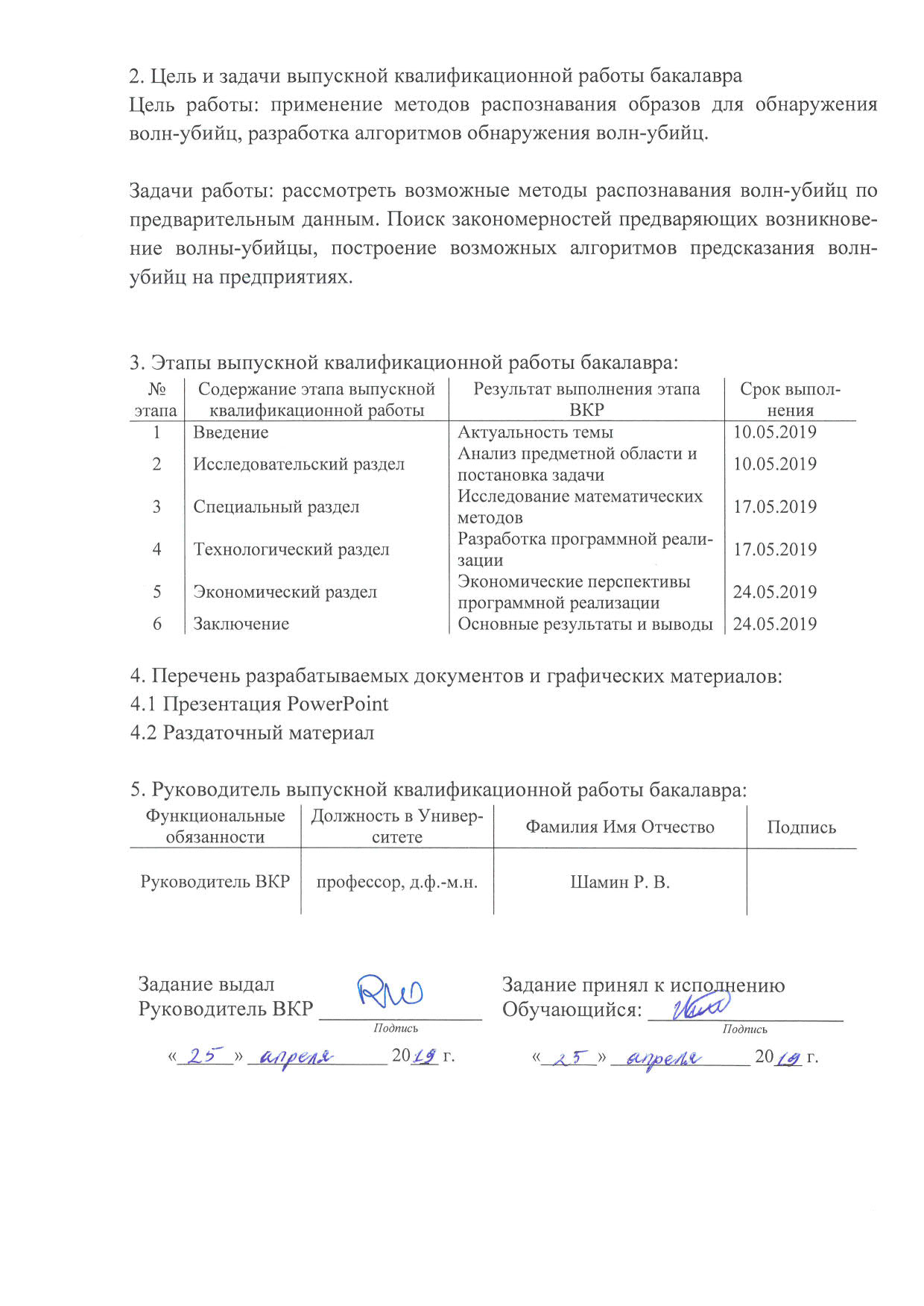
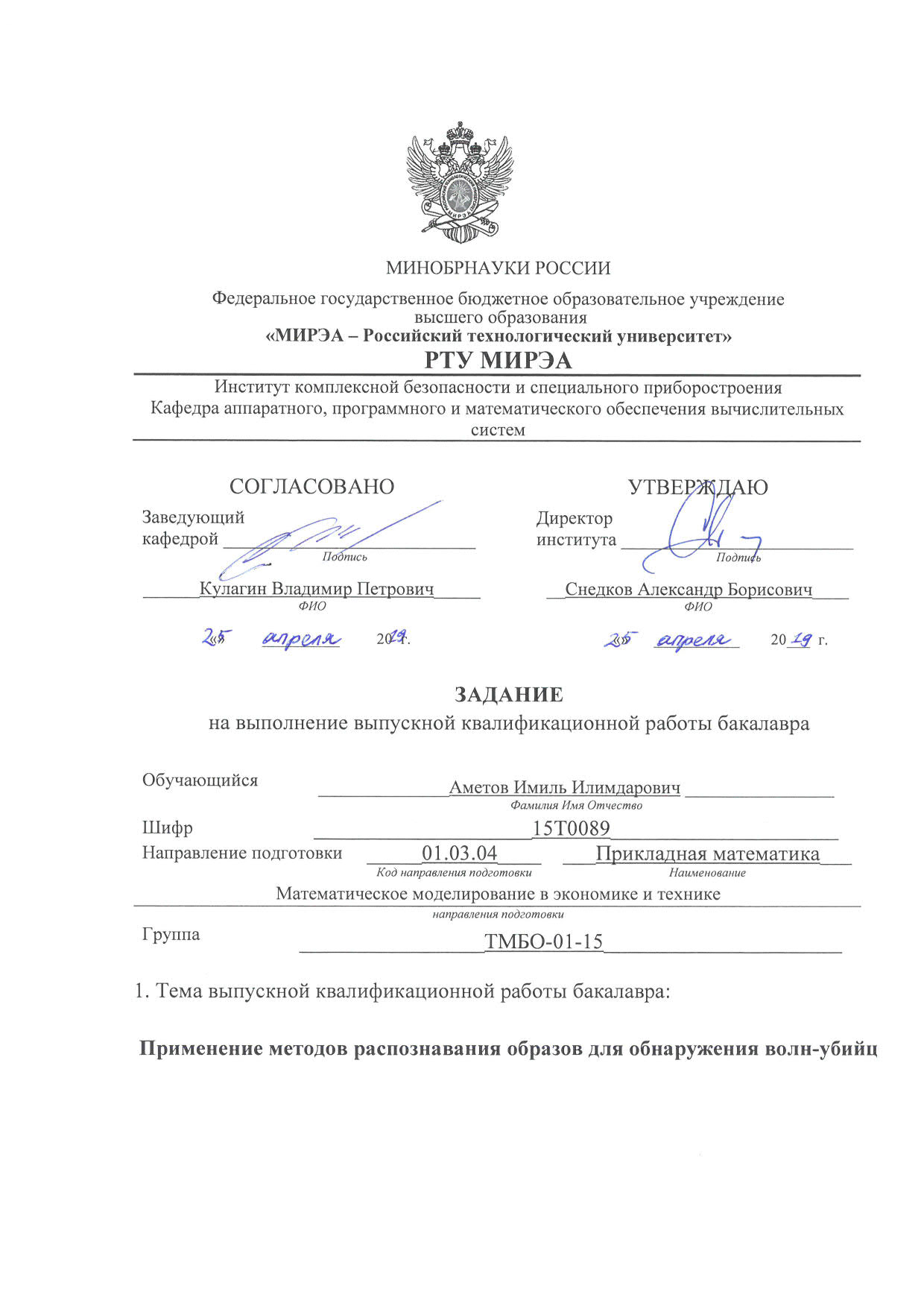


****

**АННОТАЦИЯ**

на выпускную квалификационную работу бакалавра на тему

**«Методы распознавания образов в обнаружении волн-убийц»**

студента МИРЭА – Российского технологического университета

Института комплексной безопасности и специального приборостроения

группы ТМБО-01-15

**Аметова Имиля Илимдаровича**

**Цель данной работы:** проверить применимость искусственных нейронных сетей для распознавания и прогнозирования волн-убийц.

Работа состоит из введения, четырех разделов, заключения и списков использованных источников.

Квалификационная работа содержит 37 страниц машинописного текста (без приложений), 2 таблицы, 6 рисунков, 5 приложений, 19 наименований списка использованных источников.

**Ключевые слова:** искусственный интеллект, искусственные нейронные сети, волны-убийцы, прогнозирование волн, методы распознавания образов.

ОГЛАВЛЕНИЕ

[ВВЕДЕНИЕ 5](#_Toc11915310)

[1. Исследовательский раздел 8](#_Toc11915311)

[1.1. Исторический экскурс 8](#_Toc11915312)

[1.2. Строение искусственного нейрона 13](#_Toc11915313)

[1.3. Виды искусственных нейронов 17](#_Toc11915314)

[1.4. Строение искусственной нейронной сети 17](#_Toc11915315)

[1.5. Обучение искусственной нейронной сети 20](#_Toc11915316)

[1.6. Процедура обучения методом обратного распространения ошибки 22](#_Toc11915317)

[2. Специальный раздел 24](#_Toc11915318)

[2.1. Расчёт искусственной нейронной сети, реализующей операцию XOR 24](#_Toc11915319)

[2.2. План расчёта погрешности при прогнозировании волн 29](#_Toc11915320)

[3. Технологический раздел 31](#_Toc11915321)

[3.1. Описание программной реализации 31](#_Toc11915322)

[4. Экономический раздел 35](#_Toc11915323)

[4.1. Экономические перспективы прогнозирования волн-убийц 35](#_Toc11915324)

[ЗАКЛЮЧЕНИЕ 36](#_Toc11915325)

[Список используемых источников 37](#_Toc11915326)

