ПРАВИТЕЛЬСТВО РОССИЙСКОЙ ФЕДЕРАЦИИ
ФГАОУ ВО НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет компьютерных наук
Образовательная программа «Компьютерные науки и анализ данных»

Отчет о программном проекте на тему:

Разработка инструмента для обучения агентов торговле на бирже

(промежуточный, этап 1)

Выполнили студенты:

группы БКНАД212, 3 курса	Рамусь Дмитрий Геннадьевич
группы БКНАД212, 3 курса	Давлетшин Тимур Вилевич
группы БЭК213, 3 курса	Литасов Александр Сергеевич

Принял руководитель проекта:

Горшков Сергей Сергеевич
Приглашённый преподаватель
Факультет компьютерных наук НИУ ВШЭ

Москва 2024

Содержание

Аннотация

Проект направлен на создание пользовательского фреймворка для автоматизации торговли на криптовалютных биржах. Основная цель работы заключается в разработке системы, способной анализировать рыночные данные и новостные потоки, чтобы на их основе формировать и реализовывать торговые стратегии в режиме реального времени. Проект также включает разработку пользовательского интерфейса, который позволит пользователям настраивать параметры торговых стратегий, отслеживать текущее состояние рынка и результаты работы системы.

Система будет использовать алгоритмы обработки естественного языка и машинного обучения для извлечения и анализа ключевых новостей, связанных с криптовалютами. Оценка сентимента этих новостей позволит модели адаптироваться к меняющимся рыночным условиям и предвидеть потенциальные ценовые движения, учитывая реакцию участников рынка на новостные события.

В рамках проекта планируется создание микросервисной архитектуры, которая обеспечит гибкость и масштабируемость системы, а также возможность быстрой интеграции новых функций и сервисов. Ключевые компоненты системы включают в себя модули для сбора и обработки данных с криптовалютных бирж, ML и DL модели для определения будущей справедливой цены (fair price) и модуль реализации торговых операций.

Ключевые слова

Глубинное обучение, разреживание моделей, рекуррентные нейронные сети, embedding слой, микросервис, криптовалюта, торговля на бирже

1 Введение (Литасов Александр)

С развитием технологий и увеличением объемов доступных данных, торговля на финансовых рынках становится всё более зависимой от сложных алгоритмов и автоматизированных систем. В частности, криптовалютный рынок, характеризующийся высокой волатильностью и круглосуточной доступностью, представляет уникальные возможности для алгоритмической торговли. В то же время, этот рынок требует от алгоритмизированных торговых систем способности быстро адаптироваться к изменяющимся условиям и анализировать большие объемы данных в реальном времени.

Важной особенностью современной алгоритмической торговли является использование не только количественных данных о ценах и объемах торгов, но и качественной информации, такой как новостные сообщения, которые могут оказывать значительное влияние на рыночные цены. Интеграция анализа новостного контента в торговые стратегии требует применения методов обработки естественного языка (NLP) и машинного обучения (ML), что увеличивает сложность систем и требует от разработчиков глубоких знаний в области данных и алгоритмов.

При выборе рынка для анализа и торговли перед нами стоял выбор между российским финансовым рынком и рынком криптовалют. Решение было принято в пользу криптовалютного рынка по нескольким причинам. Во-первых, криптовалютный рынок характеризуется более высокой динамикой и объемом новостей, что создает более широкие возможности для анализа сентимента и его влияния на ценовые движения. Во-вторых, на рынке криптовалют значительную долю сделок совершают физические лица, которые в большей степени опираются на новостную информацию при принятии торговых решений. Это делает анализ новостного сентимента особенно актуальным и востребованным инструментом в контексте криптовалютного рынка.

Основной задачей проекта является создание автоматизированной системы для торговли на криптовалютных биржах, способной анализировать как количественные, так и качественные данные, включая новости, и на их основе формировать и реализовывать торговые стратегии в реальном времени. Проект предполагает разработку микросервисной архитектуры, обеспечивающей высокую производительность, масштабируемость и гибкость системы, а также создание удобного пользовательского интерфейса для настройки параметров торговли и мониторинга работы системы.

