МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Брянский государственный технический университет»

Факультет информационных технологий

Кафедра «Компьютерные технологии и системы»

**ДНЕВНИК**

**ПРОИЗВОДСТВЕННОЙ ПРАКТИКИ (ПРЕДДИПЛОМНОЙ) СТУДЕНТА**

Специальность: 10.05.04 "Информационно-аналитические системы безопасности"

Специализация: "Автоматизация информационно-аналитической деятельности"

Брянск 2022

Студент: Андронов Михаил Павлович

(ФИО)

Специальность: 10.05.04 "Информационно-аналитические системы безопасности", специализация: "Автоматизация информационно-аналитической деятельности" направляется на (в) ФГБОУ ВО «БГТУ», кафедра «КТС»

(предприятие, организация, фирма, компания)

1. **Календарные сроки практики**

По учебному плану начало 01.09.21 г. конец 22.12.21 г.

Дата прибытия на практику 01.09.21 г.

Дата прибытия с места практики 22.12.21 г.

1. **Руководитель практики от БГТУ**

Кафедра «Компьютерные технологии и системы».

Ученое звание к.т.н., доцент.

Фамилия Леонов

Имя Юрий.

Отчество Алексеевич

|  |  |  |
| --- | --- | --- |
| Дата | Описание работы, выполненной студентом | Отметка руководителя от базы практики |
| 1 | 2 | 3 |
| 01.09.2021 | Этап 1. Выступление руководителя практики.  Инструктаж. Знакомство с программой производственной практики, режимом работы, перечнем отчетной документации. | Выполнено |
| 7.09.2021 | Этап 2. Разработка архитектуры информационной системы | Выполнено |
| 10.09.2021 | Этап 3. Выбор средств программирования | Выполнено |
| 21.09.2021 | Этап 4. Проектирование базы данных | Выполнено |
| 21.11.2021 | Этап 5. Разработка пользовательского интерфейса для платформы Windows и Android. | Выполнено |
| 25.11.2021 | Этап 6. Тестирование программной системы. | Выполнено |
| 30.11.2021 | Этап 7. Оценка производительности системы. | Выполнено |
| 01.12.2021 | Этап 8. Интерпретация полученных результатов. | Выполнено |
| 15.12.2021 | Этап 9. Подготовка материалов для отчетного семинара, оформление отчета по практике. | Выполнено |
| 22.12.2021 | Этап 10. Выступление с отчетной документацией на итоговом семинаре, промежуточная аттестация  (дифференцированный зачет). | Выполнено |

Руководитель практики от университета \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_

*(Подпись) (ФИО)*

**График прохождения практики (календарно-тематический план)**

| № | Этапы | Выполняемая работа | Продолжительность |
| --- | --- | --- | --- |
| 1 | Выступление руководителя практики. Инструктаж. | Этап 1. Выступление руководителя практики.  Инструктаж. Знакомство с программой производственной практики, режимом работы, перечнем отчетной документации. | 1 |
| 2 | Разработка архитектуры | Этап 2. Разработка архитектуры информационной системы | 1 |
| 3 | Выбор средств программирования | Этап 3. Выбор средств программирования | 1 |
| 4 | Проектирование БД | Этап 4. Проектирование базы данных | 2 |
| 5 | Разработка пользовательского интерфейса | Этап 5. Разработка пользовательского интерфейса для платформы Android. | 8 |
| 6 | Тестирование | Этап 6. Тестирование программной системы. | 2 |
| 7 | Оценка производительности | Этап 7. Оценка производительности системы. | 2 |
| 8 | Интерпретация полученных результатов | Этап 8. Интерпретация полученных результатов. | 2 |
| 9 | Оформление отчета | Этап 9. Подготовка материалов для отчетного семинара, оформление отчета по практике. | 1 |
| 10 | Аудиторное представление отчета | Этап 10. Выступление с отчетной документацией на итоговом семинаре, промежуточная аттестация  (дифференцированный зачет). | 1 |

Руководитель практики от университета \_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_ *(Подпись) (ФИО)*

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

ФГБОУ ВО «Брянский государственный технический университет»

Факультет информационных технологий

Кафедра «Компьютерные технологии и системы»

О Т Ч Е Т

**по производственной практике (преддипломной)**

Специальность: 10.05.04 "Информационно-аналитические системы безопасности"

Специализация: "Автоматизация информационно-аналитической деятельности"

Выполнил

Студент группы О-17-ИАС-аид-С

Андронов М.П.

Руководитель практики от БГТУ

к.т.н., доц. Леонов Ю.А.

Оценка отчета\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

«\_\_\_\_» \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.

Брянск 2022

**Федеральное государственное бюджетное образовательное учреждение**

**высшего образования**

**«Брянский государственный технический университет»**

Кафедра «Компьютерные технологии и системы»

Специальность: 10.05.04 "Информационно-аналитические системы безопасности"

Специализация: "Автоматизация информационно-аналитической деятельности"

**ИНДИВИДУАЛЬНОЕ ЗАДАНИЕ**

**на производственную практику (преддипломную)**

для Андронова Михаила Павловича\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

*(ФИО обучающегося полностью)*

Обучающегося VI курса учебная группа № О-17-ИАС-аид-С \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Место прохождения практики: БГТУ, кафедра «КТС» \_\_\_\_\_\_\_\_\_\_\_\_\_\_  
адрес организации: г. Брянск \_\_

*(указывается полное наименование структурного подразделения / профильной организации, а также их фактический адрес)*

Срок прохождения практики с 01.09.21 по 22.12.21

**Цель прохождения практики:** получение теоретических и практических результатов предпроектного обследования профессиональной деятельности и информационных потребностей автоматизируемых подразделений предприятия.

**Задачи практики:**

* Разработать архитектуру информационной системы.
* Спроектировать базу данных.
* Разработать интерфейс для платформы Windows.
* Разработать интерфейс для платформы Android.
* Протестировать разработанную систему.
* Оценить производительность разработанной системы.

|  |  |
| --- | --- |
|  | УТВЕРЖДАЮ  *\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*  *Руководитель практики от университета*  *«\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г.* |

Задание принято к исполнению: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ «\_\_\_» \_\_\_\_\_\_\_\_\_\_ 2022 г.

*(подпись обучающегося)*

Содержание

[ВВЕДЕНИЕ 3](#_Toc122300542)

[1 Архитектура информационной системы 5](#_Toc122300543)

[2 Описание основных методов серверной части программного обеспечения 15](#_Toc122300544)

[3 Проектирование базы данных 17](#_Toc122300545)

[4 Выбор формализованных методов анализа 19](#_Toc122300546)

[5 Разработка пользовательского интерфейса 21](#_Toc122300547)

[6 Тестирование программной системы 24](#_Toc122300548)

[7 Оценка производительности системы 26](#_Toc122300549)

[8 Интерпретация полученных результатов 28](#_Toc122300550)

[ЗАКЛЮЧЕНИЕ 31](#_Toc122300551)

[СПИСОК ЛИТЕРАТУРЫ 32](#_Toc122300552)

[ПРИЛОЖЕНИЕ 40](#_Toc122300553)

[Приложение 1. Листинг 40](#_Toc122300554)

# ВВЕДЕНИЕ

В наше время фондовый рынок стремительно развивается. Инвесторам для эффективного вложения средств нужна оперативная информация о том, где купить акции, по какой стоимости, как котировки ценных бумаг (например, стоимость акций Газпрома, Сбербанка, ВТБ, Роснефти) меняются в течение торговой сессии. Также вопрос инвестиций в России становится всё более популярным для обычных граждан.

Понимание поведения инвесторов на финансовом рынке в целом и на фондовом рынке, в частности, всегда было и остается важной и актуальной проблемой для стабильности современной глобальной экономики. Существует немало примеров, вошедших в историю, когда неправильное представление или игнорирование тех или иных событий влекло к значительным убыткам не только для участников торгов, но и для компаний, акции которых обращаются на биржевом рынке, и даже для людей, которые никаким прямым образом не участвуют в данной системе. Примером может послужить «крах Уолл-стрит» 1929 года, ставший началом Великой депрессии. Другой пример: глобальный финансовый кризис 2007-2008 годов, который привел к спаду на рынке жилья, падению бизнеса и безработице.

**Целью** данной работы является моделирование информационно-аналитической системы анализа фондового рынка.

Для достижения данной цели необходимо решить ряд **задач**, таких как:

* анализ существующих систем анализа фондовых рынков;
* анализ источник данных, определение необходимых математических методов, наиболее подходящих для решения данной задачи;
* организация базы данных и заполнение её достаточным объемом информации;
* разработка аналитического модуля, позволяющего прогнозировать изменения на фондовых биржах;
* разработка программного модуля для актуализации имеющихся данных о котировках акций на фондовых биржах;
* разработка удобного и интуитивно понятного интерфейса для взаимодействия с информационно-аналитической системой.

**Объектом исследования** является изменения показателей акций различных компаний на фондовых рынках.

**Предметом исследования** являются подходы к автоматизации анализа изменения показателей акций компаний на фондовых рынках.

**Методы и средства исследования.** При реализации данной задачи используются методы системно-структурного анализа и декомпозиции предметной области, объектно-ориентированное программирование, использование методологии проектирования реляционных баз данных, формализованные методы анализа данных, работы с большими данными.

**Практическую значимость** работы составляет:

* разработанный программный модуль для актуализации данных о котировках акций на фондовых биржах;
* разработанный модуль анализа влияния новостей на изменения котировок акций;
* разработанный модуль для выполнения фундаментального и технического анализов фондового рынка на основе финансовой отчетности компаний;
* Выявленные в процессе анализа котировок зависимости изменения показателей акций на фондовых биржах от событий, происходящих внутри компаний, отрасли и в мире в целом.

## Архитектура информационной системы

На этапе проектирования системы была разработана структура информационно-аналитической системы (рисунок 1).

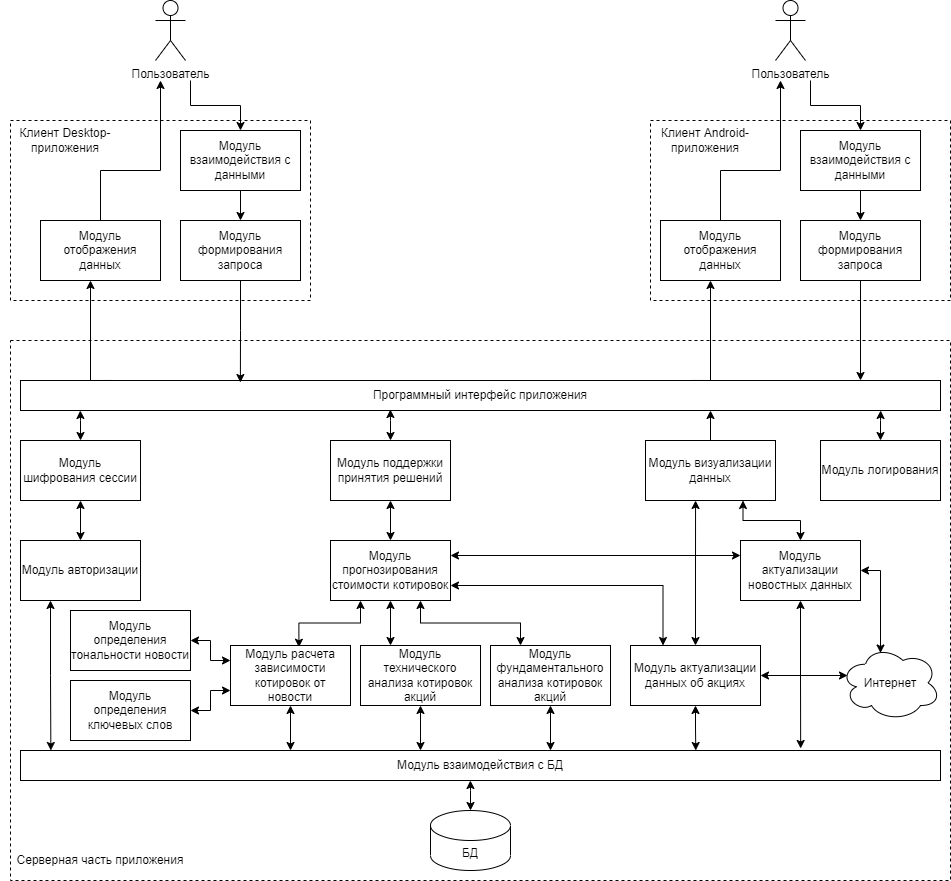


Рисунок 1 – Структура ИАС

В качестве архитектуры программного решения была выбрана архитектура «клиент-сервер». Данная архитектура подразумевает наличие серверного устройства, которое способно поддерживать одновременную работу с несколькими клиентами.

В рамках данной архитектуры на серверной части реализуется бизнес-логика приложения, модуль анализа данных, а также модуль взаимодействия с базой данных. В свою очередь модуль взаимодействия с базой данных взаимодействует с сервером СУБД, которая производит манипуляции с данными в базе.

Описание структурных элементов ИАС

Информационная система должна иметь модульную структуру. Это позволит производить манипуляции внутри отдельных модулей, не внося изменения в другие модули. Также это обеспечит возможность расширения системы путем добавления новых модулей.

**Описание серверной части ИАС**

Программный интерфейс приложения – это модуль, который служит промежуточным звеном между клиентской и серверной частями приложения. В данном случае реализован универсальный интерфейс для взаимодействия с сервером. Благодаря чему, существует возможность обеспечить доступ к серверной части приложения, не зависимо от структуры и вида клиентской части.

Модуль авторизации реализует возможность идентификации пользователя, для предоставления ему персональной информации и его сохраненных данных. Так как при авторизации используется ввод пароля, то сессия пользователя должна быть зашифрована.

Модуль шифрования необходим для того, чтобы конфиденциальные данные пользователя не передавались по сети в открытом виде. Данный модуль позволяет при необходимости шифровать данные перед их отправкой и дешифровать при получении.

