce (PredictiveLoadBalancingofReducecomputationPLB). С помощью моделирования авторы показали, что использование Tarazu значительно повышает производительность по сравнению с традиционным HadoopMapReduce в гетерогенных кластерах[9].

BlockSplit – блочный алгоритм балансировки нагрузки для уменьшения пространства поиска для задач идентификации сущностей (EntityResolution, ER). ER– это процесс определения того, относятся ли описания (или ссылки) сущности к одному объекту или к разным. Термин "сущность" означает какой-то объект реального мира. Сущности описываются атрибутами, некоторые из атрибутов могут быть идентифицирующими, то есть однозначно определяющими объект. Методы ER обычно сравнивают пары сущностей по атрибутам. ER обрабатывается в рамках двух заданий MapReduce. Оба задания основаны на одинаковом количестве задач map и одинаковом разделении входных данных. Первое задание MapReduce распределяет сущности с одинаковыми значениями выбранных (ключевых) атрибутов по блокам, ключ блока состоит из значений одного или нескольких атрибутов сущности. Также создается матрица распределения блоков (BlockDistributionMatrix, BDM), в ней указывается количество объектов для каждого блока. Последующий поиск совпадающих сущностей будет производиться уже внутри блока. Матрица используется на этапе Map второго задания MapReduce. BlockSplit анализирует количество совпадений на блок и распределяет блоки между задачами reduce. Размеры блоков могут быть разными, большие блоки могут привести к перекосу данных, поэтому они разбиваются на подблоки[15]. Этот алгоритм может быть модифицирован для работы с несколькими ключами блока [9].

Ориентированный на стойку (Rack-Aware) алгоритм динамической балансировки нагрузки для HadoopMapReduce [16]. Этот алгоритм балансирует нагрузку на стойку (rack), в то время как в предыдущих работах пытались балансировать нагрузку между отдельными узлами данных. В стандартном MapReduce и его оптимизации у Hadoop не было возможности гарантировать, что стойки с более высокими возможностями будут иметь большую рабочую нагрузку, чем стойки с более низкими возможностями. Другими словами, при назначении рабочей нагрузки узлам данных вычислительная мощность не имела значения. Для улучшения передачи данных алгоритм использует программно определяемую сеть (SoftwareDefinedNetwork SDN).

Адаптивный MapReduce

В стандартном MapReduce предполагается, что на стадии Map узлы, выполняющие функцию map() – mappers, работают изолированно. Авторы [17] предложили использовать асинхронный канал, называемый хранилищем распределенных метаданных (DistributedMetaDataSet- DMDS), для обмена информацией между mappers. Такие узлы называются Situation-Aware-Mappers (SAM). Так как на стадии Map выполняется несколько операций, создаются AdaptiveMappers, AdaptiveCombiners, AdaptiveSamplingandPartitioning. AdaptiveMappers объединяют небольшие разделы в виртуальный раздел, таким образом создавая больше разделов, что позволяет избежать частых контрольных точек и дисбаланса нагрузки. AdaptiveCombiners выполняют агрегацию на основе хэшей, а не на основе сортировки. В отличие от стандартного MapReduce, AdaptiveSampling динамически создает локальные выборки, объединяет их и создает гистограмму использования. AdaptivePartitioning может использовать глобальную гистограмму для создания разделов одинакового размера для лучшей балансировки нагрузки. Хотя SAM могут решить проблему перекоса данных, они не могут устранить перекос вычислений в Reducer. Экспериментальная оценка показала, что адаптивные методы значительно улучшают производительность MapReduce и особенно стабильность производительности [17].

Использование MapReduce для балансировки в облачных средах

Подход MapReduce может использоваться для балансировки нагрузки внутри и между центрами обработки данных (ЦОД), в этом случае сами запросы могут рассматриваться как большие данные, это особенно актуально для социальных сетей с их сотнями миллионов пользователей. Запрос с узла пользователя приходит на контроллер облака, который выбирает виртуальную машину (VM) для его обработки – узел Мастер. Задание обработки запроса пользователя может содержать несколько задач, поэтому Мастер назначает VM (рабочие узлы, slaves или workers) для выполнения задач – стадия Map задачи MapReduce. VM могут находиться и в другом ЦОД. Результаты, полученные подчиненными узлами, собираются на Мастере – стадия Reduce, и отсылаются пользователю. Например, для трехуровневого веб-приложения[18] Мастером будет являться сервер приложений, принимающий запросы от пользователя, формирующий страницу результата и отправляющий его пользователю.

В работе [19] предлагается адаптивный подход, основанный на анализе вычислительной мощности, географического расстояния от узла пользователя и загрузки ЦОД. На стадии Map сервер облака принимает запросы от пользователей и передает их балансировщику нагрузки, который группирует «похожие» запросы по параметрам – географическому положению, требуемым ресурсам, требованиями по времени и т.д. Функция map() готовит распределение похожих запросов по пакетам: создает пары «пакет, запрос». На стадии Reduce пары с одинаковым значением пакета объединяются, таким образом создаются пакеты похожих запросов. Эти пакеты рассылаются по ЦОД. С каждым ЦОД связан токен, содержащий информацию о географическом положении, рабочей нагрузке на каждый ЦОД, вычислительной мощности, времени отклика и времени задержки. Пакеты принимаются ЦОД на основании параметров заголовка токена. Если пакет согласуется с токеном, то он принимается в ЦОД для обработки запроса. В каждом ЦОД есть несколько виртуальных машин (VM) для обработки запросов. VM обрабатывают запросы от контроллера ЦОД на основе параметров токена, и результат выдается пользователю. Таким образом, большие наборы запросов разбиваются на более мелкие (пакеты), а рабочая нагрузка распределяется по каждому ЦОД на основе параметров токена. Поэтому время отклика на запрос, задержка обработки, накладные расходы сокращаются, а пропускная способность системы повышается.

Этот алгоритм оптимизирует нагрузку на трех уровнях. На уровне инфраструктуры виртуальные машины распределяются по ЦОД исходя из географического расстояния от узла пользователя, причем принимаются во внимание также пропускная способность сети и вычислительная мощность ЦОД. Для балансировки на уровне приложений выполняется задача MapReduce, запросы распределяются по пакетам, анализируется заголовок токена и выбирается ЦОД, которому будет назначена задача. Внутри ЦОД (уровень платформы) для балансировки нагрузки и минимизации сетевого трафика между узлами также используется планировщик MapReduce, контроллер ЦОД может принять или не принять задачу из пакета.

Алгоритмы балансировки, основанные на природных явлениях

Отдельный класс для балансировки нагрузки, в том числе и в задачах MapReduce, представляют алгоритмы, основанные на природных явлениях (NatureInspired Algorithms, NIA). Например, алгоритмы, основанные на поведении колоний насекомых или животных, когда путем обмена очень небольшим количеством информации между не слишком интеллектуальными особями стая птиц, рыб, рой насекомых в целом ведет себя разумным, оптимальным для выживания образом, то есть создается интеллект роя (Swarm Intelligence, SI). Если представить распределенную вычислительную систему, как рой, в котором узлы могут обмениваться информацией, можно добиться полностью децентрализованного саморегулируемого поведения системы. Такими алгоритмами являются алгоритм оптимизации колонии муравьев (AntColony Optimization, ACO), метод

оптимизации роя частиц (ParticleSwarmOptimization, PSO) и алгоритм оптимизации пчелиного роя (BeeColonyOptimization, BCO) и их вариации [9, 20, 21].

Заключение

В статье рассмотрены некоторые вопросы балансировки нагрузки для задач обработки больших данных MapReduce. Отсутствие или недостаточно тщательно проводимая балансировка нагрузки в MapReduce может привести к нарушению принципа локальности данных, к перекосу данных и вычислений на всех стадиях обработки. На балансировку нагрузки влияет планирование заданий и задач, поэтому рассматривались примеры и алгоритмов планирования, и алгоритмов балансировки. Дано краткое описание планировщиков: FIFO (простая очередь), «справедливый» планировщик, планировщик вычислительной мощности, планировщик задержек, планировщик по максимальному времени до завершения LATE(LongestApproximateTimeto End), планировщик дедлайнов, планировщик, минимизирующий количество пропущенных дедлайнов, адаптивные планировщики. Рассмотрены методы балансировки нагрузки: балансировка через статическую маршрутизацию данных Cogset, балансировка нагрузки в гетерогенных кластерах Tarazu, блочные алгоритмы балансировки с одним и несколькими ключами, динамический алгоритм балансировки нагрузки на стойку. Технология MapReduce может применяться к балансировке облачной инфраструктуры, когда поток запросов от пользователей рассматривается как источник больших данных, рассмотрен пример адаптивного подхода [19]. Исследования в этой области особенно перспективны, принимая во внимание возрастающий трафик, например, социальных сетей. Также следует отметить, что при разработке новых алгоритмов балансировки нагрузки, помимо традиционных времени отклика и использования ресурсов, должны приниматься во внимание такие метрики, как энергопотребление и углеродная эмиссия центров обработки данных.

Литература

1. Атаманов, Ю. С. Введение в BigData / Ю. С. Атаманов, В. С. Гончарук, С. Н. Гордеев. // Молодой ученый. — 2017. — № 11 (145). — С. 33–34. — URL: <https://moluch.ru/archive/145/40562/> (дата обращения: 25.10.2022).
2. ShayeganFard, Mohammad & Vakili, Meysam.. Load balancing in MapReduce on homogeneous and heterogeneous clusters: An in-depth review. International Journal of Communication Networks and Distributed Systems - 2015)..
3. Ivanisenko, I.N., Radivilova, T.A., 2015. Survey of Major Load-balancing algorithms in Distributed System. Information Technologies in Innovation Business Conference(ITIB). – 2015
4. David Taylor. What is MapReduce in Hadoop? Big Data Architecture. 2022 URL: <https://www.guru99.com/introduction-to-mapreduce.html> / (дата обращения: 25.10.2022).
5. <https://hadoop.apache.org/>
6. M. Jones. Scheduling in Hadoop. An introduction to the pluggable scheduler framework / URL: <https://developer.ibm.com/articles/os-hadoop-scheduling/> (дата обращения: 25.10.2022).
7. Zaharia, M., Borthakur, D., Sarma, J.S.,etal.:‘Job scheduling for multi-user mapreduce clusters’, Technical Report, UCB/EECS-2009-55, Electrical Engineering and Computer Sciences, University of California at Berkeley, 2009.
8. Zaharia, M., Borthakur, D., Sarma, J.S., 2010. Delay Scheduling: A Simple Technique for Achieving Locality and Fairness in Cluster Scheduling, in Proceedings of the European conference on Computer systems (EuroSys'10), 265–278.

9. EinollahJafarnejadGhomia, Amir MasoudRahmani and NooruldeenNasihQaderb (2017), "Load-balancing algorithms in cloud computing: A survey", Journal of Network and Computer Applications Volume 88, Pages 50-71.
10. Bok, K., Hwang, J., Jongtae Lim, J., Kim, Y., Yoo, J., 2016. An efficient MapReduce scheduling scheme for processing large multimedia data. *Multimed. Tools Appl.*,1–24.
11. Yang, S.J., Chen, Y.R., 2015. Design adaptive task allocation scheduler to improve MapReduce performance in heterogeneous clouds. *J. Netw. Comput. Appl.* 57, 61–70.
12. Ghoneem, M., Kulkarni, L., 2016. An Adaptive MapReduce Scheduler for Scalable Heterogeneous Systems. *Proceeding of the International Conference on Data Engineering and Communication Technology*, 603–6011.
13. Benifa, J.V.B., Dejeu, 2017. Performance improvement of MapReduce for heterogeneous clusters based on efficient locality and Replica aware scheduling (ELRAS) strategy. *Wirel. Personal. Commun.*, 1–25
14. Valvåg, S.V., 2011. Cogset: A High-Performance MapReduce Engine. Faculty of Science and Technology Department of Computer Science, University of Tromsø, 14.
15. Kolb L., Thor A., Rahm E., 2012. Load Balancing for MapReduce-based Entity Resolution, IEEE In: *Proceedings of the 28th International Conference on Data Engineering*, 618-629.
16. R. Vernica, A. Balmin, K. S. Beyer, V. Ercegovac, "Adaptive MapReduce using Situation-Aware Mappers," 2012.
17. Hou, X., Kumar, A., Varadharajan, V., 2014. Dynamic Workload Balancing for Hadoop MapReduce. *Proceeding of International Conference on Big data and Cloud Computing*, 56-62.
18. Гайдамако,В.В. Моделирование производительности многоуровневого веб-приложения на основе исчисления реального времени / В.В.Гайдамако // Проблемы автоматки и управления. 2020.–№2 (39) – С. 23–32.
19. N, Sowmya&Aparna, Manikonda&Tijare, Poonam &Nalini, N. (2015). An adaptive load balancing strategy in cloud computing based on Map reduce. 86-89. 10.1109/NGCT.2015.7375088.
20. Леонид Черняк. Интеллект роя для ИТ. – URL: <https://www.osp.ru/os/2014/02/13040047>

УДК 004

*Д. Авельцов, dmitry.aveltsov@gmail.com**Институт машиноведения и автоматизации НАН КР, г. Бишкек*

К ВОПРОСУ ПАРСИНГА БОЛЬШИХ ОБЪЕМОВ ДАННЫХ ФОРМАТА JSON СРЕДСТВАМИ ЯЗЫКА GO

В работе проведено тестирование производительности популярных библиотек, написанных на языке go, таких как: jstream, simdjson-go и стандартной библиотеки encoding/json. Рассмотрены ключевые особенности работы перечисленных библиотек посредством анализа исходного кода и приведены примеры использования данных библиотек. Тестирование производительности проводилось при помощи стандартных инструментов языка go из пакета testing. Основное внимание уделяется работе с файлами большого размера, от 1GB. Результаты тестирования говорят о том, что основная проблема популярных готовых решений для парсинга формата json заключается в работе с данными посредством ОЗУ вместо использования чтения из потока, что очевидно дает прирост в производительности, но не подходит для работы с большими данными.

Ключевые слова: формат данных json, язык программирования go, парсинг, тестирование производительности, библиотеки: jstream, encoding/json, simdjson-go.

Введение

JSON является одним из самых используемых форматов для передачи данных, данный формат удобен для использования программистами, работающими на различных языках, он легко читается и хорошо структурирован, но за удобство приходится платить свою цену, операция декодирования данных json формата достаточно ресурсоемкая. Разработчики языка go с каждым минорным релизом стараются усовершенствовать стандартную библиотеку языка go [1] для работы с данным форматом, но при работе с большими объемами данных она все еще оставляет желать лучшего с точки зрения производительности.

В экосистеме языка go существует множество альтернативных решений этой задачи, каждое из них имеет определенные преимущества и ограничения, и большинство решений не может быть использовано для декодирования больших файлов по причине того, что они не поддерживают потоковое чтение данных. Поддержка потоковых операций крайне важна для парсинга формата json. Размер входящих данных обычно заранее неизвестен, и буферизация всего файла в памяти – рискованная операция, так как может вызвать задержку в работе программы, и, кроме того, данные могут заполнить всю доступную память в системе. С использованием ограниченного буфера также существуют некоторые проблемы – для того, чтобы декодировать множество объектов json, требуется разделять файл не по размеру, а по логически целым частям данных. Так как данные в формате json могут иметь любую степень вложенности, эта задача индивидуальна для каждого отдельного типа файла.

В JSON формате не используются метки длины данных, и мы не можем заранее знать, где закончится объект, который начинаем читать. Чтобы узнать это, нужно прочитать его полностью, также мы не можем знать количество данных объектов заранее. Это означает, что при чтении входных данных мы должны определить, где начинаются и заканчиваются структуры при определении начальных и конечных токенов в потоке.

В описании многих решений для парсинга JSON формата мы можем найти сравнение производительности с аналогичными решениями, и на первый взгляд каждое

такое решение кажется привлекательным для использования в своих проектах. В настоящей работе поставлена задача разобрать некоторые из самых популярных решений и выяснить, каким образом они подходят к разбору данных и насколько они будут полезны в ситуации, когда стоит задача работать не с короткими сообщениями, а с файлами большого размера от 1GB. Для анализа библиотек используются стандартные средства языка go. Тестируются файлы как относительно небольших размеров, так и файлы большего размера там, где это возможно. Проводятся измерения по таким параметрам, как количество наносекунд, затраченных на операцию, количество байт, выделенных в оперативной памяти на операцию, и количество аллокаций памяти на операцию.

JSON токены

JSON основан на двух структурах данных [2]:

- Коллекция пар ключ/значение. В разных языках эта концепция реализована как объект, запись, структура, словарь, хэш, именованный список или ассоциативный массив.
- Упорядоченный список значений. В большинстве языков это реализовано как массив, вектор, список или последовательность.

Но с точки зрения парсинга из потока JSON – это набор токенов, чтобы читать данные, мы должны разбить поток, ориентируясь на его токены.

Для примера возьмем строку:

```
{"int_key": 1, "bool_key": true, "array_key": [1, "two", null]}
```

Данная строка – это поток следующих символов:

- `{`, открывающая скобка, обозначающая набор пар имя (ключ) / значение;
- `"int_key"`, строка `int_key`, в данном случае также является ключом;
- `:`, двоеточие, разделитель между ключом и значением в паре ключ / значение;
- `1`, число, является значением для ключа `int_key`;
- `,`, запятая, разделитель между парами ключ / значение;
- `"bool_key"`, строка `bool_key`;
- `:`, двоеточие, разделитель между ключом и значением в паре ключ / значение;
- `true`, логическое значение для `true`;
- `,`, запятая, разделитель между парами ключ / значение;
- `"array_key"`, строка `array_key`, также является ключом;
- `:`, двоеточие, разделитель между ключом и значением в паре ключ / значение;
- `[`, открывающая квадратная скобка, обозначающая упорядоченный список значений;
- `1`, число, является значением упорядоченного списка `array_key`;
- `"two"`, строка `two`, является значением упорядоченного списка `array_key`;
- `null`, пустое значение;
- `]`, закрывающая квадратная скобка, завершает список значений;
- `}`, закрывающая фигурная скобка, завершает коллекцию ключей / значений.

Инструменты тестирования производительности

Язык Go позволяет тестировать производительность с помощью пакета `testing`. `Go-benchmark` позволяет протестировать производительность различных участков кода. Каждый тест выполняется по несколько раз, проводятся измерения времени на каждой

итерации. На основе полученных результатов данные усредняются. По завершении каждого теста мы получаем 3 значения: имя теста, количество итераций и ns/op.

Здесь ns/op означает число наносекунд, затраченных на операцию. Добавление флага – `benchmark` позволяет получить дополнительную информацию о том, сколько было выделено байт (B/op), а также сколько раз за операцию выделялась память (allocs/op).

Обзор и тестирование библиотек

Библиотека `encoding/json`

`Encoding/json`[3] – это json парсер, поддерживающий чтение из потока, стандартный пакет библиотеки языка go. Пакет `json` реализует кодирование и декодирование JSON, как определено в RFC 7159.

Декодирование из потока. Существуют два способа декодировать JSON. Используя `json.Decoder`, который позволяет декодировать поток байтов при помощи интерфейса `io.Reader` [4]

```
type Decoder struct {}
```

```
func NewDecoder(r io.Reader) *Decoder
```

```
func (dec *Decoder) Decode(v interface{}) error
```

В качестве альтернативы можно декодировать из байтового среза с помощью функции `json.Unmarshal()`

```
func Unmarshal(data []byte, v interface{}) error
```

Декодирование происходит в два этапа, сначала `scanner` преобразовывает входные байты в токены, а затем `decodeState` конвертирует токены в объекты go.

Сканирование JSON. Компонент `scanner` – это внутренний конечный автомат, используемый для анализа JSON. Он работает в несколько этапов. Сначала проверяется первый байт значения, чтобы определить тип токена. Если это символ "{", то необходимо работать с объектом, если это символ "[", то необходимо работать с массивом. По такому же принципу осуществляется определение простых типов, двойная кавычка обозначает начало строки, символы "t" или "f" – начало булевого значения, 0–9 обозначает начало числа.

Как только тип значения определен, работа передается функции, определенной для сканирования данного типа. Для сложных объектов используется отслеживание закрывающего токена.