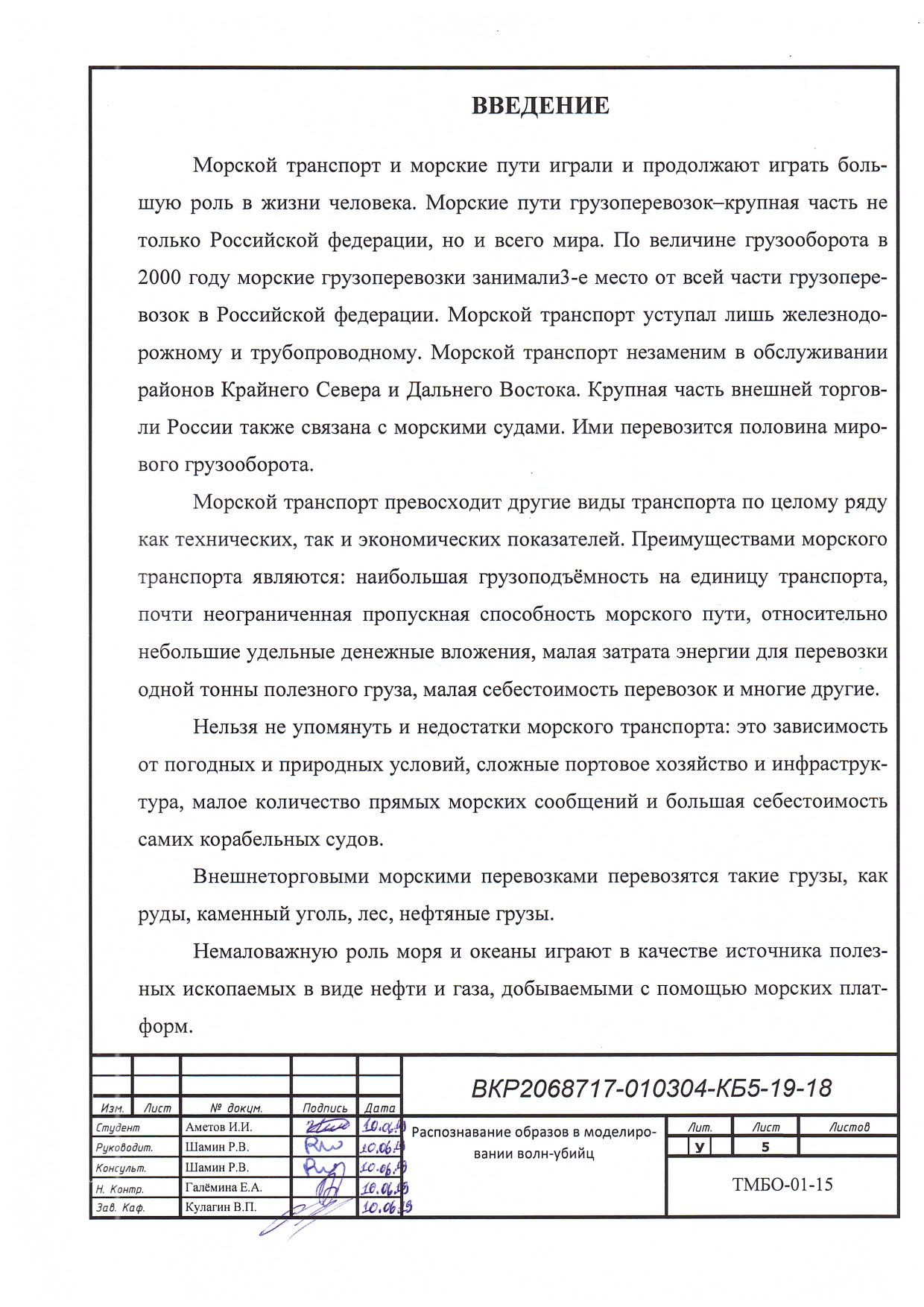
[Приложение А. Листинг программного кода 39](#_Toc11915327)

[Приложение Б. Презентация 41](#_Toc11915328)

[Приложение В. Результат проверки на оригинальность текста 52](#_Toc11915329)

[Приложение Г. Заключение руководителя об оригинальности текста 55](#_Toc11915330)

[Приложение Д. Отзыв научного руководителя 56](#_Toc11915331)



Нефтяные платформы − это инженерные комплексы, предназначенные для бурения скважин и добычи нефти и газа, залегающих под дном морей или океанов.

В США количество платформ от Backer Hughes (крупного производителя нефтяных платформ) составляет 1047 штуки, в Канаде: 212 штук, международных: 1023 штук [2]. При этом минимальная цена на одну нефтяную платформу составляет около двухсот миллионов долларов США.

Велики затраты, которые несут производители на выпуск и техническое обслуживание морских судов. За один только 2018 год в Российской Федерации было произведено 89 крупных и средних судов и кораблей почти на 170 миллиардов рублей[12].

Подытоживая вышесказанное можно сказать, что морские транспортные пути играют большую роль в жизни человечества. В связи с этим возникает необходимость в изучении физики морей и океанов.

Изучению течений и физике потоков издавна уделяется большое внимание науки и техники.

Океанология или океанография – это наука изучающая взаимодействие океана и атмосферы, физическое, химическое и биологическое взаимовлияние океанов и материков, флору и фауну океана и их экологическое взаимодействие, различные процессы происходящие на дне, глобальные и локальные явления и многое другое.

Одно из явлений, происходящее в океанах и морях называется волнами-убийцами.

Волны-убийцы – это название гигантских одиночных волн, возникающих в океане. Высота таких волн составляет 20 – 30 метров. Кроме того, волны-убийцы обладают необычным для морских волн поведением. Такие волны представляют большую опасность для различных морских судов и сооружений. При столкновении такой волны с судном, корпус судна может не выдержать массы воды, что проводит к потоплению судна.

Ещё одной особенностью волн-убийц является их внезапность появления.

Долгое время волны-убийцы считались одной из частью фольклора моряков поскольку не было достаточно убедительных доказательств существования этих волн. Первым зарегистрированным столкновением с волной-убийцей стало происшествие на платформе для добычи нефти «Дропнер» произошедшее 1 января 1995 года у побережья Норвегии в Северном море. Платформа «Дропнер» получила небольшие повреждения после обрушения на неё волны-убийцы. При этом в тот день наблюдались волны около 12 метров высотой, и среди них внезапно возникла волна высотой 25,6 метра [5]. В последующем, в рамках европейского исследовательского проекта «MaxWave» были проведены как спутниковые, так и подводные и надводные наблюдения, выявившие за три недели более десяти гигантских одиночных волн, высота которых превышала 25 метров [9].

До сих пор не до конца понятны механизмы и причины возникновения волн-убийц. В книге «Математические вопросы волн-убийц» Р.В. Шаминым в качестве одной из гипотез возникновения волн-убийц приводится фокусировка волновой энергии в некоторых структурах поверхностных течений [19].

Естественно, что изучение возможности заранее определять и предсказывать возникновение волн-убийц также является важной задачей.

# Исследовательский раздел

Рядом с островом Сахалин на дне океана установлены датчики высоты волн, регистрирующие с заданной периодичностью уровень воды.

Полученные данные сохраняются на карту памяти датчика. С периодичностью раз в год датчики поднимают со дна и заменяют карту памяти.

Требуется обработать полученный массив и на основе полученной информации написать программу, которая была бы способна предсказывать появление аномально высоких волн.

Одним из возможных способов исследования больших массивов данных является применение методов так называемого «искусственного интеллекта».

## Исторический экскурс

В середине 1950-х годов были сконструированы первые электронно-вычислительные машины первого поколения. Практически сразу же на них стали решать задачи, которые теперь относят к интеллектуальным. Тогда же начали проводиться исследования, оказавшие значительное влияние на возникновение такого научного направления, как искусственный интеллект.

Ниже перечислены некоторые из областей исследования в области искусственного интеллекта:

1. Машинный перевод. 7-го января 1954 года в США был проведён эксперимент, подготовленный корпорацией IBM и Джорджтаунским университетом. В этом эксперименте была продемонстрирован полностью автоматический перевод с русского языка на английский более шестидесяти предложений[6]. Этот эксперимент произвёл сенсацию в среде специалистов. И хотя перевод был далёк от идеального, этот эксперимент на деле доказал что применение электронно-вычислительных машин для обработки текстов на естественных языка возможно и сулит хорошие перспективы. Дальнейший путь развития этой области был полон крупными разочарованиями и скромными достижениями. Одним из факторов было отсутствие должных знаний о проектировании больших программных комплексов, помимо этого были плохо разработаны точные лингвистические модели. В первых версиях алгоритмов машинного перевода пытались совместить в одной программе как информацию о языках, так и правила для перевода. В шестидесятых годах были разработаны системы, в которых для перевода между двумя языками начал применяться специализированный язык-посредник, применение которого существенно упростило сопоставление между собой конструкций различных языков. Позже, во второй половине 70-х годов, были разработаны системы третьего поколения, и язык-посредник трансформировался в модели глубинных семантик, описывающие семантические взаимосвязи различных естественных языков.

Важными результатами в исследованиях машинного перевода стали возникновение и развитие морфологического анализа, глубинный семантический анализ, поверхностный семантический анализ, глубинный лексический анализ, прагматический анализ[11].

1. Автоматизированное реферирование и информационный поиск. Если машинный перевод представлялся как пофразовый перевод, то для автоматизированного реферирования чаще обращали внимание на крупные участки текста (например, абзацы), где смысл текста сходился к одной и той же тематике. Иначе говоря, основное внимание исследователей было обращено на обнаружение и выявление закономерностей из которых формировался смысл текста. В первых работах наиболее часто использовались подходы, основанные на статистическом выявлении закономерностей в терминах, найденных в тексте и их взаимного расположения [15]. В последующих работах выявилась тенденция к обнаружению и использованию внутренней структуры текста и обнаружения того информационного базиса, вокруг которого строится весь текст [16].

Среди достижений в области автоматизированного реферирования и информационного поиска стоит отдельно упомянуть следующие пункты, так или иначе используемые практически всеми пользователями электронно-вычислительных машин и интернета: поисковые системы, системы поиска и нахождения заимствований из различных источников.

1. Автоматическое доказательство теорем. Доказательство теорем является такой деятельностью человека, автоматизация которой осуществляется легче всего. Причиной этому является высокая степень формализации математических доказательств. Особенно хорошо это выражено в разделах математики связанных с формальными системами. На первом этапе развития автоматизации доказательств было возможно находить только простые доказательства с помощью правил вывода к начальным аксиомам. В СССР одной из первых систем автоматического доказательства теорем была разработанная под руководством Н.А. Шанина программа АЛПЕВ.В скором времени логики обратили внимание на проблему вывода и поиска эффективных процедур доказательств. Во второй половине 60-х годов были найдены две процедуры, которые прочно вошли в состав средств, используемых в исследованиях по искусственному интеллекту. Первой процедурой был метод резолюций предложенный британским философом и логиком Джоном Аланом Робинсоном в 1965 году. Спустя два года Масловым Сергеем Юрьевичем (СССР) была предложена вторая процедура ‑обратный метод доказательства. Оба этих метода дали возможность искать доказательства в исчислении предикатов и являются основополагающими идеями. На базе этих методов возникли множество других модификаций. В 1936 году Алонсо Чёрч и Алан Тьюринг независимо друг от друга показали, что не существует универсальной процедуры доказательства тождественной истинности произвольной формулы в исчислении предикатов первого порядка, если заранее не было известно, была ли она таковой. В связи с этим особый интерес вызывает нахождение процедур вывода, предназначенных для доказательства утверждений в тех проблемных областях, в которых за счёт их специфических особенностей становится возможным построение эффективных процедур доказательств.