Модуль расчета зависимости котировок акций от новостей служит для выявления степени влияния новости на изменения стоимости акции. Функционирование данного модуля напрямую связана с модулями анализа текста новости. Данные, полученные в результате расчетов, фиксируются в базе данных.

Для анализа текста новости применяются модуль определения тональности текста и модуль определения ключевых слов. Под определением тональности текста подразумевается поиск в тексте новости обособленных сущностей и определение эмоционального отношения, с которым данные сущности упоминаются. Модуль определения ключевых слов позволяет выделить ключевые слова из текста новости, для проведения более полноценного анализа.

Модули технического и фундаментального анализа необходимы для автоматизации соответствующих видов финансового анализа. Результаты работы данных модуль представляют их себя рассчитанные коэффициенты, значения мультипликаторов, а также значения индикаторов, влияющие на стоимость акции.

Задача модуля прогнозирования стоимости котировок акции заключается в расчете будущего значения стоимости акции исходя из данных и показателей, полученных от вышеупомянутых модулей. Кроме того, данный модуль обеспечивает взаимодействие с модулями актуализации данных, что позволяет гарантировать актуальность результатов работы данного модуля.

Модуль поддержки принятия решений реализует формирование комплекса рекомендаций, влияющих на конечный выбор пользователя. Исходя из данных прогноза модуль предлагает алгоритм действий пользователю, подкрепляя данные рекомендации результатами расчётов и анализа.

Модули актуализации новостных данных и данных об акциях обеспечивают поддержание актуального состояния базы данных. За счет автоматического мониторинга данных в сети Интернет. Кроме того, данные модули являются источниками данных для модуля визуализации данных. Являясь промежуточным звеном для информации из базы данных, они при необходимости дополняют данные актуальной информацией и затем сохраняют ей в базе данных.

Модуль визуализации данных необходим для формирования структуры и формата данных для последующей передачи их клиенту.

Модуль логирования осуществляет мониторинг работы системы и фиксацию в лог-файл аномальных состояний, а также состояний, при которых появляется угроза нормального функционирования программного продукта. Записи в лог-файле необходимы для определения причин перехода системы в аномальные или аварийные состояния, с целью как можно быстрее внести соответствующие корректировки в настройки система или программный код.

**Описание клиентской части ИАС**

Клиентская часть разрабатываемой системы подразумевает универсальную структуру вне зависимости от вида реализации клиентской части разрабатываемого продукта. Структура клиентской части включает в себя три модуля.

Модуль взаимодействия с данными определяет возможность пользователя управлять данными системы посредством понятных для пользователя действий. Тем самым данный модуль реализуют интерфейс взаимодействия пользователя с системой.

Модуль формирования запросов позволяет на основе действий пользователя сформировать запрос на получение или изменение данных. Данный модуль осуществляет связь клиентской и серверной частей программного продукта.

Для демонстрации данных используется модуль отображения данных. Функция данного модуля заключается в интерпретации данных с сервера и представлении их в удобном для пользователя виде. В частности, данный модуль обеспечивает возможность пользователя взаимодействовать с различными диаграммами, таблицами и графиками.

При создании структурной схемы необходимо использовать стандарт IDEF0 (ICAM Definition – integrated computer aided manufacturing definition).

IDEF0 – методология функционального моделирования и графическая нотация, предназначенная для формализации и описания бизнес-процессов. Отличительной особенностью IDEF0 является ее акцент на соподчиненность объектов. В IDEF0 рассматриваются логические отношения между работами, а не их временная последовательность.

Данный стандарт позволяет представить программный комплекс в виде набора функциональных блоков, каждый из которых может осуществлять взаимодействие с другими функциональными блоками посредством четырех видов интерфейса (входа, управления, механизма, выхода).

Интерфейс входа (слева) описывает исходные данные или объекты для выполнения функций. Интерфейс управления (сверху) описывает правила и ограничения. Интерфейс механизма (снизу) описывает ресурсы, используемые в процессе выполнения функции (ресурсы не должны изменяться). Интерфейс выхода (справа) описывает данные или объекты, являющиеся результатом выполнения функции.

Схема IDEF0 делиться на несколько уровней, первый уровень (рисунок 2) представляет из себя один блок «Прогноз стоимости акции». На вход поступают данные от пользователя, а именно название выбранной компании и период, на который будет производится анализ. Выходными данными является результат прогноза стоимости акции.

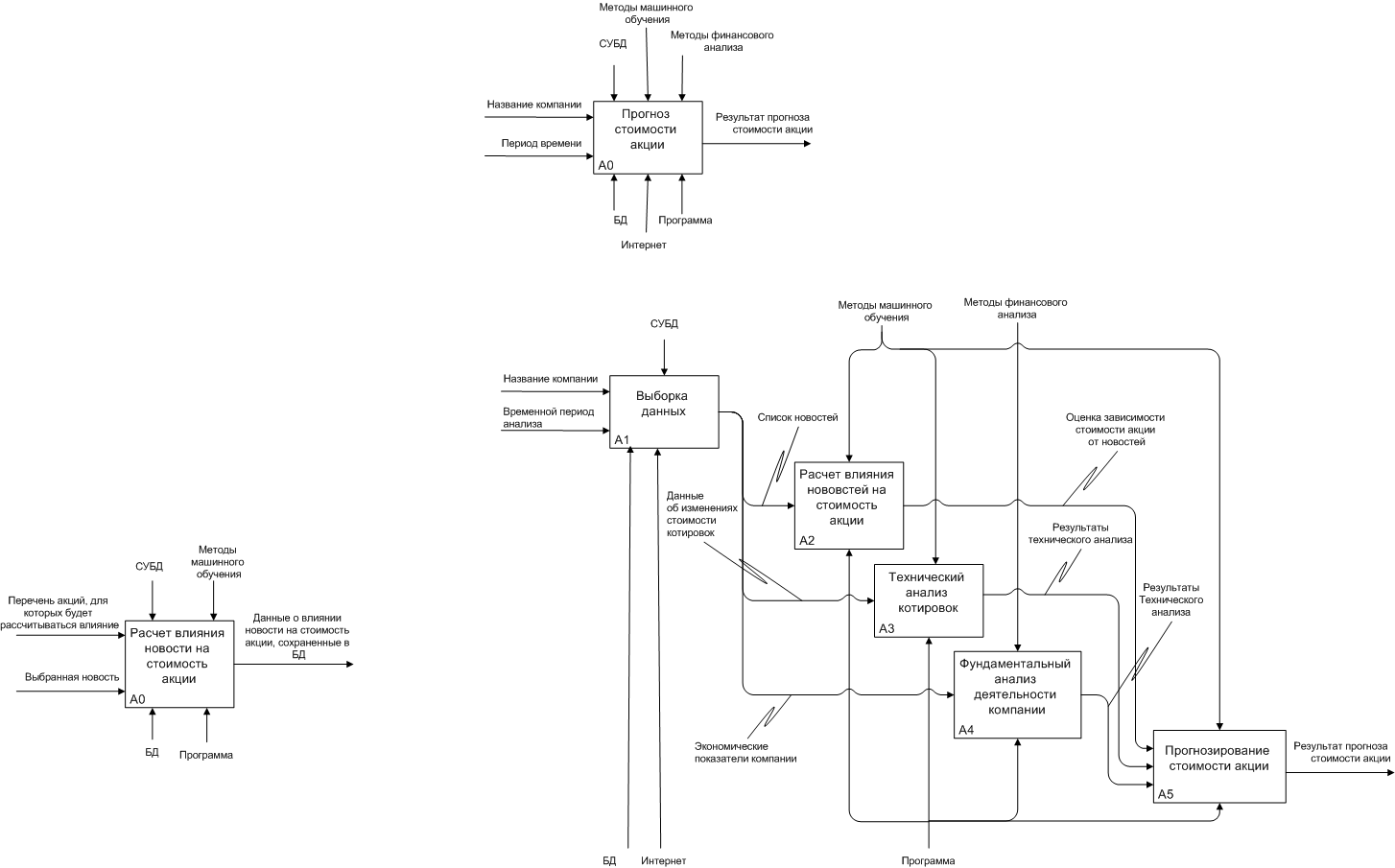


Рисунок 2 – Функциональная диаграмма системы (уровень 1)

На втором уровне детализации функциональной диаграммы процессы работы программы отображены более детально (рисунок 3). На данном уровне представлены блок по формированию данных для анализа, блок с расчета влияния новостей на стоимость акции, блок технического анализа котировок, блок фундаментального анализа компании и блок прогнозирования стоимости акции.

На этапе формирования данных для анализа программа получает входные данные от пользователя и необходимые данные из базы данных и из интернета.

На следующих этапах проводится комплексный анализ данных, а именно происходит расчет изменения стоимости акции в зависимости от новостей за определенный временной период. Оценка проводится с использованием методов машинного обучения, в данном случае эти методы реализованы в виде обученной нейронной сети.

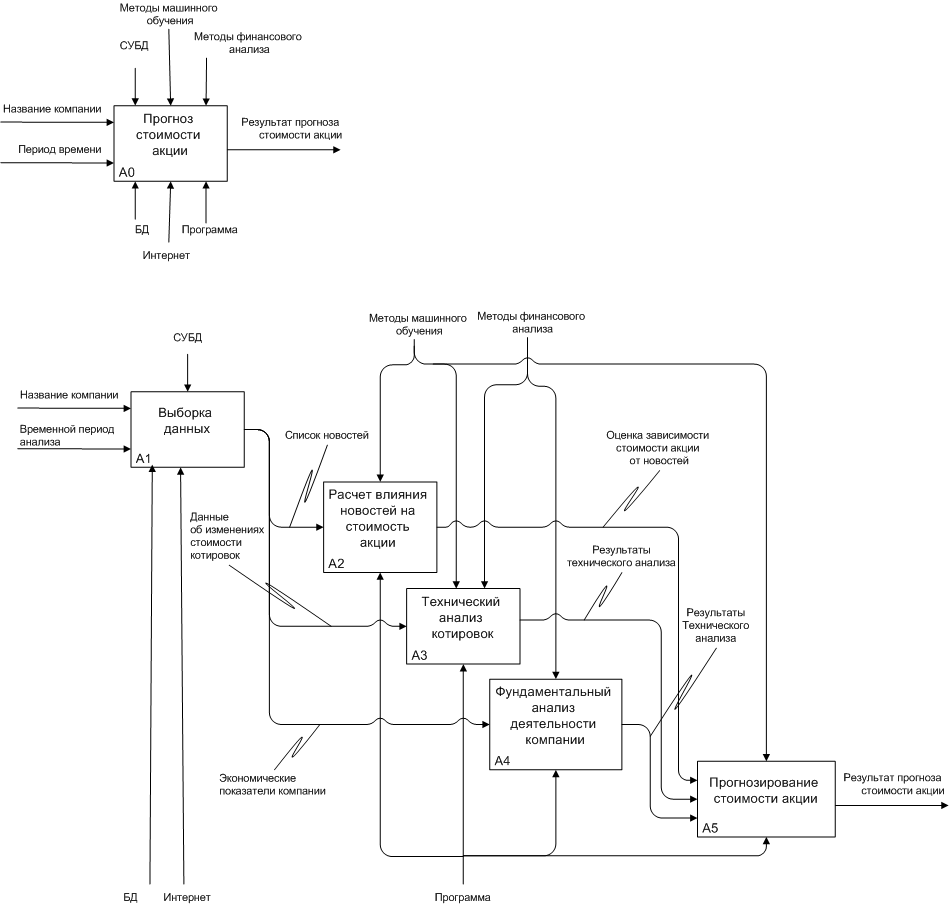


Рисунок 3 – Функциональная диаграмма системы (уровень 2)

Кроме того, данные проходят этап технического анализа. Анализ проводится на основе методов финансового анализа с применением методов машинного обучения. В качестве входных данных для анализа используются исторически данные об изменении стоимости котировок акции за определенный период времени.

В блоке фундаментального анализа деятельности компании производится оценка экономических показателей и расчет мультипликаторов для определения финансового состояния компании. В качестве входных данных используются экономические показатели компании, полученные на основе финансовой отчетности компании.

Блок прогнозирования стоимости акции агрегирует в себе результаты анализа, полученные из вышеуказанных блоков, и на основе этих данных формирует результирующую оценку будущей стоимости акции.

При анализе стоимости котировок акции одним из блоков схемы IDEF0 является блок «Расчет влияния новости на стоимость акции» (рисунок 4). На вход в данные блок поступает перечень акций для расчета влияния, а также выбранная новость, влияние которой рассчитывается. Результатом работы блока является сохранение данных о влиянии новости на стоимость котировок акции.

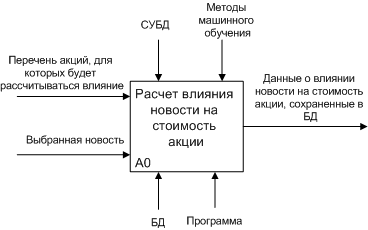


Рисунок 4 – IDEF-диаграмма расчета влияния новости на стоимость акции (уровень 1)

На втором уровне детализации блок «Расчет влияния новости на стоимость котировки акции» разделяются на 4 блока: «Выборка данных», «Определение тональности текста новости», «Определение влияния новости на стоимость акции» и «Сохранение оценки влияния в базу данных» (рисунок 5).

Блок «Выборка данных» на вход получает новость и перечень акций, для котировок которых будет производиться расчет влияния. Результатом работы данного блока является сформированный набор данных для последующего анализа.

На вход блока «Определение тональности текста новости» подается сформированный набор данных. Далее при помощи методов машинного обучения производится определение тональности текст новости, затем полученные данные о тональности передаются в следующий блок.