Декодирование токенов. После того как токен просканирован, он должен быть интерпретирован. Для этого используется `decodeState`. На данном этапе происходит сопоставление значений, которые поступили на вход с типами данных, с теми, в которые они должны быть декодированы. Это означает, что если на вход подается тип структура (`struct`), то декодер будет ожидать, что первым символом будет открывающая фигурная скобка "{". Любой другой токен приведет к тому, что декодирование вернет ошибку.

Также можно обрабатывать токены в потоке, используя методы `Decoder.Token()` и `Decoder.More()`.

Тестирование производительности `encoding/json`. В тестировании использовались файлы формата JSON размером 410kb и 1GB. Структура `json`:

```

{
  "key": [
    {"key": "value", ...},
    {"key": "value", ...},
    ...
  ],
  "key": [
    {"key": "value", ...},
    {"key": "value", ...},
    ...
  ],
  ...
}

```

Результаты приведенных ниже тестов следует читать следующим образом:

Пример результатов тестирования:

```

cpu: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
BenchmarkStandard-12  1 12292917480 ns/op   5970553504  B/op      58151873
allocs/op
PASS
ok      command-line-arguments  12.295s,

```

где

cpu – характеристики процессора;
 ns/op – наносекунды на одну итерацию бенчмарка;
 B/op – количество на итерацию;
 allocs/op – количество аллокаций на итерацию;
 s – количество секунд на выполнение всего теста.

Тест производительности чтения из потока стандартной библиотеки encoding/json

Тест для файла размером 410kb:

```

cpu: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
BenchmarkStandard-12  1 0.004256 ns/op  0 B/op  0 allocs/op
PASS
ok      command-line-arguments  0.029s

```

Как видно из результатов тестирования, при проведении операции декодирования json данных в тип данных `interface{}` языка go на сравнительно небольшом объеме данных при одной итерации не выделяется дополнительная память, т.к. чтение происходит из потока. Но приложение работает не в идеальных условиях, и, скорее всего, будет обрабатываться более одного запроса или файла. Увеличим количество итераций.

```

cpu: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz
BenchmarkStandard-12  249 4206079 ns/op  0.00 MB/s  1836233 B/op  24405 allocs/op
PASS
ok      command-line-arguments  1.556s

```

Здесь мы видим, что на 249 итераций понадобилось уже 4206079 наносекунд и использование памяти, т.к. даже при чтении из потока тестируемая библиотека имеет свой внутренний буфер для чтения на один символ вперед.

Результат тестирования файла размером 1GB:

Одна итерация:

cpu: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz

*BenchmarkStandard-12 1 25356751291 ns/op 11940661944 B/op 116302358
allocs/op*

PASS

ok command-line-arguments 25.362s

Результаты 5 итераций

cpu: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz

*BenchmarkStandard-12 5 74238424830 ns/op 35823492088 B/op 348912176
allocs/op*

PASS

ok command-line-arguments 74.244s

Программный код тестирования библиотеки:

```
func BenchmarkStandard(b *testing.B)
(b.ReportAllocs)
for i:= 0; i<= b.N; i++
(b.StopTimer)
f, err := os.Open("mid.json")
if err != nil {
    Mpanic(err)
}
{vari interface}
dec := json.NewDecoder(bufio.NewReader(f))
b.StartTimer()
for {
    // read first { and list key
t, err := dec.Token()
if err != nil && err != io.EOF {
    panic(err)
}
if t != nil && reflect.TypeOf(t).Kind() == reflect.String && strings.HasSuffix(t.(string), "List") {
    // read [
if _, err = dec.Token(); err != nil && err != io.EOF {
    panic(err)
}
}
for dec.More() {
err = dec.Decode(&i)
if err != nil && err != io.EOF {
    panic(err)
}
}
```

```

// read ]
    _, err = dec.Token()

if err != nil && err != io.EOF {
    panic(err)
}

if err == io.EOF {
    break
}
}
f.Close()
}
}

```

Библиотека **jstream**

Jstream [6] – это потоковый парсер JSON и библиотека извлечения значений в структуры данных языка go.

В отличие от большинства парсеров JSON, jstream учитывает положение и глубину документа. Это позволяет извлекать значения на заданном уровне вложенности, устраняя накладные расходы на выделение памяти для массивов или объектов.

Результаты сравнительного тестирования со стандартной библиотекой на файле 410kb:

BenchmarkStandard-12 – стандартная библиотека

BenchmarkJstream-12 – jstream

cpu: Intel(R) Core(TM) i7-10750H CPU @ 2.60GHz

BenchmarkStandard-12 100 0.04707 ns/op 0 B/op 0 allocs/op

BenchmarkJstream-12 100 0.04633 ns/op 0 B/op 0 allocs/op

PASS

ok command-line-arguments 1.084s

Как мы можем видеть, стандартная библиотека уступает в скорости работы на одну итерацию, но тестирование невозможно на файлах размером в 1GB, т.к. автор библиотеки jstream использует буфер на более глубоких уровнях вложенности, что сказывается на объеме занимаемой оперативной памяти. Использование данной библиотеки весьма ситуативно, если структура файла имеет большое количество объектов уровнем ниже, чем нужно прочитать, все эти объекты будут записаны в оперативную память.

Библиотека **simdjson-go**

Simdjson-go [5] – это синтаксический анализатор с возможностью валидации. Это означает, что он среди прочего проверяет числовые и логические значения, которые после синтаксического анализа доступны как соответствующие представления int, float, bool.

В simdjson используется подход двухэтапного синтаксического анализа, с помощью которого можно достичь большей производительности на одном ядре центрального процессора. Использование SIMD [7] позволяет обеспечить параллелизм на уровне данных, что значительно повышает производительность.

Simdjson-go является портом для языка go оригинальной библиотеки simdjson, написанной на языке Си. Данный порт показывает 40–60% производительности от оригинальной библиотеки.

Simdjson-go следует тому же двухэтапному дизайну, что и simdjson. На первом этапе такие структурные элементы, как ({}), [,],: и ,) обнаруживаются и передаются в буфер сообщений на второй этап. На втором этапе строится ленточный формат структуры документа JSON. Для повышения производительности оба этапа выполняются одновременно как отдельные подпрограммы, для связи между двумя этапами используются go каналы [8].

Этап 1. Преобразован из исходного кода C (содержащего встроенные функции SIMD) в сборку Golang с использованием с 2goasm. Данный этап состоит из пяти шагов.

- *find_odd_backslash_sequences*: обнаруживает символы обратной косой черты, используемые для экранирования кавычек;
- *find_quote_mask_and_bits*: генерирует маску для символов, которые находятся между кавычками;
- *find_whitespace_and_structurals*: генерирует маску для пробелов плюс маску для структурных символов;
- *finalize_structurals*: объединяет маски, сгенерированные на предыдущих шагах в финальную маску, где каждый активный бит представляет позицию структурного символа во входном сообщении;
- *flatten_bits_incremental*: вывод активных битов в финальной маске как инкрементальные смещения.

Этап 2. Строится иерархия документа. По сути, на данном этапе выполняется единственная функция, которая находит различные структурные символы и строит иерархию документа JSON.

Пример использования библиотеки:

```
for {
  typ := iter.Advance()
  switch typ {
  case simdjson.TypeRoot:
    if typ, tmp, err = iter.Root(tmp); err != nil {
      return
    }
  case simdjson.TypeObject:
    if obj, err = tmp.Object(obj); err != nil {
      return
    }
  }
  e := obj.FindKey(key, &elem)
  if e != nil && elem.Type == simdjson.TypeString {
    v, _ := elem.Iter.StringBytes()
    fmt.Println(string(v))
  }
}
```

default:

```
return
  }
}
```

Тестирование данной библиотеки на файле большого размера не представляется возможным, т.к. для работы скрипта, указанного выше, необходимо прочитать в оперативную память весь файл. Данная библиотека имеет возможность работать с потоком, но данная часть поддерживает ограниченный формат NDJson (JSON с использованием новой строки для разделения записей) [9].

Заключение

Для решения задачи парсинга больших объемов данных формата JSON рассмотрено использование библиотек языка goencoding/json, jstream, simdjson-go. Существуют также другие библиотеки для парсинга JSON файлов на языке go, например, json-iterator, easyjson, fastjson, но подавляющее большинство этих библиотек решают задачу через оперативную память, что в случае объема данных, превышающего объем оперативной памяти, не работает. В момент написания статьи не обнаружено ни одной библиотеки, которая могла бы обрабатывать файлы большого размера с учетом ограниченности ресурсов и без предварительной подготовки данных. На данный момент наиболее эффективный способ работы с такими файлами – преобразование данных в формат NDJson с учетом структуры файла и передача этих данных на обработку библиотеке simdjson-go. Значимость данной проблемы с годами только возрастает, данных становится все больше и задач с разбором больших объемов потоковых данных появляется также больше [10]. Сегодня формат JSON принято воспринимать как множество небольших по объему сообщений (~1кб), которые можно с легкостью прочитать и разобрать, но когда перед разработчиком на языке go становится задача разобрать файл накопленной информации в формате JSON, то в данной ситуации каждый разработчик вынужден искать решение самостоятельно и писать собственную библиотеку под данную задачу, т.к. существующие решения никак не могут отвечать требованиям такого рода задач.

Литература

1. Adam Freeman. Pro Go. The Complete Guide to Programming Reliable and Efficient Software Using Golang. ApressBerkeley. 2022 (1076 с)
2. Tom Marrs. JSON at Work: Practical Data Integration for the Web. O'ReillyMedia, Inc. 2016 (376 с)
3. <https://pkg.go.dev/encoding/json> (дата обращения 25.08.2021)
4. <https://pkg.go.dev/io> (дата обращения 28.08.2021)
5. <https://github.com/bcicen/jstream> (дата обращения 28.08.2021)
6. <https://github.com/minio/simdjson-go> (дата обращения 29.08.2021)
7. Geoff Langdale, Daniel Lemire. Parsing Gigabytes of JSON per Second / researchgate.com. 2019 (С. 3–5)
8. <https://gobyexample.com/channels> (дата обращения 29.08.2021)
9. <http://ndjson.org/> (дата обращения 29.08.2021)
10. Толпинская, Н.Б., Сычев, А.М. Алгоритмы обработки потоковых данных. / Н.Б.Толпинская, А.М.Сычев // Проблемы автоматизации и управления. – 2019 – №1(36). – С. 110–117. – EDN ZXPAXD.

УДК 004

*Д. Амельцов**Институт машиноведения и автоматизации НАН КР, г. Бишкек*dmitry.aveltsov@gmail.com

ПРИМЕНЕНИЕ ПРОТОКОЛА СЕРИАЛИЗАЦИИ СТРУКТУРИРОВАННЫХ ДАННЫХ PROTOBUF В МИКРОСЕРВИСНОЙ АРХИТЕКТУРЕ

В работе обсуждаются вопросы обмена данными между сервисами при разработке приложений с микросервисной архитектурой, в частности, применение протокола сериализации данных Protocol Buffers и удаленного вызова процедур gRPC. Приводятся примеры использования Protocol Buffers и фреймворка Go-Micro. Проводится сравнение Protocol Buffers с популярными аналогами XML и JSON, даются рекомендации по их применению.

Ключевые слова: удаленный вызов процедур gRPC, Protocol Buffers, сериализация, десериализация, Golang, микросервисы, обмен данными, фреймворк go-micro.

Введение. В настоящее время в разработке информационных систем наблюдается тенденция перехода с монолитной на микросервисную и смешанную архитектуру. Основная причина этого заключается в необходимости масштабирования, то есть способности информационной системы сохранять работоспособность при значительном возрастании нагрузки. Это стало особенно актуальным с появлением множества устройств, которые имеют доступ в интернет и могут пересылать, получать и даже обрабатывать данные. Социальные сети, криптовалюты, интернет вещей – наглядный пример. С ростом количества устройств, количества клиентов растет и нагрузка на информационные системы. Повышать производительность за счет использования более мощных вычислительных устройств, более быстрых коммуникаций дорого и долго, к тому же возможности аппаратуры небезграничны, масштабирование должно происходить с помощью применения более продуманных алгоритмов и более эффективных методов построения систем. Просто отказаться от монолитной архитектуры в пользу микросервисной недостаточно, должны быть продуманы и оптимизированы все аспекты разработки и функционирования системы. Немаловажную роль в работе микросервисов играет вопрос передачи данных, и, если в передаче данных от сервиса к пользователю разработчики ориентируются на пользователей, то передача данных между сервисами определяется разработчиком и должна ориентироваться на максимальную эффективность.

Обмен данными. Сериализация и десериализация данных. Микросервисная архитектура – это способ разработки приложений в виде набора сервисов. Каждый сервис решает отдельную задачу, представляет собой отдельный процесс, может быть написан на наиболее подходящем для задачи языке программирования и развернут в независимой среде, то есть в контейнере. Это освобождает программистов от необходимости вести разработку только на одном языке. Набор таких сервисов представляет собой единую систему, все сервисы работают на достижение общей цели, и, скорее всего, им будет необходимо обмениваться информацией, как минимум для синхронизации. Поэтому очень важно определить способ и формат передачи данных для межсервисного обмена. Чаще всего для передачи данных в микросервисных системах используется прикладной уровень сетевой модели передачи данных – TCP/IP и протокол HTTP, но возможно использовать и транспортный уровень передачи данных – протоколы TCP или UDP. Все эти протоколы работают поверх протокола IP с автоматической маршрутизацией.

Сериализация – это процесс преобразования структурированных данных или состояния объекта в поток байтов для целей передачи, хранения и распространения на физических устройствах. Десериализация – это обратный процесс построения структуры данных/состояния объекта из потока последовательных байтов (рис.1). Самыми популярными форматами сериализации данных на сегодняшний день являются JSON (JavaScript Object Notation), XML(eXtensible Markup Language) и Protocol Buffers (Protobuf). Простота JSON способствует все более широкому его распространению и падению популярности XML. В то же время, если рассматривать вопрос в контексте обмена данными между сервисами, следует обратить внимание на Protobuf.

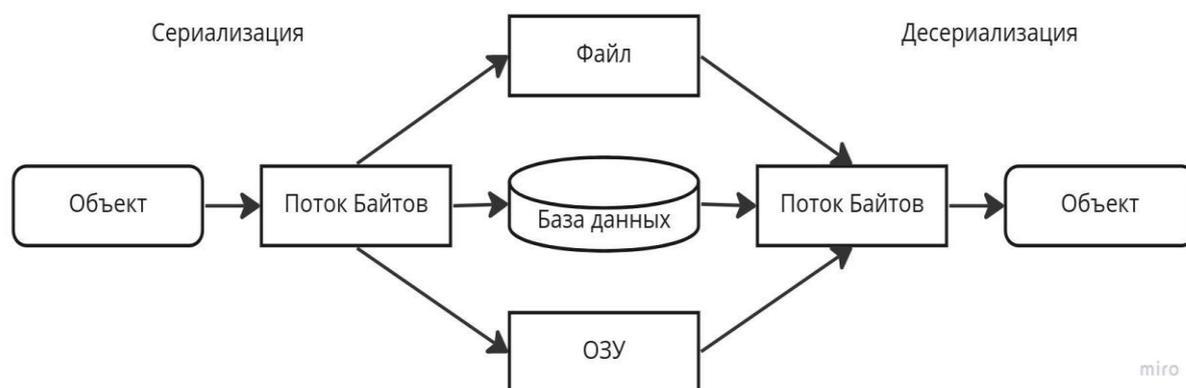


Рисунок 1. Сериализация и десериализация

Язык описания интерфейса Protocol Buffers обеспечивает независимый от языка и платформы расширяемый механизм сериализации структурированных данных с прямой и обратной совместимостью. На сегодняшний день поддерживает генерацию кода в Java, Python, Objective-C, C++, C#, JS, Go и других языках программирования. Дает возможность определить структуру данных, затем генерируется исходный код для записи и чтения структурированных данных, при этом могут использоваться различные языки программирования [1].

Формат протокола. Формат протокола – это закодированная последовательность полей, состоящих из ключа и значения, где ключ – это номер поля сообщения в proto-файле, а значение – это данные, которые требуется передать. Номер и тип поля кодируются в формате varint. В отличие от таких типов данных как int, int32, uint и тд., строки, повторяющиеся сообщения, наборы байтов не позволяют с точностью определить количество памяти, которая потребуется для их хранения, оно варьируется в зависимости от размера данных. В случаях, когда размер занимаемой памяти невозможно определить исходя из типа данных, размер данных указывается в формате varint вслед за указанием самого типа и номера поля.

При кодировании числа в формате varint в последовательность байтов у всех байтов, кроме последнего, старший бит выставляется в 1. При преобразовании в стандартное представление старший бит каждого байта отбрасывается, а оставшиеся 7-битные составляющие соединяются друг с другом в обратном порядке. Например, число 1 будет закодировано как 01. Это один байт и старший бит не устанавливается:

0000 0001

Для сравнения возьмем число 150, оно кодируется как 9601

10010110 00000001 [2]

Формат восьмиразрядного varint был выбран для уменьшения размера пакета при передаче небольших чисел. Так, если число меньше 128, то оно будет занимать лишь 1 байт. Однако, числа, близкие к максимально возможным, будут занимать больше места,

чем в обычном формате. Например, максимальное значение, которое можно сохранить в 8-ми байтах, в формате varint – 10 байтов. Отрицательные числа в формате varint всегда занимают наибольший размер, в зависимости от типа, поскольку старший бит у знакового числа выставляется в 1 [3].

Производительность. Форматы JSON и XML – текстовые, в отличие от Protobuf, который является бинарным. Процесс преобразования данных, хранящихся в оперативной памяти, в текстовый формат (маршалинг/демаршалинг) создает интенсивную нагрузку на центральный процессор, что, в случае небольшого количества передаваемых данных, не вызовет заметного снижения производительности. Но, если программная система принимает данные от большого количества устройств, и эти данные при обработке должны пройти через множество сервисов, производительность системы может существенно пострадать. Если каждый сервис выполняет небольшое количество операций, внутри системы придется пересылать большие объемы данных, причем каждый раз они могут требовать преобразований. Если система разворачивается в облачном хостинге, это приведет и к повышению финансовых затрат, так как облачные хостинги берут плату за процессорное время. Если же компонент системы запущен на устройстве с ограничениями по питанию, например, на мобильном устройстве, нагрузка на центральный процессор приведет к быстрому разряду батареи.

Сравнение Protobuf и XML/JSON. Ниже приведены примеры простого сообщения в различных форматах с использованием различных типов данных.

XML

```
<?xml version="1.0" encoding="UTF-8" ?>
<position>
  <min_position>true</min_position>
  <?has_more_items>>false
</?has_more_items>
<items_to_parse>none</items_to_parse>
<count>0</count>
<data>
  <length>200</length>
  <text>Text Field Example</text>
</data>
<numericalArray>4</numericalArray>
<numericalArray>5</numericalArray>
<StringArray>An</StringArray>
<StringArray>Array</StringArray>
<objArray>
  <name>Object</name>
  <smaple>9</smaple>
</objArray>
<objArray>
  <name>name</name>
  <num>3</num>
</objArray></position>
```

JSON

```
{ "position" : {
  "min_position": true,
  "?has_more_items": false,
```

```

    "items_to_parse": "none",
    "count": 0,
    "data": {
    "length":200,
    "text":"Text Field Example"
    },
    "numericalArray" : [4,5],
    "StringArray" : ["An","Array"],
    "objArray" : [{"name":"Object","smaple":9}, {"name":"name","num":3}]
  }
}

```

Protobuf

```

message position {
  required bool min_position = 0
  optional bool has_more_items = 1
  required string items_to_parse = 2
  required double count = 3
  message DATA {
    required double length = 0
    required string text = 1
  }
  required DATA data = 4
  repeated float numericalArray = 5
  repeated string StringArray = 6
  message OBJARRAY {
    required string name = 0
    required double age = 1
  }
  repeated OBJARRAY objArray = 7
}

```

Как видно, форматы XML и JSON являются “самоописательными” форматами, и описание происходит путем добавления данных. В отличие от них Protobuf требует сначала определить структуру самого сообщения, после чего на основе описания структуры генерируются классы или функции для работы с данными на различных языках.

