

МИНОБРНАУКИ РОССИИ

Федеральное государственное бюджетное образовательное учреждение
высшего образования

**«САРАТОВСКИЙ НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ Н. Г. ЧЕРНЫШЕВСКОГО»**

Кафедра математической кибернетики и компьютерных наук

**РАЗРАБОТКА КЛИЕНТ-СЕРВЕРНОЙ СИСТЕМЫ ДЛЯ АНАЛИЗА
КРИПТОВАЛЮТНОГО РЫНКА: МОБИЛЬНОЕ ПРИЛОЖЕНИЕ И
АЛГОРИТМЫ КОРРЕЛЯЦИОННОГО АНАЛИЗА**

АВТОРЕФЕРАТ БАКАЛАВРСКОЙ РАБОТЫ

студента 4 курса 451 группы
направления 09.03.04 — Программная инженерия
факультета КНиИТ
Чиндина Даниила Михайловича

Научный руководитель
доцент

Б. А. Филиппов

Зав.кафедрой,
к. ф.-м. н., доцент

С. В. Миронов

Саратов 2024

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1 Изучение предметной области	6
1.1 Криптовалюты	6
1.2 Криптовалютная биржа	6
1.3 Биржевой стакан	6
1.4 Курсы криптовалют	6
2 Описание алгоритмов и используемых технологий	8
2.1 Кросс-корреляционный анализ	8
2.1.1 Корреляции Пирсона и Спирмена	8
2.1.2 Функционал модуля кросс-корреляционного анализа	8
2.1.3 Метод вычисления меры кросс-корреляции между двумя временными рядами	9
2.1.4 Алгоритм фильтрации пар временных рядов	9
2.1.5 Метод визуализации кросс-корреляции	9
2.1.6 Метод составления пулов скоррелированных курсов по- сле алгоритма фильтрации	10
2.2 Асинхронный подход к решению задачи скрапинга данных	10
2.2.1 Библиотеки <code>asyncio</code> и <code>aiohttp</code>	10
2.2.2 Преимущества асинхронного подхода над распараллели- ванием действий	10
2.2.3 Пример использования <code>asyncio</code> и <code>aiohttp</code>	11
2.2.4 API бирж для получения данных	11
2.3 Мобильное приложение	11
2.3.1 Протокол <code>WebSocket</code>	11
2.3.2 Библиотека <code>React</code>	11
2.3.3 Хуки	12
3 Разработка модуля кросс-корреляционного анализа и мобильного при- ложения	13
3.1 Корреляционный анализ	13
3.1.1 Метод вычисления меры кросс-корреляции между двумя временными рядами	13
3.1.2 Метод для визуализации кросс-корреляционной функции ..	13
3.1.3 Алгоритм фильтрации пар временных рядов	13

3.1.4	Метод составления пулов скоррелированных курсов	13
3.2	Реализация скраперов	14
3.3	Разработка мобильного приложения.....	14
3.3.1	Токены авторизации	14
3.3.2	Реализации логики API	14
3.3.3	Провайдеры	14
3.3.4	Компоненты	15
3.3.5	Компоновка мобильного приложения.....	15
3.3.6	Конечный результат мобильного приложения.....	15
ЗАКЛЮЧЕНИЕ		16

ВВЕДЕНИЕ

Существующее поле информационных технологий продолжает стремительно расширяться в сторону развития инновационных решений в различных сферах индустрии. Так в последнее время обрели популярность технологии блокчейнов и криптовалют в частности. Развитие этой области в рамках IT привлекло внимание большого количества инвесторов, индивидуальных предпринимателей и крупных компаний, и, как следствие, привело к образованию обособленной части рынка. Как и множество других частей сферы финансов, торговля криптовалютами так же может быть подвержена применению различных методов анализа. Таковыми, например, являются выявление арбитражных ситуаций за счет разницы курсов одних и тех же пар токенов на разных биржах, а также нахождение взаимосвязей между различными криптовалютными активами. Конечно, постоянно работающий с данной предметной областью специалист может довести действия по анализу рынка практически до автоматизма, но возникает вопрос: как обрабатывать большой объем информации о все время пополняющихся новыми экземплярами криптовалютах и все большем количестве ситуаций для анализа рынка? В ручном режиме оценить все возможные варианты образовавшихся на текущий момент таких ситуаций не представляется возможным.