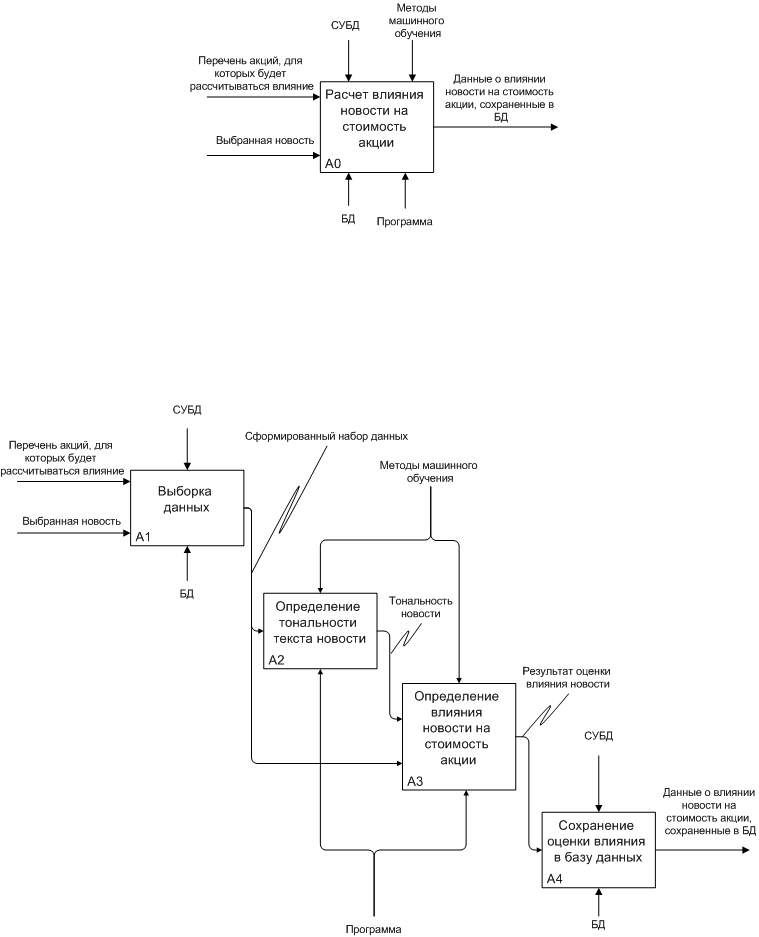


Рисунок 5 – IDEF-диаграмма расчета влияния новости на стоимость акции (уровень 2)

Блок «Определение влияния новости на стоимость котировки акции» получает на вход сформированный набор исторических данных о котировках акций, а также тональность новости, определенная в предыдущем блоке. Результатом данной области является результат оценки влияния новости.

Следующий блок «Сохранение оценки влияния в базу данных» получает на вход результат оценки влияния новости на изменение стоимости котировки акции. А затем сохраняет её в базу данных, для дальнейшего анализа. Поэтому результатом работы данного блока является данные о влияние новости на стоимость котировки акции, сохраненные в базу данных.

При разработки информационной системы необходимо учитывать какие данные и каким образом будут перемещаться между модулями и блоками системы. Диаграмма потоков данных (DFD – Data Flow Diagram) подразумевает визуализацию хранилищ данных, процессов, происходящих в информационной системе, соединенных потоками данных, а также внешние сущности, взаимодействующие с системой.

Ниже представлен фрагмент диаграммы потоков данных разрабатываемой ИАС, на ней продемонстрированы процессы, связанные с анализом стоимости акций и прогнозирования их стоимости в будущих периодах (рисунок 6).

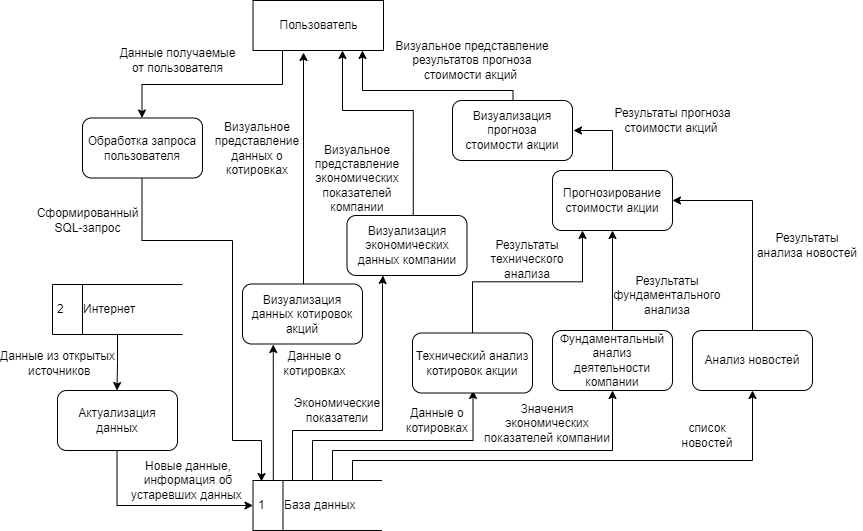


Рисунок 6 – Диаграмма потоков данных (фрагмент)

В качестве хранилищ данных в системе используются база данных и сеть Интернет. При этом основным хранилищем является база данных, получение новой информации происходит из сети Интернет в процессе актуализации данных. Важно отметить, что актуализация данных подразумевает под собой не только добавление новой информации, но и редактирование уже существующих данных.

Внешней сущностью системы является пользователь. Он является инициатором отправки данных на сервер, и он же является конечной точкой для отправляемых данных.

Данные от пользователя проходя процесс обработки, при котором происходит их преобразование в набор необходимых SQL-запросов. Полученные запросы обрабатываются базой данных. В результате обработки на выходе формируется набор данных, который в зависимости от цели перенаправляется в модули визуализации информации или в модули анализа информации. К визуализации информации можно отнести процессы визуализации данных о котировках акций и визуализации экономических данных компании.

В случае выполнения анализа данных, сформированные данные из базы данных используются в процессах технического и фундаментального анализов, а также в процессе анализа новостей. Полученные на выходе данные используются для составления прогноза стоимости акции. Данные полученные в процессе прогнозирования отправляются в модуль визуализации для представления пользователю в удобном для восприятия виде.

## Описание основных методов серверной части программного обеспечения

Так как разрабатываемая информационная система построена по принципам клиент-серверной архитектуры, то необходимо реализовать возможность обмена информацией между клиентской и серверной частями.

Одним из способов такого обмена является разработка программного интерфейса приложения или API (Application Programming Interface). Данный подход подразумевает набор URL-адресов (эндпоинтов), которые при обращении возвращают необходимые данные в универсальном формате или запускают определенный код обработки данных на сервере.

Главным плюсом использования API является его универсальность. Благодаря тому, что передача данных происходит посредством протокола HTTP, обращение к API можно производить независимо от конфигурации клиентского приложения. Это может быть как web-сайт или мобильное приложение, так и внешний сервис, которые взаимодействует с разрабатываемой системой.

В разрабатываемой ИАС программный интерфейс можно разделить на несколько блоков в зависимости от обрабатываемой информации (рисунок 7).

Блок «Auth» представляет собой набор эндпоинтов для осуществления регистрации пользователя, авторизации и процесса разлогирования. Также в данном блоке содержатся эндпоинты, необходимые для работы личного кабинета пользователя, например получения списка избранных акций.

Эндпоинты из блока «World news» используются для получения списка мировых новостей, а также информации о конкретной новости.

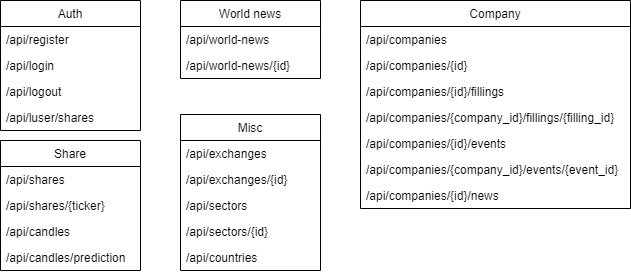


Рисунок 7 – Структура программного интерфейса приложения

Блок «Share» содержит в себе набор эндпоинтов для работы с акциями и их котировками. Например, получения списка акций, получение детальной информации об акции, получения котировок акций или получения результатов анализа и прогноза стоимости котировок акций.

В блоке «Company» объединены эндпоинты для получения данных о компании. В частности, это эндпоинты для получения списка компаний, детальной информации о компании, включающую экономические показатели, получение финансовой отчетности компании, получение списка новостей и событий, происходящих в компании.

Различные сервисные эндпоинты сгруппированы в блоке «Misc». В данном блоке перечислены эндпоинты для получения списка фондовых бирж и перечня акций, представленных на конкретной бирже; получения списка секторов экономики и списка компаний, входящих в их состав; а также списка стран, которые используются для фильтрации и группировки компаний и акций.

Реализуемые эндпоинты доступны по протоколу HTTP и возвращают ответ от сервера в формате JSON. Так как данный формат данных является одним из часто используемых форматов, используемых при построении API. Это обеспечит возможность интеграции разрабатываемой системы с различными сервисами.

## Проектирование базы данных

На основе проведенного анализа предметной области была спроектирована база данных, в которой будет храниться вся необходимая информация для разрабатываемой системы. Данная структура представлена в виде отдельных взаимосвязанных таблиц (рисунок 8).

Таблица «companies» содержит в себе информацию о компании, эмитенте акций. Данная информация включает в себя общую информацию о компании, такую как название компании, описание её деятельности, контакты, численность работников и т.д. Также в данной таблице содержатся экономические характеристики компании, например доходы и расходы компании, её стоимость, выручку и прочие экономические параметры. Таблица «companies» связана с таблицами «company\_filings» и «company\_ events» связями один-ко-многим, так как внутри одной компании публикуется множество документов и происходит множество событий.

Таблица «company\_filings» содержит данные об экономических отчетах компании. В таблице содержится информация о различных бухгалтерских отчетах, например бухгалтерские балансы, отчеты об изменении капитала, движении денежных средств и т.д. Для этого таблица содержит такие поля как дата публикации, тип отчета, ссылка на сам документ и т.д.

В таблице «company\_events» хранится информация о событиях, происходящих в компании, например выплаты дивидендов и даты их проведения. Учитываются такие свойства события как его типа, дата наступления события, название и ссылку для переходя к дополнительной информации.

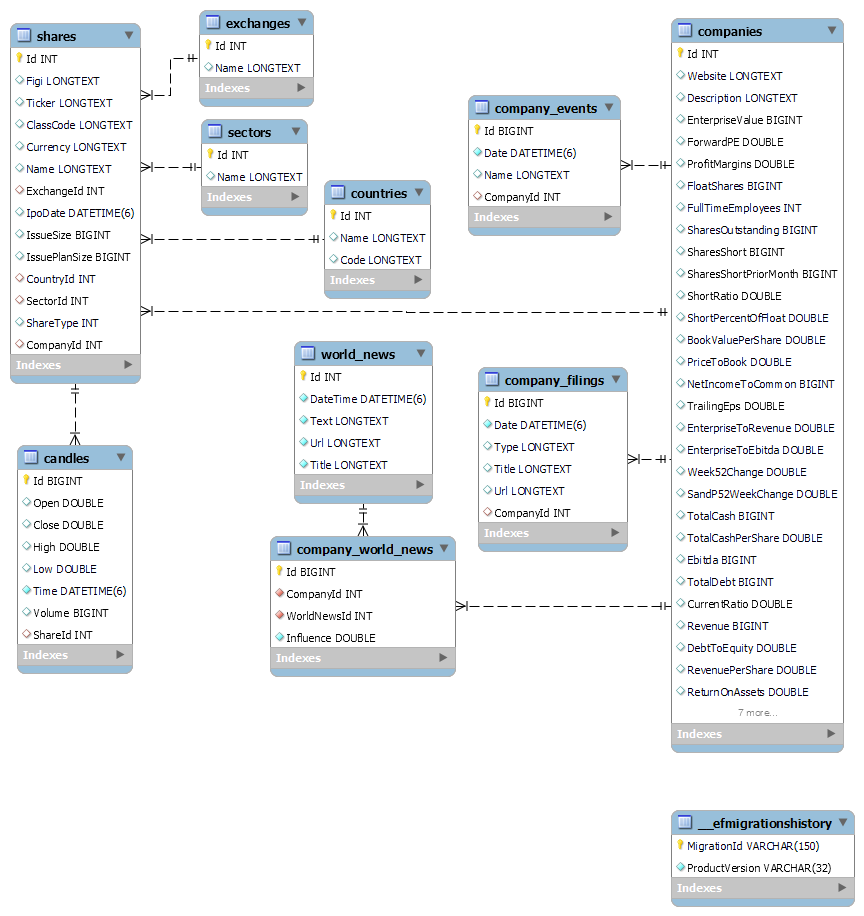


Рисунок 8 – ER-диаграмма (фрагмент)

Для хранения данных об акциях была использована таблица «shares». Данная таблица является промежуточной между компаниями и биржами. Она необходима, так как акции одной и той же компании могут размещаться на различных биржах. Таблица содержит информацию о дате IPO, количестве выпущенных акций и т.д. Кроме того, таблица содержит внешние ключи на таблицы «countries», «exchanges» и «sectors», которые хранят информацию о стране, фондовая биржа и отрасли, к которым относится конкретная акция. Данные таблицы необходимы для фильтрации и группировки акций компаний.

Таблица «candles» хранит в себе исторические данные о котировках. Данные включают в себя количество проданных акций, стоимость акции на момент открытия, закрытия, а также наибольшая и наименьшая стоимости акции, за конкретную дату. Также в данной таблице хранится ссылка на таблицу «shares» реализуя связь один-ко-многим.

Таблица «company\_world\_news» является связующей таблицей между компаниями и мировыми новостями, которые влияют на показатели изменения котировок акций. Она содержит в себя ссылки на таблицы «companies» и «world\_news», также данная таблица хранит в себе оценку влияния мировых новостей на изменения котировок акций.

## Выбор формализованных методов анализа

Одной из основных задач, которые должна решать разрабатываемая ИАС, является задача прогнозирования стоимости акции. При декомпозиции задачи прогнозирования были выделены три под задачи: прогнозирование на базе технического анализа, фундаментального анализа и анализа новостей.