В рамках проекта будут использованы современные технологии и подходы, включая машинное обучение и обработку естественного языка для анализа новостей, а также различные стратегии и алгоритмы для реализации торговых операций. Ожидается, что разработанная система позволит участникам рынка эффективно управлять своими портфелями, улучшая результаты торговли за счет использования автоматизированного анализа данных и адаптации стратегий к текущим рыночным условиям.

В случае успешной реализации и демонстрации эффективности наших моделей и платформы для автоматизации торговли на криптовалютных биржах, планируется дальнейшее масштабирование проекта. Это будет включать в себя расширение функциональности платформы, улучшение производительности и добавление новых возможностей, соответствующих потребностям и предпочтениям пользователей.

2 Постановка цели и задач (Рамусь Дмитрий)

2.1 Цель проекта

Главная цель проекта заключается в создании автоматизированного инструмента для обучения агента торговле на бирже с последующим возможным примнением новостей в качестве улучшения точности предсказывания.

2.2 Задачи

- Реализовать связку микросервисов для сборки и обработки данных: 1) исторических, такие как лоты, свечи и т.д. с криптобиржи; 2) новостных с профильных и не только ресурсов (tradingview.com, cryptonews-api.com, соцсети и остальные источники данных); а также для обучения модели и применения её в реальном трейдинге
- Использовать существующую, написать свою или воспользоваться предсказанной стратегией от нейронной сети в алгортрейдинге
- Автоматизировать управление проектом благодаря использованию систем оркестрации, CI/CD и мониторинга
- Создать клиентский интерфейс для комфортной и быстрой работы с проектом
- Проведение эксперемента с обучением и использованием нейронной сети на основе архитектуры transformer на виртуальном счете со сторны пользователя, для проверки работоспособности и переходу к масштабируемости всего проекта

3 Распределение ролей в команде

Распределение ролей и обязанностей на курсовом проекте произошло в соответствии с сильными сторонами каждого из участников. Выбор лидера был произведён голосованием.

1 Рамусь Дмитрий

- Основные: бэкенд и инфраструктура, управление проектом
- Дополнительные: ML

2 Тимур Давлетшин

- Основные: ML и тестирование гипотез
- Дополнительные: backend

3 Литасов Александр

- Основные: Аналитика и подбор стратегий
- Дополнительные: ML и документирование

4 Обзор литературы (Давлетшин Тимур)

- Исследована статья *Can ChatGPT Forecast Stock Price Movements?* [5], где авторы применили версию ChatGPT v4 для анализа новостей на американском фондовом рынке и последующего осуществления торговых операций на его основе. В результате годового эксперимента была получена прибыль в размере 500%, что существенно превысило показатели индекса NASDAQ, потерявшего в цене 27% за аналогичный период. Авторы осуществляли торговлю вручную, не прибегая к автоматизации процесса. Такой подход предполагает зависимость проекта от работоспособности единственного инструмента, что является его уязвимой точкой.

- Проведен анализ работы *Giving Content to Investor Sentiment: The Role of Media in the Stock Market* [8], в котором изучали влияние тональности новостей на динамику фондового рынка. В результате была выявлена прямая корреляция между настроением, передаваемым средствами массовой информации, и торговыми операциями на фондовом рынке.

- Статья *LSTM Architecture for Oil Stocks Prices Prediction* [2] посвящена использованию архитектуры долгосрочной кратковременной памяти (LSTM [3]) в рамках рекуррентных нейронных сетей (RNN [4]) для прогнозирования цен на акции нефтяных компаний. Анализируя влияние глобально значимых индексов, таких как цены на WTI, золото и доллар, авторы пришли к выводу об ограниченном их воздействии на точность прогнозов. В заключение подчеркивается, что несмотря на отсутствие интерпретируемости RNN, данное исследование иллюстрирует потенциал методов машинного обучения для прогнозирования фондового рынка и указывает на возможное влияние косвенных факторов на точность прогнозов.