Проблема отсутствия автоматизированного инструмента, предназначенного для анализа криптовалютного рынка, в частности решающего задачу выявления взаимосвязей между валютными парами с помощью встроенного модуля корреляционного анализа и позволяющего работать с этим инструментом с помощью удобного интерфейса мобильного приложения, является актуальной.

Целью данной дипломной работы является реализация модуля кросс-корреляционного анализа при разработке клиент-серверной системы для анализа криптовалютного рынка, а также мобильного приложения, предоставляющего удобный интерфейс для взаимодействия с функционалом системы.

В рамках данной работы были поставлены следующие задачи:

1. разработка скраперов данных о криптовалютных курсах с бирж;
2. детальный анализ поведения рынка для выявления паттернов взаимосвязей между криптовалютными курсами, необходимых при проектировании модуля;
3. проектирование логики работы и реализация модуля корреляционного

- анализа и составления пулов скоррелированных курсов;
4. разработка мобильного приложения для взаимодействия с функционалом клиент-серверной системы.

1 Изучение предметной области

1.1 Криптовалюты

Криптовалюты — это цифровые или виртуальные валюты, которые используют криптографию для обеспечения безопасности транзакций и контроля выпуска новых единиц.

1.2 Криптовбиржа

Криптовбиржа — это платформа, на которой пользователи могут покупать, продавать и обменивать криптовалюты. Они играют ключевую роль в экосистеме криптовалют, предоставляя возможность ликвидности и ценового обнаружения.

Централизованные криптобиржи — это платформы, которые управляются централизованной организацией или компанией. Они контролируют все аспекты торговли и предоставляют пользователям интерфейс для торговли криптовалютами.

Децентрализованные криптобиржи — платформы, которые работают на основе блокчейн-технологий и смарт-контрактов. Они позволяют пользователям торговать криптовалютами напрямую друг с другом без посредников.

Стоит отметить, что при разработке проекта использовался функционал именно централизованных бирж из-за простоты взаимодействия с ними через API.

1.3 Биржевой стакан

Биржевой стакан — это инструмент, который показывает текущие лимитные заявки на покупку и продажу — bid и ask ордера конкретной криптовалюты.

Для корреляционного анализа будет необходима только текущая цена токена, которая будет собираться из информации о лучших ценах из списка bid и ask.

Для анализа арбитражных ситуаций же необходима полная картина происходящего на данный момент времени в стакане.

1.4 Курсы криптовалют

Курс криптовалюты — это текущая стоимость одной единицы этой криптовалюты, выраженная зачастую в другом токене.

Изменения цен курсов могут приводить к возникновению взаимосвязей между парами курсов, что привело к мысли разработки алгоритма подсчета корреляций между курсами криптовалют.

Для простоты дальнейшего анализа набор значений курса некоторой валюты за определенный промежуток времени может быть представлен в виде временного ряда. Для решения задачи по нахождению зависимостей между криптовалютными активами необходимо рассмотреть кросс-корреляцию.

2 Описание алгоритмов и используемых технологий

2.1 Кросс-корреляционный анализ

Корреляция — это статистическая мера, которая описывает степень и направление взаимосвязи между двумя переменными.

Кросс-корреляция — это мера, которая описывает степень взаимосвязи между двумя временными рядами с учетом временного сдвига одного ряда относительно другого.

2.1.1 Корреляции Пирсона и Спирмена

Корреляция **Пирсона** измеряет линейную зависимость между двумя непрерывными переменными.