Технический анализ акций компании представляет собой систему прогнозирования цен, основанную на информации, полученной в результате рыночных торгов. Иными словами, в основе технического анализа лежит выделение и изучение определенных закономерностей в движении графика котировок. То есть принятое решение основывается только на графическом изображении линии тренда.

Для формализации данного метода анализа было принято решение использовать методы машинного обучения, а именно обученную рекуррентную нейронную сеть. Особенность рекуррентных нейронных сетей заключается в возможности передавать информацию между итерациями обработки данных (рисунок 9). Традиционные нейронные сети таким свойством не обладают, поэтому использовать их для анализа изменения стоимости акций, как временных рядом, невозможно.

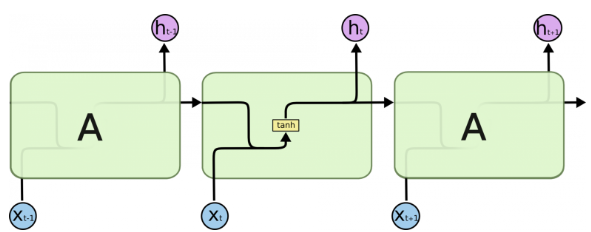


Рисунок 9 – Архитектура рекуррентной нейронной сети

Данные, полученные на предыдущей итерации нейронной сети, принято обозначать как ht-1, входные данные на текущем шаге – xt, функция зависимости результирующего показателя от входного параметра обозначается как σ. В качестве расчетной функции используется гиперболический тангенс tanh(). Таким образом значение текущей итерации (ht) нейронной сети рассчитывается как:

|  |  |
| --- | --- |
|  | (1) |

Для осуществления анализа новостей и оценки их влияния на стоимость акции необходимо произвести необходимо решить задачи, связанные с анализом естественного языка. В частности, необходимо в тексте определить перечень компаний, упоминаемых в новости, а также оценить тональность текста новости относительно каждой из обнаруженных компаний. Кроме того, необходимо выделить из текста новости ключевые слова, для формирования более полной оценки отношения встречающихся слов в тексте к упоминаемым компаниям.

В качестве средства для анализа текста наилучшим образом подойдет модель LSTM (Long short-term memory), так как для данной модели в свободном доступе имеется множество наборов данных, как на английском, так и на русском языках. Для достижения поставленной цели это использование метода замены сущности при формировании набора данных для обучения.

Например, анализируя текст новости «… компания Apple фиксирует увеличение прибыли на фоне пожара, произошедшего на заводе Samsung …», лингвистической модели будет сложно определить тональность данной новости. Поэтому при обучении сущности «Apple» и «Samsung» необходимо последовательно выделять как ключевую сущность (таблица 1).

Таблица 1 – Пример набора данных для обучения лингвистической модели

|  |  |  |
| --- | --- | --- |
| Ключевая сущность | Текст новости | Тональность текста |
| Apple | компания **Х** фиксирует увеличение прибыли на фоне пожара, произошедшего на заводе Samsung | позитивно |
| Samsung | компания Apple фиксирует увеличение прибыли на фоне пожара, произошедшего на заводе **X** | негативно |

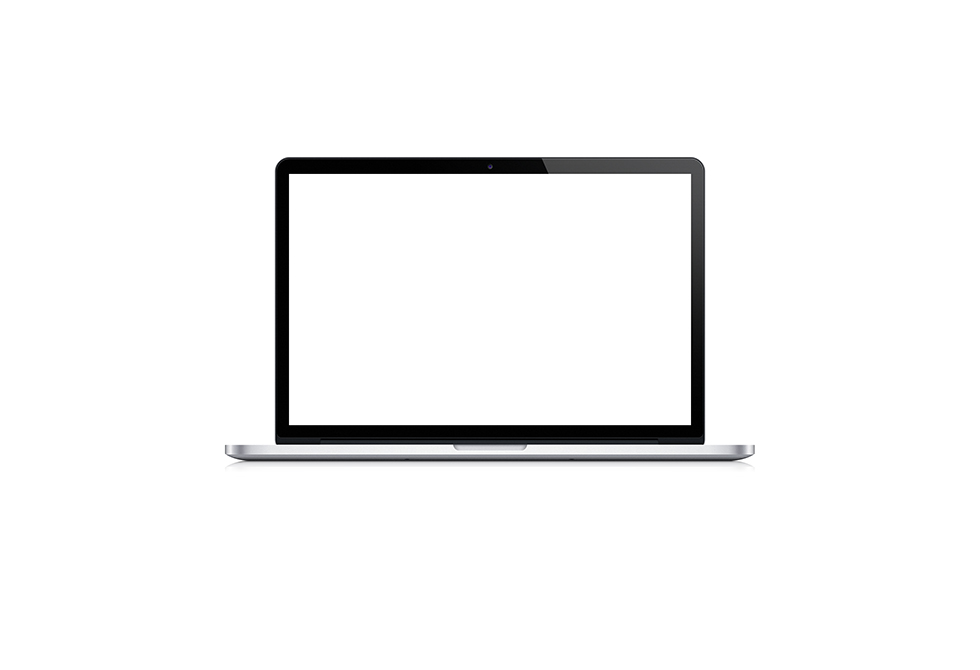
## Разработка пользовательского интерфейса

При разработке информационно-аналитической системы необходимо реализовать интерфейс взаимодействия пользователя с системой. При этом пользовательский интерфейс должен удовлетворять определенным критериям:

* Естественность интерфейса – свойство интерфейса, которое означает, что сообщения и результаты, выдаваемые приложением, не должны требовать дополнительных пояснений.
* Согласованность интерфейса позволяет пользователям переносить имеющиеся знания на новые задания, осваивать новые аспекты быстрее, и благодаря этому фокусировать внимание на решаемой задаче, а не тратить время на уяснение различий в использовании тех или иных элементов управления, команд и т. д. Обеспечивая преемственность полученных ранее знаний и навыков, согласованность делает интерфейс узнаваемым и предсказуемым.
* Принцип «обратной связи» – принцип, который означает, что каждое действие пользователя должно получать визуальное, а иногда и звуковое подтверждение того, что программное обеспечение восприняло введенную команду; при этом вид реакции, по возможности, должен учитывать природу выполненного действия.
* Простота интерфейса – представление на экране информации, минимально необходимой для выполнения пользователем очередного шага задания.

Базовым элементом, вокруг которого строится анализ и прогнозирование стоимости акции, является график изменения её стоимости во времени. Однако кроме на стоимость могут также влиять экономические показатели компании, мировые новости, а также изменения стоимости других акций.

С целью расширить возможности анализа было принято решение использовать построение диаграммы с возможность добавления на нее графиков других компаний. При этом для удобства реализована функция, которая при клике на определенную дату графика выводит список новостей за этот день (рисунок 10), что позволяет пользователю одним нажатием перейти от технического анализа к анализу новостей.



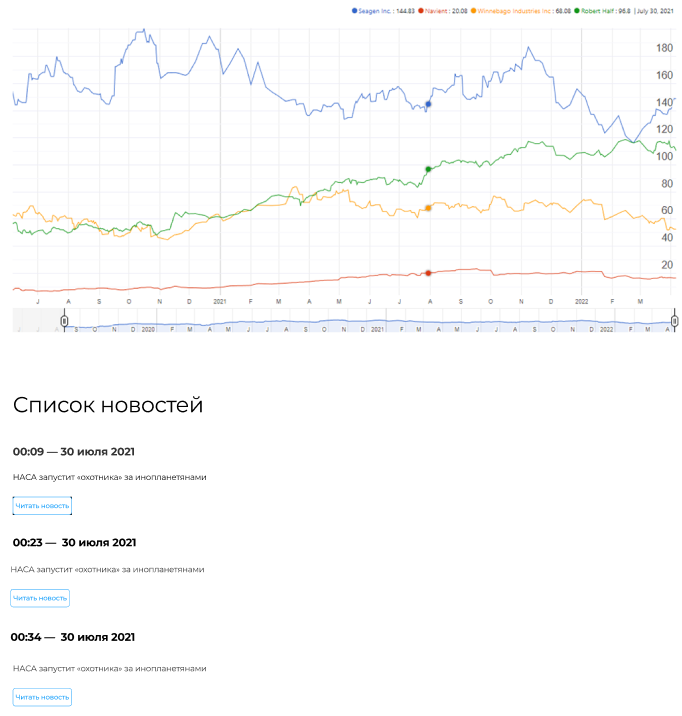


Рисунок 10 – Интерфейс программного продукта (фрагмент)

При переходе на страницу компании, пользователь может получить информацию о её деятельности, экономическом состоянии, а также оценить тренд изменения стоимости акции на основе графика.

Кроме того, пользователь должен иметь возможность управлять анализом и прогнозом стоимости акции. Для этого на странице был реализован блок «Прогнозирование стоимости». Данный блок включает в себя реализацию используемых типов анализа: технический анализ, фундаментальный анализ и анализ на основе новостей (рисунок 11).

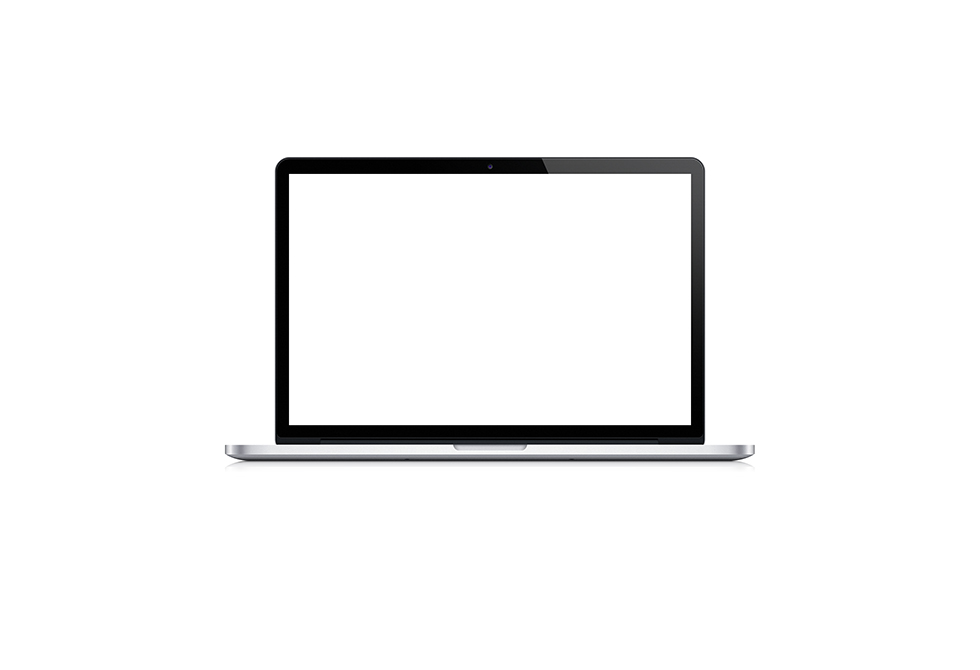
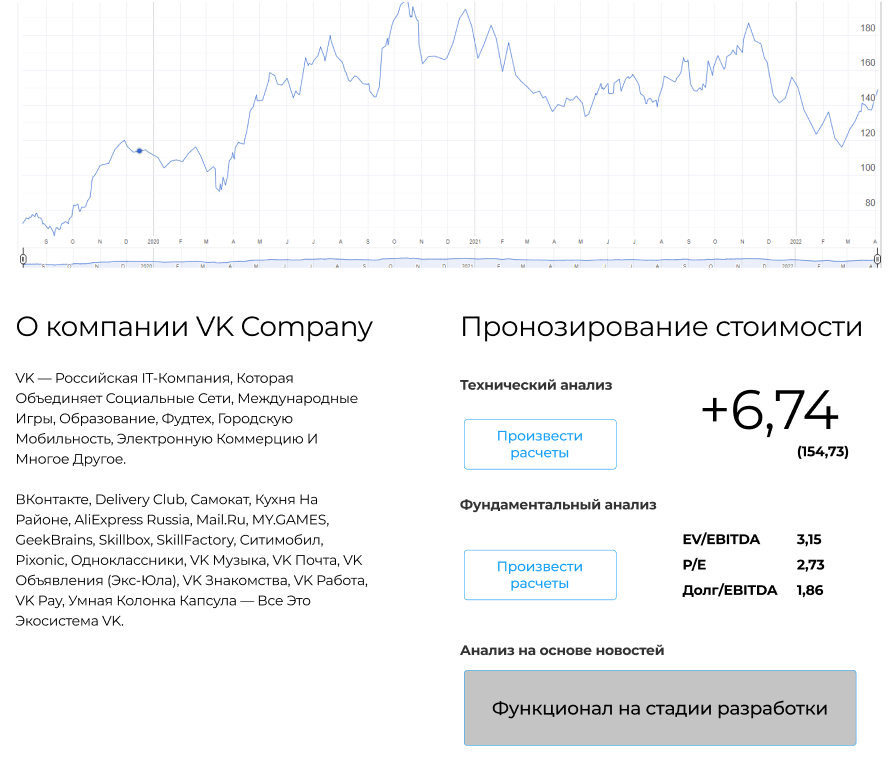


Рисунок 11 – Интерфейс программного продукта (фрагмент)

Для получения результатов анализа пользователю необходимо нажать копку «Произвести расчеты» в требуемом блоке анализа. После чего на сервер отправляется запрос для проведения анализа стоимости указанной акции. После завершения анализа результаты отправляются на клиентскую часть программного продукта, где выводятся в удобном для восприятия виде.

## Тестирование программной системы

Задача тестирования ПО нуждается в больших трудозатратах и тщательном применении выбранных подходов. Для успешного внедрения программы в рамках рабочего процесса организации-эксплуатанта необходимо обеспечение полной работоспособности системы в рамках предусмотренных задач, включающее отсутствие критических уязвимостей в процессах, обработку потенциальных ошибок, конфигурацию системы в соответствии с требованиями среды её эксплуатации.

В тестировании системы применимы следующие виды контроля:

Модульное тестирование (module testing, unit testing) – контроль исполнения функций отдельного модуля программной системы.

