Федеpальное госудаpственное обpазовательное бюджетное

учpеждение высшего обpазования

**«Финансовый унивеpситет пpи Пpавительстве Pоссийской Федеpации»**

**(Финансовый университет)**

Факультет информационных технологий и анализа больших данных

Кафедра искусственного интеллекта

Выпускная квалификационная работа на тему: «Разработка системы алгоритмической торговали криптовалютой с применением ML-технологий»

Направление подготовки 01.03.02 «Прикладная математика и информатика»

Профиль «Машинное обучение»

Выполнил студент группы ПМ21-3

Разумовский Борис Николаевич

Руководитель

к.т.н. доцент

Лабинцев Андрей Иванович

**ВКР соответствует предъявляемым требованиям**

Заведующий кафедрой

К.э.н.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ М. В. Коротеев

« »\_\_\_\_\_\_\_\_\_\_\_\_\_ 20 г.

Москва 2025

Оглавление

[ВВЕДЕНИЕ 4](#_Toc197934797)

[Глава 1. Теоретические основы алгоритмической торговли 7](#_Toc197934798)

[История развития торговых роботов 7](#_Toc197934799)

[1.2 Объект и предмет исследования 9](#_Toc197934800)

[1.3 Торговая стратегия 11](#_Toc197934801)

[1.4 Математическая постановка задачи 12](#_Toc197934802)

[1.5 Торговые индикаторы 14](#_Toc197934803)

[1.6 Методы дообучения LLM 17](#_Toc197934804)

[1.8 Как работает градиентный бустинг 21](#_Toc197934805)

[1.9 Основная проблема исследования 22](#_Toc197934806)

[Глава 2. Построение алгоритма и анализ данных 24](#_Toc197934807)

[2.1 Обзор монеты DOGECOIN 24](#_Toc197934808)

[2.2 Торговая стратегия алгоритма 26](#_Toc197934809)

[2.3 Расчёт средней цены входа 28](#_Toc197934810)

[2.4 Шаг объема по открытию следующих позиций 30](#_Toc197934811)

[2.5Определение шага для депозита 32](#_Toc197934812)

[2.6 Динамическое определение Take Profit. Улучшение алгоритма. 35](#_Toc197934813)

[2.7 Не линейное распределение капитала 36](#_Toc197934814)

[2.8 Ограничение алгоритма 40](#_Toc197934815)

[2.9 Создание LLM модели. 41](#_Toc197934816)

[2.10 Сбор датасета 42](#_Toc197934817)

[2.11 Обучение LLM 45](#_Toc197934818)

[2.11 Классический подход 47](#_Toc197934819)

[2.12 Внедрение модели в алгоритм 50](#_Toc197934820)

[2.13 Улучшение алгоритма с ML моделью 51](#_Toc197934821)

[Глава 3. Анализ полученных результатов. 54](#_Toc197934822)

[3.1 Результаты экспериментов 54](#_Toc197934823)

[3.2 Практические рекомендации 56](#_Toc197934824)

[ЗАКЛЮЧЕНИЕ 57](#_Toc197934825)

[СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ 58](#_Toc197934826)

[ПРИЛОЖЕНИЕ 60](#_Toc197934827)

ВВЕДЕНИЕ

Алгоритмическая торговля представляет собой один из самых современных подходов к управлению финансовыми активами, основанный на автоматизации анализа рынков и исполнения сделок. Особую значимость данный подход приобретает в контексте криптовалютного рынка, который отличается высокой волатильностью, круглосуточной доступностью и стремительным развитием.

Актуальность исследования обусловлена растущей ролью криптовалют в глобальной экономике, их преимуществами перед традиционными активами (децентрализация, прозрачность транзакций, низкие комиссии) и необходимостью быстрой реакции на динамичные изменения рыночной ситуации. Применение алгоритмической торговли в сочетании с технологиями машинного обучения (ML) позволяет оперативно обрабатывать большие массивы данных, точно прогнозировать рыночные движения и минимизировать влияние человеческого фактора.

Степень разработанности данной тематики подтверждается наличием широкого круга исследований и прикладных решений в области алгоритмической торговли, охватывающих как простые технические стратегии, так и сложные архитектуры с применением нейронных сетей. Однако интеграция ML-моделей в торговые системы, функционирующие на криптовалютных рынках, по-прежнему требует дополнительного анализа и адаптации, учитывая нестабильность и высокую чувствительность этих рынков к внешним факторам. Практика показывает, что уже существующие решения нередко страдают от переобучения, недостаточной адаптивности или завышенных ожиданий от моделей, не проходящих тестирование в условиях приближенных к реальному времени.

Объектом исследования в данной работе выступает криптовалютный рынок как динамично развивающаяся среда для реализации алгоритмической торговли. В качестве предмета исследования рассматриваются торговые алгоритмы, использующие методы машинного обучения для автоматизации анализа рыночных данных и выработки торговых решений в условиях высокой волатильности и неопределенности. Эти алгоритмы должны быть способными адаптироваться к изменяющейся структуре рынка и обеспечивать устойчивость при разнообразных рыночных сценариях.

**Цель исследования** — разработка и повышение эффективности торгового алгоритма автоматической торговли криптовалютами с применением методов машинного обучения.  
**Задачи исследования:**

1. Разработать торговый алгоритм на языке Python
2. Протестировать работоспособность алгоритма на обширных исторических данных.
3. Оптимизировать алгоритм и улучшить качество прогнозов с использованием ML-моделей (нейронные сети, деревья решений, методы глубокого обучения).
4. Оценить эффективность системы в условиях, приближенных к реальному рынку, и предложить направления для дальнейшего улучшения.

Круг рассматриваемых проблем охватывает разнообразие подходов к построению торговых стратегий, анализ их адаптивности и устойчивости, а также интеграцию интеллектуальных технологий для повышения точности торговых сигналов. В ходе исследования особое внимание уделяется сравнению эффективности различных ML-моделей и обоснованию выбора конкретных методов на основе эмпирических данных и анализа рыночных условий.

Основные положения, обоснованию которых посвящена работа, включают обоснование применимости алгоритмической торговли на криптовалютных рынках, целесообразность использования ML-моделей для повышения точности и адаптивности торговых стратегий, а также подтверждение работоспособности и результативности предложенной системы на примере исторических и синтетических данных.

Практическая значимость исследования выражается в возможности применения разработанного алгоритма для генерации пассивного дохода в условиях реального рынка. Его внедрение может быть полезным как для частных инвесторов, стремящихся минимизировать риски и автоматизировать свою торговую деятельность, так и для профессиональных участников рынка, заинтересованных в повышении эффективности существующих стратегий. Кроме того, результаты работы могут стать основой для дальнейших исследований и развития инновационных решений в области финансовых технологий и управления цифровыми активами.

Глава 1. Теоретические основы алгоритмической торговли

История развития торговых роботов

Алгоритмическая торговля (алготрейдинг) — это автоматизированный процесс исполнения торговых заявок с использованием компьютерных программ, основанных на математических алгоритмах. Она позволяет трейдерам минимизировать человеческий фактор, повысить скорость и точность операций, что особенно важно на высоковолатильных рынках, таких как криптовалютные. История алготрейдинга охватывает несколько десятилетий, начиная с внедрения электронных бирж и заканчивая современными системами с искусственным интеллектом.

Развитие алгоритмической торговли началось в 1970-х годах с появлением электронных бирж, таких как NASDAQ, которые автоматизировали обработку торговых заявок. В 1984 году Нью-Йоркская фондовая биржа (NYSE) внедрила системы DOT и SuperDOT, ускорившие исполнение ордеров и заложившие основу для автоматизации торговых процессов [1]. Эти технологии позволили сократить время обработки заявок и уменьшить ошибки, связанные с ручным вводом данных.

В 1990-х годах крупные институциональные инвесторы начали использовать автоматические торговые системы (АТС), которые выполняли функции советников, анализируя рыночные сигналы. Полная автоматизация стала возможной благодаря развитию интернета и улучшению коммуникационных технологий, что позволило торговым платформам оперативно взаимодействовать с биржевыми терминалами[2]. Эти системы были первыми шагами к созданию сложных алгоритмов, способных самостоятельно принимать торговые решения.

К 2000-м годам алгоритмическая торговля пережила бум благодаря высокочастотной торговле (HFT), которая предполагает выполнение тысяч сделок за доли секунды. К 2009 году HFT составляла около 73% объема торгов на американских фондовых рынках [3]. В России, по данным Московской биржи, в 2015 году доля алгоритмических операций достигала 30% на основном рынке, 60% на рынке деривативов (45% — HFT) и 62% на валютном рынке. Успех таких систем демонстрирует хедж-фонд Renaissance Technologies, который в 2008 году заработал 2,8 миллиарда долларов, используя исключительно алгоритмические стратегии .

Развитие алготрейдинга было поддержано регуляторными изменениями. В США Regulation ATS (1998), decimalization (2000) и Regulation NMS (2005) увеличили конкуренцию и стимулировали рост HFT. В Европе директива MiFID (2007) сыграла аналогичную роль[4]. Биржи также инвестировали в технологии: NYSE внедрила автоматическое котирование в 2003 году, Deutsche Boerse снизила латентность до 10 мс в 2007 году, а NYSE Euronext — до 100–400 мкс в 2008–2009 годах, что позволило HFT доминировать на рынках.

С 2010 года алготрейдинг начал интегрировать машинное обучение и искусственный интеллект для анализа больших данных, включая новостные потоки и социальные медиа. В 2012 году компания Dataminr разработала систему, которая генерировала торговые сигналы из социальных сетей, опережая традиционные источники на 54 минуты[3]. В 2023 году системы на базе ИИ, такие как ChatGPT, начали создавать торговые стратегии, повышая адаптивность алгоритмов к рыночным условиям.

Алгоритмическая торговля сопряжена с рисками, такими как программные сбои и рыночные аномалии. Флэш-крэш 2010 года, вызвавший крупнейшее внутридневное падение индексов США, и инцидент Knight Capital 2012 года, приведший к убыткам в 460 миллионов долларов за 45 минут, подчеркивают необходимость строгого тестирования и регулирования [4].