Формула корреляции Пирсона следующая:

$$r_{xy} = \frac{n \sum xy - \sum x \sum y}{\sqrt{[n \sum x^2 - (\sum x)^2] \cdot [n \sum y^2 - (\sum y)^2]}} \quad (1)$$

Корреляция **Спирмена** измеряет монотонную зависимость между двумя переменными, основываясь на рангах данных, а не на их значениях. Коэффициент ранговой корреляции Спирмена рассчитывается по формуле:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}$$

В связи с тем, что вычисление корреляции Спирмена более требовательно с точки зрения затрат ресурсов на вычисления, а также в связи с отсутствием необходимости анализа нелинейных зависимостей между курсами в текущей конфигурации проекта, было решено в дальнейшем использовать именно корреляцию Пирсона.

2.1.2 Функционал модуля кросс-корреляционного анализа

Блок кросс-корреляционного анализа должен иметь следующий функционал для реализации идеи поиска взаимосвязанных курсов:

1. метод вычисления меры кросс-корреляции между двумя временными рядами, переданными на вход;
2. алгоритм фильтрации пар временных рядов по вычисленным значениям кросс-корреляций;

3. метод визуализации вычисленной кросс-корреляции двух рядов;
4. метод составления пулов скоррелированных курсов после фильтрации;

2.1.3 Метод вычисления меры кросс-корреляции между двумя временными рядами

Приходящие на вход массивы цен курсов каждой пары токенов необходимо преобразовать в формат `ndarray`. Далее для каждого курса в новый словарь заносилась запись следующего вида:

```
{
  Символ_торгуемой_пары: [
    сумма_элементов ,
    сумма_квадратов ,
    ndarray_массив_значений_курса_в_окне
  ]
}
```

Если формула вычисления корреляции выгладит как показано в (1), то существует вариант динамического вычисления корреляции для каждого тайм-лага между двумя рядами. Более подробно алгоритм вычисления кросс-корреляции будет разобран в следующей части работы.

2.1.4 Алгоритм фильтрации пар временных рядов

Должно выполняться сразу несколько условий, чтобы пара курсов прошла этап фильтрации:

1. Модуль максимального значения из списка кросс-корреляций должен превышать определенное установленное значение.
2. Тайм-лаг максимального по модулю значения из списка кросс-корреляций не должен быть по модулю меньше установленного значения.
3. Максимальное по модулю значение кросс-корреляции в определенной заранее окрестности вокруг нулевого тайм-лага не должно также превышать определенного значения.

2.1.5 Метод визуализации кросс-корреляции

Необходимо так же создать инструмент, позволяющий визуализировать динамику изменения корреляции между парами в зависимости от тайм-лага. Для реализации такой функции была выбрана библиотека `matplotlib`, а именно ее модуль `rplot`, методы которого позволяют наносить на фигуру зависимость значений одного массива от другого.

2.1.6 Метод составления пулов скоррелированных курсов после алгоритма фильтрации

Необходима реорганизация структуры пар курсов таким образом, чтобы было возможным решение следующих задач:

1. Составление пулов скоррелированных курсов с выделением ведомых и ведущих курсов внутри с расстановками тайм-лагов между курсами.
2. При нахождении скачков на конкретном курсе из какого-то пула перед созданием сигнала пользователю необходимо проверить дополнительно заранее установленное количество находящихся в том же пуле курсов на соблюдение условия появления скачка в курсе.

2.2 Асинхронный подход к решению задачи скрапинга данных

Необходимо реализовать решение проблемы большого количества запросов к API бирж и длительного ожидания ответов от них, что тормозит работу системы.

2.2.1 Библиотеки `asyncio` и `aiohttp`

Для решения вышеописанных проблем вполне подходят методы таких библиотек, как `asyncio` и `aiohttp`.

`asyncio` — это стандартная библиотека Python для написания асинхронного кода с использованием `async` и `await`, позволяющая реализовывать логику одновременного выполнения задач, не блокируя основной поток программы.