5 System Design (Рамусь Дмитрий)

5.1 Архитектура ПО

Архитектура ПО системы задает ее структуру и механизмы взаимодействия компонентов, влияя на ее производительность, качество и надежность. Разработчики часто использовали монолитную архитектуру, где приложение является единым блоком с централизованной функциональностью и логикой, что облегчает быструю разработку и деплоймент, но увеличивает риски при внесении изменений. Микросервисная архитектура, предлагающая разделение на независимые сервисы с отдельной логикой и базами данных, обеспечивает гибкость в обновлении, тестировании, развертывании и масштабировании.

В нашем проекте правильным вариантом использования является микросервисная архитектура, так как она уменьшает риски сбоев всего цикла работы, в отличие от монолитной архитектуры.

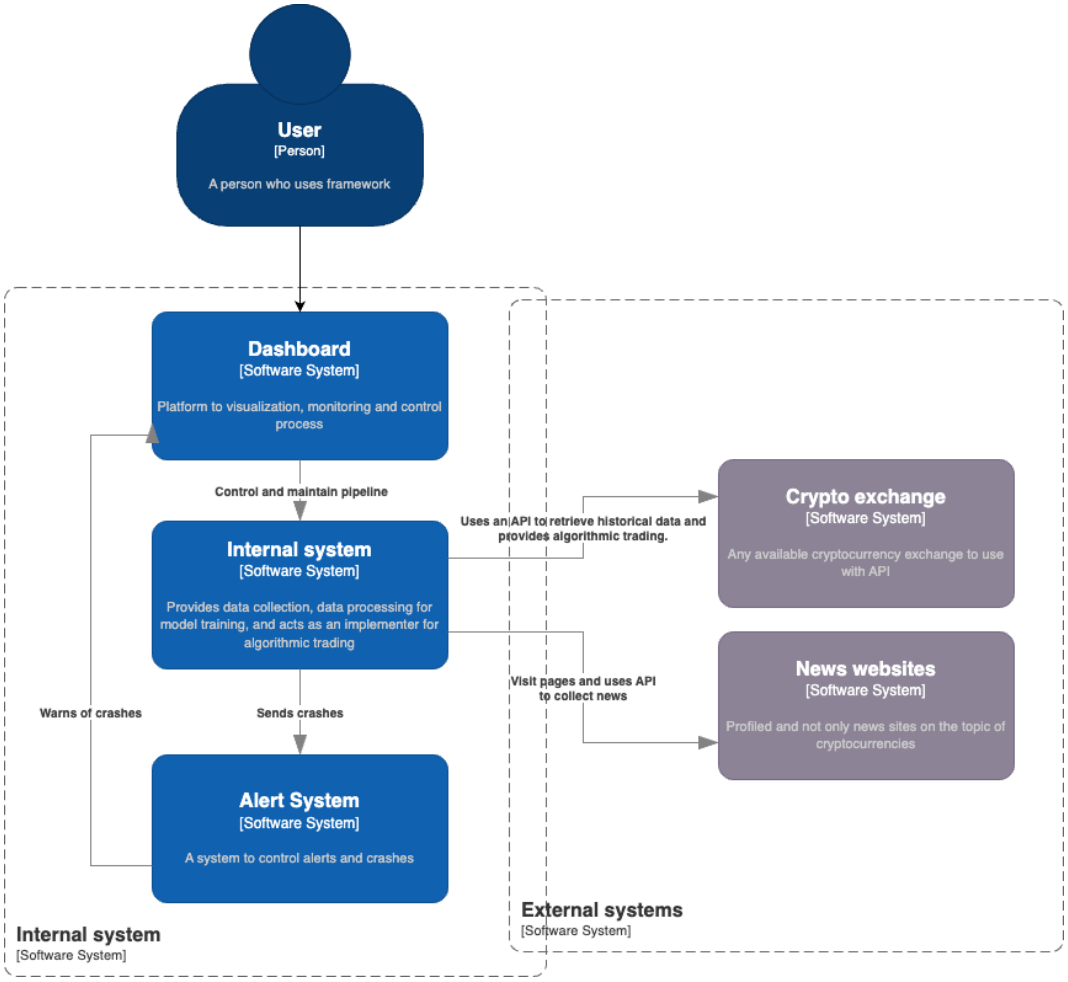

Рис. 5.1: C4: контекстная диаграмма

5.1.1 Выбор инструментов

В рамках разработки курсового программного проекта была принята микросервисная архитектура, обеспечивающая гибкий выбор технологий для реализации разнообразных задач. В проекте активно используются языки программирования Python и Go, каждый из которых выбран с учетом его специфических преимуществ и областей применения.