Рассмотрим основные преимущества использования Protobuf перед JSON/XML:

Protobuf более компактен и быстрее работает для представления структур данных, чем XML/JSON;

JSON/XML за счет текстового представления более требовательны к ресурсам при процессе маршалинга/демаршалинга. Предопределенная схема данных Protobuf упрощает процесс преобразования данных, что позволяет избежать ошибок на этапе компиляции;

Работа с массивами, строками и объектами в JSON/XML требует последовательного сканирования всего множества элементов, входящих в структуру данных, следовательно, нет никакой возможности узнать количество элементов, не прочитав всю структуру данных до конца;

Валидация структуры данных. При работе с JSON может возникнуть ситуация, когда какой-либо элемент данных отсутствует или в названии ключа появляется опечатка. В такой ситуации приходится явно проверять наличие самого ключа, что

тоже несет накладные расходы, в то время как Protobuf гарантирует, что структура сообщения останется неизменной.

Несмотря на перечисленные преимущества Protobuf, все еще достаточно много веб-сервисов используют XML/JSON по следующим причинам:

Protobuf сложнее в изучении, чем XML/JSON;

XML/JSON легко читаются и редактируются человеком. Сообщения в формате Protobuf невозможно прочитать без специальных преобразований;

В случае работы с веб-интерфейсом, когда в качестве основного языка программирования используется javascript, формат JSON является предпочтительным, так как может напрямую конвертироваться в структуры данных javascript. JSON и создавался как формат описания объектов JavaScript (javascript object), и преобразование JSON в объект является естественным для javascript.

Случаи, когда рекомендуется использование Protobuf:

Protobuf используется для передачи данных между сервисами, а не для взаимодействия с веб-интерфейсами на стороне клиента.

Если необходимо достичь более высокого уровня производительности в обмене данными между элементами системы.

При переходе с архитектурного стиля взаимодействия компонентов REST (*Representational State Transfer*, «передача репрезентативного состояния», определяет взаимодействие компонентов распределенного приложения) на gRPC (система удаленного вызова процедур Google), где Protobuf является рекомендуемым стандартом.

К вопросу о gRPC и Protocol Buffers в микросервисной архитектуре.

Ответственность сервисов в программной системе можно условно разделить на две группы. Часть сервисов (внешние сервисы) отвечает за внешнее взаимодействие с пользователями, то есть предоставляет программный интерфейс для взаимодействия с системой. Здесь для передачи данных часто, в силу распространенности, используются форматы REST (передача репрезентативного состояния) и JSON. Вторая группа сервисов (внутренние сервисы) предназначена для взаимодействия внутренних компонентов системы, эти сервисы взаимодействуют только друг с другом. Внутренние сервисы могут использовать либо синхронный, либо асинхронный режим связи. В некоторых случаях используется синхронный режим связи посредством REST API по HTTP, но в качестве оптимизированного протокола межсервисного взаимодействия более подходящим является бинарный протокол gRPC.

Что такое RPC. Удаленный вызов процедур (иногда вызов удаленных процедур; RPC от англ. remote procedure call) – класс технологий, позволяющих программам вызывать функции или процедуры в другом адресном пространстве (на удаленных узлах, либо в независимой сторонней системе на том же узле). Обычно реализация RPC-технологии включает два компонента: сетевой протокол для обмена в режиме клиент-сервер и язык сериализации объектов (или структур для необъектных RPC). Различные реализации имеют отличающуюся друг от друга архитектуру и возможности: одни реализуют архитектуру SOA, другие – CORBA или DCOM. На транспортном уровне RPC используют в основном протоколы TCP и UDP, однако, некоторые построены на основе HTTP (что нарушает архитектуру ISO/OSI, так как HTTP изначально не является транспортным протоколом).

Технология вызова удаленных процедур распространяет механизм передачи управления и данных между программами, выполняющимися на одном узле, на передачу управления и данных через сеть на другой узел. Удаленный вызов процедур используется для организации распределенных вычислений и создания распределенных клиент-серверных приложений. Использование RPC наиболее эффективно в тех

приложениях, где удаленные компоненты взаимодействуют с небольшим временем отклика и относительно малым количеством передаваемых данных. Такие приложения называются RPC-ориентированными [4, 5]. Проще говоря, RPC позволяет напрямую вызывать методы в серверном приложении на другом компьютере, так же, как если бы это был локальный объект.

Что такое gRPC. gRPC (Remote Procedure Calls) (рис. 2) – это система удалённого вызова процедур (RPC) с открытым исходным кодом, первоначально разработанная в Google в 2015 году. В качестве транспорта используется HTTP/2, в качестве языка описания интерфейса – Protocol Buffers. gRPC предоставляет такие функции, как аутентификация, двунаправленная потоковая передача и управление потоком, блокирующие или неблокирующие привязки, а также отмена и тайм-ауты. Генерирует кроссплатформенные привязки клиента и сервера для многих языков. Чаще всего используется для подключения служб в микросервисном стиле архитектуры и подключения мобильных устройств и браузерных клиентов к серверным службам.

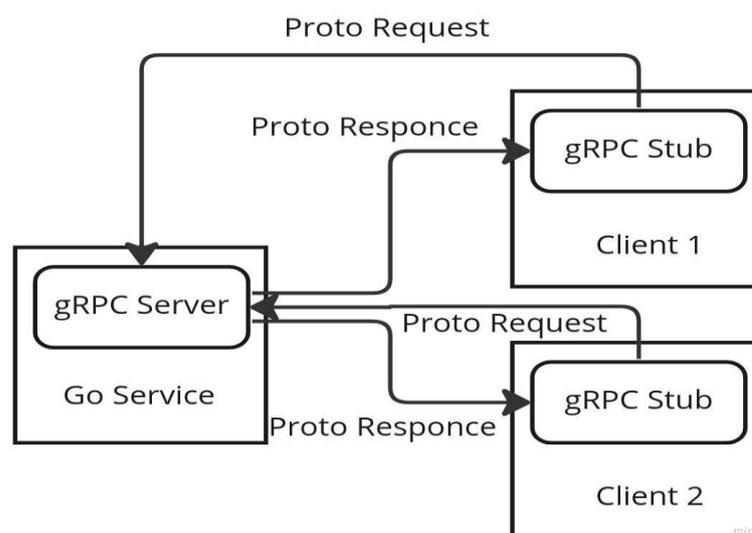


Рисунок 2. Удалённый вызов процедур gRPC

gRPC позволяет определить **четыре типа методов обслуживания**:

Унарный RPC (Unary RPC) – клиент отправляет один запрос на сервер и получает один ответ обратно, как при обычном вызове функции (рис. 3.1);

Серверные потоковые RPC (Server streaming RPC) – когда клиент отправляет запрос на сервер и получает обратно поток для чтения последовательности сообщений. Клиент читает из возвращенного потока, пока не кончатся сообщения. gRPC гарантирует упорядочение сообщений в рамках отдельного вызова RPC (рис. 3.2);

Клиентские потоковые RPC (Client streaming RPC), в которых клиент записывает последовательность сообщений и отправляет их на сервер, снова используя предоставленный поток. После того, как клиент закончил писать сообщения, он ждет, пока сервер прочтает их и вернет свой ответ. И здесь gRPC гарантирует упорядочение сообщений в рамках отдельного вызова RPC (рис. 3.3);

RPC с двунаправленной потоковой передачей (Bidirectional streaming RPC), где обе стороны отправляют последовательность сообщений, используя поток чтения-записи. Два потока работают независимо, поэтому клиенты и серверы могут читать и писать в любом порядке: например, сервер может дождаться получения всех клиентских сообщений, прежде чем записывать свои ответы, или он может поочередно читать сообщение, а затем писать ответное сообщение, или может быть какая-то другая

комбинация чтения и записи. Порядок сообщений в каждом потоке сохраняется (рис. 3.4) [6].

В каких случаях использовать gRPC. Как уже было сказано выше, gRPC применяется во внутренней инфраструктуре программной системы, которая не предоставляется для доступа внешним пользователям:

В случае, когда эффективность передачи данных между множеством микросервисов имеет критическое значение. gRPC обеспечивает связь с низкой временной задержкой и высокую пропускную способность.

В системе, где используется несколько языков программирования. gRPC поддерживается инструментарием различных языков программирования, что позволяет достаточно легко интегрировать данную технологию и не привязываться к одному языку для всей системы.

gRPC позволяет управлять двунаправленной потоковой передачей данных, что позволяет исключить ожидание ответа и отправлять сообщения в режиме реального времени.

Маломощные сети с низкой пропускной способностью. За счет использования protobuf в качестве формата передачи данных в gRPC.

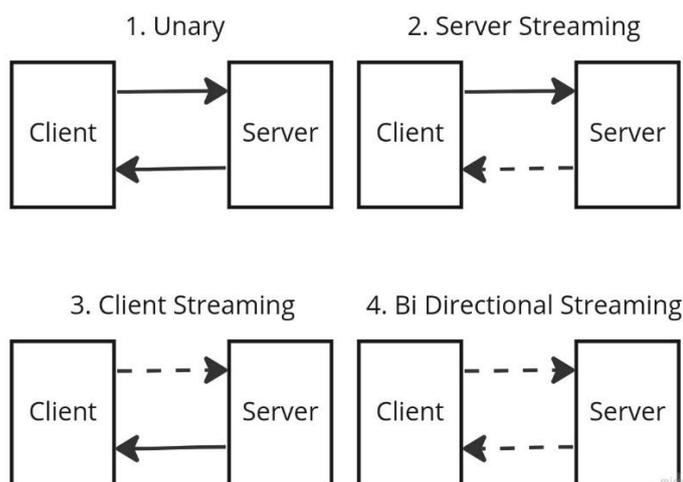


Рисунок 3. Методы обслуживания gRPC

Пример использования Protocol Buffers и gRPC в программной системе виртуализации сенсорных устройств [7]

В качестве основного языка разработки программной системы был выбран язык программирования Go [8]. В качестве фреймворка для связи между микросервисами был выбран go-micro. Go-Micro фреймворк предназначен для разработки распределенных систем, включает в себя gRPC и управление событиями [9].

Пример описания protobuf сообщения для взаимодействия с реестром данных о шаблонах сенсорных устройствах.

```

syntax = "proto3";
package sensorset.service.template_registry;
service TemplateRegistryService {
  rpc Stream(StreamingRequest) returns (stream StreamingResponse) {}
  rpc PingPong(stream Ping) returns (stream Pong) {}
  rpc NewSensorTemplate(Request) returns (Response) {}
}

```

```

message Request{
  string user_uuid = 1;
  string template = 2;
}
message Response {
  string msg = 1;
}
message StreamingRequest {
  int64 count = 1;
}
message StreamingResponse {
  int64 count = 1;
}
message Ping {
  int64 stroke = 1;
}
message Pong {
  int64 stroke = 1;
}

```

Здесь представлены три процедуры, доступные для удаленного вызова.

Stream. Служит для передачи потока данных о шаблоне сенсорного устройства. (Запись/чтение).

PingPong. Служит для проверки состояния сервиса, используется для мониторинга состояния системы.

NewSensorTemplate. Используется для регистрации нового шаблона.

Ниже приведен пример сгенерированного кода для сообщения типа Request.

```

type Request struct {
  UserUuid string `protobuf:"bytes,1,opt,name=user_uuid,json=userUuid,proto3"
json:"user_uuid,omitempty"`
  Template string `protobuf:"bytes,2,opt,name=template,proto3"
json:"template,omitempty"`
  XXX_NoUnkeyedLiteral struct{} `json:"- "`
  XXX_unrecognized []byte `json:"- "`
  XXX_sizecache int32 `json:"- "`
}
func (m *Request) Reset() { *m = Request{} }
func (m *Request) String() string { return proto.CompactTextString(m) }
func (*Request) ProtoMessage() {}
func (*Request) Descriptor() ([]byte, []int) {
  return fileDescriptor_be913841faaa2c9e, []int{0}
}
func (m *Request) XXX_Unmarshal(b []byte) error {
  return xxx_messageInfo_Request.Unmarshal(m, b)
}
func (m *Request) XXX_Marshal(b []byte, deterministic bool) ([]byte, error) {
  return xxx_messageInfo_Request.Marshal(b, m, deterministic)
}
func (m *Request) XXX_Merge(src proto.Message) {

```

```

xxx_messageInfo_Request.Merge(m, src)
}
func (m *Request) XXX_Size() int {
return xxx_messageInfo_Request.Size(m)
}
func (m *Request) XXX_DiscardUnknown() {
xxx_messageInfo_Request.DiscardUnknown(m)
}
var xxx_messageInfo_Request proto.InternalMessageInfo
func (m *Request) GetUserUuid() string {
if m != nil {
return m.UserUuid
}
return ""
}
func (m *Request) GetTemplate() string {
if m != nil {
return m.Template
}
return ""
}
func init() {
proto.RegisterType((*Request)(nil), "sensorset.service.template_registry.Request")
}

```

Как выглядит код для RPC

```

type TemplateRegistryService interface {
Stream(ctx context.Context, in *StreamingRequest, opts ...client.CallOption)
(TemplateRegistryService_StreamService, error)
PingPong(ctx context.Context, opts ...client.CallOption)
(TemplateRegistryService_PingPongService, error)
NewSensorTemplate(ctx context.Context, in *Request, opts ...client.CallOption)
(*Response, error)
}
type templateRegistryService struct {
c client.Client
name string
}
func NewTemplateRegistryService(name string, c client.Client)
TemplateRegistryService {
return &templateRegistryService{
c: c,
name: name,
}
}
func (c *templateRegistryService) NewSensorTemplate(ctx context.Context, in
*Request, opts ...client.CallOption) (*Response, error) {
req := c.c.NewRequest(c.name, "TemplateRegistryService.NewSensorTemplate", in)
out := new(Response)
err := c.c.Call(ctx, req, out, opts...)
}

```

```

    if err != nil {
        return nil, err
    }
    return out, nil
}
// Server API for TemplateRegistryService service
type TemplateRegistryServiceHandler interface {
    Stream(context.Context, *StreamingRequest, TemplateRegistryService_StreamStream)
error
    PingPong(context.Context, TemplateRegistryService_PingPongStream) error
    NewSensorTemplate(context.Context, *Request, *Response) error
}
func RegisterTemplateRegistryServiceHandler(s server.Server, hdlr
TemplateRegistryServiceHandler, opts ...server.HandlerOption) error {
    type templateRegistryService interface {
        Stream(ctx context.Context, stream server.Stream) error
        PingPong(ctx context.Context, stream server.Stream) error
        NewSensorTemplate(ctx context.Context, in *Request, out *Response) error
    }
    type TemplateRegistryService struct {
        templateRegistryService
    }
    h := &templateRegistryServiceHandler{hdlr}
    return s.Handle(s.NewHandler(&TemplateRegistryService{h}, opts...))
}
func (h *templateRegistryServiceHandler) NewSensorTemplate(ctx context.Context, in
*Request, out *Response) error {
    return h.TemplateRegistryServiceHandler.NewSensorTemplate(ctx, in, out)
}

```

Регистрация сервиса в системе.

```

import (
    "context"
    "encoding/xml"
    "fmt"
    "github.com/micro/go-micro/v2"
    "github.com/mufasadev/sensorset-ogc-models/model/sml"
    pb "github.com/mufasadev/sensorset-sensor-template-registry/proto"
    "github.com/mufasadev/sensorset-sensor-template-registry/src/db"
    sh "github.com/mufasadev/sensorset-sensor-template-registry/src/handler"
    "io/ioutil"
    "log"
    "os"
    "strconv"
)
const SERVICE_NAME = "sensorset.service.sensor-template-registry"
const SERVICE_API_CLIENT_NAME = "sensorset.service.sensor-internal-api"
func main() {
    srv := micro.NewService(
        micro.Name(SERVICE_NAME),

```

```

)
srv.Init()
xmlFile, err := os.Open("3.xml")
if err != nil {
    fmt.Println(err)
}
defer xmlFile.Close()
byteValue, _ := ioutil.ReadAll(xmlFile)
var component sml.PhysicalSystem
xml.Unmarshal(byteValue, &component)
ctx := context.Background()
ctx = context.WithValue(ctx, db.MgHostKey, "127.0.0.1")
ctx = context.WithValue(ctx, db.MgUsernameKey, "****")
ctx = context.WithValue(ctx, db.MgPasswordKey, "****")
ctx = context.WithValue(ctx, db.MgPortKey, strconv.Itoa(27017))
client, err := db.InitDB(ctx, 0)
if err != nil {
    log.Panic(err)
}
defer client.Disconnect(ctx)
collection :=
client.Database(db.MgDatabase).Collection(db.MgCollectionPhysicalComponent)
    handler := &sh.Handler{Repository: &db.TemplateRegistryRepository{Collection:
collection}}
pb.RegisterTemplateRegistryServiceHandler(srv.Server(), handler)
// Run the server
if err := srv.Run(); err != nil {
    log.Panic(err)
}
}

```

Жирным шрифтом выделены строки, отвечающие за регистрацию обработчика для удаленного вызова процедуры. Как видно, все что требуется для регистрации сервиса, это несколько строк кода. Для передачи данных – описание формата сообщений. Данные инструменты упрощают разработку приложений с использованием микросервисной архитектуры, gRPC и protocol buffers.

Заключение. Несмотря на плюсы приведенных выше технологий, всегда следует понимать, что не существует единого решения для всех типов систем. В случае, если основная работа системы заключается в обслуживании веб интерфейсов, gRPC будет не лучшим выбором. Если же требуется высокая производительность во внутренних компонентах системы, то следует присмотреться к описанным в статье инструментам. Это позволит избежать множества проблем с миграцией данных и синхронизацией протоколов взаимодействия отдельных компонентов систем. Также следует учитывать компетенцию команды, которая работает на проекте, возможно, если команда хорошо знакома с одной технологией, то переход на другую может вызвать ряд проблем и временных затрат. Protocol Buffers и gRPC не имеют такого широкого распространения, как JSON и REST, поиск новых специалистов также может быть затруднен. Подводя итоги, можно сказать, что основной причиной использования данных технологий может послужить необходимость повышения производительности программной системы в контексте обмена данными между сервисами.

Литература

1. <https://developers.google.com/protocol-buffers> (дата обращения: 10.10.2022)
2. <https://developers.google.com/protocol-buffers/docs/encoding> (дата обращения: 10.10.2022)
3. https://en.wikipedia.org/wiki/Protocol_Buffers (дата обращения: 12.10.2022)
4. https://en.wikipedia.org/wiki/Remote_procedure_call (дата обращения: 13.10.2022)
5. <https://www.techtarget.com/searcharchitecture/definition/Remote-Procedure-Call-RPC> (дата обращения: 13.10.2022)
6. <https://en.wikipedia.org/wiki/GRPC> (дата обращения: 16.10.2022)
7. Гайдамако, В.В., Авельцов, Д.О. Разработка модуля виртуализации сенсорных устройств для распределенных информационно-измерительных систем / В.В.Гайдамако, Д.О.Авельцов // Проблемы автоматки и управления. – 2020 – №1(38). – с. 89-103.
8. <https://go.dev/> (дата обращения: 18.10.2022)
9. <https://github.com/go-micro/go-micro> (дата обращения: 18.10.2022)

УДК 004

В.Гайдамако, Е.Д. Киселев*Институт машиноведения и автоматизации Национальной академии наук,
Бишкек***РАЗРАБОТКА СИСТЕМЫ МОНИТОРИНГА АППАРАТНО-ПРОГРАММНОЙ ИНФРАСТРУКТУРЫ ОРГАНИЗАЦИИ**

В статье представлен опыт разработки системы мониторинга аппаратно-программной инфраструктуры организации, а также вспомогательных компонент для ее работы, в число которых входит система оповещений, система запуска интервальных задач, система управления плагинами. Система реализована в клиент-серверной архитектуре.