Тестирование интеграции (integration testing) – контроль обеспечения работоспособности взаимодействия между отдельными функциональными компонентами системы.

Тестирование внешних функций (external functions testing) – контроль взаимодействия системы с применяемыми при эксплуатации функциями внешнего программного обеспечения.

Приемочное тестирование (acceptance testing) – проверка системы на соответствие техническому заданию и требованиям конечного эксплуатанта (заказчика).

Комплексное тестирование (system testing) – контроль и / или испытание полного содержания процесса эксплуатации системы по отношению к исходным целям её создания. При выполнении в искусственной (моделируемой) среде данный вид тестирования следует относить к процессам контроля, тогда как выполнение данного вида тестирования при внедрении на предприятие относится к процессам испытания.

Тестирование настройки (installation testing) – проверка корректности процесса установки системы для эксплуатации.

Для проверки системы могут использоваться статическое и динамическое тестирование. Статическое тестирование предполагает проверку исходного кода программы на наличие структурных дефектов, потенциально приводящих к неисправностям в процессе эксплуатации (корректность применяемой логики, взаимодействия методов и классов). Динамическое тестирование предполагает использование написанного, скомпилированного и запущенного программного кода. Анализируется поведение системы во время выполнения рабочих задач.

К основным принципам организации тестирования относятся:

* Необходимой частью каждого теста должно являться определение предполагаемых результатов работы системы для быстрого устранения выявленных недостатков;
* Следует по возможности избегать тестирования программы ее автором, т.к. процесс реальной эксплуатации системы может значительно отличаться от предполагаемого автором (отладка программы при этом входит в задачи автора программного кода).
* Должны являться правилом доскональное изучение результатов каждого теста, чтобы не пропустить малозаметную на поверхностный взгляд ошибку в программе.
* Необходимо проводить проверку как корректных (предусмотренных) входных данных, так и некорректных (непредусмотренных).
* При анализе результатов каждого теста необходимо проверять, не выполняет ли программа не предусмотренных процессом эксплуатации функций.

## Оценка производительности системы

Для оценки программного обеспечения была произведена оценка использования ресурсов и быстродействия персонального компьютера разработанной системы.

Для проведения тестирования были выбраны замеры скорости загрузки модуля работы с БД, время загрузки модуля формирования web-страниц, скорость загрузки модуля серверов для разных платформ, а также полное время работы алгоритма анализа данных. Кроме выше озвученных тестов, был проведен замер потребления оперативной памяти и загрузки процессора в заданных конфигурациях.

Таблица 1 - Производительность системы

| № | Конфигурация | Загрузка модуля работы с БД, мс | Загрузка модуля формирования web-страницы, мс | Загрузка  модуля серверов для разных платформ, мс | Скорость работы алгоритма анализа |
| --- | --- | --- | --- | --- | --- |
| 1 | CPU 3.6 ГГц на 12 потоков, 16 ГБ RAM, Windows 10 | 4 578 | 21 367 | 1 964 | 3 мин 19 с |
| 2 | CPU 2.5 ГГц на 4 потока, 4 ГБ ОЗУ, Windows 10 | 10 256 | 34 546 | 2 566 | 5 мин 15 с |
| 3 | CPU 2.7 ГГц на 8 потоков, 8 ГБ ОЗУ, Windows 10 | 7 436 | 27 843 | 2 344 | 4 мин 58 с |

В результате поведенного тестирования, можно сделать вывод о том, что наибольшее время в работе ИАС занимает модуль формирования web-страниц, предназначенный для визуализации исторических данных и данных проведенного анализа. В соответствии с конфигурацией (табл. 1) был проведен замер потребления ОЗУ и нагрузки на процессор (табл. 2).

Таблица 2 - Нагрузка системы

| № | Конфигурация | Потребление ОЗУ | Нагрузка на процессор |
| --- | --- | --- | --- |
| 1 | CPU 3.6 ГГц на 12 потоков, 16 ГБ RAM, Windows 10 | 693МБ | 25% |
| 2 | CPU 2.5 ГГц на 4 потока, 4 ГБ ОЗУ, Windows 10 | 747МБ | 79% |
| 3 | CPU 2.7 ГГц на 8 потоков, 8 ГБ ОЗУ, Windows 10 | 645МБ | 56% |

В соответствии с тестированием нагрузки, были сформулированы минимальные системные требования к серверной части программного обеспечения: процессор не менее 4 ядер/8 потоков и 2.70 ГГц, не менее 8 ГБ ОЗУ.

## Интерпретация полученных результатов

При разработке программного обеспечения и реализации поиска аномалий в данных была сформирована выборка данных для каждой избирательной кампании.

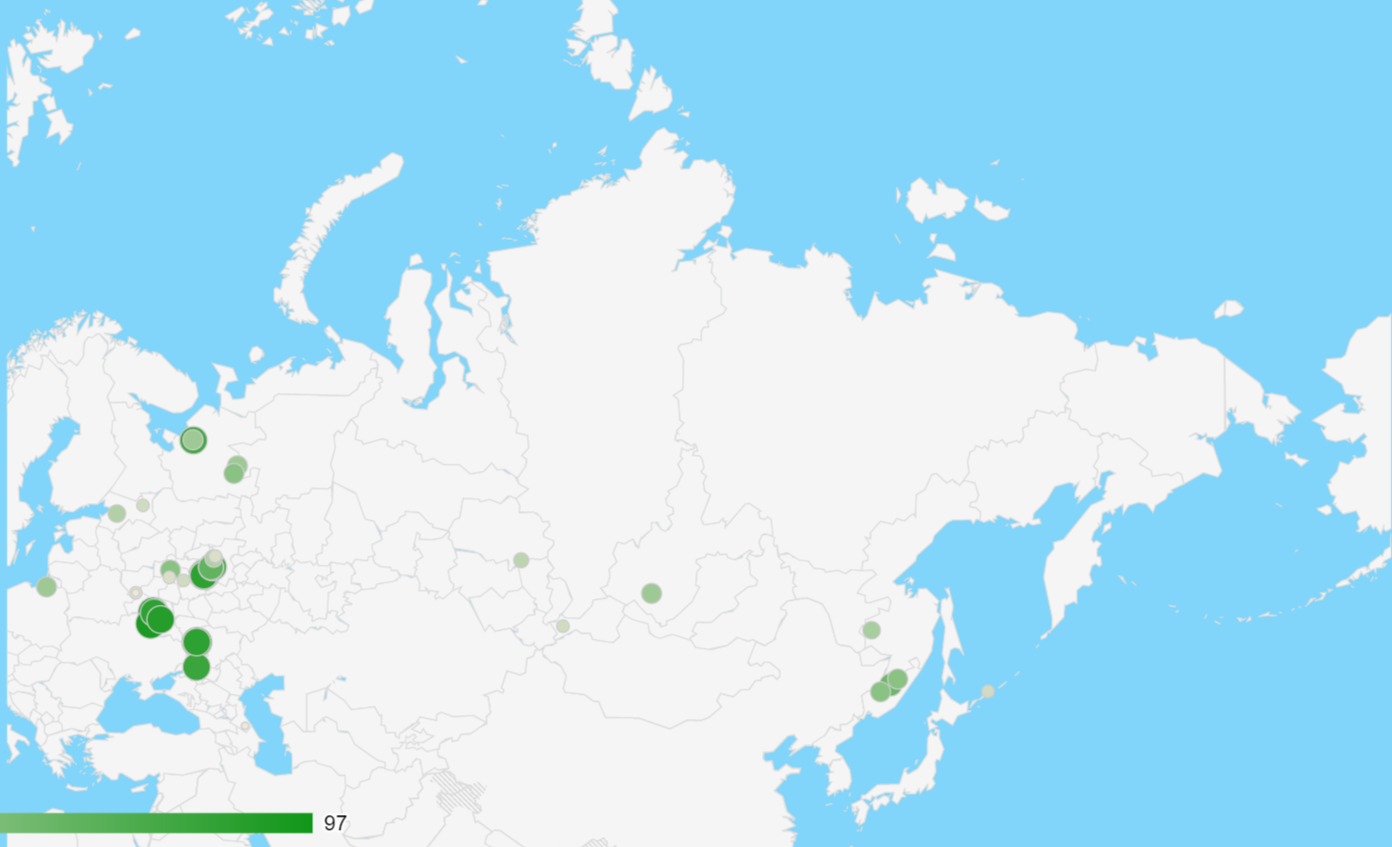
Для определения аномалий явки была сформирована отдельная выборка включающая в себя количество избирателей, зарегистрированных на избирательном участке, процент явки и регион. Размер датасета составляет 284 202 записи для избирательных кампаний с 2011 по 2021 года.

Для определения аномалий голосования за партию была сформирована выборка включающая в себя номер ТИКа, явку, номер партии и процент голосов за данную партию. Размер датасета составляет 97 818 записей.

В результате проведения анализа для поиска аномалий в явке (рисунок 8) были получены следующие результаты. На примере выборов 2021 года рассмотрим полученные результаты. Было найдено 89 участков с аномалиями в явке. Вероятность аномалий составляет от 70% до 97%. Как можно наблюдать, с помощью карты распределения аномалий на карте (рисунок 9), основные аномалии явки располагаются на Европейской части РФ. Так же используя карту явки (рисунок 10) можно сделать вывод о том, что аномалии явки наблюдаются в регионах с явкой 80%.