Для работы с нейронными сетями будет использован Python, так как это самый развитый язык программирования в сфере машинного и глубокого обучения. В качестве инструмента для написания и обучения моделей будем использовать Pytorch [7] - самый популярный и гибкий фреймворк глубокого обучения нейросетей.

В контексте задач, связанных с взаимодействием с биржевыми платформами и требующих высокоскоростной обработки транзакций для среднечастотного трейдинга, предпочтение было отдано языку Go. Выбор обусловлен его прекрасными возможностями по многопоточности и параллелизму и высокой скорости разработки, в отличие например от C++. Go также обеспечивает высокую производительность. Таким образом, Go способствует созданию эффективных микросервисов, способных оперативно обрабатывать большие данные и реагировать на рыночные изменения, быстро выполняя транзакции в режиме реального времени.

5.2 Выбор архитектуры модели (Давлетшин Тимур)

Проанализировав задачу, мы пришли к выводу, что мы имеем дело с предсказанием цены на основе истории торгов и новостей. Это есть ничто иное как предсказание временного ряда с использованием текста. Для такой задачи подойдет модель на основе RNN/transformer [9]. Но так как мы сильно ограничены по времени исполнения модели, то было решено отбросить RNN подобные архитектуры, так как они имееют асимптотику времени выполнения хуже, нежели transformer подобные. Часть нейросети, принимающая текста, будет основана на embedding слоях - их смысл перевод слов (дискретных величин) в векторное представление [1].

5.3 Способы взаимодействия и обмена данными

API (Application Programming Interface) определяется как комплекс правил и методов, обеспечивающих взаимодействие и обмен данными между программами [11]. Рассмотрим наиболее известные типы API:

- REST — архитектурный подход к созданию веб-сервисов, использующий HTTP для коммуникации. Это позволяет клиентам получать доступ к ресурсам, управлять ими и получать ответы в JSON или XML. REST API выделяется своей гибкостью, масштабируемостью и безопасностью передачи данных.
- Websockets — поддерживает двустороннее соединение между клиентом и сервером с минимальной задержкой, позволяя обмениваться данными в реальном времени без перезагрузки страницы.
- gRPC (Google Remote Procedure Call) — технология, позволяющая выполнять вызов удаленных процедур на сервере. Она обеспечивает высокую производительность, хотя данные в ней представлены в формате, сложном для чтения человеком, что усложняет анализ и тестирование.

На этапе MVP проекта предпочтение отдается REST API, так как он удовлетворяет всем нашим требованиям. Возможно использование gRPC с целью оптимизации скорости взаимодействия между микросервисами. Для обеспечения обмена данными в реальном времени используется протокол WebSockets.

5.4 Система управления базами данных (СУБД)

PostgreSQL [10], как свободная объектно-реляционная СУБД, предлагает структурированное хранение данных в таблицах, доступ к которым осуществляется через SQL запросы. Эта система выделяется своей популярностью, бесплатностью и удобством в использовании.

Преимущества PostgreSQL перед MySQL и MongoDB для mid frequency algo trading на основе анализа новостей включают:

- Надежность и безопасность: PostgreSQL выделяется своей надежностью и безопасностью, предоставляя механизмы для гарантии целостности данных, что критически важно для финансовых приложений, где точность данных имеет первостепенное значение.
- Масштабируемость: Способность к обработке больших объемов данных благодаря эффективной репликации и кластеризации делает PostgreSQL идеальным выбором для торговли, где объемы данных могут быть значительными и требуют быстрого обновления и доступа.
- Расширенные возможности SQL: Поддержка сложных SQL-операций, таких как оконные функции, CTE и полнотекстовый поиск, предоставляет гибкие инструменты для анализа и обработки данных, что необходимо при работе с новостными потоками и их влиянием на торговые стратегии.