Таблица 1.1 Ключевые этапы развития алгоритмической торговли:

|  |  |  |
| --- | --- | --- |
| Период | Событие/Развитие | Пример/Статистика |
| 1970-е | Появление электронных бирж | NASDAQ; DOT и SuperDOT на NYSE (1984) |
| 1990-е | Первые АТС для институциональных инвесторов | Ограниченные функции советников |
| 2000-е | Рост HFT | 73% объема торгов в США (2009); 60% деривативов в России (2015) |
| 2010-е | Интеграция новостей и соцмедиа | Dataminr (2012), опережение на 54 мин. |
| 2020-е | Применение ИИ | ChatGPT для торговых стратегий (2023) |

История алгоритмической торговли демонстрирует эволюцию от простых систем автоматизации до сложных алгоритмов с машинным обучением, таблица 1.1 это хорошо демонстрирует. Регуляторные изменения и технологические инновации сыграли ключевую роль в ее развитии, но риски, связанные с программными сбоями, требуют дальнейшего совершенствования систем контроля.

1.2 Объект и предмет исследования

Современные торговые алгоритмы можно классифицировать по различным признакам, включая структуру стратегии, способ принятия решений и уровень автоматизации. Рассмотрим основные типы [5]:

1. Сеточные боты автоматически размещают ордера на покупку и продажу на определённых ценовых уровнях, создавая "сетку" ордеров. Они особенно эффективны на рынках с боковым движением, где цена колеблется в пределах узкого диапазона.

**Преимущества:**

* Автоматизация торговли без необходимости постоянного мониторинга.
* Возможность получения прибыли на колебаниях цены в узком

**Недостатки:**

* Неэффективность в условиях сильного тренда или высокой волатильности.
* Риск значительных убытков при резких ценовых движениях.

1. **Не сеточные торговые боты -** эти боты основываются на различных стратегиях [5], таких как трендовые, скальпинг, арбитраж и другие. Они анализируют рыночные данные и принимают решения о входе и выходе из сделок на основе заданных алгоритмов.

**Преимущества:**

* Гибкость в выборе стратегий и адаптация к различным рыночным условиям.
* Возможность использования технических индикаторов и других аналитических инструментов.

**Недостатки:**

* Необходимость регулярной настройки и оптимизации параметров.
* Риск переоптимизации и снижения эффективности стратегии со временем.

3. Торговые боты с применением искусственного интеллекта (AI Trading Bots**) -** Эти боты используют методы машинного обучения и искусственного интеллекта для анализа больших объёмов данных [6], включая исторические цены, новости и поведение рынка. Они способны адаптироваться к изменяющимся условиям рынка и улучшать свои стратегии со временем.

**Преимущества**:

* Способность выявлять сложные паттерны и зависимости в данных.
* Адаптация к новым рыночным условиям без необходимости ручной настройки.

**Недостатки**:

* Высокая сложность разработки и необходимость значительных вычислительных ресурсов.
* Риск переобучения модели и снижение её эффективности на новых данных.

Большинство существующих торговых ботов были разработаны для традиционных финансовых рынков, таких как Forex. Однако криптовалютный рынок обладает рядом уникальных характеристик:

* **Высокая волатильность**: Цены на криптовалюты могут резко меняться в короткие промежутки времени, что увеличивает риск убытков при использовании традиционных стратегий.
* **Круглосуточная торговля**: В отличие от традиционных рынков, криптовалютный рынок работает 24/7, что требует постоянного мониторинга и адаптации стратегий.
* **Отсутствие регулирования:** Многие криптовалютные биржи не регулируются так строго, как традиционные финансовые институты, что увеличивает риски для трейдеров.

Из-за этих особенностей многие традиционные торговые боты показывают низкую эффективность на криптовалютном рынке. Кроме того, количество эффективных торговых алгоритмов, использующих методы машинного обучения, остаётся ограниченным.

1.3 Торговая стратегия

Дисциплинированность в трейдинге играет ключевую роль и имеет огромное значение для достижения успеха [8]. Благо, что роботам не присущи эмоций человека, они действуют согласно прописанным инструкциям. Потому, очень важно чаятельно подойти к торговой стратегии.

В рамках данной работы будет рассмотрен алгоритм, основанный на методе «сеточной торговли» (Grid Trading). Данный алгоритм разрабатывается для спотового рынка. Это означает, что реализуется концепция: «дешево купил – дорого продал». Обратные сделки - «дорого купил, дешево продал», свойственные фьючерсным рынкам в данной работе не рассматриваться.

Сеточная торговля предполагает использование системы ордеров, размещённых на определённых ценовых уровнях. Если после открытия позиции цена актива движется в противоположном направлении, выставляется дополнительный ордер, что позволяет усреднить стоимость позиции. Такой подход смещает среднюю точку входа, снижая амплитуду движения цены, необходимую для достижения безубыточности или фиксации прибыли.

Однако реализация сеточной стратегии сопряжена с рядом важных вопросов, требующих тщательного анализа:

1. Определение момента входа в позицию.

2. Расстояние между ордерами для усреднения.

3. Увеличение объёма позиции в последующих сделках. Стоит ли наращивать объём при добавлении новых ордеров, и если да, то по какому принципу — фиксированный шаг, прогрессия (например, по Мартингейлу) или иной метод?

4. Точка выхода из позиции. Как определить момент закрытия сделки?

Каждый из этих вопросов является очень важным, основополагающим для стратегии. Сеточная торговля, несмотря на свою кажущуюся простоту, является сложным инструментом, который при неправильном использовании может привести к значительным убыткам, особенно в случае затяжного тренда против позиции.

1.4 Математическая постановка задачи

Сеточные торговые алгоритмы демонстрирует высокую эффективность в условиях консолидации рынка, известной как «боковик» или «флет» (далее — «боковик»), когда цены колеблются в узком диапазоне без выраженного направления. Однако в периоды сильных направленных движений, или трендов, производительность алгоритма существенно снижается, что приводит к затежным сделкам.

Первичный подход, предполагающий использование алгоритмических условий для определения рыночного состояния, сталкивается с серьезными ограничениями. Крипто рынки характеризуются высокой степенью вариативности: тренды могут быть как постепенными и устойчивыми, так и резкими и волатильными. Формализация таких состояний с помощью жестких правил оказывается практически невыполнимой, поскольку рыночные условия изменяются во времени и зависят от множества факторов.

Задача заключается в том, чтобы внедрить в торговый алгоритм элемент, способный действовать аналогично "мозгу трейдера", то есть наделить его способностью принимать решения на уровне человеческого опыта и интуиции. Идея состоит в очеловечивании алгоритма за счет интеграции моделей искусственного интеллекта (LLM), что позволит эффективно распознавать и адаптироваться к различным состояниям рынка. В результате такой подход должен повысить эффективность торговли, автоматически деактивируя алгоритм в периоды тренда и активируя его исключительно при боковом движении рынка.

Однако, если применение подхода на основе LLM окажется неэффективным или технически неосуществимым, задача сведется к разработке специализированной системы, способной идентифицировать наличие тренда на крипторынке. В этом случае основная функция будет заключаться в автоматическом отключении торгового механизма при выявлении тренда и его активации в условиях боковика.

Для оценки эффективности разработанного торгового алгоритма необходимо ввести систему количественных метрик, позволяющих объективно сравнивать различные модели и подходы. Ключевыми показателями качества алгоритма в данном исследовании предлагается считать следующие:

1. F(x) → max — совокупная прибыль на исторических данных (бэктесте).  
   Эта метрика отражает общую доходность алгоритма за выбранный период тестирования. Она позволяет оценить способность стратегии генерировать устойчивую прибыль в условиях различной рыночной динамики.
2. N — максимальное количество одновременно открытых ордеров.  
   Особенно актуальный параметр при оценке работы сеточных стратегий. Большое значение N может указывать на агрессивное поведение алгоритма и потенциально высокий уровень риска, связанный с перегрузкой депозита и снижением ликвидности.
3. Среднее время удержания позиции.  
   Данная метрика демонстрирует, насколько долго алгоритм, в среднем, удерживает открытые сделки.
4. Максимальное время удержания одной позиции (в днях).  
   Позволяет выявить случаи, когда алгоритм "застревает" в убыточной или неоптимальной сделке.

Совокупный анализ этих метрик позволяет всесторонне оценить качество и стабильность торгового алгоритма, выявить потенциальные риски и определить направления для дальнейшей оптимизации. В частности, важным аспектом будет стремление к максимизации прибыли при одновременном контроле за числом одновременных позиций и средним временем удержания сделок — показателях, тесно связанных с рисками и ликвидностью стратегии.

1.5 Торговые индикаторы

Алгоритмическая торговля невозможно без использования торговых индикаторов. Именно на их работе строиться большинство торговых алгоритмов. В данном исследовании также будут использоваться торговые индикаторы.

Индикатор Bollinger Bands (BB) предназначен для анализа распределения цен относительно их среднего значения и оценки текущей волатильности рынка. Он состоит из трёх линий: средней полосы, верхней полосы и нижней полосы. Средняя полоса представляет собой простую скользящую среднюю (SMA), которая отражает среднюю цену актива за определённый период. Верхняя и нижняя полосы рассчитываются с учётом стандартного отклонения от средней линии, что позволяет им динамически адаптироваться к изменениям волатильности: они расширяются в периоды высокой активности рынка и сужаются при снижении волатильности.

Стандартные настройки индикатора Bollinger Bands включают следующие параметры:

1. Средняя полоса: 20-дневная простая скользящая средняя (SMA), которая служит базовой линией.
2. Верхняя полоса: рассчитывается как сумма 20-дневной SMA и удвоенного значения 20-дневного стандартного отклонения.
3. Нижняя полоса: определяется как разность между 20-дневной SMA и удвоенным значением 20-дневного стандартного отклонения.

Формально эти линии выражаются следующими математическими формулами:

1. Средняя полоса (SMA):

(1)

где— период усреднения, ​ — цена закрытия на i-й день.

1. Верхняя полоса:

(2)

где— 20 -дневное стандартное отклонение цен, ​ — стандартный множитель.

1. Нижняя полоса:

(3)

где— 20 -дневное стандартное отклонение цен, ​ — стандартный множитель.

1. Стандартное отклонение () рассчитывается по формуле:

(4)

Индикатор Bollinger Bands предоставляет трейдерам чёткие сигналы для входа в позицию на основе положения цены относительно полос. На рисунке 1 изображен пример работы индикатора Bollinger Bands. Красная стрелочка сигнал к продаже, белая – к покупке.

1 - пример работы индикатора Bollinger Bands.