Ключевые слова: мониторинг, ИТ-инфраструктура, динамическая загрузка модулей, плагины

Введение

Мониторинг информационной инфраструктуры (ИТ-инфраструктуры), то есть аппаратного и программного обеспечения – серверов, сетей, сервисов, необходим в любой компании, будь это небольшой офис или огромный дата-центр. Ответственные лица должны быть оповещены о сбоях и проблемах в инфраструктуре раньше или хотя бы одновременно с пользователями. Сбои и неисправности желательно прогнозировать и предотвращать, информация о них должна сохраняться и анализироваться, персонал должен вовремя оповещаться даже о тех сбоях, которые были устранены автоматически. Основной задачей системы мониторинга является представление актуальной информации для анализа состояния ИТ-инфраструктуры, быстрого обнаружения возникшей неисправности, ее оперативного устранения и восстановления работоспособности элемента инфраструктуры [1, 2]. Постоянный мониторинг помогает избежать простоев в работе ИТ-отдела, поддерживать все ИТ-сервисы в рабочем состоянии и сохранять необходимый уровень их качества. Рассматриваемая система разрабатывалась для мониторинга состояния серверной части используемых приложений банка, но может использоваться и в других организациях.

Задачи поддержки ИТ-инфраструктуры

Рассмотрим более детально деятельность отдела поддержки инфраструктуры банка. По сути, круг задач ничем не отличается от того, что делают в организациях другого типа (интернет-провайдеры, операторы сотовой связи и др.). Они также используют веб-сервера, базы данных, публичные сервисы, балансировщики нагрузки, файловые сервера, прокси-сервера, VPN-адаптеры и многое другое. Но у банка существует ряд особенностей, вытекающих из специфики предоставляемых услуг. Так как банк располагает достаточно большим объемом клиентской информации, особое внимание уделяется базе данных. Ее пропускной способности, скорости получения информации и потреблении оперативной памяти. Также большое внимание уделяется целостности информации и ее конфиденциальности. Последний пункт особенно пристально берется под контроль сразу несколькими отделами банка. Ведь несанкционированный доступ к банковской информации – это очень серьезный удар по репутации банка. Поэтому за безопасность отвечает несколько отделов, каждый из которых берет под свой контроль определенный периметр (безопасность сети, безопасность базы данных, антивирусная защита, антиспам, политики паролей, охрана физического периметра, целостность цифровых накопителей информации и др.). В связи с таким пристальным вниманием к информационной безопасности и многими другими ограничениями к квалификации членов команды поддержки инфраструктуры предъявляются более строгие требования. Требования по безопасности должны соблюдаться всеми сотрудниками банка [2]. Команда поддержки инфраструктуры должна тесно сотрудничать со смежными командами и обязана в любой момент ответить на вопрос о состоянии инфраструктуры. Казалось бы, банк может

закрывать базовые потребности в мониторинге, установив какую-либо из популярных систем мониторинга. На рынке присутствуют множество как свободно распространяемых, так и проприетарных (защищенных авторским правом) систем в числе которых: ManageEngine [3], Zabbix [4], Nagios [5], Cacti [6], Munin [7]. Но проблема всех этих систем в том, что они или не обладают достаточной гибкостью, чтобы удовлетворить все потребности команды поддержки. Также банк использует автоматизированную банковскую систему (АБС), которая предъявляет свои требования.

Цель создания программной системы состоит в том, чтобы упростить работу ИТ-специалистов, занимающихся мониторингом серверов, предоставить им инструмент для централизованной проверки состояния серверной части, а также для предотвращения сбоев путем заблаговременного уведомления ответственных специалистов о нарушении допустимых значений параметров системы. Разрабатываемое приложение «Мониторинговая система» должно обеспечить автоматизацию проверок серверных машин на состояние различных показателей работоспособности, обеспечить единый канал получения информации о состоянии системы и позволить производить гибкую настройку для различных конфигураций. Программа в реальном времени с определенной периодичностью осуществляет проверки параметров системы и сравнивает их с допустимыми значениями, выдавая оповещение при нарушении допустимых границ с помощью механизма уведомлений. Приложение «Мониторинговая система» разрабатывалось в рамках проекта автоматизации процесса мониторинга серверной части программного обеспечения компании.

Архитектура системы

Приложение «Мониторинговая система» строится на основе архитектуры клиент-сервер. Взаимодействие клиента и сервера осуществляется с помощью стандартных либо же собственных протоколов, а данные передаются через сеть передачи данных. Сервер хранит, использует, модифицирует текущую конфигурацию для выполнения мониторинга. В соответствии с конфигурацией сервер запускает проверки системы, рассылает оповещения при сбоях, сохраняет результаты проверок для последующего вывода их в удобном для просмотра и анализа виде – таблицы, графики, отчеты. Клиент не хранит никакой информации, кроме той, что находится в оперативной памяти, то есть предназначенной для отображения. Основное предназначение клиента – графический интерфейс пользователя для конфигурирования сервера и просмотра результатов мониторинга. Система представляет из себя множество независимых сервисов, которые управляются сервисным менеджером. На каждом сервере для запуска нужных сервисов мониторинга устанавливается и настраивается сервисный менеджер. При изменении конфигурации серверов достаточно изменить настройки запуска сервисов на целевой конфигурации. У каждого сервисного менеджера имеется файл с настройками, там указано, какие именно мониторинговые сервисы необходимо запускать. После редактирования файла требуется только перезапуск сервисного менеджера. При сбое в работе сервера или по достижении критических значений параметров через различные мессенджеры системным администраторам будут отправлены сообщения с краткой информацией о проблеме. Сообщения, полученные от сервисных менеджеров каждой системы серверов, сохраняются в единой базе данных. Серверной частью является REST API [8], развернутая на web-сервере IIS [9] и написанная на языке программирования C#. В качестве хранилища данных используется СУБД MS SQL Server [10]. Клиентская часть: MVC-приложение [11], написанное в IDE Visual Studio с использованием таких технологий, как .NET Framework и ASP.NET MVC [11].

Функциональность продукта

«Мониторинговая система» – программный продукт, состоящий из нескольких компонентов, задача которых обеспечить непрерывную проверку аппаратных и программных ресурсов ИТ-организации. Основной подсистемой мониторинговой системы является набор компонентов по сбору и сохранению метрик с проверяемых устройств.

Мониторинговая система должна удовлетворять следующим требованиям:

- сбор метрик с конечных устройств;
- сохранение результатов проверок;
- представление информации о выполненных проверках;
- оповещение пользователей.

Показатели качества системы:

- масштабируемость. Добавление новых серверов не должно вызывать трудностей у системных администраторов;
- отказоустойчивость. Сбой в работе одного из сервисов не должен повлечь за собой сбои в работе других сервисов, другими словами, система должна быть ортогональной;
- непрерывная доставка. Новые модули мониторинга, написанные другими программистами, должны внедряться в систему без перезапуска работающих сервисов. Их доставка должна быть простой и прозрачной для системных администраторов;

Обеспечение развития и расширяемости системы:

- возможность написания собственных плагинов. Система должна быть расширяемой извне. То есть нужно создать механизмы для сторонних разработчиков по реализации того функционала, который необходим их заказчикам;
- расширяемое количество метрик. Система должна иметь гибкий механизм настроек, позволяющий включать и выключать сбор каких-либо метрик. Таким образом, любая организация сможет настроить систему мониторинга под свои текущие нужды;
- уведомления по различным каналам связи. Данный функционал позволит предприятию гибко настроить модуль отправки уведомлений по тем каналам связи, которые привычны для сотрудников. Таким образом для команды не придется устанавливать какое-то новое ПО, изучать его, а самое главное, что отправка уведомлений легко вольется в основной рабочий процесс команд;
- невысокие требования к ресурсам. Система мониторинга не должна забирать на себя лишние вычислительные ресурсы того сервера, на котором она установлена. Ведь система мониторинга – это не основное ПО предприятия, и дополнительные затраты на вычислительное оборудование могут поставить под сомнение целесообразность реализации самой системы.

Пользователи системы

У системы есть один основной пользователь, это системный администратор – разворачивает необходимые сервисы мониторинга на целевых серверах, просматривает сообщения, производит настройку сервисов.

Акторами системы являются:

1. Системный администратор.
2. Разработчик.
3. Менеджер проекта.

Системному администратору предоставляется возможность:

- Настраивать интервальные проверки;
- Запускать (останавливать) интервальные проверки;
- Просматривать исторические данные о прошлых проверках;
- Настраивать оповещения для пользователей.

Разработчик в данной системе имеет такие же возможности, как системный администратор, а также возможность разрабатывать и подключать новые плагины.

Менеджер проекта осуществляет управление процессом разработки с целью получения готового продукта в отведенные сроки; контроль временного графика проекта; подготовку отчетности о его состоянии; разработку, поддержку и исполнение сводного плана и календарного графика проекта; организацию управления рисками.

Диаграмма вариантов использования [12] представлена на рис. 1.

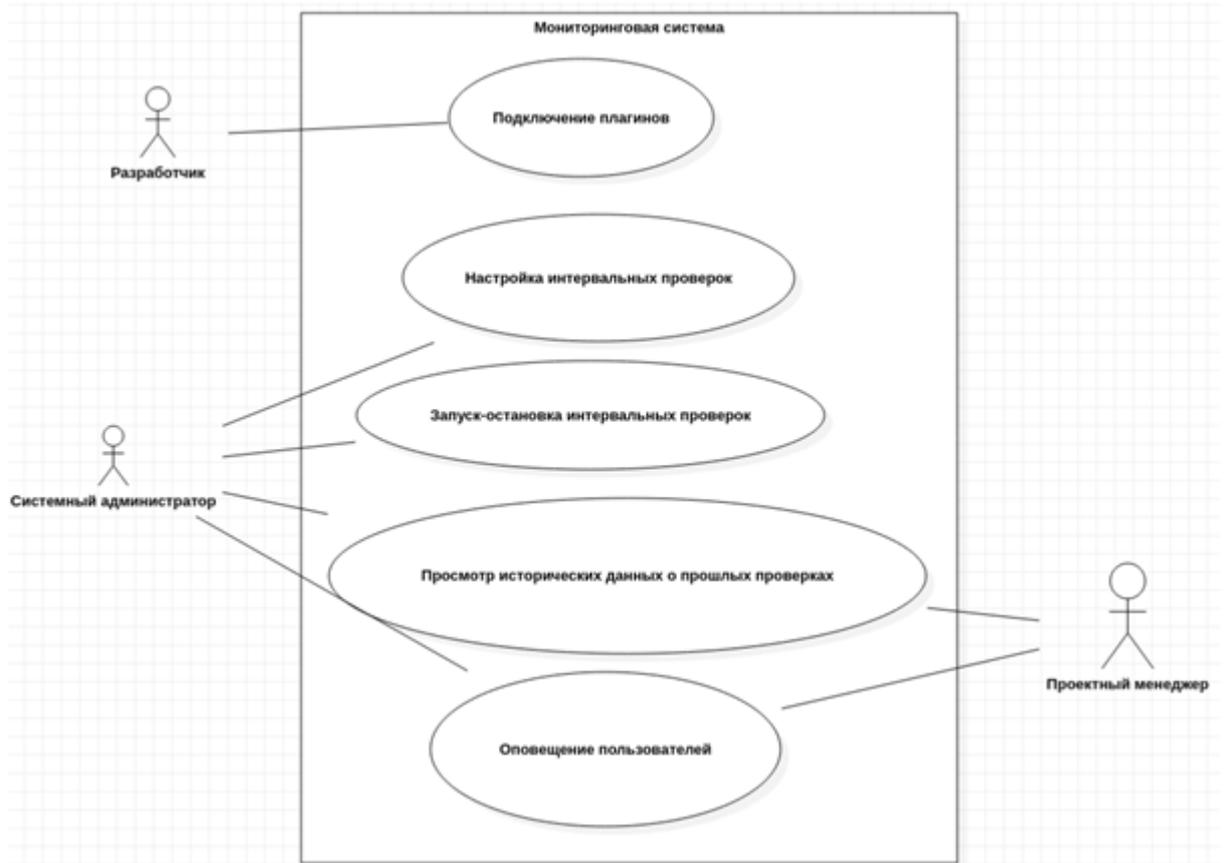


Рисунок 1 – Диаграмма вариантов использования

Разработка диаграмм деятельности

На этапе развития системы разрабатываются сценарии для вариантов использования, диаграммы последовательности (формализующие текстовые представления сценариев), проектируются диаграммы классов и планируется содержание следующего этапа разработки [12, 13]. На рисунке 2 представлен сценарий “Настройка интервальных проверок”. Здесь осуществляется настройка интервальной проверки через интерфейс пользователя (период запуска, проверяемый сервер, какие виды оповещений будут задействованы и т.д.) Далее заполняются последовательно основные параметры, без которых интервальная проверка не запустится, и вспомогательные (опциональные) параметры. Последним шагом происходит валидация введенных параметров. При успешной валидации интервальная проверка успешно запускается, иначе пользователю нужно проверить введенные настройки и повторить попытку.

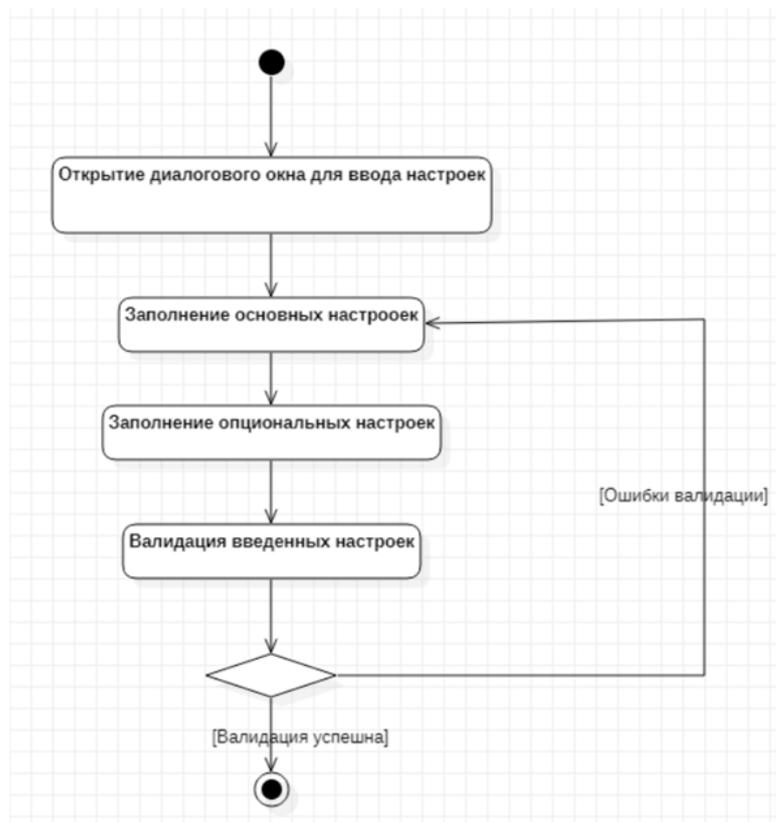


Рисунок 2 – Сценарий “Настройка интервальных проверок” в виде диаграммы деятельности

На рис. 3 представлен сценарий тестового запуска интервальной проверки. После успешной валидации параметры отправляются в виде запроса на сервер. Сервер в свою очередь запускает тестовую проверку и отправляет результаты тестирования обратно пользователю.

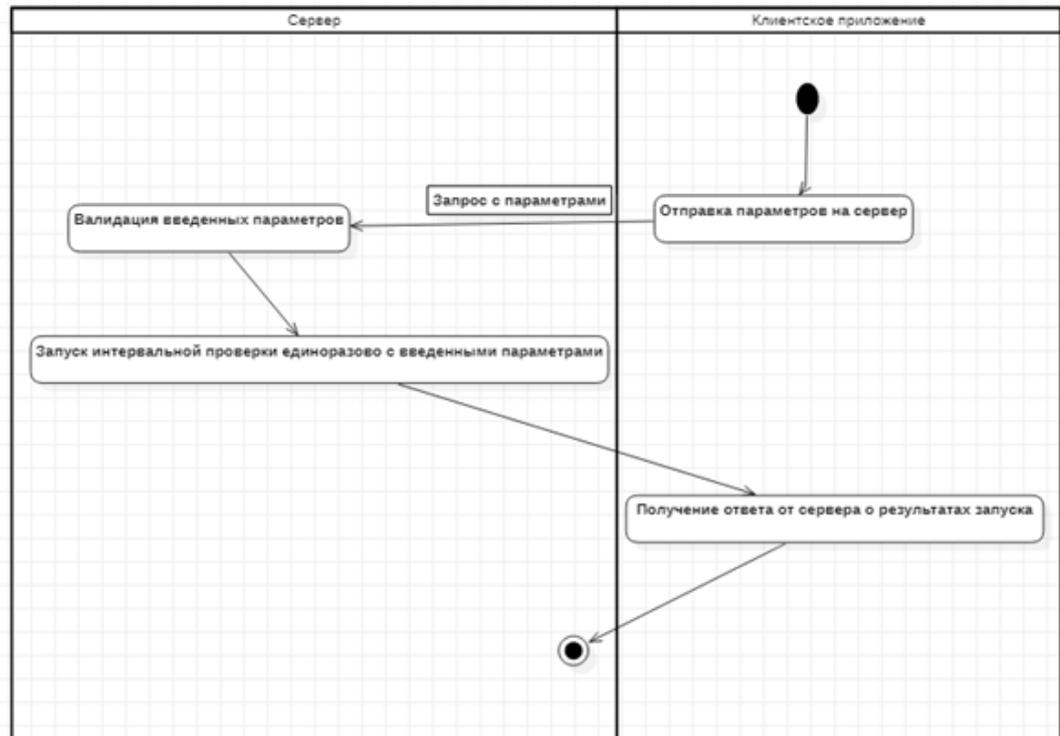


Рисунок 3 – Диаграмма деятельности для сценария “Тестовый запуск интервальной проверки”

Разработка диаграммы классов [12]

Для проектирования сценариев программируются следующие операции класса **Plugin** (рис 4.):

GetSettings() – получение всех настроек из плагина;

GetSettingByName() – получение определенной настройки по ее имени;

GetInterval() – получение значения из настройки с типом данных “временной интервал”;

GetString() – получение строкового представления настройки плагина;

GetInt() – получение числового представления настройки плагина.

В ходе реализации перечисленных операций выясняется необходимость и программируется содержание вспомогательных операций.

Для класса PluginSetting:

DisplayName – имя настройки, которое будет выведено на интерфейс. Представлены все узлы, необходимые для работы системы. Сервер непрерывной интеграции служит «мостиком» между средой разработки и производственной средой. Главной его задачей является сборка кода и развертывание артефактов на соответствующих узлах инфраструктуры. База данных, код приложения и код плагинов специально вынесены на отдельные узлы в целях увеличения производительности и отказоустойчивости. Рабочих серверов может быть большое множество. Особенностью данной системы, исходя из диаграммы, является ее центричность (так называемая PUSH-модель). Система ставится на один сервер (master), который связан сетью с остальными узлами в инфраструктуре. Никакого дополнительного программного обеспечения ставить на рабочие сервера не нужно. И, как видно из диаграммы развертывания, наличие системы никак не влияет на существующую инфраструктуру предприятия, таким образом можно сделать вывод, что представляемая мониторинговая система может быть внедрена как в новую инфраструктуру, так и в уже запущенную и функционирующую инфраструктуру, а модульность позволит осуществить внедрение последовательно (инкрементно) без нарушения работы программных узлов предприятия.