Благодаря исследованиям в области автоматического доказательства теорем возникли такие высокоуровневые языки программирования и продукты как Agda [10], Coq [1] и другие.

Автоматическое доказательство теорем находит применение не только в научных, академических кругах, но и при разработке широкого круга промышленного программного обеспечения. В этом случае автоматическое доказательство теорем служит в качестве проверки корректности алгоритмов.

Сферы применения критически важного программного обеспечения весьма широки: аэрокосмическая индустрия [7], операционные системы компании Microsoft [3], встраиваемые системы [4], разработка оптимального управления в сфере оказания медицинских услуг [8].

1. Распознавание образов. Задачи распознавания образов близки к задачам классификации: требуется найти такую совокупность признаков, благодаря которым стало бы возможным выносить решение о том, относятся ли те или иные отдельные объекты к некоторой совокупности других объектов или же порождают новую совокупность. Распознавание образов – одна из наиболее активно развивающихся направлений в проблематике искусственного интеллекта. Одна из основных причин развития исследований по направлению распознавания образов – это её важное прикладное значение. Различными исследователями в этой области были разработаны большое количество методов, основанных на идеях классификации по признакам. Такие методы оказались очень близки к искусственному интеллекту. Методы классификации продолжают развиваться и оказывать влияние на последующие исследования [17].

Одним из инструментов в области классификации образов является искусственная нейронная сеть.

Искусственные нейронные сети построены по тем же основным принципам и методу функционирования, как и естественные биологические нейронные сети. Искусственные нейронные сети применяются во многих областях информационной науки: распознавание текстов, лиц, регистрация номеров нарушителей правил дорожного движения, поиск лиц по базе данных преступников, управление беспилотными автомобилями, для прогнозирования различных экономических процессов, в системах поддержки принятия решений [14].

Появление самого термина «нейрон» произошло в середине 50-х годов XXвека.

В 1943 году нейрофизиологом Уорреном МакКаллоом и логиком-математиком Уолтером Питтсом была предложена модель нейрона и выдвинуты гипотезы о работе головного мозга.

Нейропсихолог и физиолог Дональд Олдинг Хебб в работе «The organization of Behavior» 1949 г. выдвинул идеи об объединениях нейронов в головном мозге и их взаимодействии. Им же были предложены правила обучения нейронной сети.

В 1957 году американский психолог, нейрофизиолог и разработчик искусственного интеллекта Фрэнк Розенблатт создал математическую модель искусственного нейронной сети – персептрон, воплощённую в виде компьютера «Марк-1».

В 1959 году нейробиологи Дэвид Хантер Ньюбел и Торстен Нильс Визель доказали, что в биологических нейронных сетях информация хранится и обрабатывается распределённым образом.

В 1960 году профессором Бернардом Уидроу и его студентом Маршианом Эдвардом Хоффом младшим была разработана упрощённая нейронная сеть Адалин на основе мемисторов. Адалин до сих пор является одним из компонентов систем обработки сигналов.

В 1969 году математиками Марвином Ли Минским и Сеймуром Пейпертом было доказано, что возможности персептронов являются ограниченными.

В СССР с 1970 по 1976 годы под патронажем военных ведомств проводились исследовательские работы в области персептронов.

В 1987 году начались разработки искусственного интеллекта в Японии и Европе.

В 1990 году ряд советских исследовательских организаций начинают исследования в области искусственного интеллекта. Среди них: Институт кибернетики имени Глушкова (Киев), Институт многопроцессорных вычислительных систем (Таганрог), Институт нейрокибернетики (Ростов-на-Дону) и другие.

Интерес научных и промышленных предприятий к разработкам нейронных сетей и к искусственному интеллекту весьма высок. Каждый год выходят новые публикации, разрабатываются новые программы в области искусственного интеллекта, интерес к этой тематике продолжает шириться и развиваться.

## Строение искусственного нейрона

Нейронная сеть состоит множества нейронов каким-либо образом объединённых друг с другом. Наименьшим элементом нейронной сети является нейрон. На рисунке 1 показана структура нейрона.

Нейрон состоит из трёх элементов: синапсов (умножителей), сумматора и нелинейного преобразователя.

Рисунок1 – Пример содержимого файла высоты волн

На вход синапсов подаются некоторые значения , , …, , …, . Эти значения соответственно умножаются на некоторые весовые коэффициенты , , …, , …, . Значение – это некоторое значение сдвига. После чего полученные значения суммируются в сумматоре  и полученное значение  передаётся на вход некоторому нелинейному преобразователю . Функция  называется функцией активации или передаточной функцией нейрона. В математической записи нейрон можно записать в таком виде:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | , | (1) |
|  | | , | (2) |
| где | – вес синапса, ;  – значение смещения;  – результат суммирования;  – компонент входного вектора, ;  – выходной сигнал нейрона;  –число входов нейрона;  – нелинейное преобразование (функция активации). | | |

Значения , , …, , …,  подаваемые на вход нейрона из формулы (1) представляют некоторые оцифрованные характеристики исследуемого объекта. В каждом конкретном случае следует подбирать свой набор данных наиболее хорошо соответствующих задаче. Например, для распознавания предмета на изображении такими данными могут быть некоторые или все пиксели этого изображения.

Весовые коэффициенты , , …, , …,  служат для усиления или ослабления соответствующего значения . Переменную  можно рассматривать как ещё один весовой коэффициент  для ещё одного входа , причём  всегда равен .

Функция активации (2) должна зависеть от. В таблице 1 приведены некоторые наиболее широко используемые функции активации нейронов.

Функции активации не ограничиваются только теми функциями, что приведены в таблице 1. Окончательное решение о выборе функции активации остаётся за разработчиком искусственной нейронной сети.

|  |  |  |
| --- | --- | --- |
| Таблица 1 – Функции активации нейронов | | |
| Название | Формула | Область значений |
| Линейная |  |  |
| Полулинейная |  |  |
| Логистическая  (сигмоидальная) |  |  |
| Гиперболический тангенс (сигмоидальная) |  |  |
| Экспоненциальная |  |  |
| Синусоидальная |  |  |
| Сигмоидальная  (рациональная) |  |  |
| Шаговая  (линейная с насыщением) |  |  |
| Пороговая |  |  |
| Модульная |  |  |
| Знаковая  (сигнатурная) |  |  |
| Квадратичная |  |  |

При выборе функции активации необходимо учитывать такие факторы, как область решаемой задачи, вычислительная сложность задачи, соответствие функции активации поставленной задаче. Нередко функция активации подбирается методом проб и ошибок таким образом, пока результаты нейронной сети не будут удовлетворять тестовым наборам.

При всём многообразии функций активаций наиболее часто используется функция активации под названием сигмоида, она же логистическая функция

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | (3) |
| где | –некоторая постоянная. | | |

При уменьшении значения  сигмоида принимает всё более и более пологий вид, при пределе  сигмоида вырождается в горизонтальную линию на уровне , а при увеличении значения  сигмоида приближается к виду функции единичного скачка.

Из вида формулы (3) видно, что значение всего нейрона будет лежать в промежутке .

Кроме того, сигмоидальная функция является нелинейной, гладкой, непрерывно дифференцируемой на всём множестве вещественных чисел . Ещё одно удобство сигмоиды – это возможность простого вычисления производной через саму же функцию  и постоянную. Покажем это.

Возьмём производную от:

|  |  |
| --- | --- |
|  | (4) |

Можно выполнить следующее преобразование:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Учитывая формулу (3) выясняем, что формулу (4) можно записать таким образом:

|  |  |  |
| --- | --- | --- |
|  |  |  |

Помимо простого выражения производной у сигмоиды есть свойство хорошего усиливания слабых сигналов и плохого усиливания сильных сигналов [14].

## Виды искусственных нейронов

Нейроны, из которых состоит нейронная сеть, по своему назначению делятся на три вида:

- входные нейроны, которые непосредственно получают вектор сигнала из внешней среды. Обычно в таких нейронах не выполняются какие-либо вычисления и полученные значения передаются к последующим нейронам;

- выходные нейроны, численные результаты от этих нейронов представляют собой результаты работы нейронной сети;

- промежуточные нейроны, эти нейроны собственно представляют основной вычислительный механизм нейронной сети.

Работа выходных и промежуточных нейронов описывается формулами (1) и (2) [14].

## Строение искусственной нейронной сети

Работа нейронной сети во многом обуславливается тем, как нейроны, участвующие в вычислительном процессе взаимодействуют друг с другом.

В целом выделяют три вида связей нейронов:

- полносвязные (рис. 2);

- многослойные или слоистые (рис. 3);

- слабосвяные (рис. 4);

Рисунок2 – Схематическое изображение полносвязной нейронной сети

Рисунок3 – Схематическое изображение полносвязной нейронной сети

Рисунок4 – Схематическое изображение полносвязной нейронной сети

В полносвязных нейронных сетях каждый нейрон передаёт свой выходной сигнал остальным нейронам, в том числе и самому себе. Все входные значения передаются всем нейронам. В таких нейронных сетях выходными сигналами могут быть все или некоторые сигналы нейронов спустя некоторое количество тактов работы сети.

В многослойных нейронных сетях нейроны составляются в несколько слоёв. Слой состоит из некоторого множества нейронов с едиными входными сигналами. Количество нейронов в слое может быть произвольным и никак не зависит от количества нейронов в других слоях. В целом сеть состоит из  слоёв. Внешние входные сигналы подаются на входы нейронов входного слоя, а выходами сети являются сигналы последнего слоя. Помимо входного и выходного слоёв в многослойных нейронных сетях существуют один или несколько скрытых слоёв.

Многослойные нейронные сети сами делятся на несколько разновидностей.

1. Монотонные многослойные нейронные сети.

В таких сетях существуют дополнительные условия на связи и нейроны. В частности, это разбиение каждого слоя кроме выходного на два блока: возбуждающий и тормозящий. Связи между блоками также разделяются на тормозящие и возбуждающие. Если от нейронов некоторого блока к нейронам другого блока ведут только возбуждающие связи, то это значит, что любой выходной сигнал этого блока представляет собой монотонную неубывающую функцию любого выходного сигнала из первого блока. В случае с тормозящими сигналами любой выходной сигнал из второго блока является невозрастающей функцией любого сигнала из первого блока.

1. Сети без обратных связей.

В подобных сетях сигналы передаются от входных нейронов к скрытым слоям без какого-либо обратного влияния на предыдущие нейроны.

1. Сети с обратными связями.