Основной принцип интерпретации заключается в следующем:

* Если цена пересекает нижнюю полосу и находится ниже неё, это указывает на состояние «перепроданности» актива. В таких условиях рынок часто демонстрирует стремление к возврату к средней линии, что интерпретируется как сигнал к покупке (Buy).
* Если цена пересекает верхнюю полосу и оказывается выше неё, это свидетельствует о состоянии «перекупленности». В данном случае предполагается вероятность коррекции вниз, к средней линии, что служит сигналом к продаже (Sell).

Использование индикатора Bollinger Bands в сеточной торговле позволяет определить начальную точку входа и направление сделки. Например, при касании ценой нижней полосы трейдер может открыть длинную позицию (покупку), ожидая роста к средней линии. Аналогично, при пробое верхней полосы открывается короткая позиция (продажа).

Следующий важный индикатор, который также применяется в данном исследовании является индикатор Average True Range (ATR). Индикатор ATR, разработанный Дж. Уэллсом Уайлдером, измеряет среднюю истинную волатильность актива за определённый период времени. Его суть заключается в определении диапазона ценового движения с учётом не только разницы между максимальной и минимальной ценой за период, но и возможных разрывов (гэпов) между закрытием предыдущего периода и открытием текущего. Это делает ATR более точным показателем волатильности по сравнению с простым диапазоном цен.

Истинный диапазон (, TR) рассчитывается как максимальное значение из трёх величин:

1. Разница distractions между максимальной и минимальной ценой текущего периода (5)

2. Абсолютная разница между максимальной ценой текущего периода и ценой закрытия предыдущего периода . (6)

3. Абсолютная разница между минимальной ценой текущего периода и ценой закрытия предыдущего периода (7)

Формула для :

(8)

На основе истинного диапазона рассчитывается сам ATR как экспоненциальная скользящая средняя (EMA) значений TR за выбранный период (обычно 14 дней):

(9)

где — текущий истинный диапазон, — период (обычно 14).

С помощью значения ATR в дальнейшем алгоритм будет строить сетку из ордеров.

1.6 Методы дообучения LLM

За последнее время LLM проделали большой путь в своем развитии. Сейчас многие генеративные модели умеют хорошо рассуждать и решать сложные математические задачи. LLM применяется во многих сферах, в том числе и в финансах. Однако готовой модели, которая способно торговать или хотя бы качественно проводить анализ крипто рынка в свободном доступе нет. Поэтому, возникает вопрос, как же сделать такую LLM? Существует несколько подходов с дообучением моделей. Рассмотрим наиболее распространены RAG (Retrieval-Augmented Generation) и Fine-Tuning.

RAG представляет собой метод, сочетающий возможности языковой модели с механизмом поиска и извлечения информации из внешних источников. В процессе работы RAG модель сначала запрашивает релевантные данные из базы знаний (например, текстовых документов, баз данных или других структурированных источников), а затем использует эти данные для генерации более точного и информативного ответа. Этот подход особенно эффективен в задачах, где требуется доступ к актуальной или специализированной информации, которая может отсутствовать в обучающих данных модели.

RAG является хорошим решением в ситуациях, когда модель должна предоставлять ответы на основе конкретных данных, например, при работе с корпоративной документацией или базами знаний. Однако RAG имеет ограничения в задачах, требующих глубокого анализа или интерпретации сложных данных, таких как графики, диаграммы или другие визуальные представления. Для таких случаев модель должна быть способна не только извлекать информацию, но и понимать контекст, связанный с визуальными данными, что выходит за рамки возможностей стандартного RAG без дополнительной доработки.

Fine-Tuning — это процесс дообучения предварительно обученной языковой модели на специализированном наборе данных для адаптации к конкретной задаче. Традиционный подход к Fine-Tuning предполагает обновление всех весов модели, что позволяет достичь высокой точности в узкоспециализированных задачах. Однако этот метод имеет существенные недостатки: высокую вычислительнцю стоимость Длительность обучения и Риск переобучения.

Fine-Tuning с полным обучением всех весов оправдан в редких случаях, когда требуется радикальная адаптация модели к новой предметной области. Однако в большинстве сценариев, особенно при ограниченных ресурсах, этот подход оказывается неэффективным.

Наиболее перспективным и эффективным методом для повышения точности LLM является LoRA (Low-Rank Adaptation). LoRA предлагает альтернативу полному Fine-Tuning, позволяя адаптировать модель к конкретной задаче с минимальными затратами ресурсов. Вместо обновления всех весов модели LoRA добавляет небольшие матрицы низкого ранга к определенным слоям трансформера, что позволяет "настроить" модель, сохраняя ее базовые знания. Преимущества LoRA включают:

* Экономия ресурсов: LoRA требует значительно меньше вычислительных мощностей и памяти, так как обновляется лишь небольшая часть параметров модели.
* Скорость обучения: Благодаря компактности обновляемых параметров процесс дообучения занимает гораздо меньше времени по сравнению с полным Fine-Tuning.
* Гибкость: LoRA позволяет создавать несколько адаптированных версий модели для разных задач, не затрагивая оригинальные веса. Это упрощает тестирование и развертывание.
* Сохранение обобщающей способности: Поскольку базовые веса модели остаются неизменными, LoRA минимизирует риск переобучения и сохраняет способность модели к обработке широкого спектра запросов.

1.7 Механизм работы LoRA

Большие языковые модели, основанные на архитектуре трансформера, содержат миллиарды параметров, представленных в виде весовых матриц. Например, в слоях трансформера (таких как полносвязные слои или механизмы внимания) используются матрицы весов которые определяют, как входные данные преобразуются в выходные. Полное дообучение требует обновления всех элементов W, что вычислительно дорого.

LoRA предполагает, что изменения, необходимые для адаптации модели к новой задаче, могут быть выражены через низкоранговое обновление матрицы весов. Вместо изменения всей матрицы W LoRA добавляет к ней матрицу обновления , которая имеет низкий ранг. Это позволяет значительно сократить количество обучаемых параметров, сохраняя при этом способность модели адаптироваться к новой задаче.

В LoRA матрица обновления представляется в виде произведения двух низкоранговых матриц:

(10)

где:

* — матрица, проецирующая входные данные в пространство меньшей размерности ( r );
* — матрица, возвращающая данные в исходное пространство;
* ( r ) — ранг матрицы
* Итоговая матрица весов, используемая моделью во время инференса или обучения, выглядит следующим образом:

(11)

Здесь ( W ) — это оригинальная (замороженная) матрица весов, которая не изменяется в процессе дообучения, а ( A ) и ( B ) — обучаемые параметры, которые оптимизируются для конкретной задачи.

Во время дообучения с использованием LoRA:

* Оригинальные веса W остаются неизменными, что снижает требования к памяти и предотвращает деградацию базовых знаний модели.
* Оптимизируются только матрицы A и B, которые содержат значительно меньше параметров, чем W
* Обучение проводится стандартными методами оптимизации (например, Adam), минимизируя функцию потерь для целевой задачи.

Несмотря на свои преимущества, LoRA имеет ограничения. Низкий ранг r может ограничивать выразительную способность обновления , что в некоторых случаях приводит к менее точной адаптации по сравнению с полным Fine-Tuning. Однако, LoRA представляет собой эффективный метод дообучения больших языковых моделей, основанный на низкоранговом обновлении весовых матриц. Используя формулу, LoRA позволяет адаптировать модель к новым задачам, минимизируя вычислительные затраты и сохраняя оригинальные знания модели.

1.8 Как работает градиентный бустинг

Также в исследовании применяться и классический метод машинного обучения - **градиентный бустинг. «**Бустинг (объединение) - это комбинация множества простых слабых моделей (обычно небольших деревьев решений), где каждая новая исправляет ошибки предыдущих. Каждая следующая модель пытается улучшить результат, фокусируясь на недочётах своих предшественников.» [9] В результате ансамбль корректирует неточности шаг за шагом, постепенно приближаясь к минимальной ошибке на тренировочных данных.

Пусть у нас есть выборка:

(12)

где — признаки, а — истинное значение (целевая переменная, например, изменение цены).

Модель строится в виде суммы слабых моделей (например, деревьев решений):

(13)

где:

— итоговая модель,

— слабый ученик (обычно дерево),

— вес модели.

На каждом шаге , новая модель обучается приближать **антиградиент функции потерь:**

(14)

где:

— функция потерь (например, квадратичная или логистическая),

— псевдо-остатки (ошибки), на которых обучается новая модель.

Каждая новая модель минимизирует:

(15)

после чего ансамбль обновляется:

(16)

Градиентный бустинг представляет собой мощный инструмент для построения точных торговых сигналов в условиях алгоритмической торговли. Его устойчивость к выбросам особенно важна в условиях высокой рыночной волатильности. Способность учитывать сложные нелинейные зависимости между рыночными признаками делает его эффективным при анализе сложных паттернов поведения рынка. Кроме того, градиентный бустинг хорошо справляется с небалансированными данными, что критично при прогнозировании редких событий, таких как резкие ценовые импульсы или сигналы на вход в позицию. Гибкость настройки — через глубину деревьев, скорость обучения и количество итераций — позволяет контролировать переобучение и адаптировать модель под особенности конкретного торгового инструмента. В рамках данного исследования градиентный бустинг будет использован для анализа и прогнозирования направлений движения цены на основе исторических рыночных данных.

1.9 Основная проблема исследования

На сегодняшний день одной из ключевых проблем в сфере алгоритмической торговли является отсутствие прозрачных и объективных способов оценки качества торговых роботов, разработанных другими участниками рынка. Несмотря на большое количество публично доступных стратегий и ботов, размещённых на различных платформах и форумах, их эффективность остаётся под вопросом.

Во-первых, подавляющее большинство таких алгоритмов реализует крайне простые и базовые торговые идеи, зачастую основанные на одном-двух технических индикаторах. Это делает их уязвимыми к рыночным шумам и снижает устойчивость при изменении рыночной структуры.

Во-вторых, более продвинутые торговые роботы часто предлагаются за значительную плату, однако их внутренние принципы работы остаются закрытыми. Это исключает возможность объективного анализа, сравнения и верификации заявленных результатов. При этом цена не всегда коррелирует с качеством решения.

В-третьих, стоит отметить, что на текущий момент крайне мало торговых систем, в которых применяются методы машинного обучения. Даже если такие алгоритмы и существуют, они, как правило, не имеют открытого исходного кода и не сопровождаются документацией, позволяющей понять, как именно используются ML-модели в процессе принятия торговых решений.