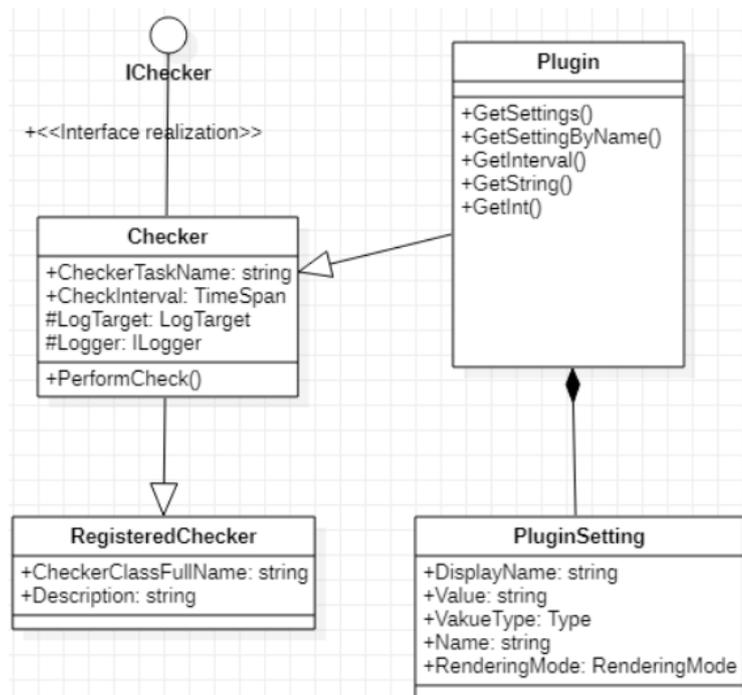


Рисунок 4 – Диаграмма классов для сценариев “Настройка интервальных проверок” и “Запуск-остановка интервальных проверок”

Особенности конструирования продукта

Создание REST API позволило сильно ускорить разработку взаимодействия клиента и сервера, так как шаблонное создание однотипных команд облегчало тестирование, вся кодовая база выглядела наглядно и стройно, процессы код-ревью (обзоры кода) не занимали много времени у остальных коллег.

Было принято решение реализовывать систему динамического рендеринга [14]. Также были задействованы атрибуты и рефлексия, но со своими особенностями, присущими данному проекту. Динамический рендеринг позволил двигаться разработке плагинов для мониторинговой системы с удвоенной скоростью, ведь теперь не нужно было ждать, когда будет готов нужный компонент интерфейса, а использовать текущий, на нем тестировать работу плагина, а в процессе запуска можно было подменить уже измененным плагином с новыми разработками.

Систему, основанную на плагинах, выбрали не случайно. Такая система могла быть расширяема “на ходу”, т.е. не нужно было ждать общего обновления системы, чтобы начать пользоваться новой функциональностью. К тому же разработка плагинов могла вестись отдельной командой независимо от команды разработки ядра мониторинговой системы. Таким образом, запуск новых механизмов проверок осуществлялся уже на работающей системе, без вмешательства в работу остальной части приложения.

Также серьезным техническим шагом была миграция системы с .NET Framework на .NET Core. Необходимость данной миграции возникла из-за новых требований заказчика, суть которых заключалась в том, чтобы мониторинговую систему можно было запустить на серверах под управлением операционной системы Linux (.NET Framework запускался только под Windows).

Особенностью разработки данной системы являлась ее мозаичность. Имеется в виду, что систему с определенного этапа можно было реализовывать с любого компонента, параллельно, а не последовательно. Таким образом, система на начальном этапе разработки росла вглубь – появлялись механизмы запуска задач, получения настроек, взаимодействие с хранилищем данных, клиент-серверное взаимодействие, инструменты динамического рендеринга. А затем система росла вширь – разрабатывались новые плагины и механизмы работы с ними. Включение новых плагинов не составляло труда, даже для слабо квалифицированных специалистов.

Тестирование

При тестировании мониторинговой системы была выбрана следующая стратегия: функционал проходил первичное тестирование локально на компьютере разработчика, затем функционал попадал на сервер заказчика и проводилось тестирование в условиях, близких к реальным. Специально для нужд тестирования было настроено окружение, которое было частью боевого, но при этом технически работало обособленно, некая “песочница” для тестирования готового функционала. В опорных точках календарного плана проводилось общее тестирование работоспособности системы. Реагирование на дефекты происходило следующим образом: если дефект не влиял на дальнейшую разработку функционала, то исправление дефекта переносилось на следующую итерацию, если же влияние дефекта приводило к замедлению или невозможности продолжать дальнейшую разработку, то все силы бросались на исправление дефекта [15].

При проектировании архитектуры будущей системы, а особенно при расширении системы плагинами, заранее подготавливались сценарии тестирования и данные для тест-кейсов.

Все тестовые сценарии и тест-кейсы выполнялись на всех окружениях (локально у разработчика и у заказчика). Бывало так, что успешно пройденные тест-кейсы на локальной машине никак не хотели выполняться на серверах заказчика. Благодаря нескольким окружениям уменьшилась вероятность пропуска дефекта в производственную среду.

Также были настроены процессы CI/CD (Continuous Integration/Continuous Delivery) [16], которые позволили держать кодовую базу в стабильном состоянии. Суть процесса заключается в том, что при завершении задачи по разработке написанный код отправлялся в основную ветку репозитория контроля версий. Сервер CI/CD подтягивал эти изменения и запускал конвейер (набор шагов по сборке и тестированию проекта). Такой конвейер запускается достаточно часто, что позволяет оперативно решать проблемы с проектом (буквально при их возникновении). Также сервер CI/CD готовил архив с готовым объектным кодом, которым можно обновить сервер заказчика, таким способом новый функционал попадал в боевое окружение.

На поздних этапах жизненного цикла программного продукта для совместного тестирования были привлечены сотрудники заказчика. К этому моменту уже отрабатывались мелкие правки, так как процесс внутри приложения и основной функционал вступили в фазу устойчивой и стабильной работы.

Заключение

В процессе разработки мониторинговой системы были пройдены все этапы жизненного цикла программного продукта, начиная от инициации проекта до сопровождения. На данный момент мониторинговая система развивается как программный продукт и внедряется новыми клиентами. При проектировании системы были собраны и проанализированы наиболее приоритетные требования, практически полностью соответствовавшие ожиданиям заказчика относительно будущего программного продукта. Декомпозиция требований и системный анализ позволили создать программную архитектуру, на базе которой предстояло разработать будущее приложение. Тесное взаимодействие с заказчиком как один из факторов успеха привел к тому, что завершающие этапы проекта проходили эффективно. Тестирование и внедрение программного продукта показало слаженную работу команд по обе стороны. Разработанный программный продукт выполняет следующие функции:

- Осуществляет проверки через заданные пользователем интервалы времени;
- Результаты проверок сохраняются в файлах журналов (логах);
- Создание и сохранение настроек;
- Отправка уведомлений по различным каналам связи;
- Запуск проверок по желанию пользователей;
- Наличие пользовательского интерфейса для удобного взаимодействия с системой;
- Возможность подключения новых способов проверок «на ходу» без перезапуска и уж тем более без перекомпиляции проекта.

В дальнейшем функционал приложения будет расширяться и совершенствоваться в следующих направлениях:

- Увеличение производительности операций проверок;
- Интерфейс для просмотра журналов проверок;
- Реализация новых типов мониторинга;
- Добавление авторизации и прав доступа для пользователей системы.

Программный продукт успешно перешел на сопровождение в оговоренные сроки. При сопровождении развитие продукта не остановилось. Появились новые требования к продукту, появились новые клиенты, которые также хотят иметь возможность получать информацию о состоянии IT-инфраструктуры.

Литература

1. Ворожцов А.С., Тутова Н.В., Тутов А.В. Оценка производительности облачных центров обработки данных. // Т-Comm - Телекоммуникации и транспорт, 2014. URL: <https://cyberleninka.ru/article/v/otsenka-proizvoditelnosti-oblachnyh-tsentrov-obrabotki-dannyh> (дата обращения 25.10.2022)

2. Корякин, С. В. Современные тенденции развития систем информационной безопасности / С. В. Корякин // Проблемы автоматизации и управления. – 2017. – № 2(33). – С. 82-91. – EDN ZXPAXD
3. ManageEngine Applications Manager // URL: https://www.manageengine.com/ru/applications_manager/(дата обращения 25.10.2022)
4. Zabbix // URL: <https://www.zabbix.com/ru/>(дата обращения 25.10.2022)
5. Nagios. The Industry Standard In IT Infrastructure // URL: <https://www.nagios.org/> (дата обращения 25.10.2022)
6. Cacti // URL: <https://www.cacti.net/> (дата обращения 25.10.2022)
7. Munin // URL: <https://munin-monitoring.org/> (дата обращения 25.10.2022)
8. Что такое RESTful API? // URL: <https://aws.amazon.com/ru/what-is/restful-api/> (дата обращения 25.10.2022)
9. Home : The Official Microsoft IIS Site // URL: <https://www.iis.net/> (дата обращения 25.10.2022)
10. Microsoft SQL Server // URL: <https://www.microsoft.com/en-us/sql-server/> (дата обращения 25.10.2022)
11. Руководство по ASP.NET Core MVC // URL: <https://metanit.com/sharp/aspnetmvc/>
12. Л. А. Мацяшек. Анализ и проектирование информационных систем с помощью UML 2.0. – М:Вильямс, 2008.
13. В. И. Грекул, Н. Л. Коровкина, Ю. В. Куприянов. Проектное управление в сфере информационных технологий. 2015. 337 с.
14. Что такое динамический рендеринг и как он влияет на SEO? // URL: <https://webprosite.net/what-is-dynamic-rendering-how-does-it-impact-seo/>
15. В.П. Котляров, Т.В. Коликова. Основы тестирования программного обеспечения. // СПб, Бином. Лаборатория знаний, 2006.
16. Вольф Эберхард. Continuous delivery. Практика непрерывных апдейтов – СПб:Питер, 2017.

З.Т. Маликова, zirek.malicova@mail.ru,

Г.Т.Мамажакынова, gulzara.mamazhakypova@mail.ru

Ошский технологический университет имени акад.М.М.Адышева

ТЕХНОЛОГИЯ ПОСТРОЕНИЯ ГЕОИНФОРМАЦИОННОЙ МОДЕЛИ УДС ГОРОДА ОШ НА ОСНОВЕ ДАННЫХ ИЗ ОТКРЫТЫХ ИСТОЧНИКОВ

В данной статье описан метод получения геопространственных данных из картографического сервиса OpenStreetMap, разработанный на основе технологии краудсорсинга. Данный сервис был использован с учетом того, что можно бесплатно экспортировать большие объемы достоверной картографической информации. Этот сервис удобен тем, что поддерживает систему координат WGS 84 (EPSG:3857), являющуюся наиболее популярной географической системой координат, и имеет векторный формат данных, благодаря которым можно работать с векторными типами данных. Была также рассмотрена технология разработки геоинформационной модели улично-дорожной сети на примере города Ош. Геоинформационная модель в свою очередь состоит из связанного графа и его основных характеристик. Для получения окончательных результатов исследования была использована платформа ArcGIS и ее дополнительный модуль NetworkAnalyst. С помощью инструментов модуля NetworkAnalyst был разработан набор сетевых данных, где основу составили узлы и вершины со всеми дугами поворотов. Результат исследования послужит инструментом для дальнейших исследований в области транспортного планирования и моделирования.

Ключевые слова: геоинформационная модель; геопространственные данные; геоинформационные системы; картографический сервис; улично-дорожная сеть; граф; узел; вершина; краудсорсинг; OpenStreetMap; ArcGIS; Network Analyst.

Введение

В последние годы в мире из-за быстрого роста количества автомобильного парка в городах возникла проблема оптимального планирования транспортной инфраструктуры. Транспортная инфраструктура – одна из важнейших систем, обеспечивающих комфортную доступность территорий города, безопасность и надежность организации движения, оптимизацию движения транспортных, пешеходных и пассажирских потоков. Эта проблема на данный момент актуальна как для крупных городов, так и для небольших. Для решения подобных проблем применяется транспортное моделирование с построением математических транспортных моделей, основанных на геопространственных данных.

Моделирование транспортных систем так или иначе не обходится без получения геопространственных данных об улично-дорожной сети. Такие данные важны для оценки характеристик транспортной системы, таких как протяженность, плотность, площадь улично-дорожной сети, ее связность и пропускная способность. Получение этих данных является одной из основных задач подготовительного этапа математического моделирования транспортных систем [1].

Набор картографических данных об улично-дорожной сети города с заданными характеристиками называется геопространственной моделью улично-дорожной сети. Такая модель представляет собой связный граф, каждой вершине которого сопоставлена пара географических координат, а некоторым вершинам и ребрам поставлены в соответствие числовые и качественные характеристики [2].

Баранов Д.А. в своих исследованиях привел другое определение: “... граф представляет собой набор векторных геопространственных данных о положении в пространстве элементов улично-дорожной сети, каждому из которых задаются необходимые параметры. Граф состоит из узлов и ребер. Узлы, как правило, находятся в местах пересечения отрезков сети (связей) или их концов. Ребра графа условно соответствуют трассам линейных сооружений, составляющих транспортную сеть, или маршрутам движения транспорта по ним”. [1]

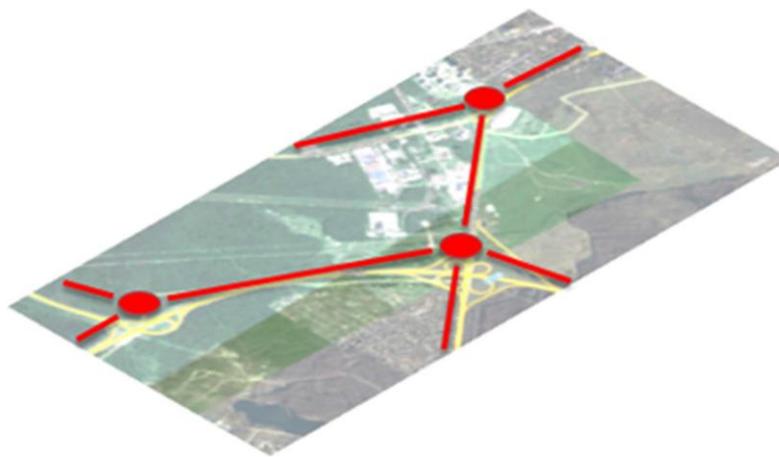


Рисунок 1 – Принцип построения дорожного графа [1]

Наличие и доступность геопространственных данных о существующей улично-дорожной сети зависит от уровня развития инфраструктуры хранения открытых данных и результатов деятельности государственных или иных организаций, компетентных в этой области. В различных странах и их регионах качество этих данных значительно различается от полного отсутствия до высочайшей детализации и точности [1].

В существующем геопортале Кыргызстана получение достоверной информации геопространственных данных об улично-дорожной сети в большинстве случаев недостаточно, так как отсутствуют данные. И это сильно затрудняет построение точных математических моделей транспортных систем и требует больших трудозатрат по исследованию улично-дорожной сети.

В настоящее время спрос на геопространственные данные, в том числе на данные дистанционного зондирования, неуклонно растет из-за того, что они являются одним из наиболее оперативных и достоверных источников информации. Все эти данные объединены и сохранены в так называемых картографических сервисах. К картографическим сервисам относятся: Google Maps, Microsoft Bing Maps and Live Local, Yahoo Maps, Яндекс-карты и OpenStreetMap. Все эти сервисы, кроме OpenStreetMap, представляют собой набор приложений, построенных на основе бесплатного картографического сервиса и технологий, предоставляемых компаниями Google, Microsoft, Yahoo и Яндекс.

Единственный картографический сервис OpenStreetMap представляет собой программное обеспечение с открытым исходным кодом, основанный на технологии краудсорсинга. В этом сервисе все геопространственные данные вносятся пользователями, владеющими некоторыми специфическими знаниями и опытом на основе данных GPS-навигаторов.

OpenStreetMap (OSM) – открытая картографическая основа, создаваемая силами энтузиастов. Распространяется в нескольких известных векторных форматах в нарезке по регионам и странам СССР. Включает готовые проекты для QGIS [3].

Данный ресурс является некоммерческим веб-картографическим проектом, который создается, обновляется и дополняется силами сообщества участников-пользователей бесплатных географических карт. В данном проекте каждый зарегистрированный пользователь может вносить любую информацию, которую уточняют картографические данные. Данные проекта распространяются на условиях свободной лицензии OpenDatabaseLicence. Для внесения данных в проект OpenStreetMap используются данные с GPS-трекеров, видеозаписи и спутниковые снимки [4].

Применение открытых источников данных о транспортной сети, основанных на краудсорсинге, имеет как положительные, так отрицательные стороны. К положительным можно отнести открытость и максимальную доступность информации, а к отрицательным

– высокая вероятность ошибок и зависимость качества данных от активности пользователей сервиса [1].

Интерфейс сервиса OSM позволяет получать данные лишь по выбираемым пользователем участкам небольшого размера. Данные OSM имеют хорошо проработанный, стандартизированный формат, позволяющий хранить огромное количество разнообразной информации о пространственных объектах. Помимо стандартных функций сервиса, имеются различные сторонние инструменты получения данных. Например, сервис Geofabrik позволяет получать большие объемы данных по регионам или странам в виде архивов [1].

Далее рассмотрены метод получения данных из сервиса OSM и технология построения геоинформационной модели улично-дорожной сети на платформе ArcGIS.

Методы исследования

Для облегчения работ по созданию модели УДС вполне оправданно использовать геоинформационные системы, получившие широкое распространение в последнее время. Такие системы предназначены для работы с электронной картой. Одним из представителей геоинформационных систем является программный продукт ArcGIS компании ESRI.

Геоинформационная платформа ArcGIS – наиболее популярная в мире информационная система для картографии и пространственного анализа на настольных компьютерах, в корпоративных и SaaS-приложениях. Все компоненты платформы неразрывно связаны между собой и поддерживают цифровую трансформацию организаций любого размера.

ArcGIS – картографическая и аналитическая платформа, предусматривающая сбор, организацию, управление и анализ, обмен и распределение географической информации. Является мировым лидером среди платформ для построения и использования геоинформационных систем. ArcGIS используется людьми по всему миру для применения географических знаний в практической сфере государственного управления, бизнеса, науки, образования и СМИ. Платформа ArcGIS позволяет публиковать географическую информацию для доступа и использования любыми пользователями. Система доступна в любой точке, где возможно использование веб-браузеров, мобильных устройств в виде смартфонов, а также настольных компьютеров [5].

Для построения модели УДС используют следующие способы:

- ручной;
- автоматизированный;
- комбинированный.

При разработке модели УДС важным аспектом становится назначение класса автомобильной дороги. Выбор класса определяет такие параметры дороги, как пропускная способность, скоростной режим движения и др. Эти параметры влияют на результаты расчетов прогнозируемой интенсивности движения. [6]

При ручном способе разработки графа все параметры УДС задаются транспортным инженером с нуля, что не исключает влияния человеческого фактора и ведет к высоким трудозатратам. При таком способе разработки графа класс автомобильной дороги задается в соответствии с классификацией, представленной в нормативных документах определенной страны.

При автоматизированном способе уже готовая сеть УДС загружается в программное обеспечение геоинформационных систем из открытых картографических источников.

Комбинированный способ сочетает характеристики двух способов разработки графа: сеть загружается из открытых картографических источников и корректируется, дополняется вручную [6].

В процессе исследования был выбран автоматизированный способ построения графа. Пример экспортирования улично-дорожной сети города из открытого

картографического источника OSM и дальнейшего импорта в программный продукт ArcGIS представлен на рис. 2 и 3.