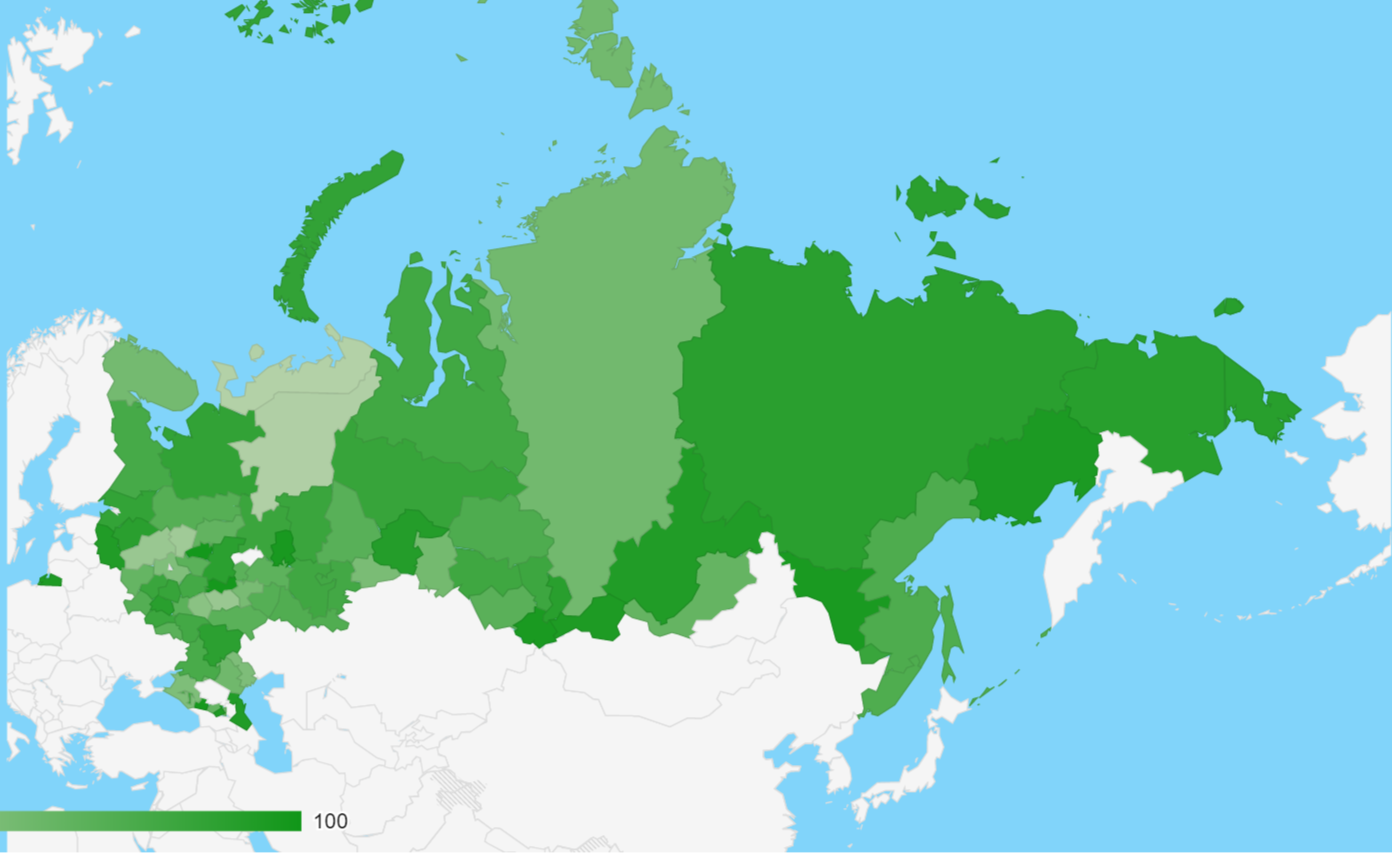


*Рисунок 15 - Аномалии явки*

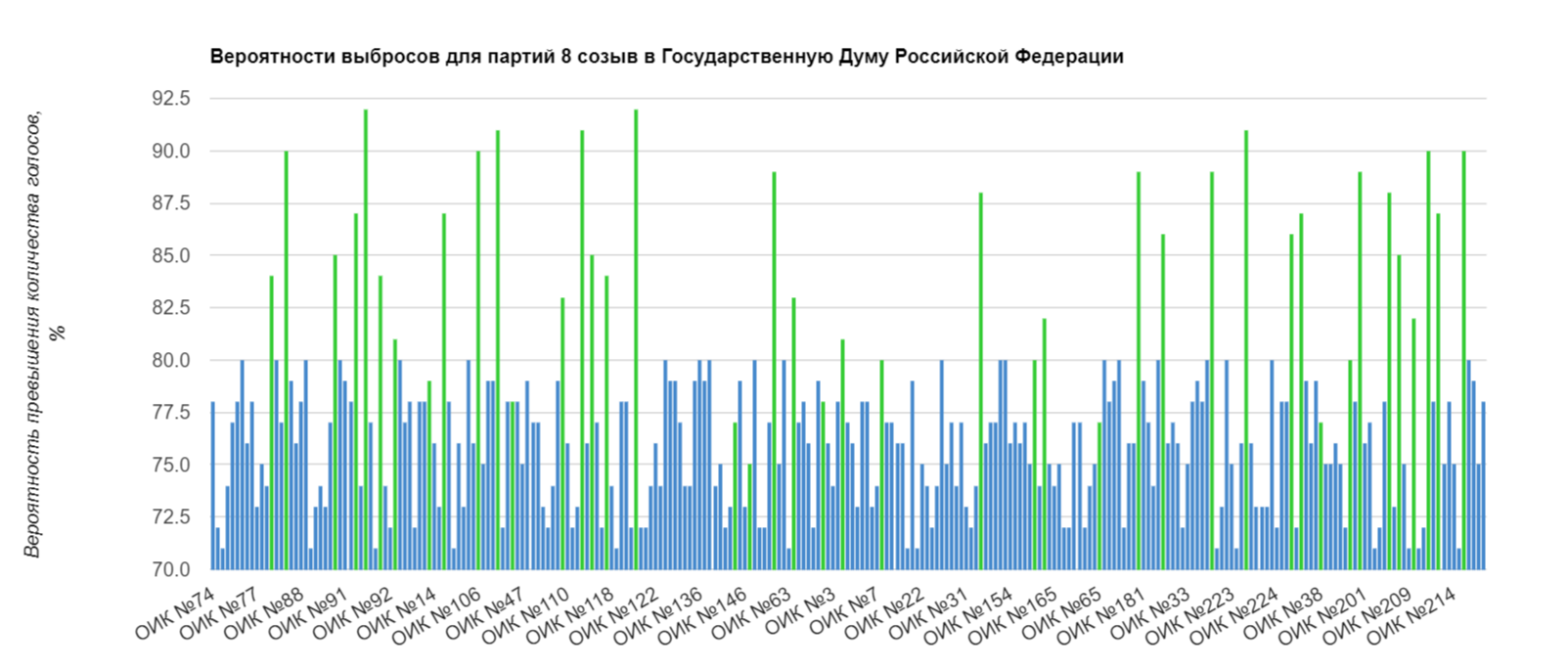


*Рисунок 16 - Аномалии голосования на карте*

В результате выявления аномалий в голосовании за партию (рисунок 10), были получены следующие результаты. Для избирательной кампании 2021 года, были выявлены аномалии в голосовании за партию «Яблоко» и «ЛДПР». Для партии «Яблоко» вероятности аномалий (рисунок 12) составляют от 75% до 92%. Для партии «ЛДПР» вероятность аномалий (рисунок 13) составила от 71% до 80%.



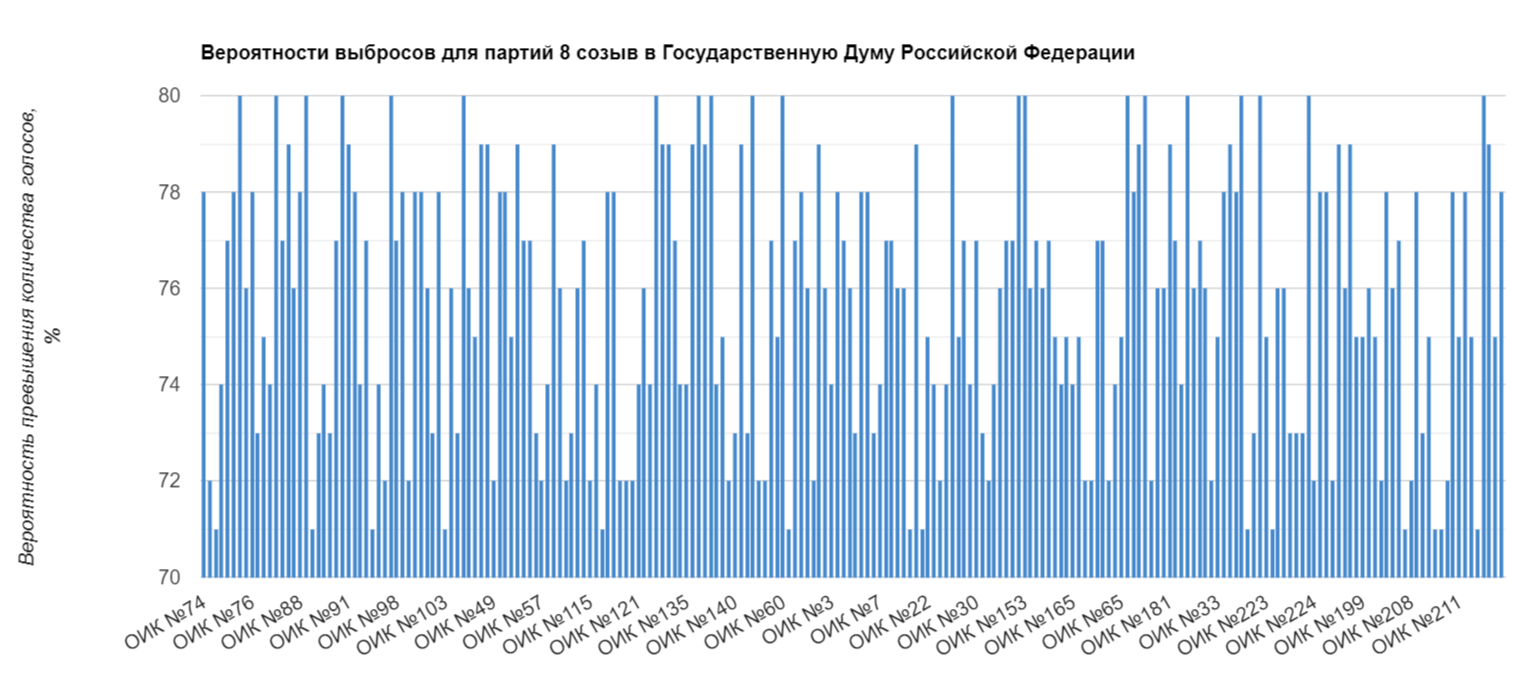
*Рисунок 17 - Явка по регионам*



*Рисунок 18 - Аномалии голосования за партию*



*Рисунок 19 - Аномалии голосования за партию «Яблоко»*



*Рисунок 20 - Аномалии голосования за партию «ЛДПР»*

Таким образом, данные, представленные в полученной системе, пригодны для анализа при помощи метода SRCNN; удалось выявить ряд аномалий в явке и итоговым результатам голосования за партию.

# ЗАКЛЮЧЕНИЕ

В результате прохождения производственной (преддипломной практики) была разработана информационно-аналитическая система для анализа фондового рынка. В работе была обоснована актуальность данной темы, исследована предметная область и проанализированы существующие решения.

На основе анализа были выставлены функциональные требования к программному продукту и составлено техническое задание. Также для реализации данного программного продукта были реализованы методы анализа изменения котировок акций на фондовых биржах.

Разработанная аналитическая система позволяет прогнозировать изменения на фондовых биржах, а также выявлять зависимость изменения показателей акций на фондовых биржах от событий, происходящих внутри компаний, отрасли и в мире в целом.

# СПИСОК ЛИТЕРАТУРЫ

1. Абрамян, А. В. Разработка пользовательского интерфейса на основе системы Windows Presentation Foundation : учебник / А. В. Абрамян, М. Э. Абрамян. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2017. — 301 c. — ISBN 978-5-9275-2375-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/87487.html> — Режим доступа: для авторизир. Пользователей
2. Спицина, И. А. Применение системного анализа при разработке пользовательского интерфейса информационных систем : учебное пособие / И. А. Спицина, К. А. Аксёнов ; под редакцией Л. Г. Доросинского. — Екатеринбург : Издательство Уральского университета, 2018. — 100 c. — ISBN 978‑5‑7996‑2265‑7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/106498.html> — Режим доступа: для авторизир. пользователей
3. Назаркин, О. А. Разработка графического пользовательского интерфейса в соответствии с паттерном Model-View-Viewmodel на платформе Windows Presentation Foundation. Основные средства WPF : учебное пособие по дисциплине «Проектирование человеко-машинного интерфейса» / О. А. Назаркин. — Липецк : Липецкий государственный технический университет, ЭБС АСВ, 2014. — 61 c. — ISBN 978-5-88247-679-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/55141.html> — Режим доступа: для авторизир. Пользователей
4. Федин, Ф. О. Анализ данных. Часть 2. Инструменты Data Mining : учебное пособие / Ф. О. Федин, Ф. Ф. Федин. — Москва : Московский городской педагогический университет, 2012. — 308 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/26445.html> — Режим доступа: для авторизир. Пользователей
5. Воронов, В. И. Data Mining - технологии обработки больших данных : учебное пособие / В. И. Воронов, Л. И. Воронова, В. А. Усачев. — Москва : Московский технический университет связи и информатики, 2018. — 47 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/81324.html> — Режим доступа: для авторизир. пользователей. - DOI: <https://doi.org/10.23682/81324>
6. Чубукова, И. А. Data Mining : учебное пособие / И. А. Чубукова. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 469 c. — ISBN 978-5-4497-0289-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89404.html> — Режим доступа: для авторизир. Пользователей
7. Воронова, Л. И. Big Data. Методы и средства анализа : учебное пособие / Л. И. Воронова, В. И. Воронов. — Москва : Московский технический университет связи и информатики, 2016. — 33 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/61463.html (дата обращения: 25.12.2021). — Режим доступа: для авторизир. Пользователей
8. Проектирование клиент-серверных приложений : учебное пособие для проведения практических занятий / составители П. В. Лобзенко, И. В. Щербань. — Ростов-на-Дону : Северо-Кавказский филиал Московского технического университета связи и информатики, 2018. — 54 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89512.html> — Режим доступа: для авторизир. Пользователей
9. Васюткина, И. А. Разработка клиент-серверных приложений на языке C# : учебное пособие / И. А. Васюткина. — Новосибирск : Новосибирский государственный технический университет, 2016. — 112 c. — ISBN 978-5-7782-2932-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/91508.html (дата обращения: 25.12.2021). — Режим доступа: для авторизир. пользователей
10. Разработка обслуживаемых программ на языке С#. [Электронный ресурс]. – Режим доступа: https://dmkpress.com/catalog/computer/ programming/dot\_net/978-5-97060-446-5/
11. Базы данных и СУБД. [Электронный ресурс]. – Режим доступа: https://timeweb.com/ru/community/articles/bazy-dannyh-i-subd-1
12. Цильковский И.А. Методы анализа знаний и данных [Электронный ресурс] : конспект лекций / И.А. Цильковский, В.М. Волкова. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2010. — 68 c. — 978-57782-1377-7. — Режим доступа: http://www.iprbookshop.ru/45385.html
13. Закон Российской Федерации "Об информации, информационных технологиях и о защите информации" от 27.07.2006 № 149-ФЗ // Российская газета. 2006 г. с изм. и допол. в ред. от 02.12.2019
14. Учебник. Обнаружение аномалий в данных – Режим доступа: https://docs.microsoft.com/ru-ru/dotnet/machine-learning/tutorials/sales-anomaly-detection
15. Грамотная клиент-серверная архитектура: как правильно проектировать и разрабатывать web API. [Электронный ресурс]. – Режим доступа: <https://tproger.ru/articles/web-api/>
16. Нестеренко Дмитрий Юрьевич Объектно-ориентированное программирование на примере языка Java // Научный журнал. 2016. №8 (9). URL: https://cyberleninka.ru/article/n/obektno-orientirovannoe-programmirovanie-na-primere-yazyka-java
17. Разработка обслуживаемых программ на языке С#. [Электронный ресурс]. – Режим доступа: https://dmkpress.com/catalog/computer/  
    programming/dot\_net/978-5-97060-446-5/
18. Павлов С.Н. Системы искусственного интеллекта. Часть 1 [Электрон-ный ресурс] : учебное пособие / С.Н. Павлов. — Электрон. текстовые данные. — Томск: Томский государственный университет систем управления и радиоэлек-троники, Эль Контент, 2011. — 176 c. — 978-5-4332-0013-5. — Режим доступа: http://www.iprbookshop.ru/13974.html
19. Цильковский И.А. Методы анализа знаний и данных [Электронный ресурс] : конспект лекций / И.А. Цильковский, В.М. Волкова. — Электрон. текстовые данные. — Новосибирск: Новосибирский государственный технический университет, 2010. — 68 c. — 978-57782-1377-7. — Режим доступа: http://www.iprbookshop.ru/45385.html
20. Открытый курс машинного обучения. Тема 5. Композиции: бэггинг, случайный лес [Электронный ресурс]. – Режим доступа: https://habr.com/ru/company/ods/blog/324402/#2-sluchaynyy-les, свободный. – (дата обращения: 16.12.2021).
21. Васюткина, И. А. Технология разработки объектно-ориентированных программ на JAVA : учебно-методическое пособие / И. А. Васюткина. — Новосибирск : Новосибирский государственный технический университет, 2012. — 152 c. — ISBN 978-5-7782-1973-1. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/45047.html> — Режим доступа: для авторизир. Пользователей
22. Мухаметзянов, Р. Р. Основы программирования на Java : учебное пособие / Р. Р. Мухаметзянов. — Набережные Челны : Набережночелнинский государственный педагогический университет, 2017. — 114 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/66812.html> — Режим доступа: для авторизир. пользователей. - DOI: <https://doi.org/10.23682/66812>
23. Программирование на языке Java : конспект лекций / А. В. Гаврилов, С. В. Клименков, А. Е. Харитонова, Е. А. Цопа. — Санкт-Петербург : Университет ИТМО, 2015. — 123 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/68692.html (дата обращения: 26.12.2021). — Режим доступа: для авторизир. Пользователей
24. Васильев, А. Н. Самоучитель Java с примерами и программами / А. Н. Васильев. — 4-е изд. — Санкт-Петербург : Наука и Техника, 2017. — 367 c. — ISBN 978-5-94387-745-2. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: https://www.iprbookshop.ru/73048.html (дата обращения: 26.12.2021). — Режим доступа: для авторизир. пользователей
25. Федин, Ф. О. Анализ данных. Часть 1. Подготовка данных к анализу : учебное пособие / Ф. О. Федин, Ф. Ф. Федин. — Москва : Московский городской педагогический университет, 2012. — 204 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/26444.html> — Режим доступа: для авторизир. Пользователей
26. Федин, Ф. О. Анализ данных. Часть 2. Инструменты Data Mining : учебное пособие / Ф. О. Федин, Ф. Ф. Федин. — Москва : Московский городской педагогический университет, 2012. — 308 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/26445.html> — Режим доступа: для авторизир. Пользователей
27. Воронова, Л. И. Machine Learning: регрессионные методы интеллектуального анализа данных : учебное пособие / Л. И. Воронова, В. И. Воронов. — Москва : Московский технический университет связи и информатики, 2018. — 82 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/81325.html> — Режим доступа: для авторизир. пользователей.
28. Горожанина, Е. И. Нейронные сети : учебное пособие / Е. И. Горожанина. — Самара : Поволжский государственный университет телекоммуникаций и информатики, 2017. — 84 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/75391.html> — Режим доступа: для авторизир. Пользователей
29. Павлова, А. И. Информационные технологии: основные положения теории искусственных нейронных сетей : учебное пособие / А. И. Павлова. — Новосибирск : Новосибирский государственный университет экономики и управления «НИНХ», 2017. — 191 c. — ISBN 978-5-7014-0801-0. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/87110.html> — Режим доступа: для авторизир. пользователей. - DOI: <https://doi.org/10.23682/87110>
30. Барский, А. Б. Введение в нейронные сети : учебное пособие / А. Б. Барский. — 3-е изд. — Москва, Саратов : Интернет-Университет Информационных Технологий (ИНТУИТ), Ай Пи Ар Медиа, 2020. — 357 c. — ISBN 978-5-4497-0309-5. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/89426.html> — Режим доступа: для авторизир. Пользователей
31. Барский, А. Б. Искусственный интеллект и логические нейронные сети : учебное пособие / А. Б. Барский. — Санкт-Петербург : Интермедия, 2019. — 360 c. — ISBN 978-5-4383-0155-4. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/95270.html> — Режим доступа: для авторизир. Пользователей
32. Свёрточная нейронная сеть [электронный ресурс] Режим доступа: <https://ru.wikipedia.org/wiki/Свёрточная_нейронная_сеть>
33. Документация по ML.NET [электронный ресурс] Режим доступа: <https://docs.microsoft.com/ru-ru/dotnet/machine-learning/>
34. ChromiumWebBrowser Class [электронный ресурс] Режим доступа: https://cefsharp.github.io/api/51.0.0/html/T\_CefSharp\_WinForms\_ChromiumWebBrowser.htm
35. Зудилова, Т. В. Web-программирование HTML / Т. В. Зудилова, М. Л. Буркова. — Санкт-Петербург : Университет ИТМО, 2012. — 70 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/65748.html> — Режим доступа: для авторизир. Пользователей
36. Петрунина, Е. Б. Основы HTML : учебно-методическое пособие / Е. Б. Петрунина, Е. Г. Селина. — Санкт-Петербург : Университет ИТМО, 2013. — 47 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/67488.html> — Режим доступа: для авторизир. Пользователей
37. Кириченко, А. В. Динамические сайты на HTML, CSS, Javascript И Bootstrap. Практика, практика и только практика / А. В. Кириченко, Е. В. Дубовик. — Санкт-Петербург : Наука и Техника, 2018. — 272 c. — ISBN 978-5-94387-763-6. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/77578.html> — Режим доступа: для авторизир. Пользователей
38. Беликова, С. А. Основы HTML и CSS: проектирование и дизайн веб-сайтов : учебное пособие по курсу «Web-разработка» / С. А. Беликова, А. Н. Беликов. — Ростов-на-Дону, Таганрог : Издательство Южного федерального университета, 2020. — 174 c. — ISBN 978-5-9275-3435-7. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/100186.html> — Режим доступа: для авторизир. Пользователей
39. Зудилова, Т. В. Web-программирование JavaScript / Т. В. Зудилова, М. Л. Буркова. — Санкт-Петербург : Университет ИТМО, 2012. — 68 c. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/65749.html> — Режим доступа: для авторизир. Пользователей
40. Никольский, А. П. JavaScript на примерах. Практика, практика и только практика / А. П. Никольский. — Санкт-Петербург : Наука и Техника, 2018. — 272 c. — ISBN 978-5-94387-762-9. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/78103.html> — Режим доступа: для авторизир. Пользователей
41. Кулькова, Л. И. Задачи и упражнения по JavaScript : учебное пособие / Л. И. Кулькова, С. И. Салпагаров. — Москва : Российский университет дружбы народов, 2018. — 102 c. — ISBN 978-5-209-08646-8. — Текст : электронный // Электронно-библиотечная система IPR BOOKS : [сайт]. — URL: <https://www.iprbookshop.ru/104199.html> — Режим доступа: для авторизир. Пользователей