`aiohttp` — библиотека для выполнения асинхронных HTTP-запросов. Она основана на `asyncio` и позволяет легко выполнять асинхронные сетевые операции.

2.2.2 Преимущества асинхронного подхода над распараллеливанием действий

Ниже приведены основные преимущества `asyncio` и `aiohttp` над применением потоков:

- Асинхронный код с `asyncio` и `aiohttp` позволяет запускать тысячи запросов параллельно, не создавая тысячи потоков.
- Асинхронные операции не требуют создания новых потоков и переключения контекста, что уменьшает накладные расходы и повышает эффективность работы.

- Асинхронный код избегает многих сложностей, связанных с многопоточностью, таких как блокировки и гонки данных.

2.2.3 Пример использования `asyncio` и `aiohttp`

В этом разделе приведен фрагмент кода, использующий `asyncio` и `aiohttp` и выполняющий несколько запросов на биржу `Binance`.

2.2.4 API бирж для получения данных

Почти каждая централизованная криптобиржа предоставляет пользователям собственное публичное и приватное `Web-API` для взаимодействия с функционалом удаленно, через логику запросов с помощью стандартных протоколов, таких как `HTTP`.

Скраперы для всех бирж будут реализованы в соответствии со спроектированной структурой, описанной в смежной дипломной работе другого разработчика проекта:

1. Первый этап скрапинга:

- Часть скрапера по получению информации о всех торгуемых парах токенов на бирже.
- Часть скрапера, получающая детальную информацию о торгуемых парах.

2. Второй этап скрапинга: Должен собирать детальную информацию с биржевого стакана, переданной в качестве аргумента торгуемой пары токенов из найденной потенциальной арбитражной цепочки.

2.3 Мобильное приложение

2.3.1 Протокол `WebSocket`

`WebSocket` — это технология, позволяющая устанавливать постоянное соединение между клиентом и сервером для обмена данными в реальном времени.

2.3.2 Библиотека `React`

`React` — это `JavaScript`-библиотека с открытым исходным кодом, разработанная `Facebook`, предназначенная для создания пользовательских интерфейсов. `React` позволяет создавать сложные интерфейсы из небольших, изолированных блоков кода, называемых «компонентами».

2.3.3 Хуки

Хуки в React — это функции, которые позволяют использовать состояние и другие возможности React в функциональных компонентах. Они предоставляют удобный способ управления состоянием и жизненным циклом компонентов без необходимости использовать классы. Основные хуки в React — `useState` и `useEffect`.

`useState` — позволяет добавлять состояние в функциональные компоненты. Он возвращает пару значений: текущее состояние и функцию для его обновления.

`useEffect` — позволяет выполнять побочные эффекты в функциональных компонентах. Это могут быть, например, данные, получаемые с сервера.

3 Разработка модуля кросс-корреляционного анализа и мобильного приложения

3.1 Корреляционный анализ

3.1.1 Метод вычисления меры кросс-корреляции между двумя временными рядами

Данные приходят на вход алгоритму в виде словаря `pair_dict` следующей сигнатуры:

```
{
    Символ_торгуемой_пары: [
        сумма_элементов,
        сумма_квадратов,
        ndarray_массив_значений_курса_в_окне
    ]
}
```

далее эта структура реорганизовывается в следующий вид описанным в данном разделе алгоритмом:

```
{(Пара_1, Пара_2): список_текущих_значений_скалярных_произведений}
```

После чего этот же алгоритм производит динамическое вычисление кросс-корреляции между всеми парами валют.

3.1.2 Метод для визуализации кросс-корреляционной функции

В данном разделе описана реализация метода для визуализации кросс-корреляционной функции пары валют с использованием инструментов модуля `matplotlib.pyplot`, а также произведено сравнение созданного на предыдущем этапе алгоритма с уже имеющейся в методах `NumPy` функцией `corrcoef` при ее применении в вычислении кросс-корреляций.