Эти аспекты делают PostgreSQL более предпочтительным для применения в проекте алгоритмической торговли, использующую новостные данные. В отличие от MySQL, который может ограничиваться в сложности запросов и масштабируемости, и MongoDB, предназначенного для работы с документо-ориентированными данными и не предлагающего столь же мощные SQL-возможности, PostgreSQL предлагает баланс между надежностью, масштабируемостью и глубиной аналитических возможностей, необходимых для эффективной обработки и анализа данных в алготрейдинге.

5.5 CI/CD

В нашем проекте применяется система непрерывной интеграции и доставки (CI/CD) от GitLab, обеспечивающая автоматизацию ряда задач: проверка кода с помощью линтера для обнаружения и устранения ошибок, компиляция и сборка исходного кода в Docker [6] образы, их тегирование и отправка в Container Registry. Это позволит нам быстро в будущем развертывать обновления в Kubernetes, улучшая производительность и безопасность приложения. Автоматический запуск пайплайна при внесении изменений в репозиторий гарантирует, что новые версии кода тестируются, собираются и разворачиваются эффективно и без задержек.

5.6 Микросервисы

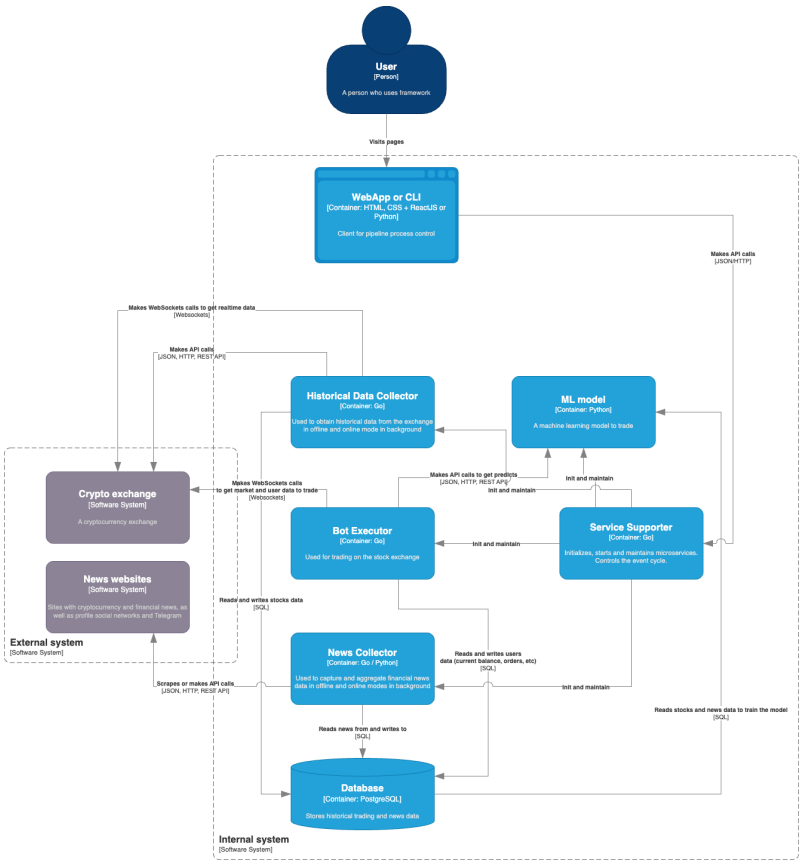

Рис. 5.2: C4: контейнерная диаграмма

- WebApp или CLI: Это клиентский интерфейс для управления процессами продукт. Пользователь посещает страницы или использует консольную утилиту, для взаимодействия с компонентами.
- ML модель: Это центральный компонент, который анализирует данные и принимает торговые решения.
- Bot Executor: Используется для торговли на криптобирже (Binance, Bybit или аналоги).
- Service Supporter: Компонент, который инициализирует, поддерживает и контролирует цикл событий микросервисов.
- Historical Data Collector: Этот компонент собирает исторические данные с биржи для их использования в режимах онлайн, подписываясь на обновления через websockets и оффлайн, делая запросы через REST API.
- News Collector: Собирает новостные данные в режимах аналогичных Data Collector: онлайн и оффлайн.