В таких сетях сигналы с последующих слоёв передаются на предыдущие. Сети с обратными связями подразделяются на следующие:

- слоисто-циклические. В таких сетях слои объединены в кольцо. Сигнал от последнего слоя идёт на вход первого слоя. Все слои имеют одинаковые права и могут как получать входные сигналы, так и выдавать выходные;

- слоисто-полносвязные сети представляют собою слои, являющиеся полносвязными сетями, сигналы передаются как от слоя к слою, так и внутри слоя. Работа таких сетей состоит из трёх частей: приём сигнала от предыдущего слоя, обмен сигналами внутри слоя, выработка выходного сигнала и передача сигналов к последующему слою.

- полносвязно-слоистые. Такие сети похожи на слоисто-полносвязные сети, но работают по-другому. В этих полносвязно-слоистых сетях не разделяются фазы обмена внутри слоя и передачи следующему, на каждом такте нейроны всех слоёв принимают сигналы как от нейронов своего слоя, так и с последующих.

После того, как исследователем было решено какой разновидностью сети следует пользоваться переходят к реализации обучения нейронной сети [16].

## Обучение искусственной нейронной сети

Обучение нейронной сети условно делят на обучение с учителем и на обучение без учителя.

В первом случае формируют наборы данных, для которых заранее вычислено требуемое значение, которое должна возвращать нейронная сеть. После чего в случайном порядке подают эти наборы на вход нейронной сети. Полученное значение от нейронной сети сравнивается с эталонным значением и в случае, когда сеть допускает слишком большую ошибку корректируются весовые коэффициенты сети. По мере того, как снижается погрешность сети, также снижается корректирование весовых коэффициентов. Когда будет достигнут приемлемый уровень погрешности процесс обучения останавливают иначе существует риск перетренировки сети. Это явление выражается в том, что сеть просто «вызубривает» эталонные образцы данных, но не может корректно работать на реальных данных.

В обучении без учителя для нейронной сети определяются некоторые ориентировочные особенности, которые она должна достигнуть. И с каждой итерацией работы сеть самостоятельно определяет погрешность и вносит коррективы в весовые коэффициенты.

Для корректирования весовых коэффициентов используется алгоритм обратного распространения ошибки. В этом алгоритме минимизируется среднеквадратичное отклонение требуемых данных.

Минимизируемой функцией является величина

|  |  |  |  |
| --- | --- | --- | --- |
|  | | , |  |
| где | ‑ реальное выходное состояние нейрона-го выходного слоя нейронной сети при подаче на её входы -го образца,  ‑ требуемое выходное состояние этого нейрона. | | |

При этом суммируются значения для всех нейронов оконечного слоя и для всех тестовых данных, обрабатываемых сетью. При использовании метода градиентного спуска настройка весовых коэффициентов выполняется с помощью формулы

|  |  |  |  |
| --- | --- | --- | --- |
|  | | , | (5) |
| где | ‑ весовой коэффициент связи соединения -го нейрона слоя  с нейроном,  ‑ коэффициент скорости обучения,. | | |

С учётом правил дифференцирования сложных функций выражение  можно записать таким образом:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | (6) |
| где | ‑ сумма входных сигналов нейрона  (аргумент активационной функции). | | |

Поскольку производная активационной функции должна быть определена на всей оси абсцисс, то активационные функции с неоднородностями (пример: функция единичного скачка) не соответствуют требуемым условиям. Удовлетворяющими требованиям (6) являются такие функции как гиперболический тангенс или сигмоида (таблица 1).

Первый множитель из формулы (5) можно записать следующим образом:

|  |  |  |
| --- | --- | --- |
|  | . | (7) |

В формуле (7) осуществляется суммирование по переменной  по нейронам слоя .

Введём такую переменную:

|  |  |  |
| --- | --- | --- |
|  | , |  |

получаем рекурсивную формулу для расчётов величин  слоя  из величин более старшего слоя . То есть,

|  |  |  |
| --- | --- | --- |
|  | . | (8) |

Для выходного слоя:

|  |  |  |
| --- | --- | --- |
|  |  | (9) |

Тогда формула (5) преобразовывается так:

|  |  |  |
| --- | --- | --- |
|  |  | (10) |

При использовании метода градиентного спуска, есть вероятность попадания в локальные минимумы минимизируемой функции. Для того чтобы суметь выбраться из локальных минимумов и дойти до глобального минимума используют так называемый «коэффициент инерционности». В этом случае формула (10) модифицируется:

|  |  |  |  |
| --- | --- | --- | --- |
|  | | , | (11) |
| где | ‑ коэффициент инерционности;  ‑ номер текущей итерации. | | |

## Процедура обучения методом обратного распространения ошибки

Обучение искусственной нейронной сети методом обратного распространения ошибки выполняется по следующим шага.

Шаг 1. На входы нейронной сети подаётся случайным образом выбранный образ и в режиме обычного функционирования нейронной сети вычисляется результат.

Шаг 2. По формуле (9) рассчитывается погрешность . По формулам (10) или (9) вычисляются изменения весов  для слоя .

Шаг 3. По формулам (8) и (10) (или (8) и (11)) вычисляются  и  для всех предыдущих слоёв .

Шаг 4. Выполняется корректировка всех весов нейронной сети:

|  |  |  |
| --- | --- | --- |
|  | . | (12) |

Шаг 5. Если ошибка сети не удовлетворяет выставленным требованиям, то переходят на шаг 1.

# 2. Специальный раздел

## 2.1. Расчёт искусственной нейронной сети, реализующей операцию XOR

Бинарная логическая операция XOR, также называемая как «исключающее или» или «исключающая дизъюнкция» представляет операцию, возвращающую истинное значение только тогда, когда отличаются её входные данные (одно значение истинное, другое значение ложное). Истинное и ложное значения часто обозначаются как  и  соответственно. В таблице 2 приведена таблица истинности операции XOR.

|  |  |  |
| --- | --- | --- |
| Таблица 2 – Таблица истинности операции XOR | | |
| Входные значения | | Выходное значение |
|  |  |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

Составим простейшую нейронную сеть из одного скрытого слоя с двумя нейронами, двух входных нейронов и одного выходного нейрона. Смотрите рисунок 5.

Для рисунка 5 приняты следующие обозначения

|  |  |
| --- | --- |
| где | , ‑ входные нейроны; |
|  | ,‑ нейроны скрытого слоя; |
|  | ‑ выходной нейрон |
|  | , , , , , ‑ весовые коэффициенты нейронных связей. |













Рисунок 5 – Пример нейронной сети реализующей операцию XOR

Пусть при этом будут даны следующие значения:

|  |  |  |
| --- | --- | --- |
|  | ;  ;  ;  ;  ;  ;  ;  . |  |

Рассчитаем вывод всей сети. Для этого рассчитаем суммы, поступающие на вход нейронов  и . Суммы будем обозначать как  и , соответственно:

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | . |  |

После подсчёта сумм необходимо посчитать выходные значения от нейронов и :

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | . |  |

Теперь подсчитаем сумму  для выходного нейрона :

|  |  |  |
| --- | --- | --- |
|  | . |  |

Найдём результирующее значение для всей сети :

|  |  |  |
| --- | --- | --- |
|  | . |  |

Но сеть, при заданных значениях  и  должна была вернуть , следовательно, имеет место ошибка. Найдём квадрат среднеквадратической ошибки ()по формуле:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  |  |
| где | ‑ ожидаемое значение. | | |

То есть, ошибка составляет около .

Очевидно, что такое положение неудовлетворительно. Поэтому нейронную сеть необходимо тренировать для улучшения результатов. Тренировка будет осуществляться с помощью полученного результата  через алгоритм обратного распространения ошибки описанный в разделе 1.6.

Найдём значение  для выходного нейрона по формуле (9):

|  |  |  |  |
| --- | --- | --- | --- |
|  | | ; |  |
| где | . | |  |

Тогда получаем:

|  |  |  |
| --- | --- | --- |
|  | ; |  |

Найдём  для нейронов  и  по формуле(8):

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | ; |  |

После нахождения  следует вычислить приращения для соответствующих нейронов по формуле (11). Для улучшения читаемости перепишем формулу (11) в таком виде:

|  |  |  |  |
| --- | --- | --- | --- |
|  | |  | (13) |
| где | ‑ коэффициент обучения (подбирается отдельно в каждом случае, здесь взято );  ‑ коэффициент момента (также подбирается отдельно для различных задач, здесь взято );  ‑значение приращения на предыдущей итерации вычисления. Значение  принимают равным нулю. | | |

Отсюда получаем следующие значения

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | ; |  |

После вычисления приращений для весов их следует сложить с соответствующими весами в соответствии с формулой (12), тем самым получив новые, улучшенные значения весов:

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | ; |  |

Поскольку во входных нейронах  и  не выполняется никаких изменений входных данных, то для  и  надо рассчитать лишь приращения весовых коэффициентов:

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | ; |  |
|  | ; |  |
|  | . |  |

После чего вычислим новые значения весовых коэффициентов , ,  и :

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | ; |  |
|  | ; |  |
|  | . |  |

Приведём новые, полученные значения для искусственной нейронной сети:

|  |  |  |
| --- | --- | --- |
|  | ;  ;  ;  ;  ;  ;  ;  . |  |

Рассчитаем значения для  и :

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | . |  |

Рассчитаем выходные значения для нейронов и :

|  |  |  |
| --- | --- | --- |
|  | ; |  |
|  | . |  |

Рассчитаем сумму  для выходного нейрона :

|  |  |  |
| --- | --- | --- |
|  | . |  |

Найдём результирующее значение для всей сети :

|  |  |  |
| --- | --- | --- |
|  | . |  |

Найдём квадрат среднеквадратической ошибки ():

|  |  |  |
| --- | --- | --- |
|  |  |  |

То есть, ошибка теперь составляет.

Этот процесс требуется повторять до тех пор, пока не будет достигнут допустимый процент ошибки.

## 2.2. План расчёта погрешности при прогнозировании волн

В нейронной сети реализующей логическую операцию XORна вход подаётся только два аргумента. Но такой подход не представляется возможным для работы с прогнозированием волн. Причина этому ‑ гипотеза о том, что волны-убийцы возникают как результат фокусировки волновой энергии в некоторых структурах поверхностных течений [19]. Поэтому предсказания должны строиться исходя из как можно большего числа аргументов.

В этой работе было решено использовать нейронные сети, принимающие на вход десять аргументов.

Наборы данных для проверки работоспособности нейронной сети хранятся в текстовом файле «learnSample.txt». На рисунке 6 приведён пример одного из тестовых наборов данных.