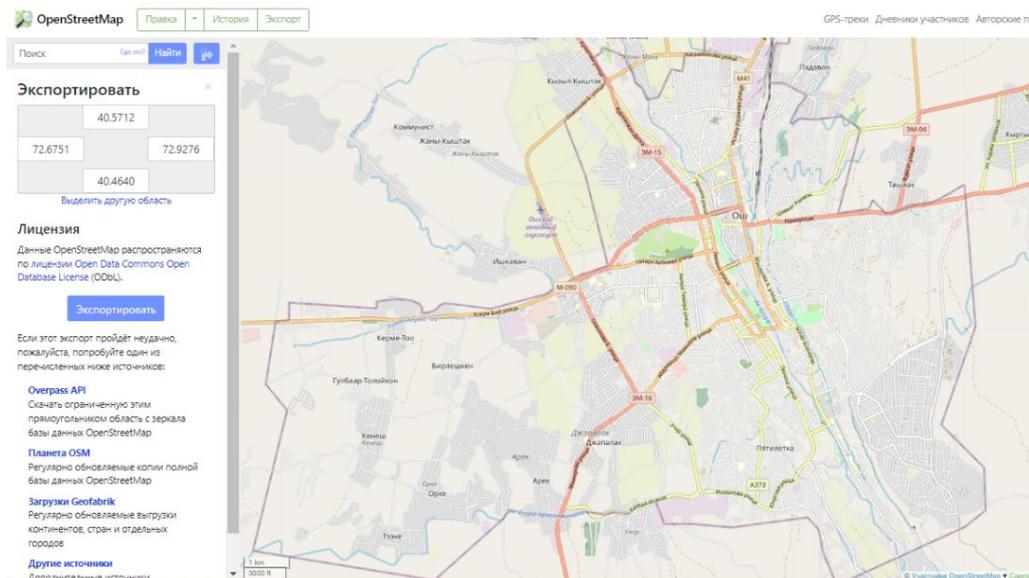


Рисунок 2 – Экспорт УДС г.Ош из OSM

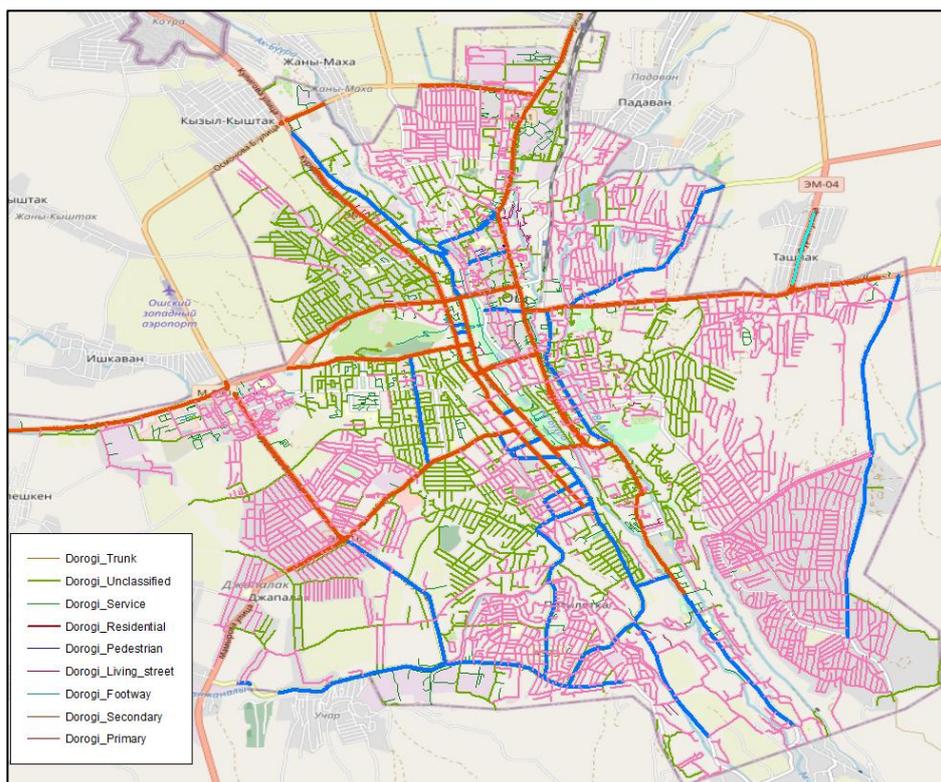


Рисунок 3 – Импорт УДС г.Ош в ArcGIS 10.7

Геопространственные данные из сервиса OSM поддерживают систему координат WGS84 (EPSG 3857) и представляют собой векторные графические данные с форматом *.shp. При получении геопространственных данных об улично-дорожной сети с сервиса OSM экспортируются также следующие характеристики: классификация улиц и дорог, информация о наличии одностороннего движения, стандартные значения разрешенной скорости и протяженность улиц и дорог (рис.4). Характеристики задаются в формате

“ключ=значение”, например, “highway=*”. Значение данного ключа – это описание класса участка улично-дорожной сети (табл. 1).

Рисунок 4 – Атрибутивная таблица, содержащая некоторые характеристики улично-дорожной сети г.Ош

Таблица 1– Общее описание классификации дорог в OSM

Название и значение ключа	Описание ключа
highway=trunk	Дорога, входящая в международную азиатскую сеть
highway=primary	Автомобильные дороги регионального значения, соединяющие крупные города и/или областные центры
highway=secondary	Автомобильные дороги областного значения, соединяющие крупные населённые пункты между собой
highway=tertiary	Более важные автомобильные дороги среди прочих автомобильных дорог местного значения, например, соединяющие районные центры с сёлами, а также несколько сёл между собой
highway=unclassified	Остальные автомобильные дороги местного значения, образующие соединительную сеть дорог, могут не иметь твёрдого покрытия (surface=unpaved)
highway=service	Служебные проезды (например, на АЗС)

Многие новые программные комплексы дают возможность моделировать изучаемые объекты. И естественно, всем ученым необходимо в первую очередь создать математическую модель для разработки алгоритма моделирования в программных системах [8].

При построении графа УДС г.Ош был использован дополнительный модуль геоинформационной платформы ArcGIS 10.7 – Network Analyst, обеспечивающий эффективный интеллектуальный анализ транспортной сети.

С помощью Network Analyst в первую очередь создается набор сетевых данных. При создании набора сетевых данных необходимо включить опцию RestrictedTurns, являющуюся источником объектов поворотов. Для определения связности сети необходимо настроить опцию “Использование полей высот”. И в самом конце необходимо настроить данные об улично-дорожной сети, и отобразится диалоговое окно с основными параметрами набора сетевых данных(рис.5).

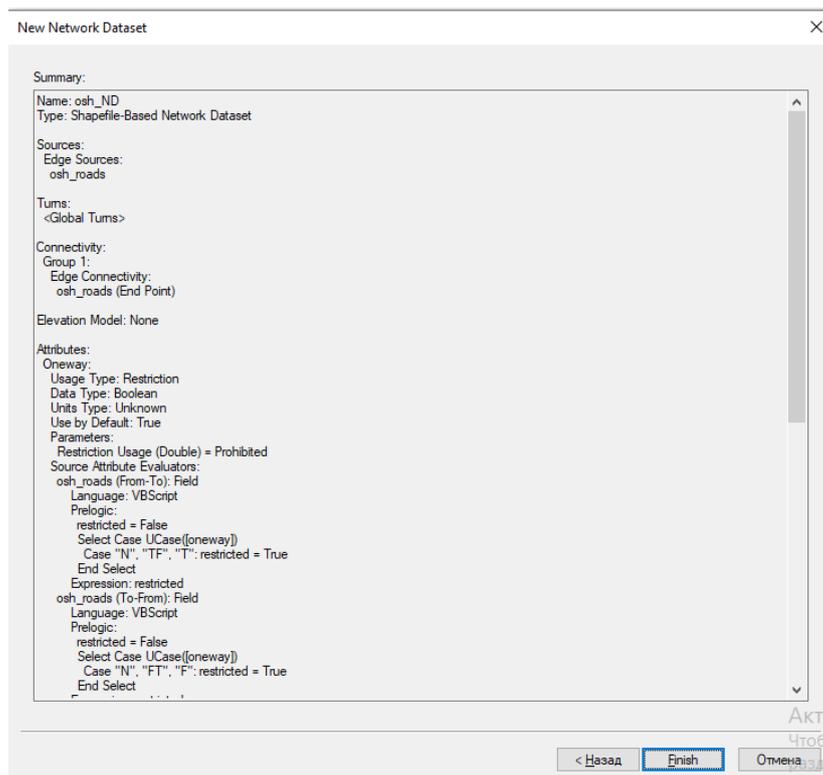


Рисунок 5 – Окно, отображающее все параметры создаваемого набора сетевых данных

В результате описанных выше действий получена полноценная, связанная и точная геоинформационная модель улично-дорожной сети. Данная модель содержит ориентированный граф УДС г.Ош, который состоит из 7140 дуг со всеми дугами поворотов и 5532 узлов (рис. 6).

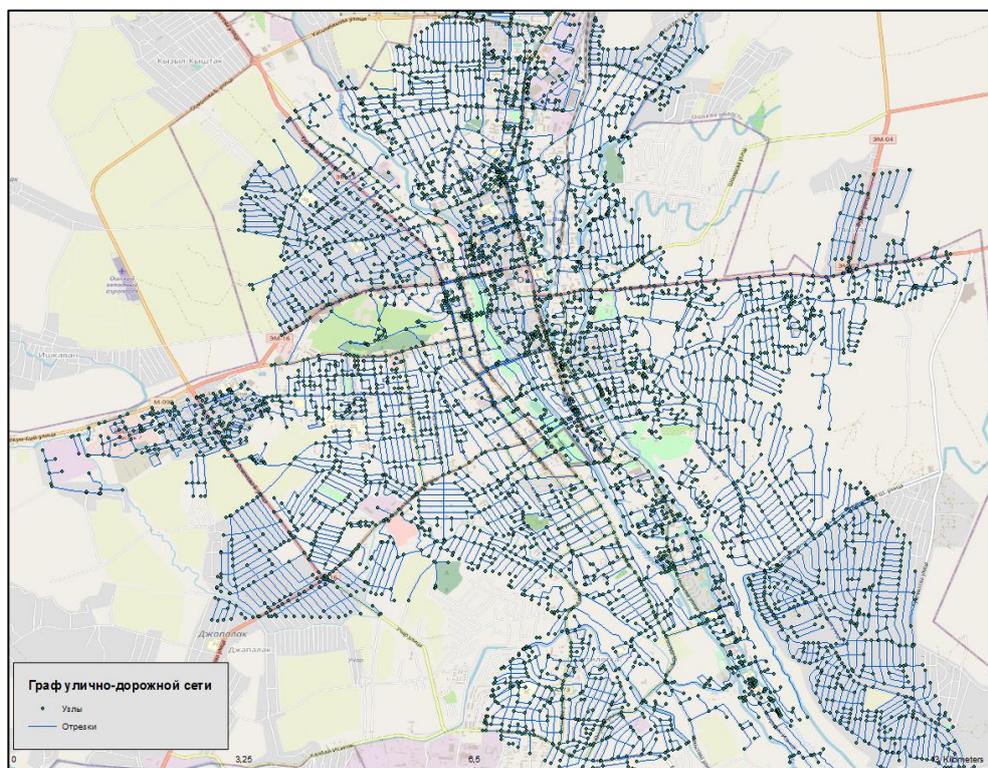


Рисунок 6 – Разработанный граф УДС г.Ош в ArcGIS 10.7

Выводы

Построенная геоинформационная модель улично-дорожной сети города Ош даст возможность быстро и легко проводить сетевой анализ с последующей публикацией в виде геоинформационной системы. Формат данных построенного графа аналогичен формату сервиса OSM, имеющего самый универсальный, открытый и экономически выгодный формат для разработки транспортных моделей. И по мнению авторов, OSM является наиболее удобным и эффективным картографическим сервисом, безвозмездно предоставляющим большой объем данных, собранных на основе технологии краудсорсинга.

Существуют множество программных обеспечений по разработке геоинформационных моделей УДС, но самой многофункциональной, понятной картографической системой для визуального анализа и решения различных транспортных задач с помощью геоинформационного анализа считается платформа ArcGIS. Дополнительный модуль Network Analyst платформы ArcGIS – это набор инструментов для решения различных логистических задач на базе модели реальной дорожной сети. С помощью этих инструментов есть возможность построения оптимизации маршрутов, нахождения ближайших объектов, построения маршрутов парка транспортных средств, оптимального распределения ресурсов, расчета зон обслуживания и вычисления матрицы расстояний.

Литература

1. Баранов Д.А. Получение данных об улично-дорожной сети из открытых источников./ Баранов Д.А. // Транспортное планирование и моделирование. – 2019. – С.10–19.
2. Мехедов И.С. Модель улично-дорожной сети на основе скелета. / Мехедов И.С., Козлов А.В. // URL: <https://studylib.ru/doc/2524184/model.-ulichno-dorozhnoj-seti-na-osnove-skeleta> (дата обращения 12.06.22)
3. Степаненко А.Э. Источники открытых геопространственных данных и способы их использования./ Колесников А.А., Степаненко А.Э.//Интерэкспо Гео-Сибирь, 2017. №2.URL: <https://cyberleninka.ru/article/n/istochniki-otkrytyh-geoprostranstvennyh-dannyh-i-sposoby-ih-ispolzovaniya/viewer>(дата обращения: 02.06.2022)
4. Википедия – электронная свободная энциклопедия. <https://ru.wikipedia.org/wiki>(дата обращения 20.04.22)
5. Официальный сайт компании ESRI. <https://www.esri-cis.ru/ru-ru/arcgis/about-arcgis/overview> (дата обращения 24.04.22)
6. Мячин В.Н., Боровикова К.С., Кривцов Д.П. Особенности классификации улично-дорожной сети при построении графа дорог в транспортной модели // Автоматика на транспорте. – 2021. – №2.– С.268–683. URL: <https://cyberleninka.ru/article/n/osobennosti-klassifikatsii-ulichno-dorozhnoj-seti-pri-postroenii-grafa-dorog-v-transportnoy-modeli> (дата обращения: 21.06.2022).
7. Официальный сайт картографического сервиса OpenStreetMap. URL: https://wiki.openstreetmap.org/wiki/RU:Highway_classification (дата обращения 10.06.22)
8. Askarbekov, R. N. Mathematical model for describing elastic deformation of rubber-metal elements / R. N. Askarbekov, T. B. Duishenaliev // Проблемы автоматизации и управления. – 2021. – No 2(41). – P. 32-39. – EDN WNGQHO.

УДК 681.3

*Сороковая А.В.**Институт машиноведения и автоматизации НАН КР, Бишкек, Кыргызстан**E-mail: nastusha24sh-g@yandex.com*

МОДЕЛЬ ПРОГРАММНОГО ИНТЕРФЕЙСА ДЛЯ РАЗРАБОТКИ ПРОГНОСТИЧЕСКИХ СИСТЕМ НА ПРИМЕРЕ РЕШЕНИЯ ЗАДАЧИ ПРОГНОЗА КЛАССА ИНДЕКСА КАЧЕСТВА ВОЗДУХА

В работе представлена модель системы, предоставляющей программный интерфейс для обучения и применения прогностических систем на примере решения задачи прогноза индекса качества воздуха. Разработанная на основе модели система позволит: управлять исходными данными, подготавливать новые данные на основе исходных для обучения и применения различных моделей прогноза, обучать и использовать различные обучаемые модели, подключать в виде плагинов другие методы обработки исходных данных и обучаемых моделей, реализующие соответствующие интерфейсы.

Ключевые слова: индекс качества воздуха, метеофакторы, программный интерфейс, обучение, прогнозирование, LSTM-классификатор, UML- диаграммы, Clean Architecture

Введение. В течение последних лет в Кыргызстане активно выполняются исследования, связанные с построением моделей прогноза загрязненности атмосферного воздуха г. Бишкек твердыми частицами PM_{2.5}. Особое внимание уделено задаче классификации индекса качества воздуха (AQI) [1]. Проведены многочисленные вычислительные эксперименты, результаты которых опубликованы в ряде работ [2, 3, 4]. Однако применение разработанных моделей для прогноза класса AQI ограничено предопределенным набором данных и инструментами разработки этих моделей. Они не являются универсальными инструментами и их использование для задач реального прогноза, тем более в он-лайн режиме, невозможно, поскольку наборы исходных данных представлены в различных форматах, среды разработки являются специфичным инструментом, требующим навыков владения ими, нет единой системы использования разных моделей, классификаторы не обладают единым интерфейсом использования.

В результате была поставлена задача: спроектировать модель программного интерфейса (ПИ) для разработки прогностических систем на примере решения задачи прогноза индекса качества воздуха AQI. Поскольку один из лучших результатов в задаче классификации AQI показала прогностическая система на основе LSTM-сети [3], именно она и положена в основу проектирования программного интерфейса. Разрабатываемая модель программного интерфейса позволит обобщить подходы к проектированию многих прогностических систем и создать единую платформу для их разработки и использования. Программная система, основанная на проектируемой модели, должна быть расширяема и поддерживаема таким образом, чтобы с минимальным знанием о системе было возможно добавлять, обучать и использовать новые модели предсказания.

1. Постановка задачи проектирования. Основные проблемы, которые привели к необходимости создания единого программного интерфейса (ПИ) для разработки прогностических систем:

представление исходных данных в различном формате, в связи с чем при получении новых данных они требуют приведения к единому определенному разработчиком формату; что приводит к большим затратам времени на процесс подготовки к обучению и использованию моделей;

необходимость менять метод обработки исходных данных или способ прогноза приводит к изменениям большей части программного кода;

использование систем прогноза людьми, не владеющими навыками разработки, невозможно, так как система не имеет интерфейса пользователя;

анализ данных о работе прогностических систем ограничивается алгоритмами, реализованными в самой программной системе; результаты исследований сохраняются в текстовые и графические файлы, что затрудняет их дальнейший анализ;

невозможность использования полученных моделей прогноза в режиме реального времени.

Проектируемый программный интерфейс позволит решить эти проблемы. Основные цели разработки:

упростить расширение прогностической системы новыми методами обработки данных и обучаемыми моделями прогноза;

ускорить процесс подготовки к обучению и использованию прогностических моделей;

хранить результаты обработки данных и обучения моделей в единой базе данных, подключившись к которой с помощью других инструментов, можно будет использовать и анализировать полную информацию о результатах обучения моделей;

предоставить возможность использования прогностических систем, полученных на основе разработанного ПИ, людьми, не являющимися программистами;

создать программно-алгоритмическую базу для прогнозов AQI в режиме реального времени.

Основные задачи, которые необходимо решить для достижения этих целей:

обобщение и разработка настраиваемых методов загрузки и обработки данных, основываясь на ранее полученных данных и их формате;

обобщение и разработка такого единого интерфейса для создания, обучения и использования прогностических систем, который будет являться точкой расширения программной системы путем создания плагинов, реализующих этот интерфейс;

разработка графического интерфейса пользователя для обучения и использования систем прогноза.

Основные этапы проектирования прогностических систем. При проектировании, обучении и использовании прогностических систем можно выделить следующие основные этапы:

сбор данных из различных источников и, как правило, в различных форматах,

обработка данных с целью приведения их к единому формату,

определение структуры прогностической модели,

формирование обучающей и тестовой выборок,

обучение модели,

формирование входных параметров для прогноза,

использование модели для предсказания.

Рассмотрим каждый из этапов подробнее для того, чтобы определить основные компоненты прогностической системы и интерфейсы их взаимодействия. В качестве примера рассматривается задача классификации AQI.

Сбор данных. Исходные данные для решения задач прогноза уровня загрязненности атмосферного воздуха г. Бишкек - следующие:

- метеорологические данные, получаемые с сайта РП5 [5], обычно в виде EXCEL-таблиц различного формата, шаг значений - 3 час; при получении новых данных необходимо переводить эти данные в единый формат, чтобы его можно было использовать для чтения;

- значения AQI, обычно поступающие в едином формате и не требующие дополнительной обработки, шаг значений - 1 час [6];

- данные об использовании угля на ТЭЦ г. Бишкек, предоставляются в различном формате, шаг значений - 1 сутки.

Данные могут отсутствовать, что ведет к необходимости исключения их из выборки. Так же данные нужно сопоставить по времени. Данные об использовании угля имеют сложный формат, но не дают большого прироста точности LSTM-сети [4]. Кроме того,

использование этих данных для прогноза в реальном времени проблематично. В первой версии системы их решено не учитывать. Остальные же данные поступают в виде классических таблиц, первая строка которых описывает, какая переменная содержится в соответствующем столбце, а остальные строки представляют собой значения этих переменных в разные моменты времени.

Таким образом, импорт данных можно представить как импорт таблиц, в которых обязательно должен присутствовать столбец, определяющий, какому моменту времени соответствуют значения соответствующей строки и заголовок каждого столбца определяет переменные строки.

Обработка данных. Данные могут поступать в различных единицах измерения. Например: данные о давлении могут быть указаны в некоторых файлах как значения в мм.рт.ст, в других - в Па; данные о направлении ветра могут быть указаны в виде аббревиатуры направления (ССВ для северо-северо-востока) или полного описания (северо-северо-восток).

Для подготовки данных к обучению прогностической модели их необходимо привести к единой системе измерений. Так же к обработке данных можно отнести нормализацию данных [7], так как она представляет собой перевод вектора из одних значений в другие с сохранением размерности.