6 Планируемые результаты и дальнейший план работ

Этап	**Дата**	**Статус**
Постановка цели и задачи	2023.12.12	Выполнено
Обзор существующих решений	2023.12.20	Выполнено
Сравнительный анализ новостных источников	2024.01.10	Выполнено
Выбор рынка и источника данных	2024.01.25	Выполнено
Реализация MVP бэкенда и инфраструктуры	2024 февраль - март	В процессе
Реализация MVP нейронной модели и тестирование гипотез	2024 февраль - март	В процессе
Реализация клиентского интерфейса	2024 февраль - март	
Поиск закономерностей, использование новостных данных и написание стратегии, тестирование	2024 март - апрель	
В случае позитивных результатов эксперемента - мастабирование на большее количество крипто-монет.	2024 март - апрель	
Создание финальной версии	2024 апрель	
Защита проекта	2024 май	

Таблица 6.1: План работ

Список литературы

[1] Antoine Bordes, Jason Weston, Ronan Collobert и Yoshua Bengio. “Learning Structured Embeddings of Knowledge Bases”. В: *Proceedings of the AAAI Conference on Artificial Intelligence* (2011). URL: https://api.semanticscholar.org/CorpusID:715463.

[2] Javad T. Firouzjaee и Pouriya Khaliliyan. *The Interpretability of LSTM Models for Predicting Oil Company Stocks: Impact of Correlated Features*. 2023. arXiv: 2201.00350 [q-fin.ST].

[3] Sepp Hochreiter и Jürgen Schmidhuber. “Long Short-Term Memory”. В: *Neural Computation* 9.8 (нояб. 1997), с. 1735—1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. eprint: https://direct.mit.edu/neco/article-pdf/9/8/1735/813796/neco.1997.9.8.1735.pdf. URL: https://doi.org/10.1162/neco.1997.9.8.1735.

[4] J J Hopfield. “Neural networks and physical systems with emergent collective computational abilities.” В: *Proceedings of the National Academy of Sciences* 79.8 (1982), с. 2554—2558. DOI: 10.1073/pnas.79.8.2554. eprint: https://www.pnas.org/doi/pdf/10.1073/pnas.79.8.2554. URL: https://www.pnas.org/doi/abs/10.1073/pnas.79.8.2554.

[5] Alejandro Lopez-Lira и Yuehua Tang. *Can ChatGPT Forecast Stock Price Movements? Return Predictability and Large Language Models*. 2023. arXiv: 2304.07619 [q-fin.ST].

[6] Dirk Merkel. “Docker: lightweight linux containers for consistent development and deployment”. В: *Linux journal* 2014.239 (2014), с. 2.

[7] Adam Paszke, Sam Gross, Soumith Chintala, Gregory Chanan, Edward Yang, Zachary DeVito, Zeming Lin, Alban Desmaison, Luca Antiga и Adam Lerer. “Automatic differentiation in PyTorch”. В: (2017).

[8] Paul C. Tetlock. “Giving Content to Investor Sentiment: The Role of Media in the Stock Market”. В: *The Journal of Finance* 62(3) (2005), с. 1139—1168. URL: https://api.semanticscholar.org/CorpusID:18509529.

[9] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser и Illia Polosukhin. *Attention Is All You Need*. 2017. arXiv: 1706.03762 [cs.CL].

[10] *СУБД PostgreSQL: почему её стоит выбрать для работы с данными и как установить*. practicum.yandex.ru. URL: https://practicum.yandex.ru/blog/chto-takoe-subd-postgresql/ (дата обр. 14.12.2022).

[11] *Что такое API?* Режим доступа: свободный. aws.amazon.com. URL: https://aws.amazon.com/ru/what-is/api/ (дата обр. 08.01.2023).