# ПРИЛОЖЕНИЕ

### Приложение 1. Листинг

class Program

{

static CEC c;

static CECResult cecResult;

static Convocation convocation;

static DEC dec;

static DECRegion decRegion;

static DECResult decResult;

static Party party;

static PEC pec;

static PECResult pecResult;

static Region region;

static Result result;

static TEC tec;

static TECResult tecResult;

static Actualization act;

static Actualization2 act2;

static void Main(string[] args)

{

c = new CEC();

cecResult = new CECResult();

convocation = new Convocation();

dec = new DEC();

decRegion = new DECRegion();

decResult = new DECResult();

party = new Party();

pec = new PEC();

pecResult = new PECResult();

region = new Region();

result = new Result();

tec = new TEC();

tecResult = new TECResult();

while (true)

{

Console.WriteLine("Введите команду сервера: ");

string[] command = Console.ReadLine().Split(' ');

Sockets s;

switch (command[0])

{

case "server":

{

switch (command[1])

{

case "-r":

{

DataModel dm = new DataModel(c, cecResult, convocation, dec, decRegion, decResult, party, pec, pecResult, region, result, tec, tecResult);

dm.InitialView();

s = new Sockets(dm);

}; break;

case "-s": s = new Sockets(); ; break;

default: Console.WriteLine("Уточните параметры сервера"); break;

}

};break;

case "actualization":

{

switch (command[1])

{

case "xml":

{

Console.Write("VRN: ");

string vrn = Console.ReadLine();

act = new Actualization(c, cecResult, convocation, dec, decResult, party, pec, pecResult, result, tec, tecResult, vrn);

}; break;

case "txt": act2 = new Actualization2(c, cecResult, convocation, dec, decRegion, decResult, party, pec, pecResult, region, result, tec, tecResult); ; break;

default: Console.WriteLine("Уточните параметры парсера"); break;

}

};break;

}

}

}

}

class Actualization

{

CEC cec;

CECResult cecResult;

Convocation convocation;

DEC dec;

DECResult decResult;

DECRegion decRegion;

Party party;

PEC pec;

PECResult pecResult;

Region region;

Result result;

TEC tec;

TECResult tecResult;

DataModel dm;

List<DataFile> dataFile;

DataFile Info;

List<string> TitleDEC;

List<string> TitleTEC;

List<string> TitlePEC;

public Actualization(CEC cec, CECResult cecResult, Convocation convocation, DEC dec, DECRegion decRegion, DECResult decResult, Party party,

PEC pec, PECResult pecResult, Region region, Result result, TEC tec, TECResult tecResult)

{

this.cec = cec;

this.cecResult = cecResult;

this.convocation = convocation;

this.dec = dec;

this.decRegion = decRegion;

this.decResult = decResult;

this.party = party;

this.pec = pec;

this.pecResult = pecResult;

this.region = region;

this.result = result;

this.tec = tec;

this.tecResult = tecResult;

dm = new DataModel(cec, cecResult, convocation, dec, decRegion, decResult, party, pec, pecResult, region, result, tec, tecResult);

dataFile = new List<DataFile>();

ParsFile();

}

public void ParsFile()

{

foreach (string line in System.IO.File.ReadLines(@"DataSet\2021.txt"))

{

dataFile.Add(new DataFile(line.Split('\t')));

}

foreach (string line in System.IO.File.ReadLines(@"DataSet\2021 INFO.txt"))

{

Info = new DataFile(line.Split('\t'));

}

long? cecID;

cecID = cec.ContainConvoction(8);

if (cecID == 0)

{

double NumberVote = dataFile.Sum(x => long.Parse(x.data[4]));

double InvalidBallot = dataFile.Sum(x => long.Parse(x.data[5]));

double ValidBallot = dataFile.Sum(x => long.Parse(x.data[6]));

cecID = cec.Add(new CEC(8, result.Add(new Result(NumberVote, InvalidBallot, ValidBallot))));

for (int i = 9; i <Info.data.Count(); i++)

{

long? partyID = party.Contain(Info.data[i]);

if (partyID == 0)

partyID = party.Add(new Party(Info.data[i], ""));

double partyCount = dataFile.Sum(x => long.Parse(x.data[i]));

double percent = Math.Round((partyCount / (InvalidBallot + ValidBallot)) \* 100, 2);

cecResult.Add(new CECResult(cecID, partyID, partyCount, percent.ToString().Replace(",", ".")));

}

}

TitleDEC = dataFile.Select(x => x.data[1]).Distinct().ToList();

ParsDEC(cecID);

//ParsOneDEC(225, 3);

}

public void ParsDEC(long? cecID)

{

int decID = 0;

for (int i = 0; i < TitleDEC.Count() / Settings.Threads + 1; i++)

{

Thread[] threads;

ThreadParams[] threadParams;

int countThread = TitleDEC.Count() - 1;

threads = new Thread[Settings.Threads];

threadParams = new ThreadParams[Settings.Threads];

int ThreadAll = 0;

int thread = 0;

for (int j = i \* Settings.Threads; j < i \* Settings.Threads + Settings.Threads; j++)

{

threads[thread] = new Thread(ParsThreadDEC);

threadParams[thread] = new ThreadParams

{

cecID = cecID,

id = decID

};

thread++;

ThreadAll++;

//ThreadID++;

decID++;

}

for (int k = 0; k < Settings.Threads; k++) threads[k].Start(threadParams[k]);

for (int k = 0; k < Settings.Threads; k++) threads[k].Join();

dm.SaveAllTAble();

Console.WriteLine($"{i} стак завершил работу");

}

}

private void ParsTEC(long? decID, List<DataFile> TitleTEC)

{

List<DataFile> temp = new List<DataFile>();

List<string> ListTECTitle = TitleTEC.Select(x => x.data[2]).Distinct().ToList();

for (int i = 0; i < ListTECTitle.Count; i++)

{

temp = TitleTEC.Where(x => x.data[2] == ListTECTitle[i]).ToList();

double NumberVote = temp.Sum(x => long.Parse(x.data[4]));

double InvalidBallot = temp.Sum(x => long.Parse(x.data[5]));

double ValidBallot = temp.Sum(x => long.Parse(x.data[6]));

long? tecID;

long? regionID;

lock (tec)

{

lock (result)

{

tecID = tec.Add(new TEC(decID, result.Add(new Result(NumberVote, InvalidBallot, ValidBallot)), ListTECTitle[i], ""));

}

}

lock (region)

{

regionID = region.Contain(temp[0].data[0]);

if (regionID == 0)

regionID = region.Add(new Region(temp[0].data[0]));

}

lock (decRegion)

{

decRegion.Add(new DECRegion(decID, regionID, 0));

}

for (int j = 9; j < Info.data.Count(); j++)

{

long? partyID;

lock (party)

{

partyID = party.Contain(Info.data[j]);

if (partyID == 0)

partyID = party.Add(new Party(Info.data[j], ""));

}

double partyCount = temp.Sum(x => long.Parse(x.data[j]));

double percent = Math.Round((partyCount / (InvalidBallot + ValidBallot)) \* 100, 2);

lock (tecResult)

{

tecResult.Add(new TECResult(tecID, partyID, partyCount, percent.ToString().Replace(",", ".")));

}

}

for (int j = 0; j < temp.Count(); j++)

{

AddPEC(temp[j], tecID);

Console.WriteLine($"{temp[j].data[1]}: {temp[j].data[2]} ({i + 1}/{ListTECTitle.Count}): {temp[j].data[3]} добавлен ({j + 1}/{temp.Count}) ");

}

Console.BackgroundColor = ConsoleColor.Yellow;

Console.WriteLine($"{ListTECTitle[i]} добавлен ({i+1}/{ListTECTitle.Count})");

Console.BackgroundColor = ConsoleColor.Black;

}

}

public void ParsThreadDEC(object o)

{

ThreadParams thread = (ThreadParams)o;

//if (dec.Contain(thread.id.ToString()) == 0)

ParsOneDEC(thread.id, thread.cecID);

Console.BackgroundColor = ConsoleColor.Green;

Console.WriteLine($"DEC {thread.id} добавлен");

Console.BackgroundColor = ConsoleColor.Black;

}

public void ParsOneDEC(int id, long? cecID)

{

List<DataFile> ListTEC = new List<DataFile>();

ListTEC = dataFile.Where(x => x.data[1] == TitleDEC[id]).ToList();

double NumberVote = ListTEC.Sum(x => long.Parse(x.data[4]));

double InvalidBallot = ListTEC.Sum(x => long.Parse(x.data[5]));

double ValidBallot = ListTEC.Sum(x => long.Parse(x.data[6]));