3.1.3 Алгоритм фильтрации пар временных рядов

В данном разделе приведена практическая реализация алгоритма фильтрации найденных скоррелированных пар курсов по описанной в теоретическом разделе логике.

3.1.4 Метод составления пулов скоррелированных курсов

После этапа фильтрации остается набор пар сильно взаимосвязанных курсов, который необходимо реорганизовать. Это необходимо сделать для дальней-

шей простоты составления сигналов на открытие сделок по сильно скоррелированным парам курсов. В данном разделе приведена реализация алгоритма, проводящего такую реорганизацию.

3.2 Реализация скраперов

В скрапере должно быть реализовано два этапа скрапинга: первый для получения названий торгуемых пар и информации о лучших bid и ask, а второй — для получения подробной информации об ордерах из стакана конкретного курса. В данном разделе приведены реализации обоих методов скраперов на примере API биржи Gate.io.

3.3 Разработка мобильного приложения

При дальнейшем описании этой части работы были указаны только самые необходимые функции, вокруг которых выстраивается вся структура мобильного приложения.

3.3.1 Токены авторизации

Сама структура приложения подразумевает такие функции, как регистрацию и авторизацию пользователей в нем. По логике при авторизации пользователю сервером должен выдаваться индивидуальный токен, который дальше будет использоваться для проверки валидности действий пользователя. В данном разделе приведена реализация создания, получения и удаления токенов авторизации.

3.3.2 Реализации логики API

Вся логика будущего приложения будет выстраиваться вокруг трех базовых конфигураций: `baseService`, `authenticationService` и `refreshService`. Первая будет использоваться при обработке действий клиента уже постфактум его авторизации в приложении, `authenticationService` — при запросах на авторизацию, `refreshService` — при запросах на обновление токена.

3.3.3 Провайдеры

В рамках разработки мобильного приложения удобно было иметь модули с описанием методов взаимодействия с определенным функционалом мобильного приложения. Далее при описании проекта такие блоки функционала будут

называться «провайдерами». В рамках этого раздела в качестве примера описан провайдер `ChainsProvider`, взаимодействующий с блоком информации об арбитражных цепочках.

3.3.4 Компоненты

Визуальная составляющая проектов на React конструируется из компонент. В данном разделе приводится пример реализации с помощью компонент используемых в приложении градиентных кнопок.

3.3.5 Компоновка мобильного приложения

В данном разделе приводится компоновка всех методов и провайдеров в одно исполняемое приложение `App`.

3.3.6 Конечный результат мобильного приложения

В данном разделе приведены описания и скриншоты разработанного мобильного приложения для взаимодействия с функционалом системы.

ЗАКЛЮЧЕНИЕ

В ходе выполнения данной дипломной работы было изучено множество новых технологий, применимых к решению задач разработки мобильных приложений, реализации асинхронных скраперов, анализу специализированных рынков методами математической статистики и data-science. Благодаря детальному изучению предметной области, анализу поведения курсов криптовалют и дальнейшей реализации всех необходимых в итоговой конфигурации системы инструментов получилось достичь поставленной в рамках данной работы цели путем успешного выполнения следующих задач:

1. проектирование и разработка алгоритма нахождения связанных между собой пар курсов путем динамического вычисления кросс-корреляционной метрики;
2. реализация алгоритма составления пулов скоррелированных курсов для дальнейшего построения модуля сигналов;
3. изучение API бирж и разработка классов скраперов с применением асинхронных функций на asyncio и aiohttp для всех бирж, с которыми ведется взаимодействие в рамках проекта;
4. разработка полноценного мобильного приложения для удобного и быстрого взаимодействия с функционалом клиент-серверной системы.

Проектирование и дальнейшая реализация проекта будет продолжаться по завершении данной дипломной работы. Планируется расширение функционала системы посредством разработки алгоритмов анализа резких изменений в ведущих курсах и прогнозирования изменений ведомых, а также усовершенствование алгоритмов кросс-корреляционного анализа.