Определение структуры прогностической модели. Параметры для определения структуры модели сильно отличаются. Например, в случае метода k-средних [8] это может быть предопределенное количество классов, на которые следует делить обучающую выборку. В случае LSTM-сети параметры могут определять количество нейронов в каждом слое сети и т.д. [9]. Таким образом, сама модель должна предоставлять метаданные о параметрах, необходимых для ее определения.

Формирование обучающей и тестовой выборки. Формирование входных параметров. Все векторы данных нужно соотнести по времени и выбрать примеры, в которых N раз подряд нет отсутствующих значений. Такая последовательность может служить для обучения или тестового прогноза. Для разделения всех примеров на тестовую и обучающую используется единый метод кросс-валидации с делением на 5 частей [10].

Обучение прогностической модели. На основе данных об инициализированной модели прогноза (то есть модели, в которой определено количество входных и выходных параметров) можно сформировать тестовую и обучающую выборки на основе исходных данных и обучить модель согласно определенной модели логики. После обучения модель считается готовой для использования. Изначально каждая модель характеризуется точностью, основанной на результатах прогноза с использованием тестовой выборки.

Использование модели для предсказания. По заданным входным параметрам модель должна быть способна прогнозировать значения выходных параметров. В случае прогноза в реальном времени возможно использование прогнозных значений входных параметров (метеопрогноз, например).

Функциональные требования к ПИ. Функциональные требования к программной системе описываются основными сценариями использования (рисунок 1).

Исходя из анализа предметной области выявлены акторы, которые будут взаимодействовать с программным продуктом:

Пользователь – человек, использующий программную систему в целях прогноза;

Источник данных – любой ресурс, который может служить источником данных, распределенных во времени;

Хранилище данных – база данных для сохранения артефактов программной системы: загруженных данных, обучающих выборок, классификаторов и предсказаний.

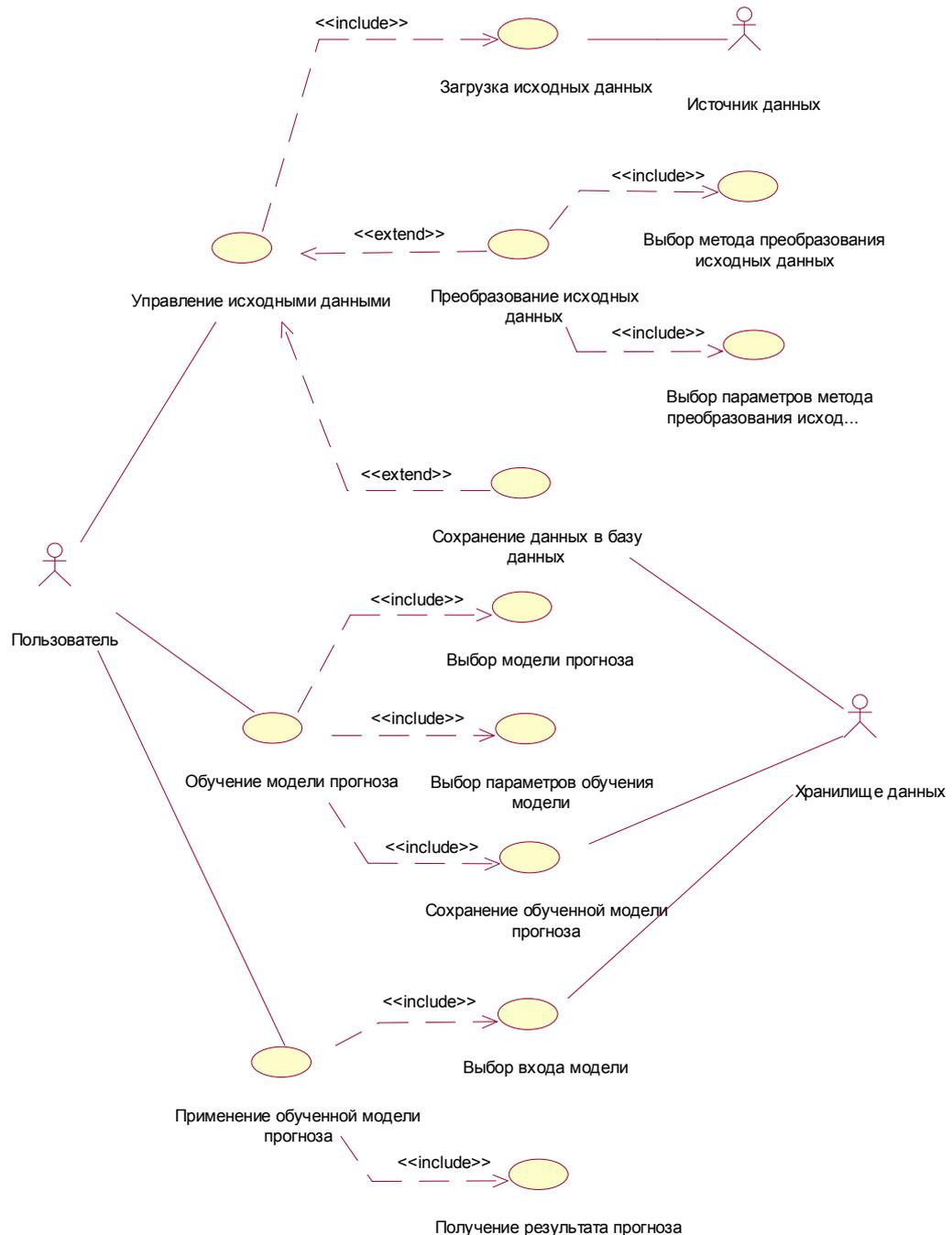


Рисунок 1 - Варианты использования программной системы для обучения и применения моделей прогноза

Выделено 3 основных сценария использования

Управление исходными данными: предполагает загрузку, обработку и сохранение исходных данных;

Обучение модели: предполагает выбор обучаемой модели, формирование обучающей выборки на основе выбранных параметров обучения и сохранение обученной модели в единой базе данных приложения;

Применение обученной модели: предполагает выбор сохраненной модели и прогноз значений выходных параметров модели по выбранным входным параметрам.

Сценарий «Управление исходными данными». Сценарий включает в себя ВИ «Загрузка данных» и расширяется ВИ «Преобразование исходных данных» и «Сохранение данных в БД». ВИ «Преобразование исходных данных» включает в себя, например, приведение строковых значений в числовые, нормализацию числовых данных и т.д. – в зависимости от выбора метода преобразования данных и от его параметров. Также должна быть предоставлена возможность сохранения любых данных (загруженных или преобразованных) в единую БД приложения (ВИ «Сохранение данных в БД»).

ВИ «Загрузка данных» предполагает загрузку данных из различных источников. Это может быть файл данных, база данных, публичное API или любой другой источник в зависимости от выбора пользователя реализации. Загрузка данных детально описана диаграммой деятельности (рисунок 2). Данная диаграмма отображает основной поток управления процессом, распределение ответственности за действия и потоки данных, возникающие в процессе загрузки данных.

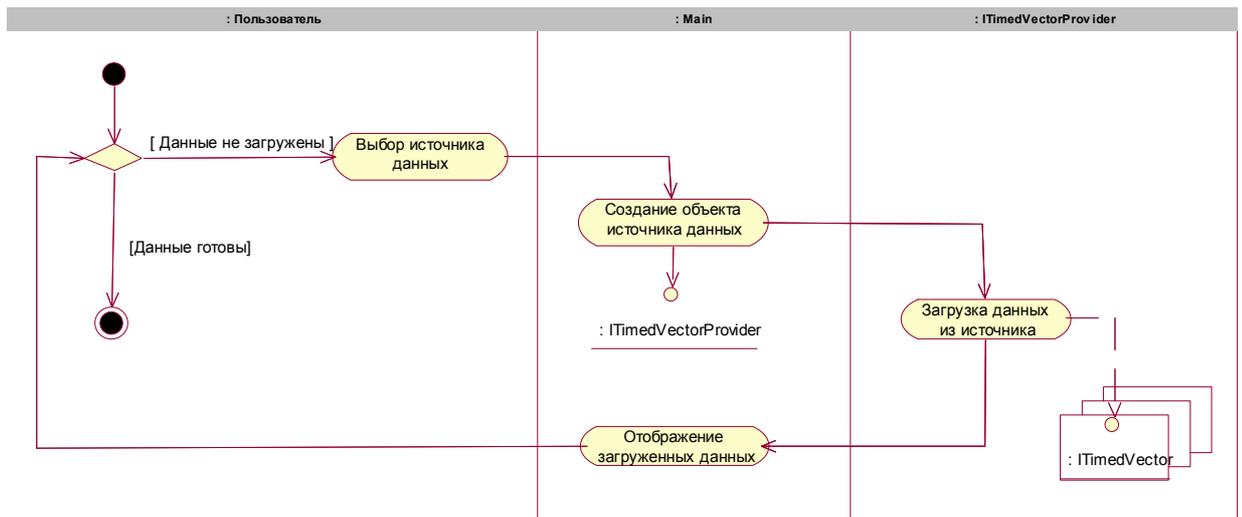


Рисунок 2 – Алгоритм загрузки данных в систему

Основной поток управления Main создает объект источника данных с интерфейсом ITimedVectorProvider, затем может передать управление созданному источнику данных для загрузки векторов данных и отобразить список загруженных векторов пользователю.

Сценарий ВИ «Преобразование исходных данных» детально представлен на рисунке 3.

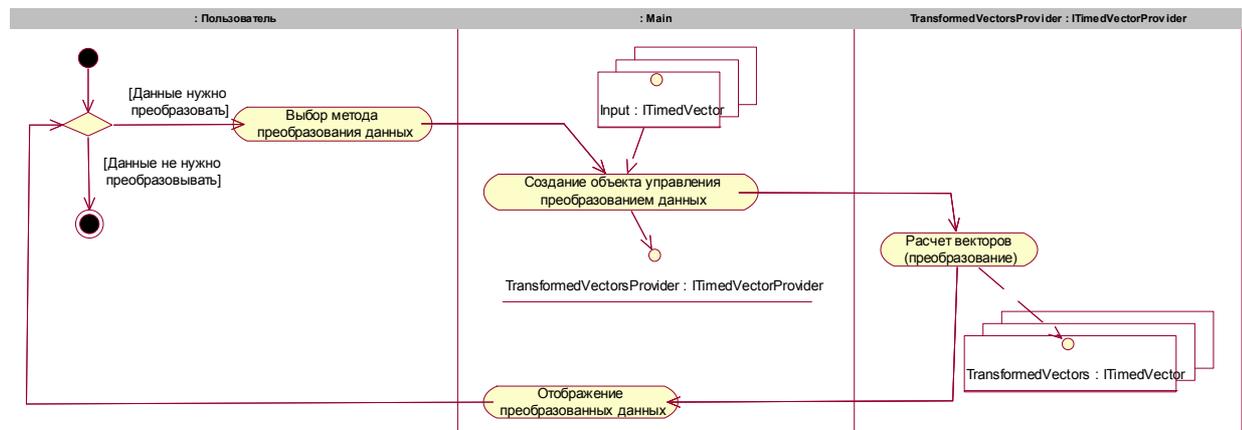


Рисунок 3 - Алгоритм преобразования исходных данных

Пользователю предлагается выбрать метод преобразования данных. На основе уже загруженных (см. ВИ «Загрузка данных») или сохраненных (см. ВИ «Сохранение данных в базу данных») векторов основной поток управления создает объект источника

преобразованных данных с интерфейсом `ITransformedVectorProvider` и передает ему управление. Объект источника преобразованных данных возвращает список преобразованных векторов основному потоку управления для отображения этих данных пользователю.

ВИ «Сохранение данных в базу данных». Сохранение в базу данных является элементарным процессом, основной задачей является определение структуры базы данных для сохранения выбранных данных. Структура отображена диаграммой классов (рисунок 4), каждый класс которой можно интерпретировать как таблицу в реляционной базе данных, а объект класса – как запись в соответствующей таблице.

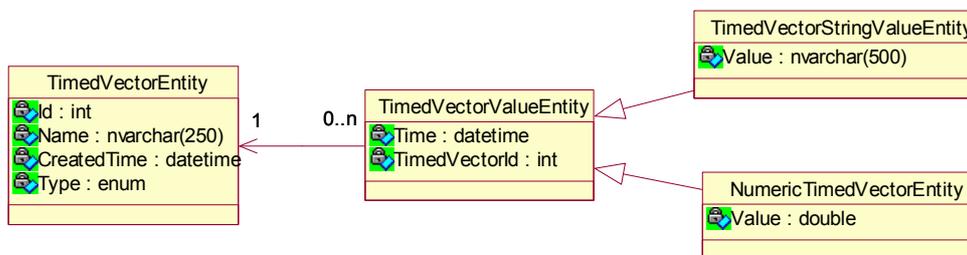


Рисунок 4 - Модель для хранения векторов данных в реляционной базе данных

Объект класса `TimedVectorEntity` хранит в себе информацию о векторе данных: его название и тип (набор строковых данных, набор числовых значений и т.п.). Объект класса `TimedVectorValueEntity` представляет собой одно значение из вектора. В первой итерации это пара время-строка (`TimedVectorStringValueEntity`) или пара время-число (`NumericTimedVectorEntity`).

Сценарий «Обучение модели прогноза». Обучение модели является основным вариантом использования программной системы (рисунок 5).

При запуске системы регистрируются все компоненты системы, в том числе доступные типы моделей прогноза, которые отображаются пользователю первым шагом в этом сценарии.

Из предоставленного списка пользователь должен выбрать тип модели прогноза, которую он хочет обучить для дальнейшего использования. Так как для создания модели нужны дополнительные данные об иницируемой структуре, обучающем наборе данных (входные и выходные вектора), а система пользуется только интерфейсом, не зная о реализации конкретных типов моделей, сама модель должна предоставить основному потоку управления структуру необходимых ей данных. Этой структуры должно быть достаточно для отображения формы ввода данных пользователем системы.

После ввода данных основной поток инициализирует объект выбранной модели прогноза, сохраняет ее и входные параметры в базу данных, инициализирует обучение.

Для дальнейшего отображения прогресса обучения система сможет использовать ту же базу данных, отмечая исходные данные для обучения как использованные.

Добавим к структуре базы данных таблицы для сохранения прогностической модели. Структура базы данных отображена на диаграмме классов (рисунок 6), где каждый класс отображает таблицу в базе данных, а объект класса – запись таблицы. Чтобы отличать классы, отображающие структуру базы данных от бизнес-моделей, к названиям класса добавлен постфикс «Entity».

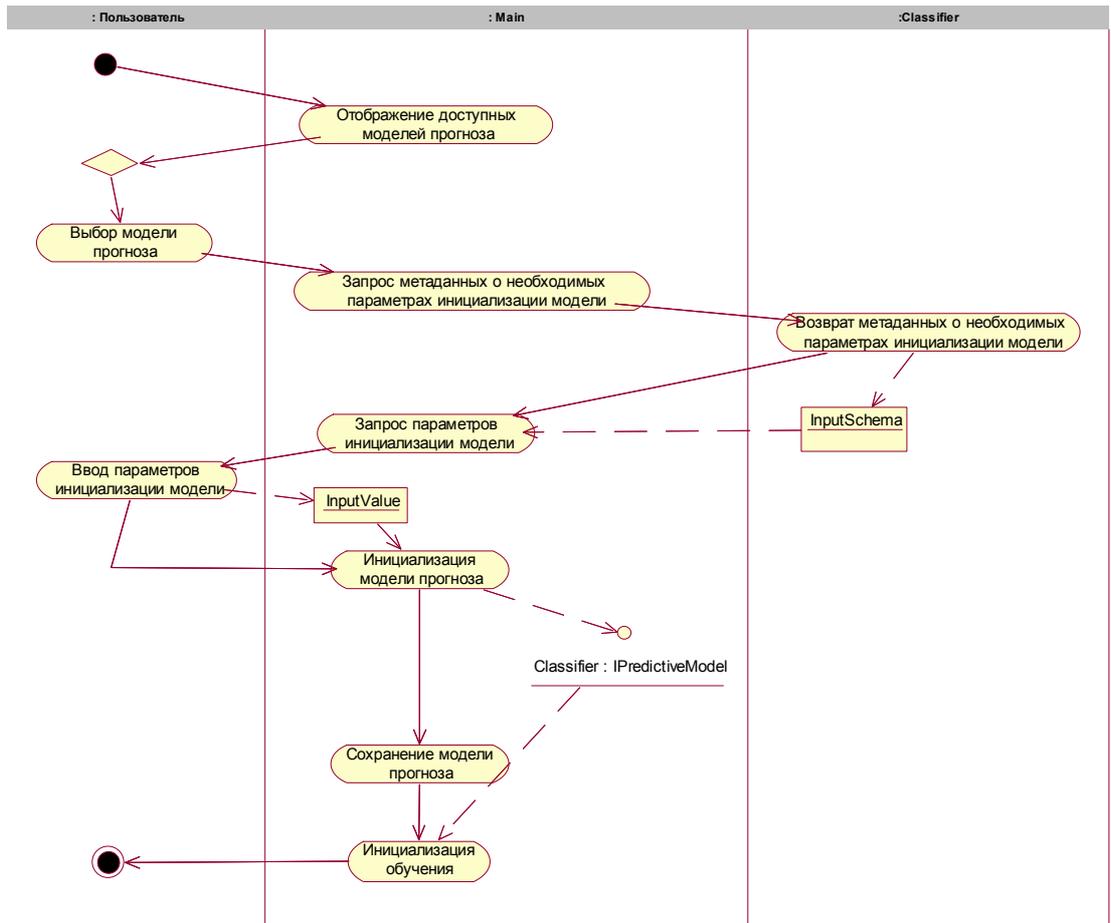


Рисунок 5 - Диаграмма деятельности, описывающая процесс создания и обучения модели прогноза

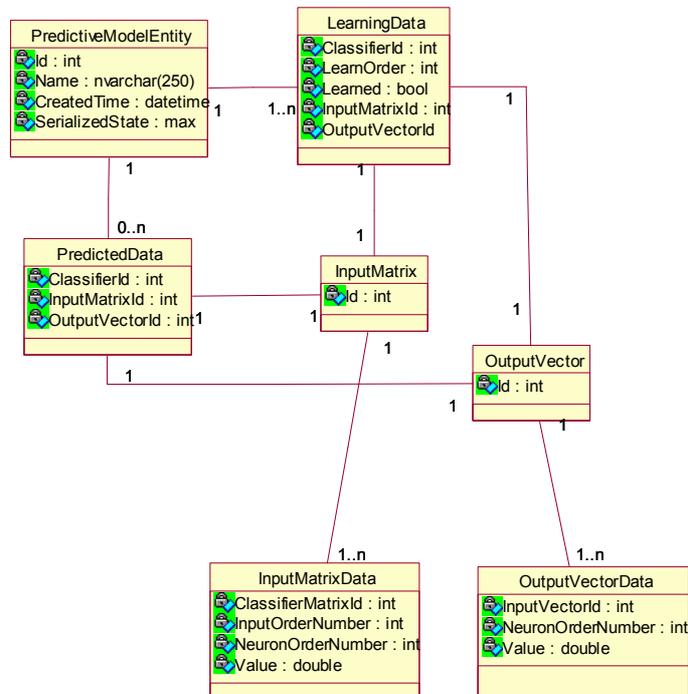


Рисунок 6 - Классы для хранения прогностической модели в реляционной базе данных

Основная таблица, сохраняющая список обучаемых / обученных моделей PredictiveModelEntity хранит сериализованную структуру модели на текущий момент времени (необученную, обученную частично или обученную) и имя, используемое пользователем программной системы.

Для использования модели необходимо знать пограничную структуру модели – описание входных и выходных параметров модели. Для этого в базе создается таблица PredictiveModelBoundary, запись которой ассоциирована с конкретной сохраненной моделью. Количество записей типа “Вход” определяют количество входных векторов прогностической модели, количество записей “Выход” – количество выходных векторов (результат прогноза). Каждая запись определяет последовательность вектора и его имя для использования в программной системе. Каждый набор вход/выход для обучения сохраняется в таблице LearningData. Вход InputMatrix представляет собой матрицу размерности $I \times S$, где I – количество факторов прогноза, а S – длина последовательности векторов размерности I , приходящей на вход прогностической системы. Выход OutputVector представляет собой вектор длины O , где O – количество векторов результата прогноза.