Таким образом, отсутствие проверяемых решений с использованием современных технологий, таких как искусственный интеллект и машинное обучение, значительно осложняет задачу выбора и внедрения эффективного торгового алгоритма. Именно по этой причине в рамках данного исследования предпринимается попытка разработать, обучить и протестировать алгоритмическую торговую систему, в основе которой лежит использование методов машинного обучения. Цель — определить, может ли ML-технология действительно повысить эффективность торговли и сформировать устойчивую стратегию в условиях высокой рыночной волатильности.

Глава 2. Построение алгоритма и анализ данных

2.1 Обзор монеты DOGECOIN

Dogecoin — это криптовалюта, которая первоначально создавалась в качестве шуточного проекта, вдохновлённого популярным мемом с собакой породы сиба-ину. Однако со временем Dogecoin превратился в полноценный финансовый феномен. История криптовалюты началась в 2013 году, когда разработчики Билли Маркус и Джексон Палмер решили предложить рынку менее серьёзную альтернативу биткоину. Благодаря активному сообществу и широкому освещению в СМИ, в том числе упоминаниям со стороны таких влиятельных фигур, как Илон Маск, Dogecoin приобрёл большую популярность и значительно превзошёл изначальные ожидания своих создателей.

Особую роль в судьбе Dogecoin играет Илон Маск, который неоднократно выражал поддержку этой криптовалюте через свой аккаунт в Twitter, провоцируя значительные скачки цены. Кроме того, Маск входит в консультативный орган под названием «Комитет эффективности D.O.G.E.», официально признанный правительством США, что придает дополнительный вес и значимость Dogecoin в финансовом сообществе.

Характерной особенностью Dogecoin является его высокая волатильность. Цена этой криптовалюты способна изменяться на сотни процентов за короткий период, что делает её крайне чувствительной к публикациям в социальных сетях и общим рыночным настроениям. Именно благодаря такой нестабильности Dogecoin представляет собой привлекательный инструмент для трейдеров и инвесторов.

Учитывая высокую волатильность Dogecoin и его значимость в финансовом сообществе, именно на основе этой криптовалюты в данном исследовании будет разрабатываться и тестироваться специализированный торговый алгоритм. Для проверки эффективности алгоритма будут использоваться исторические данные за четырёхлетний период с 19 ноября 2020 года по 19 ноября 2024 года.

Рассмотрим график монеты DOGEUSDT за 22.11.2020-22.11.2024

1 - график монеты DOGEUSDT за 22.11.2020-22.11.2024

График подчёркивает выраженную волатильность Dogecoin, что делает его подходящим объектом для применения сеточной стратегии, но одновременно требует внимательного подхода к управлению рисками.

Глядя на график, можно заметить как сильно подорожал монета. Минимальная цена актива за анализируемый период составила , а максимальная . Если бы трейдер знал эти экстремумы и вложил весь капитал в покупку на минимуме с последующей продажей на максимуме, можно было бы рассчитать гипотетическую прибыль следующим образом:

1. Количество монет при покупке на минимуме:

(17)

2. Прибыль при продаже на максимуме:

После, трейдер продаёт монеты на .

Revenue==1,325,757.58⋅0.7≈928,030.31$ (18)

Это означает, что чистая прибыль могла бы составить около , что значительно превышает потенциал торгового алгоритма.

На первый взгляд такой подход кажется привлекательным, однако он основан на ретроспективном знании глобального минимума и максимума, что в реальных условиях недоступно трейдеру. В момент совершения сделок точные точки разворота рынка неизвестны, и решение о покупке или продаже во многом зависит от удачи. В отличие от этого, торговая стратегия, использующая индикаторы Bollinger Bands и ATR, работает независимо от долгосрочных трендов, адаптируясь к текущей волатильности и обеспечивая последовательность действий. Алгоритм позволяет фиксировать прибыль в условиях как восходящего, так и нисходящего рынка, минимизируя зависимость от точного предсказания экстремумов.

Таким образом, хотя гипотетический сценарий с полной инвестицией на минимуме и продажей на максимуме демонстрирует впечатляющий результат, он не применим в реальной торговле из-за отсутствия предсказуемости. Сеточная стратегия, напротив, предлагает системный подход, где шаг депозита будет оптимизирован с учётом исторических данных.

* 1. Торговая стратегия алгоритма

Торговая стратегия алгоритма представляет собой набор простых правил. Если цена пересекает нижнюю полоску индикатора BBands, то открывается ордер на покупку.

Если начинается длинное затяжное движение, и цена актива начинает дешеветь, то каждая последующая сделка открывается по формуле:

(19)

Где N – это количество уже открытых позиций, а ATR – значение индикатора. Таким образом, линейно увеличивая сетку ордеров, можно избежать излишнего открытия сделок. На рисунке 2 изображено неправильное построение сетки. На рисунке 3 изображено линейное построение сетки.



3 – Линейное построение сетки

2 – Неправильное построение сетки

Однако, все не так прост с определением точки выхода. Совершенно неочевидно, где фиксировать прибыль. Существует несколько подходов к определению уровня Take Profit в рамках предложенной стратегии, использующей индикатор Bollinger Bands (BBands). Среди возможных вариантов:

1. Верхняя полоса Bollinger Bands: Этот уровень может служить целью в случае значительного роста цены, однако он часто оказывается слишком амбициозным, особенно в условиях умеренной волатильности.
2. Средняя полоса Bollinger Bands: Представляет собой 20-дневную простую скользящую среднюю (SMA) и отражает точку равновесия, к которой рынок стремится вернуться после отклонений.
3. Промежуточный уровень между средней и верхней полосами: Компромиссный вариант, который может учитывать как возврат к средней линии, так и потенциал более сильного движения.

Эмпирическим путем было выяснено, что использование средней полосы Bollinger Bands в качестве уровня Take Profit является оптимальным решением. Это обусловлено тем, что рынок, как правило, демонстрирует тенденцию к возвращению к средней линии после выхода за пределы верхней или нижней полосы, что делает её достижимой и стабильной целью.

После определения средней полосы BBands как целевого уровня возникает вопрос: как рассчитать Take Profit для сетки, состоящей из нескольких ордеров, учитывая их усреднённую цену входа? Для этого был разработан специальный алгоритм.

2.3 Расчёт средней цены входа

В рамках сеточной стратегии, при открытии нескольких ордеров по различным ценам, для корректного определения уровня фиксации прибыли необходимо учитывать не только цену первого входа, но и общий объём инвестиций, а также количество приобретённых активов. В этой связи используется расчет средней цены входа, которая позволяет адаптировать стратегию Take Profit к изменяющимся рыночным условиям и структуре открытых позиций.

Средняя цена входа определяется как отношение общей суммы вложенных средств к общему количеству купленных монет:

(20)

где ​ — сумма, потраченная на покупку в -м ордере, а — количество приобретённых монет в том же ордере. Такая формула позволяет учесть не только количество сделок, но и их объём, обеспечивая точное определение точки равновесия позиции.

Для определения целевого уровня выхода используется процентное соотношение между желаемым уровнем Take Profit и ценой первого ордера. Этот коэффициент рассчитывается по формуле:

(21)

— значение средней полосы Bollinger Bands в момент открытия позиции, а — цена первого ордера. Полученный коэффициент сохраняется для последующего пересчета уровня Take Profit после усреднения позиции.

При изменении средней цены входа в процессе добавления новых ордеров целевой уровень Take Profit пересчитывается с сохранением ранее установленного процента прибыли:

(22)

Такой подход позволяет адаптировать выход из позиции в соответствии с рыночной ситуацией и при этом сохранять заданный уровень доходности.

Сохранение пропорционального соотношения между входом и выходом позволяет не только повысить общую эффективность стратегии, но и минимизировать риски при реализации глубокой сетки. Это особенно актуально при работе с инструментами, подверженными резким колебаниям, такими как криптовалюты.

2.4 Шаг объема по открытию следующих позиций

Еще одним важным элементом в сеточной торговой стратегии является настройка шага открытия следующего ордера. Этот параметр определяет, каким образом будет увеличиваться объём позиции при добавлении новых ордеров, что напрямую влияет на распределение капитала, управление рисками и общую эффективность алгоритма. Существует несколько подходов к определению объёма каждой последующей сделки, каждый из которых имеет свои преимущества и ограничения. Выбор оптимального метода требует тщательного анализа, учитывающего как рыночные условия, так и доступный капитал трейдера.Возможные подходы к определению шага:

1. Геометрическое увеличение объёма (удвоение)

В этом случае объём каждой следующей сделки увеличивается в два раза по сравнению с предыдущей. Например, если первый ордер открыт на сумму , то второй будет на , третий — на и так далее. Формула для объёма го ордера:

(22)

где — номер ордера. Этот подход, известный как стратегия Мартингейла, позволяет быстро усреднить позицию, но резко увеличивает потребность в капитале и повышает риски.

2. Линейное увеличение объёма (фиксированный прирост)

Здесь к объёму каждой следующей сделки добавляется фиксированная величина ( k ). Например, если первый ордер открыт на , то второй — на третий — на Формула:

(23)

Такой метод обеспечивает более плавное наращивание позиции, но может быть менее эффективным в условиях сильных ценовых движений.

3. Фиксированный объём (константный депозит)

При этом подходе каждая сделка открывается на одинаковую сумму независимо от номера ордера:

(24)

Этот метод прост в реализации и позволяет равномерно распределять капитал, однако он может замедлить процесс усреднения при значительном падении цены.

Независимо от выбранного подхода, необходимо учитывать ограничения капитала трейдера. Сетка должна быть рассчитана таким образом, чтобы сумма средств, выделенных на все ордера, не превышала доступный депозит. Формально это можно выразить как:

(25)

где:

го ордера,

— общее количество ордеров,

— весь доступный капитал.

Например:

- Для геометрического увеличения: ,

- Для линейного увеличения,

- Для фиксированного объёма:

Таким образом, зная размер капитала, можно определить максимальное число ордеров для каждого подхода.

Как определить оптимальный подход и количество ордеров? Выбор наилучшего метода и оптимального числа ордеров зависит от множества факторов: волатильности актива, размера депозита, ожидаемой глубины просадки и целевого уровня прибыли. Геометрическое увеличение подходит для агрессивных стратегий с высоким риском и большим капиталом, линейное — для более сбалансированного подхода, а фиксированный объём — для консервативной торговли с ограниченными средствами. Точный ответ на вопрос об оптимальном шаге и количестве ордеров требует эмпирической проверки. Анализ результатов позволит выявить, какой подход обеспечивает наилучшее соотношение доходности и риска, а также определить, сколько ордеров сетка может включать без превышения капитала.