Console.WriteLine("DEC: " + TitleDEC[id]);

long? decID;

lock (dec)

{

lock (result)

{

decID = dec.Add(new DEC(cecID, result.Add(new Result(NumberVote, InvalidBallot, ValidBallot)), TitleDEC[id], id.ToString()));

}

}

for (int j = 9; j < Info.data.Count(); j++)

{

long? partyID;

lock (party)

{

partyID = party.Contain(Info.data[j]);

if (partyID == 0)

partyID = party.Add(new Party(Info.data[j], ""));

}

double partyCount = ListTEC.Sum(x => long.Parse(x.data[j]));

double percent = Math.Round((partyCount / (InvalidBallot + ValidBallot)) \* 100, 2);

lock (decResult)

{

decResult.Add(new DECResult(decID, partyID, partyCount, percent.ToString().Replace(",", ".")));

}

}

TitlePEC = ListTEC.Select(x => x.data[2]).Distinct().ToList();

ParsTEC(decID, ListTEC);

Console.WriteLine($"Загруженно DEC {id} из {TitleDEC.Count()}");

}

private void AddPEC(DataFile df, long? tecID)

{

long? pecID;

lock (result)

{

lock (pec)

{

pecID = pec.Add(new PEC(tecID, result.Add(new Result(double.Parse(df.data[4]), double.Parse(df.data[5]), double.Parse(df.data[6]))), df.data[3], "", df.data[8], df.data[7]));

}

}

for (int j = 9; j < Info.data.Count(); j++)

{

long? partyID;

lock (party)

{

partyID = party.Contain(Info.data[j]);

if (partyID == 0)

partyID = party.Add(new Party(Info.data[j], ""));

}

double partyCount = double.Parse(df.data[j]);

double percent = 0;

if (double.Parse(df.data[5]) == 0 && double.Parse(df.data[6]) == 0)

{

percent = 0;

}

else percent = Math.Round((partyCount / double.Parse(df.data[5] + double.Parse(df.data[6]))) \* 100, 2);

lock (pecResult)

{

pecResult.Add(new PECResult(pecID, partyID, partyCount, percent.ToString().Replace(",", ".")));

}

}

}

}

class TurnoutPECAnaliticView

{

public StringBuilder HTML { get; set; }

public PECResult pecResut;

public PEC PECs;

public Result Results;

public TEC tec;

List<DataAnaliticPEC> data;

MLContext mlContext;

int \_docsize = 36;

List<DataAnalitic> dataAnalitic;

List<ConvocationPercentBallot> convocationPercentBallot;

public List<ConvocationDataAnalitic> convocationDataAnalitic;

public List<ConvocationPrediction> ConvocationPredictions { get; set; }

public TurnoutPECAnaliticView(PECResult pecR, PEC pec, Result res, TEC tec)

{

pecResut = pecR;

PECs = pec;

Results = res;

mlContext = new MLContext();

this.tec = tec;

convocationDataAnalitic = new List<ConvocationDataAnalitic>();

}

public void Fill(List<ConvocationPercentBallot> convocationPercentBallot)

{

this.convocationPercentBallot = convocationPercentBallot;

ConvocationPredictions = new List<ConvocationPrediction>();

for (int i = 0; i < convocationPercentBallot.Count(); i++)

{

data = new List<DataAnaliticPEC>();

foreach (PercentBallots p in this.convocationPercentBallot[i].PercentBallotsPEC)

{

data.Add(new DataAnaliticPEC(p.NumberOfBallots, p.Percent, (double)p.LevelID));

}

StreamWriter sw = new StreamWriter("Data.tsv", false);

sw.WriteLine($"{"CountVote"}\t{"Percent"}\t{"Region"}");

foreach (var p in data)

{

sw.WriteLine($"{p.Data[0]}\t{p.Data[1]}\t{p.Data[2]}");

}

sw.Close();

var predict = Predict();

int aCounrt = predict.Count();

//ConvocationPredictions.Add(new ConvocationPrediction(convocationPercentBallot[i].ConvocationTitle, predict));

convocationDataAnalitic.Add(new ConvocationDataAnalitic(convocationPercentBallot[i].ConvocationTitle, FillDataFromHTL(predict, i)));

}

}

public List<DataAnalitic> FillDataFromHTL(IEnumerable<PredictionAnomaly> data, int index)

{

int i = 0;

dataAnalitic = new List<DataAnalitic>();

foreach (var d in data)

{

if (d.Prediction[0] == 1)

dataAnalitic.Add(new DataAnalitic((PEC)convocationPercentBallot[index].PercentBallotsPEC[i].db, convocationPercentBallot[index].PercentBallotsPEC[i].Level, convocationPercentBallot[index].PercentBallotsPEC[i].UpLevel, Math.Round((1-d.Prediction[2])\*100)));

i++;

}

return dataAnalitic;

}

public IEnumerable<PredictionAnomaly> Predict()

{

IDataView dataView = mlContext.Data.LoadFromTextFile<DataAnaliticPEC>("Data.tsv", hasHeader: true, separatorChar: '\t');

\_docsize = data.Count();

return DetectSpike(mlContext, \_docsize, dataView);

}

public IDataView CreateEmptyDataView(MLContext mlContext)

{

IEnumerable<DataAnaliticPEC> enumerableData = new List<DataAnaliticPEC>();

return mlContext.Data.LoadFromEnumerable(enumerableData);

}

public IEnumerable<PredictionAnomaly> DetectSpike(MLContext mlContext, int docSize, IDataView vote)

{

var sample = mlContext.Transforms.DetectAnomalyBySrCnn(nameof(PredictionAnomaly.Prediction), nameof(DataAnaliticPEC.Data));

var iidSpikeTransform = sample.Fit(vote);

IDataView transformedData = iidSpikeTransform.Transform(vote);

IEnumerable<PredictionAnomaly> predictions = mlContext.Data.CreateEnumerable<PredictionAnomaly>(transformedData, reuseRowObject: false);

return predictions;

}

}

public class VotePartyAnaliticView

{

List<ConvocationPercentBallot> convocationPercentBallots;

List<DataAnaliticVoteParty> dataAnaliticVotes;

List<ConvocationDataAnaliticVoteParty> dataAnalitivVoteParties;

public List<ConvocationDataAnaliticParty> convocationDataAnalitic;

List<DataAnaiticParty> anaiticParties;

MLContext mlContext;

int \_docsize = 36;

public VotePartyAnaliticView(List<ConvocationPercentBallot> convocationPercentBallots)

{

this.convocationPercentBallots = convocationPercentBallots;

convocationDataAnalitic = new List<ConvocationDataAnaliticParty>();

mlContext = new MLContext();

}

public void Fill()

{

dataAnalitivVoteParties = new List<ConvocationDataAnaliticVoteParty>();

for (int i = 0; i < convocationPercentBallots.Count(); i++)

{

dataAnaliticVotes = new List<DataAnaliticVoteParty>();

StreamWriter sw = new StreamWriter("Data.tsv", false);

sw.WriteLine($"{"DEC"}\t{"Turnout"}\t{"Party"}\t{"Percent"}");

foreach (var p in convocationPercentBallots[i].turnoutPercentVoteParties)

{

sw.WriteLine($"{p.DEC.Id}\t{p.Turnout}\t{p.Party.Id}\t{p.Percent}");

// dataAnaliticVotes.Add(new DataAnaliticVoteParty(double.Parse(p.Dec.Id.ToString()), double.Parse(p.Party.Id.ToString()), double.Parse(p.Turnout), double.Parse(p.Percent)));

}

sw.Close();

dataAnalitivVoteParties.Add(new ConvocationDataAnaliticVoteParty(convocationPercentBallots[i].ConvocationTitle, dataAnaliticVotes));

var predict = Predict();

int aCounrt = predict.Count();

convocationDataAnalitic.Add(new ConvocationDataAnaliticParty(convocationPercentBallots[i].ConvocationTitle, FillDataFromHTL(predict, i)));

}

}

public List<DataAnaiticParty> FillDataFromHTL(IEnumerable<PredictionAnomaly> data, int index)

{

int i = 0;

anaiticParties = new List<DataAnaiticParty>();

foreach (var d in data)

{

if (d.Prediction[0] == 1)

anaiticParties.Add(new DataAnaiticParty()

{

Party = convocationPercentBallots[index].turnoutPercentVoteParties[i].Party,

TEC = convocationPercentBallots[index].turnoutPercentVoteParties[i].TEC,

DEC = convocationPercentBallots[index].turnoutPercentVoteParties[i].DEC,

Percent = convocationPercentBallots[index].turnoutPercentVoteParties[i].Percent,

Probability = Math.Round((1 - d.Prediction[2]) \* 100)

});

i++;

}

return anaiticParties;

}

public IEnumerable<PredictionAnomaly> Predict()

{

IDataView dataView = mlContext.Data.LoadFromTextFile<DataAnaliticPEC>("Data.tsv", hasHeader: true, separatorChar: '\t');

\_docsize = dataAnaliticVotes.Count();

return DetectSpike(mlContext, \_docsize, dataView);

}

public IDataView CreateEmptyDataView(MLContext mlContext)

{

IEnumerable<DataAnaliticPEC> enumerableData = new List<DataAnaliticPEC>();

return mlContext.Data.LoadFromEnumerable(enumerableData);

}

public IEnumerable<PredictionAnomaly> DetectSpike(MLContext mlContext, int docSize, IDataView vote)

{

var sample = mlContext.Transforms.DetectAnomalyBySrCnn(nameof(PredictionAnomaly.Prediction), nameof(DataAnaliticPEC.Data));

var iidSpikeTransform = sample.Fit(vote);

IDataView transformedData = iidSpikeTransform.Transform(vote);

IEnumerable<PredictionAnomaly> predictions = mlContext.Data.CreateEnumerable<PredictionAnomaly>(transformedData, reuseRowObject: false);

return predictions;

}

}

public class ParametrsThread

{

public int port;

public string platform;

}

class Sockets

{

DataModel Dm;

public Sockets(DataModel dm)

{

Dm = dm;

HTMLView.Save();

ParametrsThread parametrsPK = new ParametrsThread()

{

port = Settings.PortPK,

platform = "PK"

};

ParametrsThread parametrsAndroid = new ParametrsThread()

{

port = Settings.PortAnroid,

platform = "Android"

};

Thread PK = new Thread(new ParameterizedThreadStart(Server));

Thread Android = new Thread(new ParameterizedThreadStart(Server));

PK.Start(parametrsPK);

Android.Start(parametrsAndroid);

}

public Sockets()

{

HTMLView.Load();

ParametrsThread parametrsPK = new ParametrsThread()

{

port = Settings.PortPK,

platform = "PK"

};

ParametrsThread parametrsAndroid = new ParametrsThread()

{

port = Settings.PortAnroid,

platform = "Android"

};

Thread PK = new Thread(new ParameterizedThreadStart(Server));

Thread Android = new Thread(new ParameterizedThreadStart(Server));

PK.Start(parametrsPK);

Android.Start(parametrsAndroid);

}

public void Server(object o)

{

Socket listenSocket;

ParametrsThread param = (ParametrsThread)o;

IPEndPoint ipPoint = new IPEndPoint(IPAddress.Parse(Settings.IP), param.port);

// создаем сокет

listenSocket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);

listenSocket.Bind(ipPoint);

listenSocket.Listen(10);

Console.WriteLine($"Сервер {param.platform} запущен. Ожидание подключений...");

while (true)

{

Response(param.platform, listenSocket);

}

}

public void Response(string platform, Socket listenSocket)

{

Socket handler = listenSocket.Accept();

StringBuilder builder = new StringBuilder();

int bytes = 0;

byte[] data = new byte[10240];

switch(platform)

{

case "PK":

{

do

{

bytes = handler.Receive(data);

builder.Append(Encoding.Unicode.GetString(data, 0, bytes));

}

while (handler.Available > 0);

};break;

case "Android":

{

do

{

bytes = handler.Receive(data);

builder.Append(Encoding.UTF8.GetString(data, 0, bytes));

}

while (handler.Available > 0);

};break;

}

string response = "";

string[] d = builder.ToString().Replace("\0", "").Replace("\t", "").Replace("\u0010", "").Replace("\u0018", "").Replace("\u0015", "").Replace("$", "").Replace("\v", "").Replace("\f", "").Replace("\u0011","").Split(' ');

Console.WriteLine($"{platform}: {builder.ToString().Replace("\0", "").Replace("\t", "").Replace("\u0010", "")}");

switch (d[0])

{

case "All": response = HTMLView.HTMLAll.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "TurnoutPEC": response = HTMLView.HTMLVotePEC.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "TurnoutDECFunction": response = HTMLView.HTMLVoteDEC.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "TurnoutDECPoint": response = HTMLView.HTMLBallotDEC.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "TurnoutTECFunction": response = HTMLView.HTMLVoteTEC.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "TurnoutTECPoint": response = HTMLView.HTMLBallotTEC.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "Spike": response = HTMLView.HTMLSpikeTurnout.ToString().Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "Party": response = HTMLView.HTMLVoteParty.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); break;

case "SpikeParty": response = HTMLView.HTMLSpikeParty.Replace("HEIGHTWINDOWS", (Math.Round(double.Parse(d[1]))).ToString()); ; break;

case "MapTurnout": response = HTMLView.HTMLMapTurnout; break;

case "MapSpikeTurnout":response = HTMLView.HTMLMapSpikeTurnout;break;

default: {

Console.WriteLine(" Некорректная команда");

};break;

}

switch (platform)

{

case "PK":

{

data = new byte[response.Length\*8];

data = Encoding.Unicode.GetBytes(response);

}; break;

case "Android":

{

List<Byte> temp = data.ToList();

temp.Add((byte)data.Length);

byte t = temp[0];

temp[0] = temp[temp.Count() - 1];

temp[temp[temp.Count() - 1]] = t;

data = temp.ToArray();

data = Encoding.UTF8.GetBytes(response);

}; break;

}

handler.Send(data);

handler.Shutdown(SocketShutdown.Both);

handler.Close();

}

}