Каждый набор тестовых данных состоит из десяти десятичных дробей служащих аргументами для нейронной сети, плюс одиннадцатое число служащее для сравнения с тем, что было получено в результате работы нейронной сети.

|  |  |  |
| --- | --- | --- |
|  | 0.645383759  0.744300892  0.722451211  0.175503314  0.541044312  0.443453347  0.870543496  0.8604689  0.493869107  0.911665387  2.369305436 |  |

Рисунок6 – Пример содержимого файла высоты волн

Эксперимент будет выполняться в следующем виде: поочерёдно будут созданы простые нейронные сети с разным числом слоёв  и разным числом нейронов в этих слоях .

Каждая созданная нейронная сеть будет натренирована на одинаковых наборах данных. После чего каждой нейронной сети будет передан весь набор данных и найдено среднее арифметическое абсолютных ошибок.

# 3. Технологический раздел

## 3.1. Описание программной реализации

Программа тестирования простой искусственной нейронной сети для данных, полученных с глубинного датчика была написана в универсальном редакторе Emacs на языке программирования C. При разработке программы использовалась библиотека для построения простых искусственных нейронных сетей Genann за авторством Lewis Van Winkle.

Программа служит для автоматизированной проверки нейронных сетей с различным количеством слоёв и различным количеством нейронов на работу с данными глубинных датчиков.

Программа использует данные из файла «learnSamples.txt» в качестве тренировочного набора для нейронной сети и по прошествии пятьсот циклов тренировок тестирует нейронную сеть на способность распознавать каждый из тестовых наборов. Всего для тестирования предусмотрено двадцать тестовых наборов.

Пример одного из тестовых наборов представлен на рисунке 7.

Файл «learnSamples.txt»является текстовым файлом в каждой строке которого задана высота волны в метрах.

Программа считывает данные из файла «learnSamples.txt» и располагает первые десять значений в массив input. Следующее одно значение программа помещает в массив output.Данный цикл повторяется до тех пор, пока не будут прочтены все двадцать тестовых наборов.

Для считывания данных из файла предназначена функция readFromFile. В этой функции значения, считанные из файла делятся на три. Причиной этому является то, что во всём наборе данных от глубинного датчика не было волн выше трёх метров. Кроме того, функция активации нейросети может возвращать значения только на промежутке .

После загрузки данных из файла в оперативную память начинается основной цикл создания, тренировки и тестирования нейронной сети.

Нейронные сети создаются с помощью переменных  и .

Переменная  обозначает количество скрытых слоёв в порождаемой нейронной сети. Переменная  обозначает количество нейронов в порождаемой нейронной сети.

Цикл начинается со следующих значений:  и . Конец цикла достигается при значениях  и .

Порождаемая нейронная сеть принимает на вход 10 аргументов и возвращает одно значение.

После создания нейронной сети следует цикл тренировки нейронной сети. Процесс тренировки длится пятьсот эпох. Во время каждой эпохи среди доступных двадцати образцов случайным образом выбирается один образец и инициируется процедура тренировки.

По окончании тренировки программа переходит к проверке натренированной сети. Для этого в цикле последовательно выбирается один за другим все тренировочные образцы и подаются на вход нейросети. Полученный результат от нейросети сравнивается с эталонным и находится модуль абсолютной ошибки. Все найденные модули абсолютных ошибок суммируются и вычисляется среднее арифметическое.

Полученные результаты вместе с данными о количестве слоёв и нейронов выводятся на стандартный вывод.

После чего выполняется очистка промежуточных переменных и объектов, и осуществляется переход к следующему витку цикла.

По окончании всех шагов цикла программа завершает свою работу.

Некоторые результаты работы программы с малыми модулями абсолютных ошибок показаны ниже:

Networks test result:

Layers: 1, neurons: 1, absolute error: 0.10

Layers: 1, neurons: 2, absolute error: 0.05

Layers: 1, neurons: 3, absolute error: 0.15

Layers: 1, neurons: 4, absolute error: 0.10

Layers: 1, neurons: 5, absolute error: 0.50

Layers: 1, neurons: 6, absolute error: 0.05

Layers: 2, neurons: 5, absolute error: 0.45

Layers: 2, neurons: 6, absolute error: 0.10

Layers: 2, neurons: 7, absolute error: 0.25

Layers: 2, neurons: 8, absolute error: 0.05

Layers: 2, neurons: 9, absolute error: 0.45

Layers: 2, neurons: 16, absolute error: 0.10

Layers: 2, neurons: 18, absolute error: 0.05

Layers: 2, neurons: 19, absolute error: 0.10

Layers: 3, neurons: 13, absolute error: 0.10

Layers: 3, neurons: 14, absolute error: 0.10

Layers: 3, neurons: 16, absolute error: 0.05

Layers: 3, neurons: 17, absolute error: 0.10

Layers: 3, neurons: 19, absolute error: 0.10

Layers: 3, neurons: 20, absolute error: 0.10

Layers: 4, neurons: 2, absolute error: 0.10

Layers: 4, neurons: 3, absolute error: 0.05

Layers: 4, neurons: 4, absolute error: 0.10

Layers: 4, neurons: 5, absolute error: 0.05

Layers: 4, neurons: 7, absolute error: 0.10

Layers: 4, neurons: 8, absolute error: 0.10

Layers: 4, neurons: 9, absolute error: 0.05

Layers: 4, neurons: 11, absolute error: 0.10

Layers: 5, neurons: 9, absolute error: 0.05

Layers: 5, neurons: 10, absolute error: 0.10

Layers: 5, neurons: 11, absolute error: 0.20

Layers: 5, neurons: 12, absolute error: 0.10

Layers: 5, neurons: 13, absolute error: 0.10

Layers: 6, neurons: 7, absolute error: 0.30

Layers: 6, neurons: 8, absolute error: 0.05

Layers: 7, neurons: 15, absolute error: 0.05

Layers: 9, neurons: 5, absolute error: 0.05

Layers: 11, neurons: 1, absolute error: 0.05

Layers: 11, neurons: 2, absolute error: 0.30

Layers: 11, neurons: 3, absolute error: 0.15

Layers: 11, neurons: 4, absolute error: 0.35

Layers: 11, neurons: 5, absolute error: 0.10

Layers: 11, neurons: 6, absolute error: 0.30

Layers: 11, neurons: 7, absolute error: 0.10

Layers: 11, neurons: 8, absolute error: 0.30

Layers: 11, neurons: 9, absolute error: 0.05

Layers: 11, neurons: 10, absolute error: 0.15

Layers: 11, neurons: 11, absolute error: 0.25

Layers: 11, neurons: 12, absolute error: 0.10

Layers: 11, neurons: 13, absolute error: 0.30

Layers: 11, neurons: 14, absolute error: 0.10

Layers: 12, neurons: 1, absolute error: 0.10

Layers: 12, neurons: 2, absolute error: 0.10

Layers: 12, neurons: 3, absolute error: 0.30

Layers: 12, neurons: 4, absolute error: 0.50

Layers: 12, neurons: 5, absolute error: 0.10

Layers: 12, neurons: 17, absolute error: 0.05

Layers: 15, neurons: 16, absolute error: 0.10

Layers: 15, neurons: 17, absolute error: 0.05

Layers: 15, neurons: 18, absolute error: 0.25

Layers: 15, neurons: 19, absolute error: 0.05

Layers: 17, neurons: 20, absoluteerror: 0.05

# 4. Экономический раздел

## 4.1. Экономические перспективы прогнозирования волн-убийц

Возможность прогнозирования волн-убийц представляет не только научный, но и финансовый интерес. Не секрет, что постройка и содержание морских и океанских судов стоит очень дорого. Например, по данным за 2005 год постройка нового балкера «Capesize» обходится заказчикам примерно в 68 миллионов долларов США[18].

Вместе с тем немало кораблей оказываются потерянными по тем или иным причинам.

Из 337 построенных сухогрузов «Волго-Дон макс» потерянными в различных авариях оказались 11 штук (из них 4 штуки погибли из-за перелома корпуса, 6 кораблей оказались затопленными).

Из 562 «Трёхтысячных» сухогрузов всего потеряно 33 корабля (из них 4 – из-за перелома корпуса, 18 кораблей затонуло) [13].

Чрезвычайные происшествия на морях и океанах наносят не только экономические, но и ни с чем не сравнимые человеческие потери.

Всё это в совокупности делает задачу исследования волн-убийц важной.

# ЗАКЛЮЧЕНИЕ

Нейронные сети являются мощным инструментом для решения различных задач в области искусственного интеллекта. Примерами таких задач могут служить задачи распознавания образов, попытки предсказания последующих событий на основе предыдущих явлений, классификация объектов и многое другое.

Несмотря на то, что нейронные сети с успехом используются во многих областях науки, техники и производства, эта область всё ещё остаётся недостаточно исследованной в связи с большим количеством неизвестных, входящих в формализацию той или иной задачи.

За время экспериментов с нейронными сетями удалось выработать определённые алгоритмы, позволяющие добиться значительных успехов. Но несмотря на это всё ещё нет обобщающего математического доказательства сходимости нейронной сети к требуемому результату. В этой области всё ещё есть множество белых пятен требующих исследований.

Проведённая работа показала, что точность простых нейронных сетей в вопросах прогнозирования волн-убийц оказывается недостаточной. Требуются дальнейшие изыскания по тематике волн-убийц.

Для получения новых результатов нужно собрать больше статистических данных. В этой работе был изучен только одномерный набор данных. Один из возможных способов улучшения наборов данных – это установление датчиков волн на как можно большей площади. Такие меры позволят улучшить картину природы волн, что в последующем может благоприятно сказаться на последующих исследованиях.