2.5Определение шага для депозита

Как уже отмечалось ранее, выбор шага депозита представляет собой критически важный параметр сеточной торговой стратегии, определяющий эффективность распределения капитала и управление рисками. Подробное обоснование его значимости было изложено в предыдущих разделах, поэтому здесь акцент будет сделан на практическом аспекте оптимизации. На основе ранее описанных принципов был разработан и запрограммирован торговый алгоритм, который подвергся тестированию на исторических данных пары DOGEUSDT за период с 2021 по 2024 годы. Результаты тестирования показали, что максимальное количество ордеров в сетке, необходимое для покрытия наблюдаемой волатильности, составило 10. Это значение было определено с учётом шага, рассчитанного как где достигает 10, что даёт достаточно широкий диапазон для усреднения позиций. Однако для повышения надёжности и подготовки к нештатным ситуациям, таким как неожиданные скачки волатильности, целесообразно предусмотреть возможность открытия до 11 ордеров.

На следующем этапе необходимо зафиксировать базовый параметр — общий торговый капитал. В рамках данного исследования для всех последующих расчётов будет использоваться условное значение капитала, равное 3850 долларов США. Данное значение выбрано произвольно и может быть адаптировано в реальных условиях под любой доступный бюджет.

Поставленная задача теперь сводится к построению арифметической прогрессии, в которой каждый последующий ордер увеличивает объём инвестиций на фиксированную величину.

Пусть:

— начальный депозит для первого ордера,

— фиксированный шаг увеличения депозита для каждого следующего ордера,

— общее количество ордеров (в данном случае до 11).

Сумма депозитов для всех ордеров должна быть ограничена доступным капиталом в размере. Формула суммы арифметической прогрессии для ордеров выглядит следующим образом:

(26)

(27)

где:

- Первый ордер: ,

- Второй ордер:,

- Третий ордер:

- …,

- ).

Для сумма депозитов должна удовлетворять уравнению:

(28)

Необходимо подобрать значения и при условии (начальный депозит должен превышать шаг, чтобы обеспечить логическую последовательность роста). Разделим уравнение на 11 для упрощения:

(29)

Отсюда:

(30)

Поскольку должно быть положительным, а также сумма депозитов не должна превышать 3850 $:

1 (31)

2 (32)

3. (шаг должен быть положительным). (33)

Наибольшее значение определяется минимальным из ограничений: Таким образом, должно быть целым числом, меньшим 58. Попробуем подобрать оптимальные значения:

- Если

(34)

Проверка суммы:

(35)

Сетка

- Если

(36)

Проверка суммы:

(37)

Сетка: ( 75, 130, 185, 240, 295, 350, 405, 460, 515, 570, 625 ). Условие ( S > K ) (75 > 55) выполнено.

- Если (проверка границы):

но что противоречит условию. Сумма:

(38)

Однако недопустимо, поэтому исключается.

Наилучший вариант определяется балансом между начальным депозитом и шагом, обеспечивающим равномерное распределение капитала. При , сетка выглядит логично и удовлетворяет всем условиям:

- Начальный депозит достаточно велик для первой сделки,

- Шаг позволяет плавно наращивать позицию,

- Полное использование капитала (3850 $).

Альтернатива с , также допустима, но меньший начальный депозит может быть менее устойчивым при высоковолатильных активах. Таким образом, оптимальными значениями считаются:

-

- .

Предложенная арифметическая прогрессия и обеспечивает эффективное распределение капитала в 3850 $ на 11 ордеров, сохраняя условие и готовность к нештатным ситуациям.

На основе выбранных параметров — начального депозита S = 100 $ и шага K = 50 $ — был протестирован торговый алгоритм. Результаты работы алгоритма представлены в таблице 3 под названием А1 – алгоритм первый [54].

Полученный алгоритм с описанными выше параметрами построения сетки ордеров, увеличения объема последующих сделок продемонстрировал неплохую доходность в 25 тысяч долларов за 4 года. Более подробно с алгоритмом можно будет ознакомиться в приложении. В следующих абзацах будет проведена работа над улучшением алгоритмической составляющей кода и повышение общей доходности.

2.6 Динамическое определение Take Profit. Улучшение алгоритма.

Полученная чистая прибыль в размере 25 тысячи долларов США за четырёхлетний период тестирования торгового алгоритма на исторических данных пары DOGEUSDT представляет собой значительный финансовый результат. Однако возникает вопрос: как можно улучшить результат? Есть три ключевые области, которые могут быть улучшены для повышения эффективности и увеличения прибыли алгоритма:

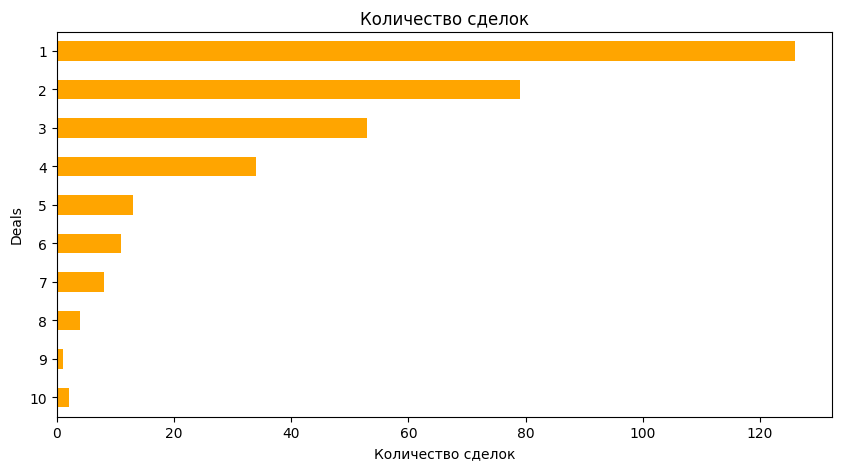
* Изменение шага сетки:Однако ранее было установлено, что линейное построение сетки с шагом, основанным на является оптимальным.
* Изменение уровня Take Profit: Адаптация целевого уровня выхода из позиции может повлиять на общую доходность, особенно в условиях значительных рыночных движений.
* Изменение размера сделки: Корректировка начального депозита и шага депозита позволяет перераспределить капитал, что может повысить потенциальную прибыль при сохранении устойчивости сетки.

Начнём с анализа второго направления — изменения уровня Take Profit, поскольку линейное построение сетки было признано оптимальным и не подлежит пересмотру на данном этапе. Текущая стратегия использует среднюю полосу Bollinger Bands в качестве целевого уровня Take Profit. Однако эмпирическое наблюдение за поведением алгоритма выявило интересную закономерность: в случаях, когда сетка включает значительное количество ордеров (7 и более), цена актива часто демонстрирует потенциал для дальнейшего роста, превышающего уровень средней полосы. Это наталкивает на идею пересчёта Take Profit в таких ситуациях, ориентируясь на верхнюю границу Bollinger Bands, которая отражает состояния перекупленности и может служить более амбициозной целью.

Возникает вопрос: с какого числа ордеров сетку следует считать «большой» и переходить к использованию верхней границы? На основе эксперимента было установлено, что пороговым значением является 8 ордеров. Сетка глубиной 8 и более считается экстремальной, что указывает на значительное отклонение цены и высокую вероятность последующего разворота или продолжения тренда. Таким образом, новая логика предполагает: если количество открытых ордеров Take Profit рассчитывается по верхней границе Bollinger Bands; в противном случае используется средняя полоса.

Сравнение результатов показывает, что переход на верхнюю границу Bollinger Bands при позволил увеличить чистую прибыль примерно на 8.55%. Хотя разница не является радикальной, она подтверждает потенциал данного подхода. Увеличение доходности достигнуто за счёт более агрессивного целевого уровня в ситуациях с экстремальными сетками, где цена имеет тенденцию к дальнейшему росту.

2.7 Не линейное распределение капитала

После успешного тестирования базового торгового алгоритма с линейной сеткой и оптимизацией уровня Take Profit возникает естественный вопрос: можно ли дальнейшим образом повысить его эффективность? Одним из перспективных направлений является экспериментальная работа с распределением капитала, выходящая за рамки линейного увеличения депозита. Интуитивно понятно, что неравномерное распределение средств между ордерами может улучшить использование капитала и адаптацию стратегии к различным рыночным условиям. Проведём анализ распределения сделок, полученного в ходе предыдущих тестов. На рисунке 4 можно наблюдать как распределяться сделки по количеству открытых позиций.

4 - распределяться сделки по количеству открытых позиций

Из данных видно, что подавляющее большинство сделок ограничиваются открытием одного ордера. При начальном капитале 3,850 долларов и стартовом депозите 100 долларов первая сделка использует всего 2.6% доступных средств, что указывает на неэффективное распределение капитала на ранних этапах. Это побуждает к пересмотру структуры сетки, начиная с изменения шага депозита.

В качестве первого шага была протестирована модификация линейной сетки с увеличением начального депозита и уменьшением шага. Предположим, что начальный депозит , а шаг . Сумма для 11 ордеров должна быть равна 3,850 $:

Тест с S = 131, K = 39 показал незначительную прибыль.

Дальнейшее улучшение предполагает отказ от строгой линейности в пользу адаптивной структуры сетки. Идея заключается в следующем:

* - Максимальное использование капитала на старте: Первые два ордера (1 и 2) открываются на значительные суммы, чтобы сразу задействовать большую часть капитала и воспользоваться начальными движениями цены.
* - Снижение капитала в середине: Ордера с 3 по 6 уменьшают депозит (отрицательный шаг), что оправдано в боковиках, где цена часто возвращается к средней полосе Bollinger Bands без необходимости глубокого усреднения.
* - Увеличение капитала в конце: Ордера с 7 по 11 наращивают депозит, снижая среднюю точку входа и повышая вероятность достижения Take Profit в условиях тренда.

Идея кусочно-заданной функции заключается в адаптации распределения капитала к фазам рыночного цикла. Начальные крупные депозиты и максимизируют использование капитала в момент входа, что эффективно в условиях быстрого возврата цены к средней полосе Bollinger Bands. Уменьшение депозита с 3-го по 6-й ордер минимизирует убытки в боковике, где дальнейшее усреднение не требуется. Наращивание капитала с 7-го по 11-й ордер (с использованием )) позволяет эффективно усреднить позицию в условиях тренда, повышая вероятность достижения верхней границы или более амбициозного Take Profit. Такой подход обеспечивает гибкость и устойчивость стратегии.