Сценарий «Применение обученной модели». На рисунке 7 показан алгоритм применения обученной модели в виде диаграммы деятельности. Пользователь из списка обученных прогностических моделей, которые отображает главный поток управления, выбирает модель, которую хочет использовать для прогноза. Главный поток управления загружает в память структуру модели и на основе данных о входных векторах предлагает пользователю ввести источник данных для каждого из входов модели. После выбора всех источников модель делает прогноз на ближайшие моменты времени и главный поток управления отображает его пользователю программной системы.

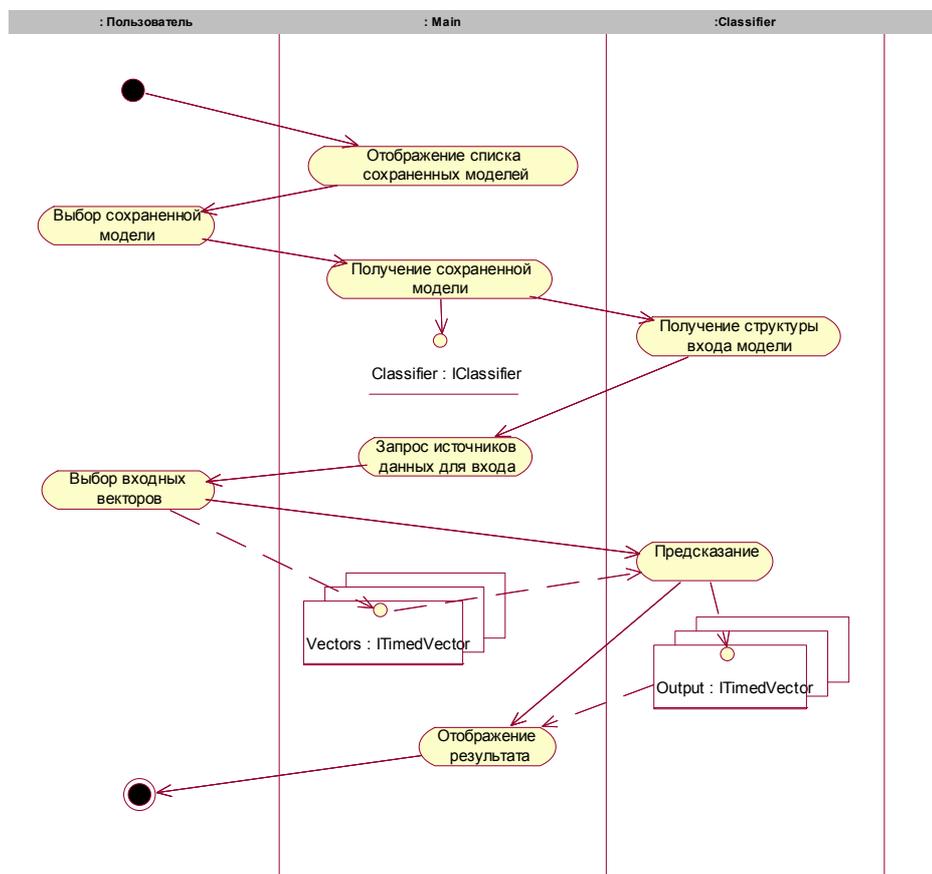


Рисунок 7 - Алгоритм применения обученной модели

4 Проектирование и конструирование библиотеки ядра программной системы. Из диаграмм деятельности можно выделить основные компоненты и интерфейсы программной системы:

Источник данных (интерфейс, способный предоставить список доступных векторов),
 Вектор (Один доступный вектор, основные методы – чтение значений время / значение),

Обработчик векторов или преобразование (класс управления, способный превращать N векторов в M других векторов согласно некоторому алгоритму с параметрами),

Прогностическая модель (класс управления, способный по последовательности S векторов предсказывать состояние других векторов в момент времени T),

Вход прогностической модели,

Выход прогностической модели.

Модель системы отображена на следующей диаграмме классов (рисунок 8).

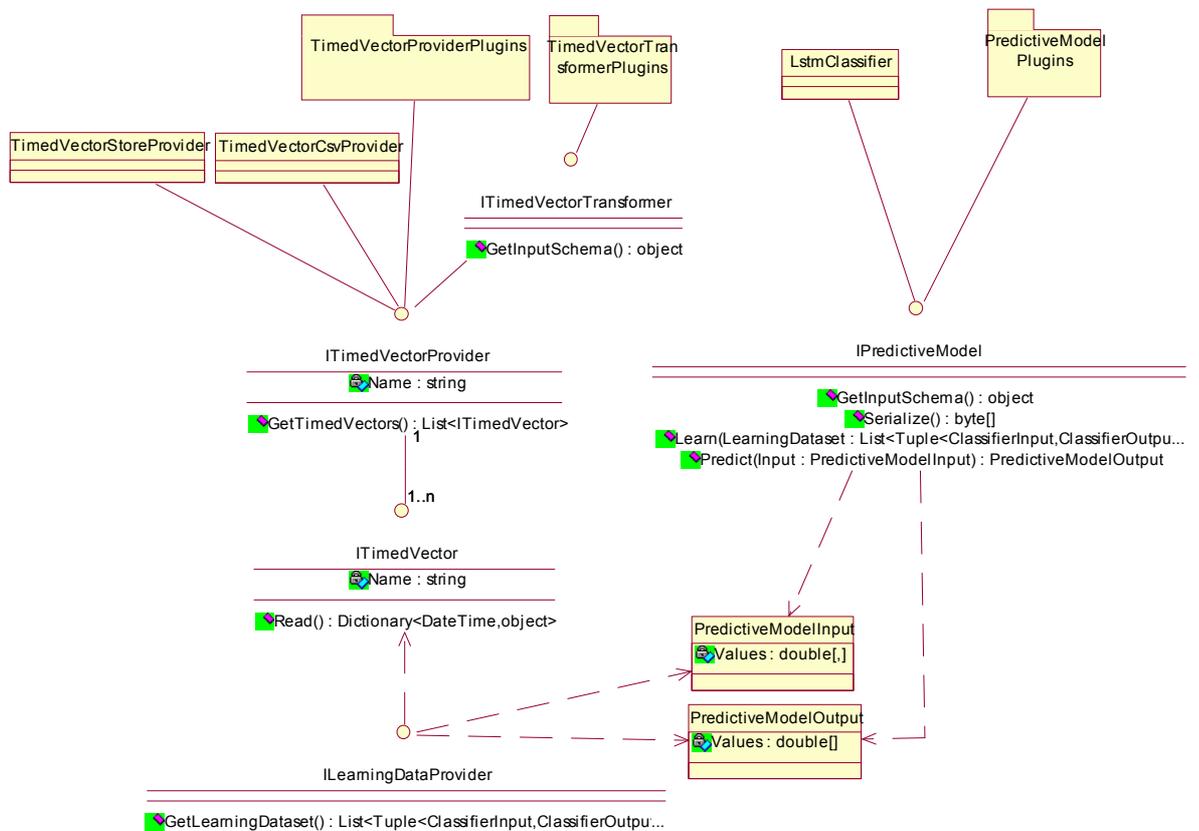


Рисунок 8- Модель программной системы для разработки, обучения и применения прогностических систем

Ядром системы служат обобщенные интерфейсы, и система не зависит от их реализаций. Основная сущность, над которой производится различные преобразования в процессе, это вектор значений во времени. За его представление отвечает интерфейс ITimedVector. Он определяет основную операцию над вектором – получение значений (метод Read). В зависимости от реализации это может быть чтение вектора из файла, базы данных, преобразованный вектор.

Исходя из предполагаемых вариантов использования, возможно выбирать и применять различные способы преобразования и обработки векторов, необходимые для получения новых значений из уже существующих. Результат обработки данных так же является коллекцией векторов, что определяет способ обработки данных как источник данных. Однако для предоставления возможности настраивать способы обработки данных

необходимо предоставить пользовательский интерфейс, что требует дополнительные метаданные о способе обработки: его параметры и ожидаемый результат. Поэтому был определен дополнительный интерфейс `ITimedVectorTransformer` который расширяет интерфейс `ITimedVectorProvider`.

Основная задача программы – прогноз некоторых значений на основе известной информации. Ответственность за обучение и прогноз несет интерфейс `IPredictiveModel`. Основные операции интерфейса `IPredictiveModel` представлены в таблице 1.

Таблица 1 - Основные операции интерфейса `IPredictiveModel`

<code>GetInputSchema()</code>	Предоставляет метаданные о параметрах модели прогноза. В случае классификатора это может быть количество слоев, количество нейронов в каждом слое, количество нейронов на входе и выходе классификатора. Т.е. все данные необходимые для определения структуры модели.
<code>Learn()</code>	Операция, доступная для запуска обучения модели. С случае обучения с учителем принимает последовательность входных данных и соответствующий каждой последовательности набор ожидаемых значений выходных векторов. В случае обучения без учителя это может быть только набор входных параметров.
<code>Predict()</code>	Операция, доступная для прогноза значений выходных параметров при заданных входных параметрах.

Также представлены два вспомогательных класса для представления входов и выходов обучаемой модели `PredictiveModelInput` и `PredictiveModelOutput`.

Класс `PredictiveModelInput` инкапсулирует структуру и логику входа прогностической модели. Предполагается, что входом прогностической модели является временная последовательность значений параметров (например, в случае рекуррентных нейронных сетей) или один набор значений параметров (например, в случае нейронной сети прямого распространения). В случае LSTM-классификатора это последовательность параметров входов нейронов. Все эти варианты можно описать матрицей $S \times N$, где S - это длина временной последовательности значений параметров, а N – количество значений в каждый момент времени (количество входных нейронов классификатора).

Класс `PredictiveModelOutput` инкапсулирует структуру и логику результата предсказаний. Например, для модели классификатора это обычно количество классов, которое способна определить модель. То есть в случае LSTM-сети `PredictiveModelOutput` – вектор с длиной O , где O – количество выходных нейронов классификатора.

В таблице 2 представлены классы и интерфейсы, которые определяют домен задачи. Их достаточно для описания бизнес-логики приложения.

Таблица 2 - Классы и интерфейсы для описания бизнес-логики программной системы.

Интерфейс	Описание	Основные операции
<code>ITimedVectorProvider</code>	Интерфейс определяет способ получения исходных данных из любых источников. Примеры источников данных: База данных, Файл, Обработчик вектора	- Получение списка доступных векторов
<code>ITimedVectorTransformer</code>	Интерфейс определяет способ преобразования одних исходных данных в новые. Является частным случаем интерфейса <code>ITimedVectorProvider</code> , наследует все его операции, но является настраиваемым, для чего предоставляет метаданные о структуре необходимых параметров.	- Получение метаданных о структуре параметров преобразования (статический метод) - Получение списка доступных векторов
<code>ITimedVector</code>	Представляет исходные данные в виде пар время / значение	- Чтение вектора из источника данных
<code>ILearningDataProvider</code>	Управляющий класс, способный из выбранных векторов входных и выходных данных генерировать данные для каждого шага обучения модели	- Получение обучающей выборки
<code>IPredictiveModel</code>	Основной класс для представления модели	-Получение мета-данных о

	прогноза в системе	параметрах модели (статический метод) - Обучение - Сериализация - Прогнозирование
--	--------------------	-----------------------------------------------------------------------------------------------

5 Общая архитектура системы. Архитектуру программной системы решено построить на базе чистой архитектуры (Clean Architecture, рисунок 9). Впервые эту архитектуру описал Роберт Сесил Мартин [11].

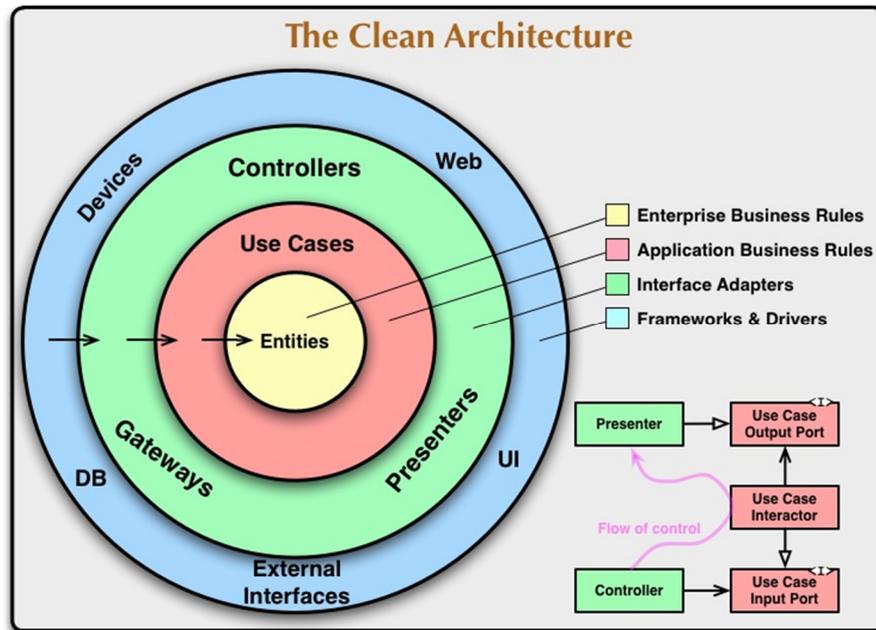


Рисунок 9 - Чистая архитектура

Концентрические круги на диаграмме представляют собой различные слои программного обеспечения.

Основой чистой архитектуры является Правило Зависимостей. Это правило гласит, что зависимости в программной системе могут указывать только во внутренние слои. Ничто из внутреннего слоя не может знать что-либо о внешнем слое. Это касается функций, классов, переменных и т.д.

Более того, структуры данных, используемые во внешнем круге, не должны быть использованы во внутреннем круге, особенно если эти структуры генерируются фреймворком во внешнем круге. Мы не используем ничего из внешнего круга, что могло бы повлиять на внутренний.

Сущности системы. Сущностями в данной системе являются основные типы данных, используемые в программном обеспечении: отображение базы данных, вспомогательные типы, такие как PredictiveModelInput и PredictiveModelOutput и т.п.

Сценарии использования. Представляют собой логику работы приложения. Слой включает в себя основные управляющие классы: источники исходных данных ITimeVectorProvider, провайдер обучающей выборки ILearningDataProvider, классификатор IPredictiveModel и т.д. Они используют сущности предметной области для реализации необходимой логики. В данной системе этот слой определяет только интерфейсы управляющих классов.

Интерфейс-Адаптеры. В проектируемой программной системе будут представлены классами, преобразующими данные из предыдущих слоев в данные удобные для представления пользователю и обратно.

Фреймворки и драйверы. Представляет собой реализацию интерфейсов, объявленных в нижних слоях системы. Таким образом детали реализации изолируются от логики приложения.

Для реализации Правила Зависимостей обычно прибегают к принципу инверсии зависимостей (механизм инверсии зависимостей). Например, предположим, что для формирования обучающей выборки (в слое Сценариев) нужно обратиться к базе данных и прочесть несколько векторов данных. Однако, этот вызов обязан быть не прямым, чтобы не нарушать Правило Зависимостей — внутренний слой не должен знать ничего о внешнем. Таким образом, источник обучающих данных `ILearningDataProvider` пользуется интерфейсом `ITimedVector` во внутреннем круге, а представление `TimedVectorStoreProvider` из внешнего круга реализует его, чтобы прочитать вектор из базы данных.

Та же самая техника используется, чтобы пересечь все границы в архитектуре. Мы воспользуемся преимуществами динамического полиморфизма для создания зависимостей в исходном коде так, чтобы поток управления соответствовал Правилу Зависимостей. Такая техника позволяет уменьшить связность классов, что способствует масштабируемости системы. Этот механизм позволит заменять компоненты системы, не изменяя структуру остального кода.

Заключение. Таким образом, в работе представлена модель системы, предоставляющей программный интерфейс для обучения и применения моделей прогноза данных. Смоделированная архитектура системы достаточна для ее реализации средствами объектно-ориентированного программирования.

Разработанная на основе модели система позволит:

управлять исходными данными, подготавливать новые данные на основе исходных для обучения и применения различных моделей прогноза,
обучать и использовать различные обучаемые модели,
подключать в виде плагинов другие методы обработки исходных данных и обучаемых моделей, реализующие соответствующие интерфейсы.

Выбор архитектуры системы обоснован требованиями к функционалу программного продукта.

В дальнейшем планируется реализовать проект адаптивного классификатора AQI на основе LSTM-сети и сделать программный код публичным. Таким образом, система сможет быть расширена дополнительным функционалом и новыми обучаемыми моделями. Трудозатраты на разработку нового функционала будут уменьшены за счет того, что разработчик нового функционала должен знать только интерфейсы, которые он должен реализовать. Детали реализации программной системы будут спрятаны и не важны для разработчика.

Литература

1. Air Quality Index (AQI) - A Guide to Air Quality and Your Health. US EPA. 9 December 2011.
2. Великанова, Л. И. Мультирегрессионные и обобщенно-регрессионные нейросетевые модели краткосрочного прогноза загрязнения PM_{2.5} в г. Бишкек с учетом метеорологических параметров / Л. И. Великанова, Н. М. Лыченко // Проблемы автоматизации и управления. – 2019. – № 2(37). – С. 16-25. – DOI 10.5281/zenodo.3594777. – EDN MNNDYU.
3. Лыченко Н.М., Сороковая А.В. [Прогнозирование классов индекса качества воздуха г. Бишкек с учетом новых данных 2020-2021 гг. на базе LSTM-нейросетевого классификатора](#) // Проблемы автоматизации и управления. - 2021. - № 2 (41). - С. 89-99.
4. Лыченко, Н. М. Модификация классификатора индекса качества воздуха г. Бишкек с учетом фактора загрязнения / Н. М. Лыченко, А. В. Сороковая

// Проблемы автоматки и управления. – 2021. – № 3(42). – С. 101-110. – EDN BXWCYS.

5. Сайт «Расписание погоды rp5.ru» [Архив погоды в Бишкеке](https://rp5.ru/%D0%90%D1%80%D1%85%D0%B8%D0%B2_%D0%BF%D0%BE%D0%B3%D0%BE%D0%B4%D1%8B_%D0%B2_%D0%91%D0%B8%D1%88%D0%BA%D0%B5%D0%BA%D0%B5)
https://rp5.ru/%D0%90%D1%80%D1%85%D0%B8%D0%B2_%D0%BF%D0%BE%D0%B3%D0%BE%D0%B4%D1%8B_%D0%B2_%D0%91%D0%B8%D1%88%D0%BA%D0%B5%D0%BA%D0%B5 (дата обращения: 05.10.2022)
6. AirNow Department of State // https://airnow.gov/index.cfm?action=airnow.global_summary#U.S._Department_of_State_Bishkek, (дата обращения: 05.10.2022).
7. Singh B. Investigating the impact of data normalization on classification performance. Applied Soft Computing Journal, 2020, vol. 97, p. 105524.
8. MacQueen J. (1967). Some methods for classification and analysis of multivariate observations. In Proc. 5th Berkeley Symp. on Math. Statistics and Probability, pages 281—297.
9. Understanding LSTM Networks. – URL: <http://colah.github.io/posts/2015-08-Understanding-LSTMs/>. (дата обращения 14.09.2020)
10. Кросс-валидация (Cross-validation). URL: <https://wiki.loginom.ru/articles/cross-validation.html> (дата обращения: 05.11.2022).
11. Роберт Мартин. Чистая архитектура. Искусство разработки программного обеспечения / Перевод Киселев А. // Питер, 2022. 352 с.

ПРОБЛЕМЫ АВТОМАТИКИ И УПРАВЛЕНИЯ

Научно-технический журнал

Литературный редактор Е.В. Комарова

Компьютерная верстка А.У. Касимова

Подписано к печати 30.11.2022 г. Формат 70/108 1/8

Печать офсетная. Объем 22,6 усл. п.л. Тираж 200 экз.

Издательство «ИМА»

720071, Бишкек, проспект Чуй, 265