Список используемых источников

1. Avigad Jeremy. Interactive Theorem Proving ‑ Department of Philosophy and Department of Mathematical Sciences Carnegie Mellon University, 2011. ‑ 66 с.
2. Baker Hughes Rig Count. Rig Count Overview & Summary Count. Электронный ресурс. Дата посещения: 20.05.2019. URL: phx.corporate-ir.net/phoenix.zhtml?c=79687&p=irol-rigcountsoverview
3. de Moura Leonardo. Satisfiability at Microsoft. Formal Methods for Industrial Critical Systems ‑ Trento, Italy: 16th International Workshop, August 2011. ‑ 270 с.
4. Ferro Luca и др. Runtime Verification of Typical Requirements for a Space Critical SoC Platform. Formal Methods for Industrial Critical Systems ‑ Trento, Italy: 16th International Workshop, August 2011. ‑ 270 с.
5. Haver Sverre. A Possible Freak Wave Event Measured at the Draupner Jacket January 1 1995 – Marine Structures and Risers, Statoil ASA, N-4035 Stavanger, Norway, 2004. – 8 с.
6. Hutchins John. Readings in machine translation ‑ Cambridge, Mass.: The MIT Press, 2003. ‑ 7 с.
7. Katoen Joost-Pieter. Towards Trustworthy Aerospace Systems: An Experience Report. Formal Methods for Industrial Critical Systems ‑ Trento, Italy: 16th International Workshop, August 2011. ‑ 270 с.
8. Mashiyat Ahmed Shah и др. Modeling and Verifying Timed Compensable Workflows and an Application to Health Care. Formal Methods for Industrial Critical Systems ‑ Trento, Italy: 16th International Workshop, August 2011. ‑ 270 с.
9. Rosenthal W. и др. Detection of Extreme Single Waves and Wave Statistics – MAXWAVE Final Meeting, Geneva, Switzerland, October 8-10, 2003. – 12 с.
10. Stump Aaron. Verified Functional Programming in Agda - The Association for Computing Machinery and Morgan & Claypool Publishers, 2016. ‑ 259 с.
11. Апресян Ю.Д., Богуславский И.М., Иомдин Л.Л. и др. Лингвистическое обеспечение в системе автоматического перевода третьего поколения. – М.: Научный Совет по комплексной проблеме "Кибернетика" при Президиуме АН СССР, 1978.
12. Веденеева А. Рубли сошли со стапелей. Газета «Коммерсантъ» №53 от 27.03.2019, стр. 8.
13. Егоров Г.В., Егоров А.Г. Фактическое списание судов смешанного река-море плавания и прогноз утилизации судов до 2025 года. Типы судов, востребованные рынком. Журнал "Морская биржа" №3(61), 2017.
14. Круглов В.В., Борисов В.В. Искусственные нейронные сети. Теория и практика. – 2-е изд., стереотип. – М.: Горячая линия – Телеком, 2002. – 382 с.
15. Михайлов А.И., Черный А.И., Гиляревский Р.С. Основы информатики. – М.: Наука, 1968., Леонов Б.П. О методах автоматического реферирования (США 1958-1974 гг.)// Научно-техническая информация, сер.2. – 1975. – №6. – С. 16-20.
16. Пащенко Н.А., Кнорина Л.В., Молчанова Т.В. и др. Проблемы автоматизации индексирования и реферирования//Итоги науки и техники. Сер. Информатика. – М.: ВИНИТИ, 1983. – Т.7. – С. 7-164., Севбо И.П. Структура связного текста и автоматизации реферирования. – М.: Наука, 1969.
17. Поспелов Д.А. Из истории искусственного интеллекта ‑ Новости искусственного интеллекта, №4, 1994. ‑ с. 70-90.
18. Романенко А. Цены на суда. Журнал "Морская биржа" №2(12), 2005.
19. Шамин Р.В. Математические вопросы волн-убийц – М.: URSS, 2016. – 168 c.

Приложение А. Листинг программного кода

#include<stdio.h>

#include <stdlib.h>

#include <time.h>

#include "genann.h"

void readFromFile (const char \*, double [][10], double []);

int main(int argc, char \*argv[]){

double input[20][10];

double output[20];

srand(time(0));

readFromFile("learnSamples.txt", input, output);

int l,n,k,i;

double tmp;

printf("Networks test result:\n");

for (l = 1; l < 21; l++)

for (n = 1; n < 21; n++) {

genann \* ann = genann\_init(10, l, n, 1);

// Training network

for (i = 0; i < 500; ++i) {

k = (double) rand() / RAND\_MAX \* 19;

genann\_train(ann, input[k], output + k, 3);

}

// Calculating sum from network

for (i = 0; i < 20; i++) {

tmp += abs (((\* genann\_run(ann, input[i])) - output[i]) \* 3) ;

}

tmp /= 20;

// Print results

printf("Layers: %d, neurons: %d, absolute error: %4.2f\n", l, n, tmp);

// Clear objects

tmp = 0;

genann\_free(ann);

}

}

return 0;

// Load data from file

void readFromFile (const char \* filename, double inArr[][10], double outArr[]) {

FILE \*fp;

float waveHeight;

fp=fopen(filename, "r");

int i, j;

for (i=0; i<20; i++) {

for (j=0; j<10; j++) {

fscanf(fp, "%f", &waveHeight);

inArr[i][j] = waveHeight / 3;

}

fscanf(fp, "%f", &waveHeight);

outArr[i] = waveHeight / 3;