Для реализации данной идеи была разработана кусочно-заданная функция, рассчитывающая депозит для -го ордера :

- Для:

(39)

Здесь начальный депозит устанавливается как базовое значение, определяющее размер первой сделки.

- Для :

(40)

Второй ордер увеличивает депозит на положительный шаг, что позволяет сразу задействовать значительную часть капитала для усиления начальной позиции.

- Для до :

(41)

В этой фазе депозит уменьшается с каждым последующим ордером за счёт отрицательного шага, что отражает стратегию минимизации капитала в условиях бокового движения или умеренного отклонения цены.

- Для до :

(42)

На заключительном этапе депозит снова наращивается с использованием положительного шага, снижая среднюю точку входа и повышая вероятность достижения Take Profit в условиях тренда.

Для поиска оптимальных значений гиперпараметров была проведена процедура перебора, охватывающая следующие наборы параметров:

* - начальный депозит для первой сделки.
* - положительный шаг увеличения депозита для второго ордера.
* отрицательный шаг уменьшения депозита для ордеров с 3 по 6, предназначенный для снижения капитала в условиях бокового движения или затяжного тренда.
* положительный шаг увеличения депозита для ордеров с 7 по 11, направленный на снижение средней точки входа и повышение шансов на прибыль.

Путем перебора наилучшими параметрами оказались следующие значения:

Пример расчета сетки для , , :

Начиная с 7-го ордера используется инкремент:

Промежуточная база:

* Сумма:

Не линейная сетка хороша тем, что позволяет более гибко распределять капитал, тем самым увеличивая прибыль. Новый алгоритм А2 (алгоритм второй) был протестирован на исторических данных. Результат показал прибыль в 34 тысячи долларов. Подробнее о результатах можно ознакомиться в таблице 3[54]. Алгоритмическому методами удалось повысить прибыль на 36%. Это очень хороший результат.

2.8 Ограничение алгоритма А2

К сожалению, текущая версия торгового алгоритма А2 демонстрирует хорошую эффективность преимущественно в условиях бокового движения рынка (флет). В такие периоды алгоритм успешно идентифицирует локальные минимумы и максимумы, открывая позиции с расчетом на краткосрочные ценовые колебания. Однако во время выраженных и продолжительных нисходящих трендов алгоритм сталкивается с рядом ограничений, которые снижают его эффективность и увеличивают риски.

Основная проблема заключается в том, что в условиях сильного тренда алгоритм продолжает открывать позиций. Алгоритм не понимает, в какой фазе рынка он находиться в данный момент. Методично, робот открывает и открывает позиции. В этом и заключается главный недостаток сеточной торговли - несмотря ни на что позиция усредняется и усредняется. Это приводит к накоплению открытых сделок, которые могут оставаться в рынке длительное время. В некоторых случаях откат цены происходит относительно быстро. Однако в других ситуациях ожидание разворота может затягиваться на недели или даже месяцы. Например, в ходе тестирования алгоритма была зафиксирована сделка, которая оставалась открытой в течение 186 дней. Такой длительный период "заморозки" средств существенно снижает общую доходность стратегии, увеличивает риски просадки капитала и ограничивает возможности для использования средств в других торговых операциях.

Кроме того, длительное удержание позиций в условиях трендового рынка может привести к накоплению свопов (комиссий за перенос позиции на следующий день), что дополнительно снижает потенциальную прибыль. Эти ограничения подчеркивают необходимость доработки алгоритма, чтобы повысить его устойчивость к различным рыночным условиям. В следующей главе будет предложен подход к решению данной проблемы

2.9 Создание LLM модели.

Для того чтобы преодалеть недостатки 2 алгоритма (А2) необходимо научить думать робота. Необхдимо, чтобы торговый бот мог сам анализировать сложившуюся обстановку на рынке и от этого принимать решения. В связи с этим, самым логичным решением было поробывать совместить торговый алгоритм с какой-нибудь LLM.

Для эффективного применения LLM в торговом алгоритме необходимо, в первую очередь, подобрать подходящую архитектуру модели. Несмотря на то, что на платформе Hugging Face представлено множество LLM, включая специализированные модели для финансовой аналитики, они ориентированы составления отчетов финансовых документов. Данные LLM не умеют анализировать крипторынки.

В связи с этим было принято решение о дообучении одной из современных мультимодальных моделей на криптовалютной тематике. В качестве базовой архитектуры выбрана модель LLaVA-1.5-7B, содержащая 7 миллиардов параметров. Ключевым преимуществом данной модели является её мультимодальность — способность обрабатывать не только текстовую, но и визуальную информацию. Это особенно важно в контексте криптотрейдинга, где анализ графиков (свечных, линейных, объёмных и т. д.) играет критическую роль в принятии решений.

После первичного запуска была проведена оценка базовой производительности модели LLaVA-1.5-7B. На вход подавался график цен (в виде изображения) и текстовый запрос: «В каком состоянии находится рынок? Это восходящий тренд, нисходящий тренд или боковик?». В одном из тестов модель ответила: «Цены растут», ошибочно интерпретировав нисходящий тренд как восходящий.

Дальнейшее тестирование на 10 примерах показало, что модель отвечает случайным образом, не учитывая ключевые показатели, такие как объём и RSI. Эти результаты свидетельствуют, что несмотря на способность работать с визуальными данными, модель не понимает их в контексте криптотрейдинга. Следовательно, требуется дообучение для повышения её точности и осмысленности ответов.

2.10 Сбор датасета

Для сбора датасета проводился отбор литературы с учетом их авторитетности и охвата ключевых аспектов трейдинга. Среди выбранных книг были труды, посвященные техническому анализу, фундаментальному анализу, психологии трейдинга и управлению рисками. Эти источники содержали как теоретические основы, так и практические примеры, что делало их подходящими для формирования обучающего датасета. Однако перед созданием датасета возник вопрос: как преобразовать текстовую информацию из книг в структурированный формат, пригодный для обучения модели?

Однако Процесс создания датасета оказался сложным и многокомпонентным, так как книги представляли собой неструктурированные текстовые данные, которые необходимо было обработать и привести к единому формату. Для этого были использованы два подхода: ручная обработка и полуавтоматическая обработка с применением локальной языковой модели.

Часть данных, особенно связанная с фундаментальными понятиями трейдинга, требовала ручной обработки. Например, такие аспекты, как определение тренда (восходящий, нисходящий или боковой), идентификация уровней поддержки и сопротивления или распознавание свечных паттернов, сложно извлечь автоматически без глубокого понимания контекста. Для этого из книг выделялись ключевые определения, правила и примеры, которые затем формализовались в виде пар "вопрос-ответ". Пример такой пары:

* Вопрос: Что такое восходящий тренд?
* Ответ: Восходящий тренд — это последовательность ценовых движений, при которой каждый последующий максимум и минимум выше предыдущего, что указывает на рост цены актива.

Ручная обработка была трудоемкой, но позволила создать высококачественные данные для базовых концепций, которые служили основой для дальнейшего обучения модели.

Для ускорения процесса обработки больших объемов текста был применен полуавтоматический подход с использованием локально развернутой языковой модели через платформу Ollama. Этот метод позволил автоматизировать извлечение ключевых идей из текстов и их структурирование. Процесс состоял из следующих этапов:

1. Разделение текста на чанки: Книги разбивались на небольшие фрагменты (чанки) размером от 500 до 2000 символов. Это обеспечивало удобство обработки и позволяло модели сосредотачиваться на отдельных логических блоках текста.
2. Извлечение ключевых идей: Каждый чанк подавался в локальную LLM (LLaMA) с инструкцией выделить основную мысль или сформулировать вопрос и ответ на основе текста. Например, для текста, описывающего понятие "скользящая средняя", модель могла сгенерировать:
3. Вопрос: Что такое скользящая средняя?
4. Ответ: Скользящая средняя — это индикатор, который сглаживает ценовые данные, усредняя значения за определенный период, и используется для определения тренда.

Форматирование в JSON: Полученные пары "вопрос-ответ" записывались в структурированный JSON-файл. Пример структуры:

[

{

"question": "Что такое восходящий тренд?",

"answer": "Восходящий тренд — это последовательность ценовых движений, при которой каждый последующий максимум и минимум выше предыдущего, что указывает на рост цены актива."

},

{

"question": "Что такое скользящая средняя?",

"answer": "Скользящая средняя — это индикатор, который сглаживает ценовые данные, усредняя значения за определенный период, и используется для определения тренда."

}

]

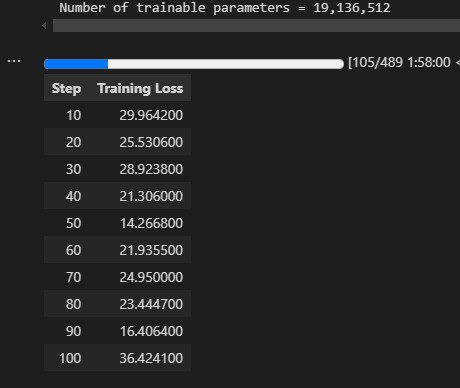
Полуавтоматический подход значительно ускорил процесс, однако требовал дополнительной проверки результатов, так как LLM иногда допускала ошибки в интерпретации сложных концепций или генерировала недостаточно точные формулировки.

На выходе был сформирован структурированный датасет в формате JSON, содержащий сотни пар "вопрос-ответ", охватывающих ключевые аспекты трейдинга. Датасет включал:

* Теоретические понятия (определения трендов, индикаторов, паттернов);
* Практические примеры (описания торговых стратегий и их применения);
* Правила управления рисками и психологии трейдинга.

Этот датасет стал основой для дообучения модели.

2.11 Обучение LLM

После подготовки датасета начался этап обучения модели. На рисунке 5 можно наблюдать процесс обучения.

5 – Процесс обучения LLM

Обучение модели проводилось на основе подготовленного датасета, содержащего структурированные пары "вопрос-ответ", извлеченные из книг по трейдингу. Процесс обучения занял 7 часов. После завершения обучения модель была протестирована. Новая дообученная модель лучше опредляла наличие тренда или его отсутствия, в отличии от базовой (родительской модели).

Для повышения эффективности модели, особенно с точки зрения скорости инференса, было применено квантование. Квантование — это процесс оптимизации модели, при котором веса и активации модели, изначально представленные в формате с высокой точностью (например, 32-битные числа с плавающей запятой, FP32), преобразуются в формат с более низкой точностью (например, 16-битные числа, FP16, или 8-битные целые числа, INT8). Это позволяет сократить объем памяти, необходимый для хранения модели, и ускорить вычисления, что особенно важно для развертывания на устройствах с ограниченными ресурсами или для задач, требующих быстрого отклика.

Квантование основано на следующем подходе: значения весов w в модели масштабируются и округляются до дискретных уровней. После квантования модель становится более компактной, а вычисления выполняются быстрее за счет использования операций с меньшей разрядностью.

В данном проекте квантование позволило значительно сократить время ответа модели. Если неквантованная модель в среднем отвечала за 3 минуты, то квантованная модель сократила это время до 8 секунд. При этом была проведена оценка качества ответов, которая показала, что квантование не только не ухудшило производительность, но и в некоторых случаях повысило уверенность модели в ответах. Это связано с тем, что квантование устраняет незначительные шумы в весах, что иногда улучшает обобщающую способность модели. Тем не менее, общее качество ответов все еще не соответствовало требованиям для полноценного "мышления как трейдер".

После тестирования стало очевидно, что модель не обладает достаточным качеством. Она не научилась качественно классифицировать наличие тренда или его отсутствия. Также Модель не научилась рассуждать, думать как трейдер. Модель скорее отвечала цитатами из книжек, нежели рассуждала над поставленным вопросом. Поэтому, было принято решение отложить интеграцию LLM и алгоритма и перейти к более простой модели градиентного бустинга.

2.12 Классический подход

В процессе исследования возможностей больших языковых моделей (LLM) для задач финансового анализа, были выявлены некоторые ограничения. Создание модели, способной эффективно функционировать в роли трейдера в условиях ограниченных вычислительных ресурсов (домашних условиях), оказалось крайне сложной задачей. Основным недостатком LLM является их низкая скорость обработки данных. Даже оптимизированные (квантованные) модели демонстрируют время отклика порядка трёх минут, что неприемлемо для высоковолатильных финансовых рынков, где решения должны приниматься в реальном времени. В таких условиях классические методы машинного обучения, отличающиеся высокой скоростью и простатой, становятся предпочтительным выбором.

Теперь, задача немого изменилась и перешла от «попытки имитации живого человека трейдера» к созданию алгоритма, способного определять наличие рыночного тренда. Если тренд отсутствует, алгоритм должен воздерживаться от торговли, что сводит задачу к бинарной классификации: тренд присутствует (класс 1) или отсутствует (класс 0).

Для решения этой задачи был разработан специальный алгоритм, так как вручную собирать датасет было бы невозможно. Брался за основу торговый алгоритм, если сделок было больше 5, то это тренд, данные сразу записывались в файл trend.csv, после того как тренд заканчивался. То последнии свечи считались не трендовыми, и записывались в файл flet.csv. После работы программы получился 2 csv файла с набором данных. После их объединения финальный датасет содержал следующие признаки:

* avg\_volume: средний объём торгов за определённый период;
* last\_volume: объём торгов в последней свече;
* RSI: индекс относительной силы (Relative Strength Index);
* Close\_price: цена закрытия свечи;
* low\_BB: нижняя граница полос Боллинджера.

Эти признаки были выбраны с учётом их значимости для анализа рыночных трендов и способности отражать динамику цен и объёмов торгов.

Перед подачей данных в модель машинного обучения была проведена их предобработка, направленная на повышение информативности признаков и упрощение задачи классификации. Были выполнены следующие шаги:

1. Бинаризация признака отношения цены закрытия к нижней полосе Боллинджера:  
   Определено, находится ли цена закрытия выше или ниже нижней полосы Боллинджера. Новый признак close\_hier\_low\_BB принимает значение 1, если Close\_price > low\_BB, и 0 в противном случае.
2. Вычисление относительного объёма торгов:  
   Рассчитан процентный прирост последнего объёма торгов (last\_volume) относительно среднего объёма (avg\_volume). Новый признак volume\_diff\_pct отражает, во сколько раз последний объём превышает средний, выраженный в процентах.
3. Оценка относительного превышения цены закрытия над нижней полосой Боллинджера:  
   Создан признак price\_diff\_pct, который показывает, на сколько процентов цена закрытия превышает нижнюю полосу Боллинджера относительно цены закрытия.

Итоговый датасет включал признаки (avg\_volume, last\_volume, RSI, Close\_price, low\_BB). Это позволило модели лучше улавливать закономерности в данных.

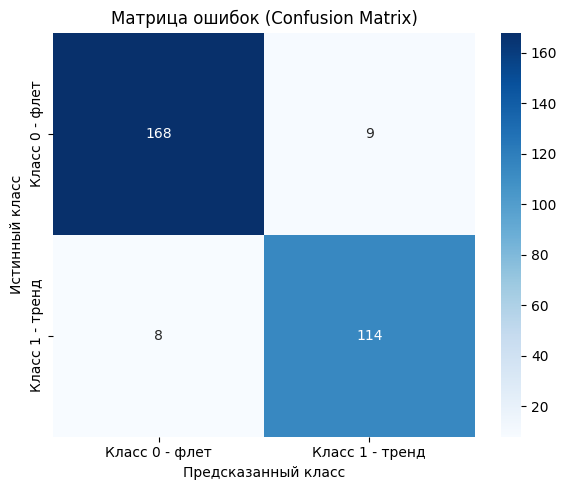
Для решения задачи классификации был выбран метод градиентного бустинга, который хорошо зарекомендовал себя в задачах с табличными данными благодаря высокой точности и устойчивости к шуму. Модель была обучена на подготовленном датасете с использованием библиотеки CatBoost. После обучения модель была протестирована на отложенной выборке, и получены следующие метрики производительности:

Рисунок – 4. Матрица ошибок

6 - Матрица ошибок

Таблица 2.

Метрики качества

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Class | Metrics | Precision | Recall | F1-Score | Support |
| 0 |  | 0.95 | 0.95 | 0.95 | 177 |
| 1 |  | 0.93 | 0.93 | 0.93 | 122 |
|  | Accuracy |  |  | 0.94 | 299 |
|  | Macro Avg | 0.94 | 0.94 | 0.94 | 299 |
|  | Weighted Avg | 0.94 | 0.94 | 0.94 | 299 |

Полученные результаты свидетельствуют о высокой производительности модели. Точность (precision), полнота (recall) и F1-мера для обоих классов составляют около 0.93–0.95, а общая точность модели достигает 94%. Это указывает на то, что модель эффективно справляется с задачей определения наличия или отсутствия рыночного тренда. Важно отметить, что градиентный бустинг не только продемонстрировал высокую точность, но и значительно превзошёл LLM по скорости обработки данных. Время отклика модели составляет доли секунды, что делает её пригодной для использования в условиях реального времени на волатильных рынках.

2.13 Внедрение модели в алгоритм

После успешного создания и обучения модели следующим этапом стало её внедрение в торговый алгоритм. Интеграция модели привела к усложнению логики открытия новых сделок, что позволило алгоритму адаптироваться к различным рыночным условиям. Новая логика включает два основных сценария, зависящих от количества активных ордеров и прогнозов модели.

Сценарий 1: Количество ордеров не превышает трёх

* Если количество открытых ордеров составляет три или менее, алгоритм продолжает функционировать в соответствии с исходной стратегией, которая основана на сеточном подходе.

Сценарий 2: Количество ордеров превышает три

* При превышении лимита в три открытых ордера алгоритм переходит к более сложной логике, включающей проверку наличия рыночного тренда с использованием прогнозов модели. Процесс принятия решения об открытии новой сделки состоит из следующих этапов:

1. Проверка наличия тренда: Модель анализирует текущие рыночные данные. Если модель классифицирует текущую ситуацию как тренд (класс 1), алгоритм воздерживается от открытия новых сделок. Это позволяет избежать торговли в условиях сильных направленных движений.
2. Ожидание завершения тренда: Алгоритм продолжает мониторить рынок до тех пор, пока модель не определит отсутствие тренда (класс 0). Отсутствие тренда сигнализирует о стабилизации рынка, что создаёт более безопасные условия для входа в позицию.
3. Проверка цены относительно расчётного уровня: После подтверждения отсутствия тренда алгоритм сравнивает текущую рыночную цену с ценой, рассчитанной в рамках сеточного подхода. Если текущая цена оказывается ниже расчётной, алгоритм открывает новую сделку (покупку) по более выгодной цене.

Все эти методы позволяет использовать коррекционные движения рынка для входа в позицию с улучшенным соотношением риска и доходности. Однако прямое соединение модели с алгоритмом А3 показало очень плохой результат. Подробнее с результатами можно ознакомиться в таблице 3 [54].

2.13 Улучшение алгоритма с ML моделью

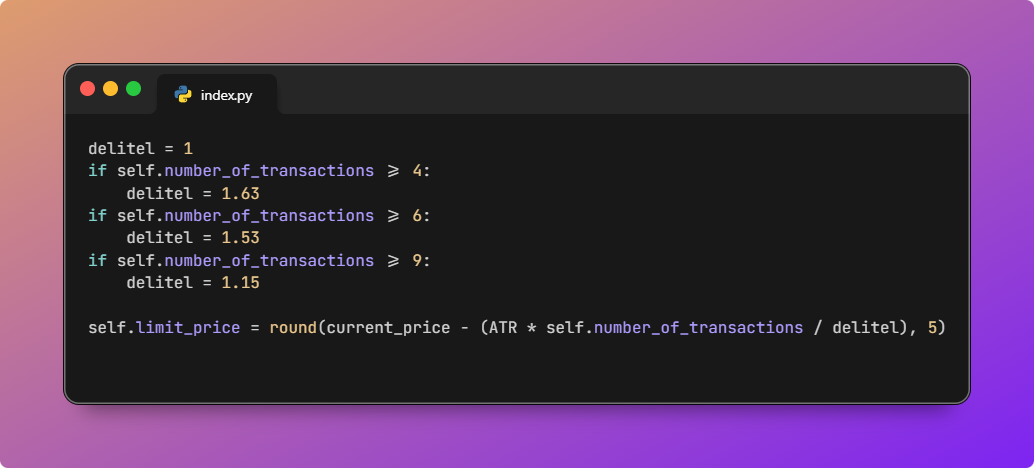
Анализ выявил, что ухудшение результатов связано с конфликтом между логикой сеточного алгоритма и прогнозами модели. В исходной версии алгоритма уже была реализована встроенная защита от трендов, основанная на расчёте цены открытия каждого последующего ордера по формуле:

(43)

где — индикатор средней истинной амплитуды (Average True Range), а — число уже открытых ордеров.