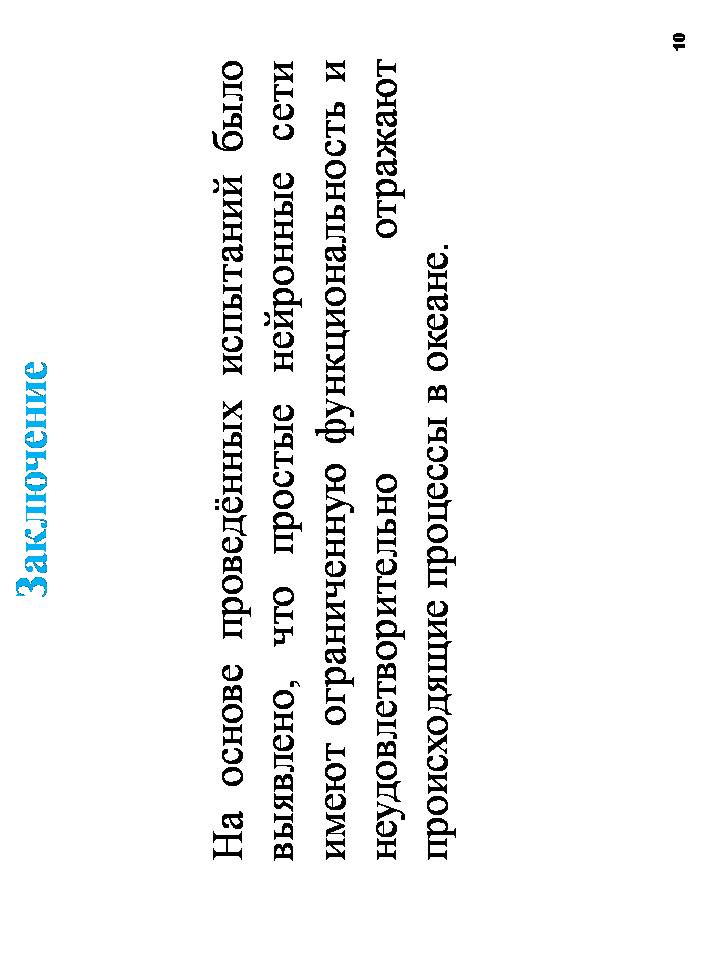
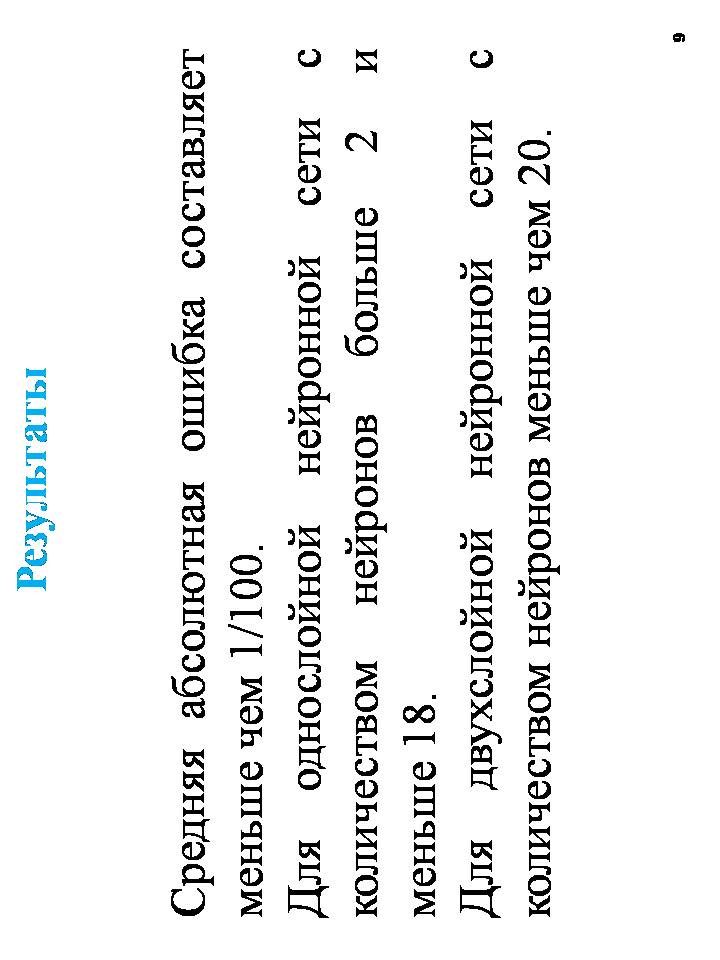
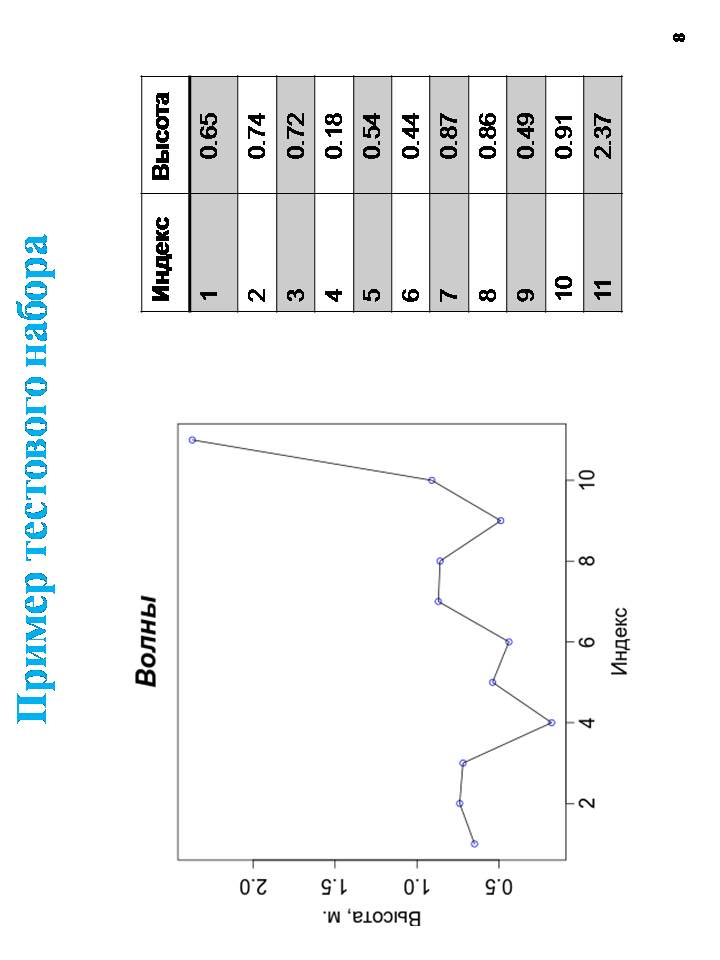
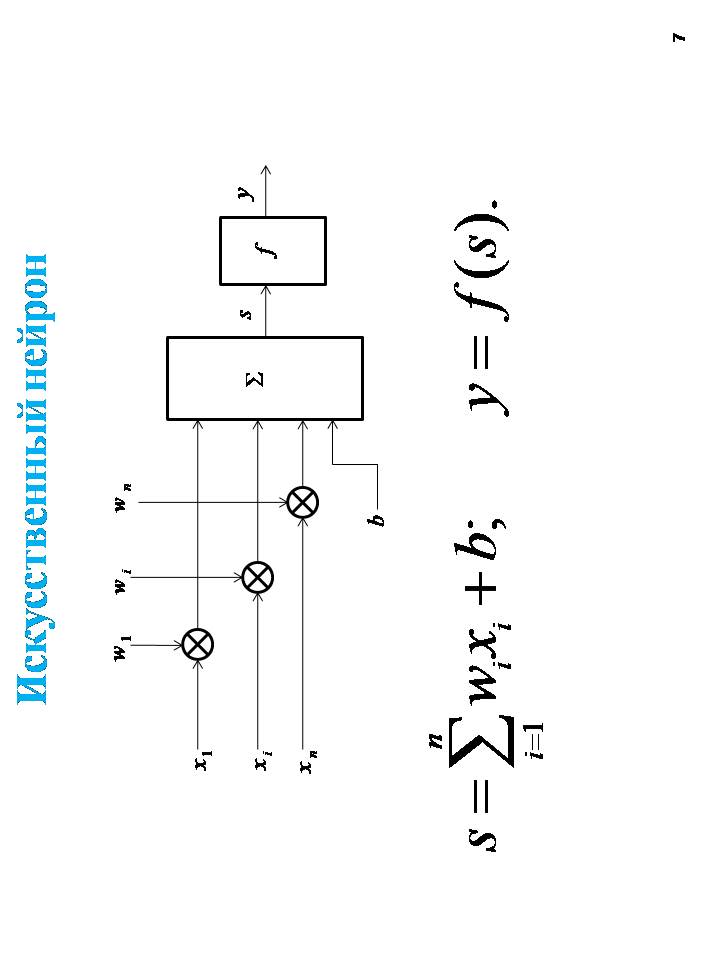
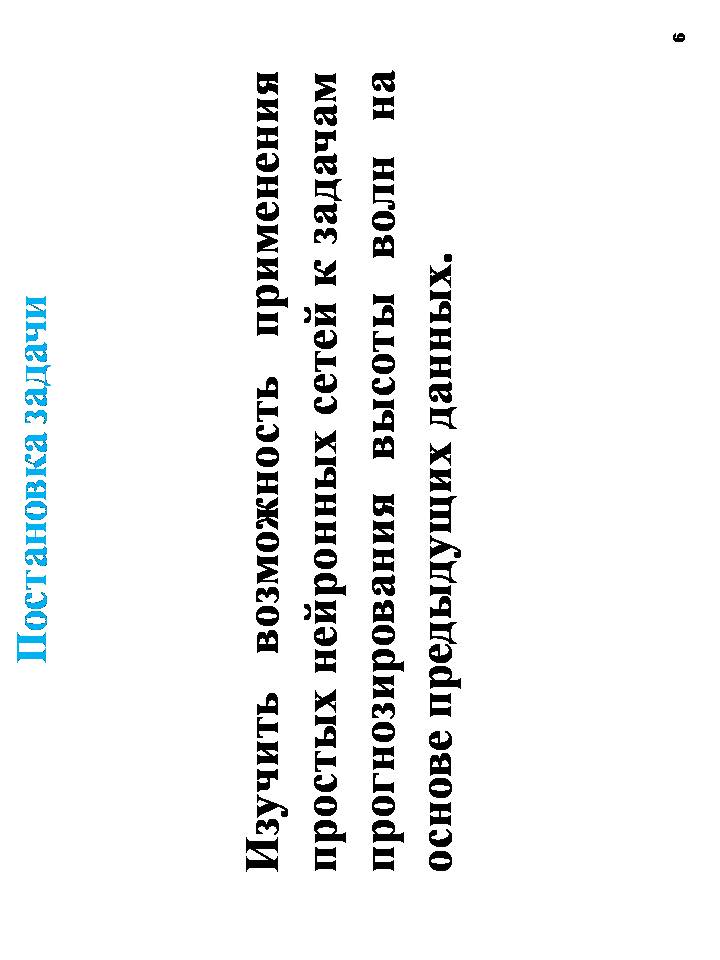
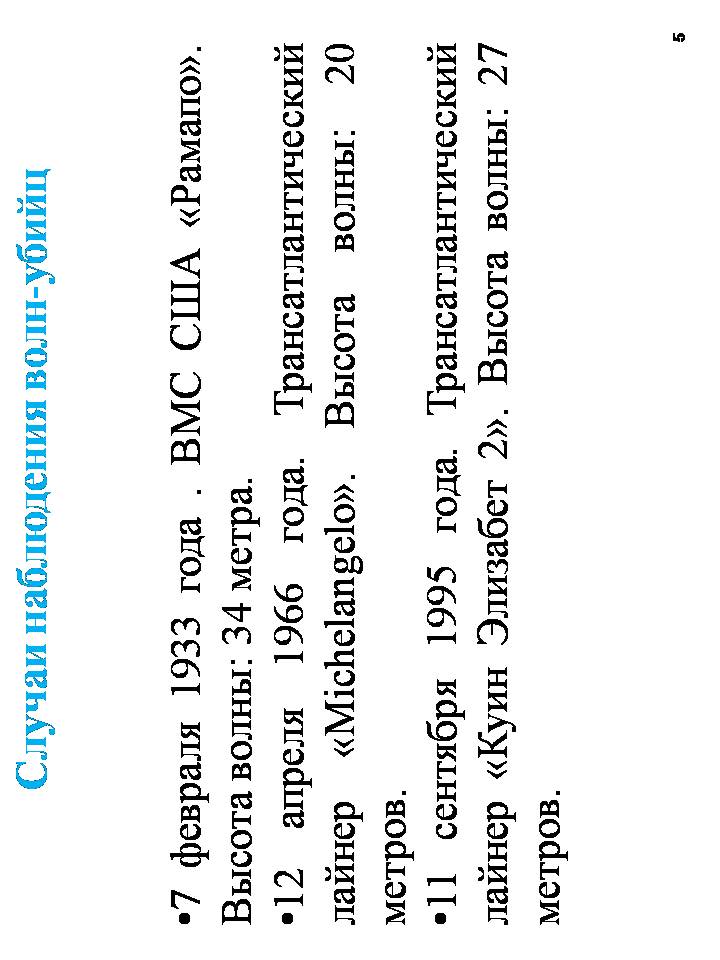
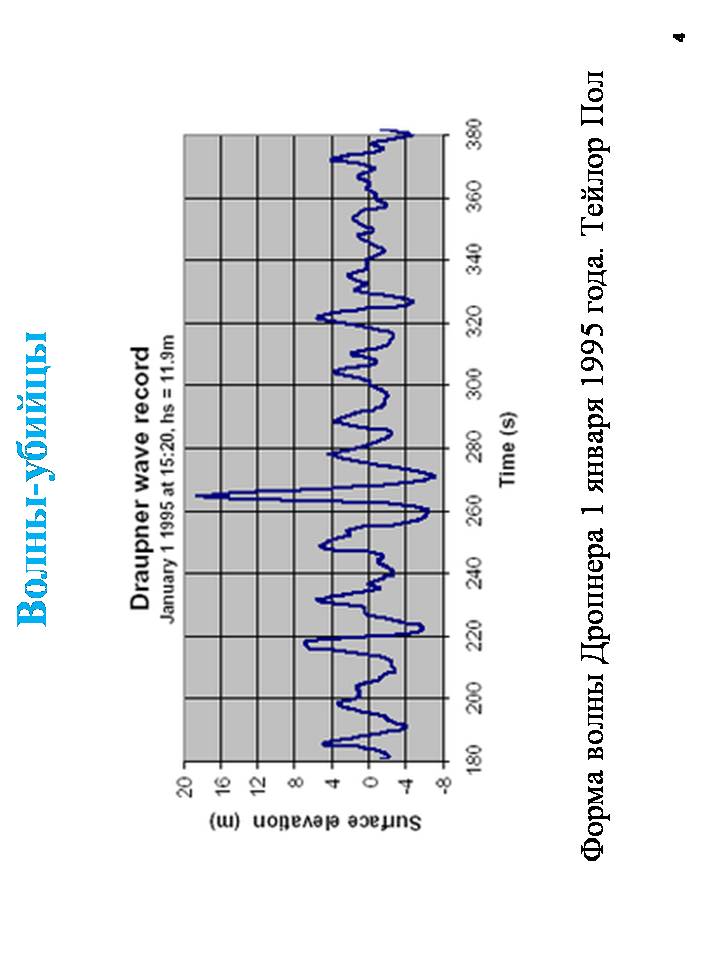
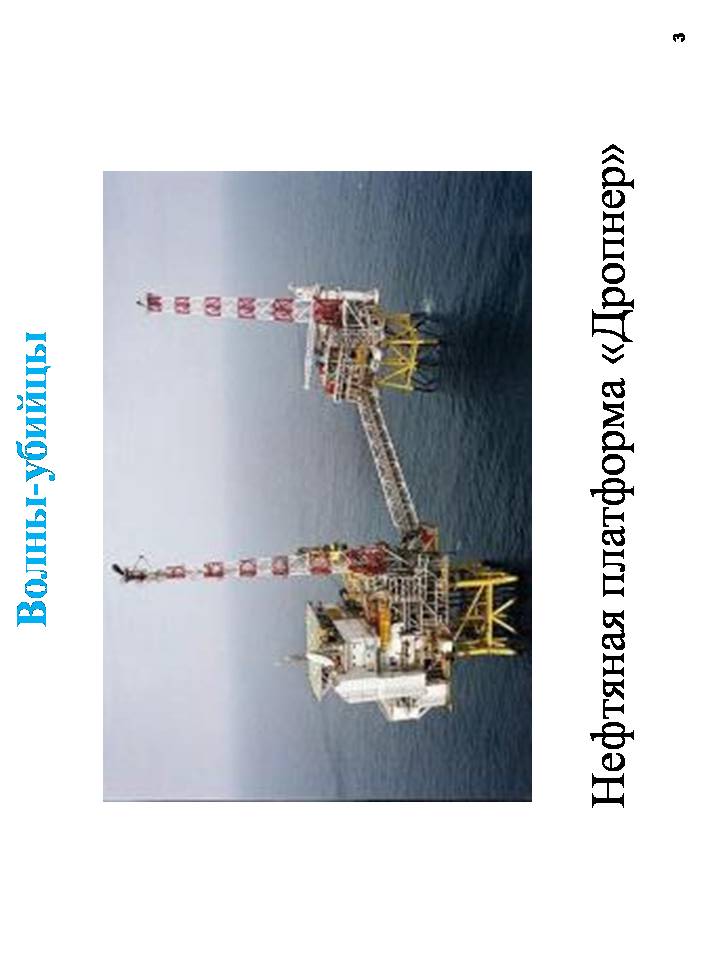
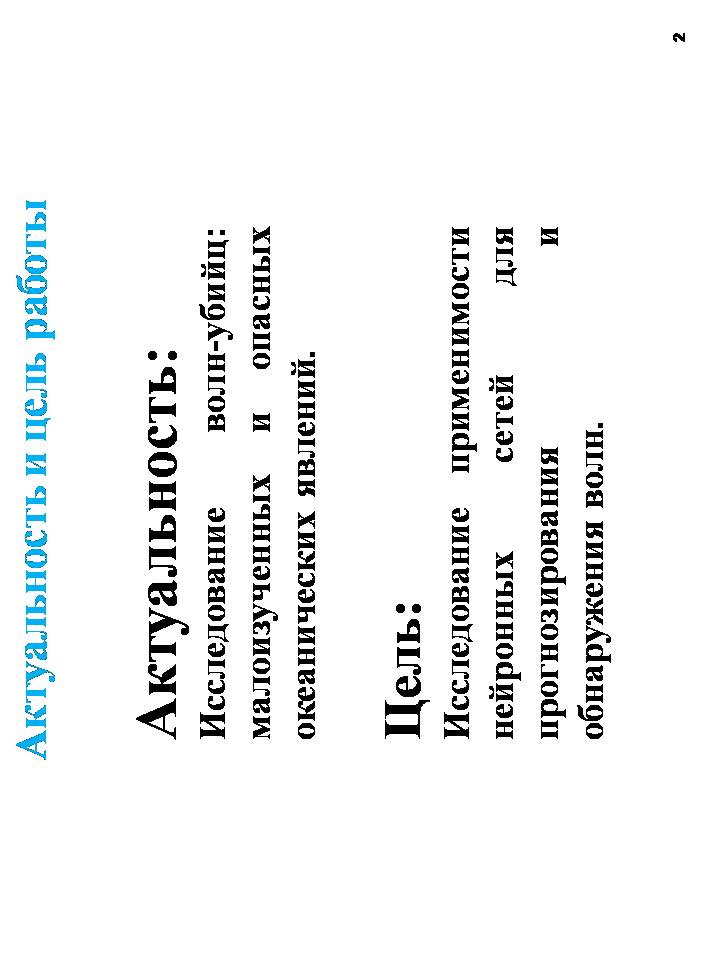
}