Однако внедрение модели, которая дополнительно фильтрует сделки на основе наличия тренда, привело к нежелательным эффектам. Если модель определяла тренд и блокировала открытие ордера, цена продолжала снижаться, отдаляясь от расчётного уровня сетки. Когда тренд завершался, новый ордер открывался уже по более низкой цене, что приводило к повторному расчёту всей сетки от этой заниженной цены. В результате ордера размещались на значительном удалении от текущих рыночных уровней, что снижало их эффективность и увеличивало время нахождения в позициях. Таким образом, модель и сеточный алгоритм вступали в конфликт, препятствуя друг другу и снижая общую производительность системы.

Для устранения выявленной проблемы была предложена модификация формулы расчёта цен лимитных ордеров, направленная на повышение плотности сетки и улучшение эффективности размещения ордеров. В новой версии алгоритма введён коэффициент делителя delitel, который изменяется в зависимости от количества открытых сделок. Логика расчёта реализована следующим образом:

Математически новая формула выглядит следующим образом:

6 – каоефицент для уплотнения сетки ордеров.

(44)

Коэффициент уменьшает шаг между ордерами при увеличении количества сделок, что делает сетку более плотной. Например, при шаг уменьшается в 1.63 раза, а при — в 1.15 раза. Это позволяет размещать ордера ближе к текущей рыночной цене, увеличивая вероятность их исполнения и сокращая время нахождения в позициях.

Ключевым аспектом данной модификации является её зависимость от модели классификации трендов. Без модели, которая предотвращает открытие ордеров во время трендов, плотная сетка могла бы привести к чрезмерному накоплению позиций в условиях направленного движения, увеличивая риски. Однако благодаря модели, которая фильтрует сделки, алгоритм открывает ордера только в благоприятных рыночных условиях, что делает модификацию безопасной и эффективной.

Глава 3. Анализ полученных результатов.

3.1 Результаты экспериментов

Теперь пришло время сравнить, оценить полученные в ходе исследования результаты. Обозначим все версии алгоритмов следующим образом:

* А1 – самый первый алгорим, без улучшений.
* А2 – улучшенная версия алгоритма
* А3 – улучшенная версия алгоритма с применением ML модели
* А4 – модифицированный алгоритм для эффективного использования ML модель

Все алгоритмы имели одинаковый начальный капитал. Алгоритмы прошли проверку на исторических данных монеты DOGECOIN с 22.11.2020 по 22.11.2024. Получены следующие результаты:

Таблица 3

Результат работы разных модификаций алгоритмов

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | А1 | А2 | А3 | А4 |
| Чистая прибыль | 25,659 $ | 33,322 $ | 12,647 $ | 44,630 $ |
| Максимальное время в позиции | 484 дня, 4 часа 45 минут | 189 дней, 13 часа 15 минут | 846 дней, 9 часов, 15 минут | 185 дней, 20 часов, 45 минут; |
| Среднее время в позиции | 3 дня, 11 часов, 32 минуты | 1 дней, 12 часов, 29 минуты | 2 дня, 19 часов, 25 минут | 1 день, 7 часов, 38 минуты |
| Количество сделок | 385 | 814 | 472 | 919 |
| Средняя прибыль с каждой сделки | 77,5$ | 40,3 $ | 26,2 $ | 48,9 $ |
| Максимальное количество открытых позиций. | 10 | 10 | 11 | 11 |

Как видно из таблицы, благодаря машинному обучению удалось увеличить прибыль примерно на 31% относительно А2 и на 75% относительно А1. Были произведены замеры, и выяснилось, что в среднем модель позволяет снижать цену входа в сделки на 2.2–2.5% по сравнению с расчётными уровнями сетки. Такое снижение является значительным достижением, поскольку даже небольшое улучшение цен входа может существенно повысить общую прибыльность стратегии. Оптимизация цен достигается благодаря способности модели точно определять моменты отсутствия тренда, что позволяет алгоритму открывать позиции по более низким ценам после коррекционных движений рынка.

Для иллюстрации эффективности модели приведены конкретные примеры, демонстрирующие, как расчётная цена, определённая сеточным алгоритмом, сравнивается с новой ценой, полученной с использованием модели. Для каждого примера рассчитан процент снижения цены, чтобы количественно оценить вклад модели. Формула для расчёта процентного снижения цены следующая:

(45)

Ниже представлены примеры с соответствующими расчётами:

Пример 1:

* Расчётная цена: 0.00231
* Новая цена: 0.0022112

Пример 2:

* Расчётная цена: 0.00213
* Новая цена: 0.0020958

Пример 3:

* Расчётная цена: 0.00182
* Новая цена: 0.0016011

Эти примеры подтверждают, что модель способна обеспечивать значительное снижение цен входа, хотя величина снижения варьируется в зависимости от рыночных условий. Благодаря внедрению модели в алгоритм удалось сократить среднее время прибывания в позиции и увеличить среднее число сделок, что в свою очередь увеличило общую прибыль.

3.2 Практические рекомендации

Для дальнейшего улучшения алгоритма и увеличения прибыли предлагается разработка дополнительной модели, направленной на прогнозирование глобальных нисходящих трендов на старших таймфреймах, таких как часовой (1H). Текущая модель, работающая на более коротких таймфреймах, эффективно определяет локальные тренды, но может не учитывать долгосрочные рыночные движения. Это позволит более эффективно растягивать сетку ордеров, пережидая длинные затяжные тренды.

Вторым этапом улучшения алгоритма может стать создание модели для определения восходящих трендов. Такая модель даст возможность в некоторых случаях не закрывать позицию по достижении первого уровня Take Profit, а дождаться окончания движения вверх, тем самым увеличив итоговую доходность.

Реализация рекомендаций даст более гибкое и контекстное управление позициями и выходами, что может существенно повысить прибыль алгоритма.

ЗАКЛЮЧЕНИЕ

В ходе проведённого исследования удалось достичь поставленных целей и успешно реализовать все основные задачи. Разработанный торговый алгоритм показал высокую эффективность при автоматической торговле криптовалютой. Использование классических методов машинного обучения, в частности градиентного бустинга, обеспечило точное определение локальных рыночных трендов, что позволило оптимизировать точки входа в рынок и значительно повысить общую прибыльность торговой стратегии.

Алгоритм был реализован на языке программирования Python с учётом особенностей криптовалютных рынков и успешно протестирован на обширных исторических данных. Результаты тестирования подтвердили его работоспособность и устойчивость в условиях высокой волатильности. Также была проведена оптимизация модели с использованием современных подходов машинного обучения, что позволило повысить точность торговых сигналов и снизить риск принятия ошибочных решений.

Практическая значимость разработанной системы заключается в её способности приносить стабильный доход при минимальном участии человека, что делает её привлекательным инструментом как для частных инвесторов, так и для профессиональных трейдеров. Полученные результаты могут служить основой для дальнейших исследований в области алгоритмической торговли и развития более совершенных торговых систем

СПИСОК ИСПОЛЬЗОВАННЫХ ИСТОЧНИКОВ

1. История алгоритмической торговли и HFT трейдинга [Электронный ресурс] // Empirix.ru. – URL: <https://empirix.ru/istoriya-algoritmicheskoj-torgovli/> (дата обращения: 10.05.2025).
2. Досенко Е. М. Тенденции развития и регулирование алгоритмической торговли // Научный журнал. – 2015. – № 3. – С. 45–50. – URL: <https://cyberleninka.ru/article/n/tendentsii-razvitiya-i-regulirovanie-algoritmicheskoy-torgovli> (дата обращения: 10.05.2025).
3. Алгоритмическая торговля [Электронный ресурс] // Википедия. – URL: [https://ru.wikipedia.org/wiki/Алгоритмическая\_торговля](https://ru.wikipedia.org/wiki/%D0%90%D0%BB%D0%B3%D0%BE%D1%80%D0%B8%D1%82%D0%BC%D0%B8%D1%87%D0%B5%D1%81%D0%BA%D0%B0%D1%8F_%D1%82%D0%BE%D1%80%D0%B3%D0%BE%D0%B2%D0%BB%D1%8F) (дата обращения: 10.05.2025).
4. Володин С. Н., Якубов А. П. Развитие алгоритмической торговли на мировых финансовых рынках: причины, тенденции и перспективы // Финансы и кредит. – 2017. – № 9. – С. 2017–2032. – URL: <https://cyberleninka.ru/article/n/razvitie-algoritmicheskoy-torgovli-na-mirovyh-finansovyh-rynkah-prichiny-tendentsii-i-perspektivy> (дата обращения: 10.05.2025).
5. Введение в алготрейдинг: роботы, стратегии и торговля [Электронный ресурс] // JumpAltitude. – URL: <https://jumpaltitude.com/woodbridge/blog/vvedenie-v-algotrejding-roboty-strategii-i/> (дата обращения: 10.05.2025).
6. Применение искусственного интеллекта и машинного обучения в криптотрейдинге [Электронный ресурс] // AnyExchange. – URL: <https://anyexchange.best/ispolzovanie-mashinnogo-obucheniya-dlya-analiza-kriptoproektov/> (дата обращения: 10.05.2025).
7. Касап Р. Виды торговых (трейдинг) роботов: топ-20 стратегий [Электронный ресурс], URL: <https://vc.ru/id1388688/593012-vidy-torgovyh-treiding-robotov-top-20-strategii> (дата обращения: 09.05.2025)
8. Dissperss. Профиль пользователя. [Электронный ресурс] // Тинькофф Инвестиции. – URL: <https://www.tbank.ru/invest/social/profile/Dissperss/c4fe361b-f998-4d4d-8d3d-21b2d55bf7bf/?author=profile> (дата обращения: 10.05.2025).
9. Градиентный бустинг для чайников [Электронный ресурс] // Habr. – URL: <https://habr.com/ru/companies/raft/articles/890802/> (дата обращения: 09.05.2025).

ПРИЛОЖЕНИЕ