fclose(fp);

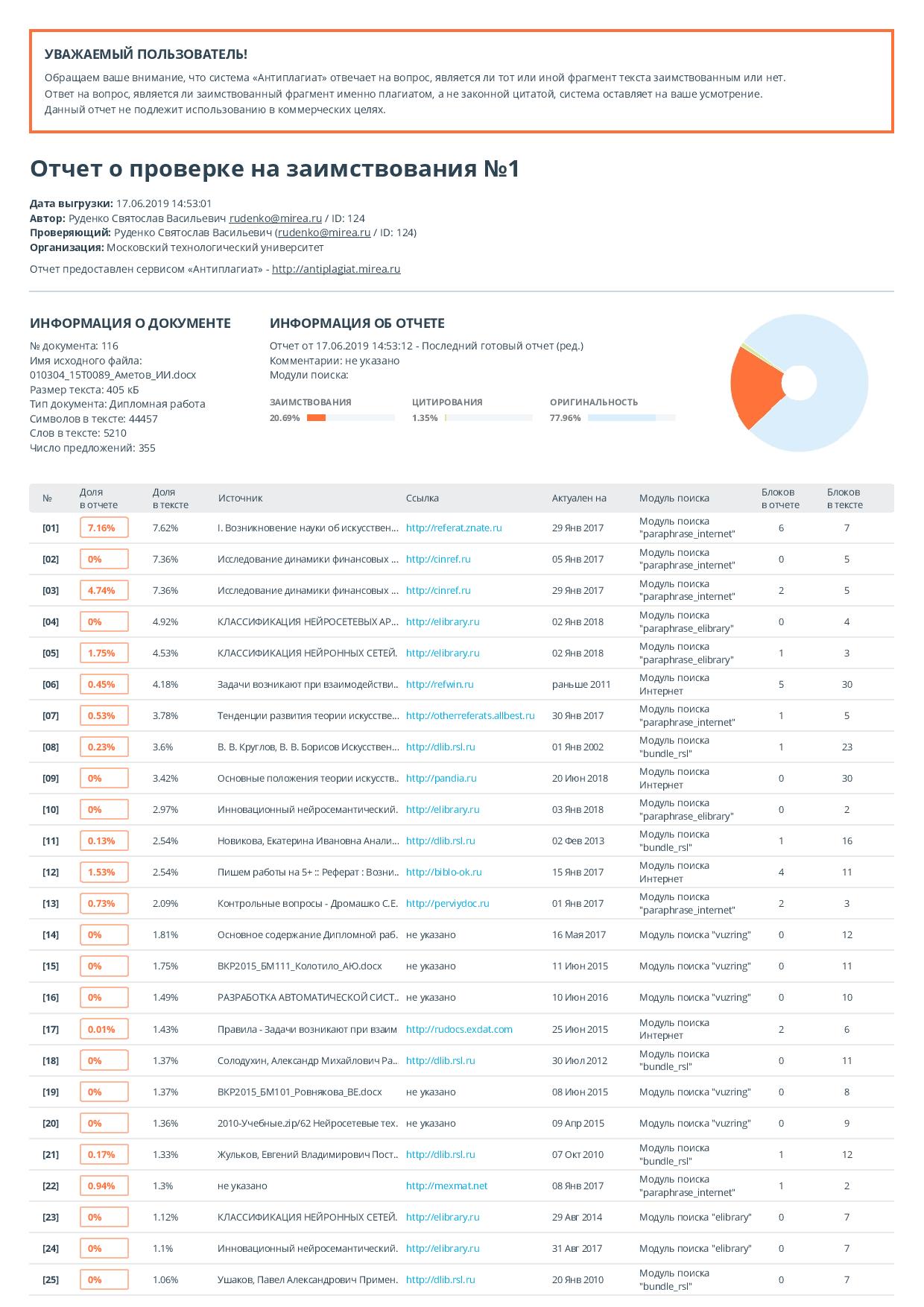
}

Приложение Б. Презентация



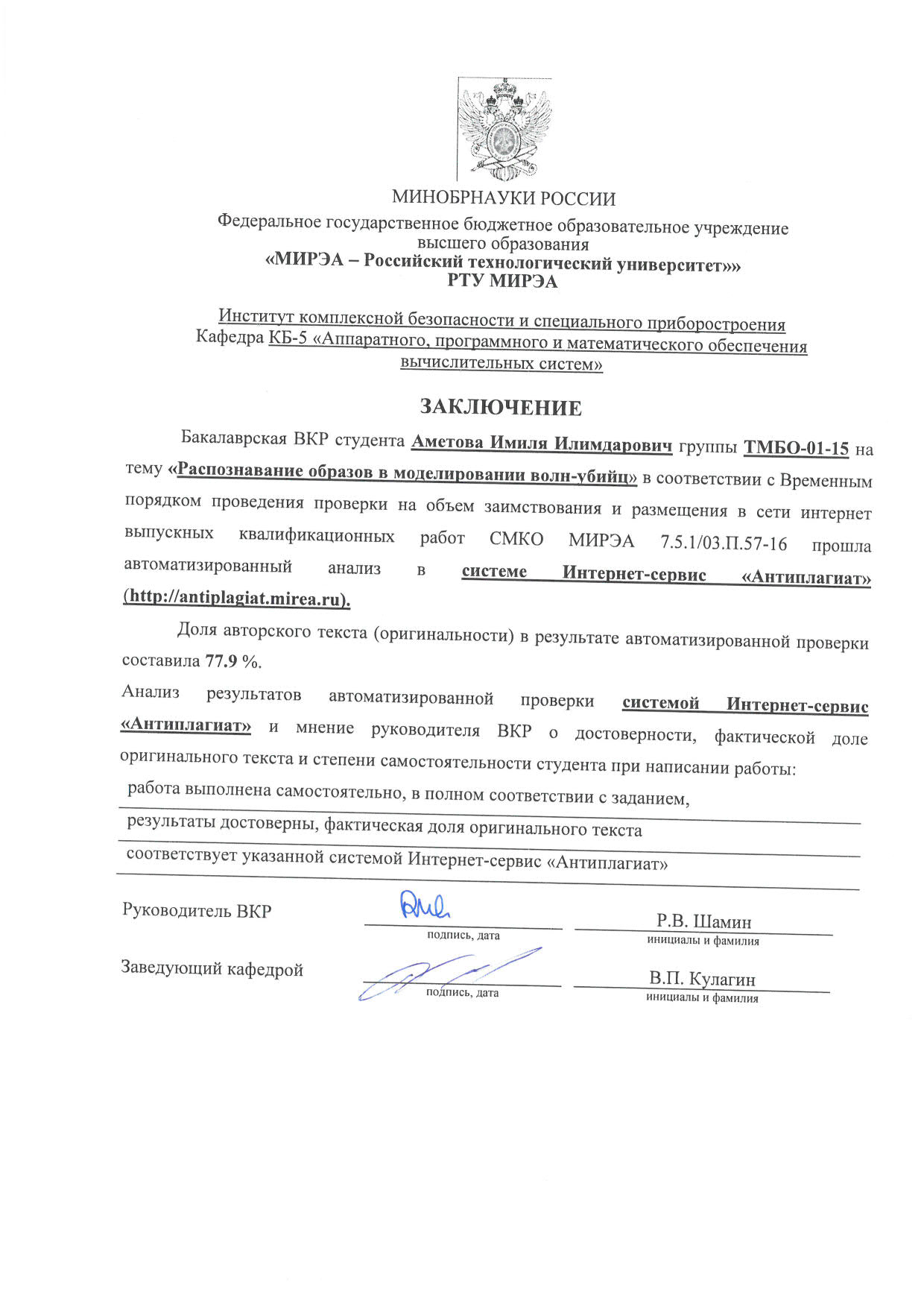


Приложение В. Результат проверки на оригинальность текста





Приложение Г. Заключение руководителя об оригинальности текста



Приложение Д. Отзыв научного руководителя

