

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ
ВЫСШЕГО ОБРАЗОВАНИЯ
НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ
«ВЫСШАЯ ШКОЛА ЭКОНОМИКИ»

Факультет информатики, математики и компьютерных наук

Соболев Данил Александрович

ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ

Курсовая работа

Руководитель
Кандидат компьютерных наук

С.В. Павлов

Нижний Новгород, 2025

СОДЕРЖАНИЕ

ВВЕДЕНИЕ	4
1. ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ РЯДОВ	6
1.1. Определение временного ряда	6
1.2. Требования к исходным данным	6
1.3. Классификация временных рядов	7
1.4. Основные свойства временных рядов	7
1.5. Декомпозиция временного ряда	8
1.6. Обработываемый интервал	9
1.7. Типы прогнозов	9
1.8. Метрики качества прогнозирования	11
1.9. Генерация дополнительных признаков временного ряда	12
1.9.1. Технические индикаторы	12
1.9.2. Статистические признаки	13
1.10. Базовые методы прогнозирования	14
1.10.1. Наивное повторение	14
1.10.2. Скользящая средняя (simple moving average, SMA)	14
1.10.3. Экспоненциальная скользящая средняя (exponential moving average, EMA)	15
1.10.4. Ансамбль скользящих средних (ensemble of moving averages)	16
1.10.5. Метод дрейфа (drift method)	17
1.11. Статистические модели	17
1.11.1. Проверка стационарности	17
1.11.2. Удаление тренда и сезонности	18
1.11.3. Стабилизация дисперсии	18
1.11.4. Разностное преобразование	18
1.11.5. Логарифмические доходности	18
1.11.6. Алгоритм приведения ряда к стационарному	19
1.11.7. ARIMA (autoregressive integrated moving average)	19
1.11.8. Auto-ARIMA (automatic ARIMA selection)	20
1.11.9. TBATS (trigonometric seasonality, Box–Cox transformation, ARMA errors, trend, seasonal components)	22
1.12. Линейные модели	23
1.12.1. DLinear (decomposed linear)	23
1.12.2. NLinear (nonlinear linear)	24
1.13. Рекуррентные модели	24
1.13.1. RNN	25
1.13.2. LSTM (long short-term memory)	25
1.14. Методы машинного обучения	26
1.14.1. Линейная регрессия (linear regression)	27
1.14.2. К ближайших соседей (K-nearest neighbors, K-NN)	27
1.14.3. XGBoost (eXtreme Gradient Boosting)	28
1.14.4. CatBoost (categorical boosting)	30
1.14.5. Random Forest Regression	31
1.14.6. Prophet	32
1.15. VAR (vector autoregression)	33
1.16. Библиотека tsai	34
1.17. FEDOT (framework for evolutionary design of optimization techniques)	35
1.18. Chronos (chronos-forecasting)	36
1.19. Прогнозирование временных рядов с помощью больших языковых моделей	38

1.19.1. Прямой подход (one-shot prompting).....	38
1.19.2. Подход StockTime.....	38
2. ДЕТЕКТИРОВАНИЕ АНОМАЛИЙ И ПОИСК ТОЧЕК ПЕРЕЛОМА В ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДАХ.....	40
2.1. Аномалии.....	40
2.2. Точки перелома.....	40
2.3. Классификация методов.....	41
2.4. Генерация аномалий.....	42
2.4.1. Оценка базовой плотности доходностей.....	42
2.4.2 Генерация базового синтетического ряда.....	42
2.4.3 Внедрение аномалий.....	43
2.4.4. Метрики оценки качества.....	43
2.5. Статистические методы детектирования аномалий.....	44
2.5.1. IQR-детектор (OutlierDetector).....	44
2.5.2. Скользящее среднее (MeanAnomalyDetector).....	45
2.5.3. Параметрический детектор (DistributionBasedAnomalyDetector).....	45
2.5.4. Хвостовой детектор (RareDistributionDetector).....	45
2.6. Прогнозный подход.....	46
2.7. Isolation Forest.....	47
2.8. Кластерный подход.....	48
2.8.1. K-Means.....	48
2.8.2. K-NN.....	48
2.9. Автоэнкодеры.....	49
2.10. Детектирование многомерных аномалий (PyOD).....	50
2.11. Ruptures.....	51
2.12. Детектирование аномалий с помощью LLM.....	52
2.12.1 Пакетный анализ.....	52
2.12.2. Детектирование с использованием прогноза.....	53
3. ОЦЕНКА ВЛИЯНИЯ НОВОСТЕЙ НА СТОИМОСТИ АКТИВОВ С ПОМОЩЬЮ LLM.....	54
4. РАЗРАБОТКА КОМБИНИРОВАННОГО ПОДХОДА ПРОГНОЗИРОВАНИЯ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ LLM.....	55
4.1. Переход к многомерным рядам.....	55
4.2. Учёт новостного фона.....	56
4.2.1. Интеграция оценок новостного фона.....	56
4.2.2. Прямое включение заголовков.....	57
4.3. Передача изображения ряда.....	57
5. ЭКСПЕРИМЕНТЫ.....	59
5.1. Прогнозирование одномерных рядов.....	59
5.2. Детектирование точечных аномалий.....	60
5.3. Оценка новостей.....	61
5.4. Генерация признаков.....	63
5.4. Прогнозирование многомерных рядов.....	64
ЗАКЛЮЧЕНИЕ.....	67
СПИСОК ЛИТЕРАТУРЫ.....	68
ПРИЛОЖЕНИЕ.....	70

ВВЕДЕНИЕ

Прогнозирование финансовых временных рядов — одна из центральных задач современной финансовой теории. Оно позволяет оценивать динамику цен акций, облигаций, валют и других инструментов, что напрямую влияет на эффективность инвестиционных решений и управление рисками. Высокая волатильность рынков, обусловленная политическими событиями и макроэкономическими шоками, делает задачу «однодневного» прогноза особенно сложной.

Традиционные методы прогнозирования прошли путь от простейших техник сглаживания (наивное повторение, SMA, EMA) к классическим статистическим моделям (ARIMA, SARIMA, TBATS) и разложенным линейным архитектурам (DLinear, NLinear). С ростом вычислительных мощностей в центр внимания вышли глубокие и рекуррентные сети (RNN, LSTM, TCN, N-BEATS), трансформеры, а также гибридные AutoML-решения и специализированные библиотеки (Auto-ARIMA, FEDOT, tsai, Chronos). Параллельно развивались методы детектирования аномалий (IQR, Z-Score, distribution-based, Ruptures, Isolation Forest, кластеризация, автоэнкодеры, PyOD) и sentiment-анализ новостного фона.

В последние годы большие языковые модели (LLM) продемонстрировали способность обрабатывать мультимодальные данные и, даже в условиях one-shot, автоматически выявлять скрытые закономерности. Исследования [1–5] подтверждают их высокую эффективность в задачах прогнозирования, детектирования аномалий и анализа корпоративных отчётов.

Цель данной работы — исследовать потенциал комбинированного мультимодального подхода на базе LLM для прогнозирования финансовых временных рядов. Основная задача — объединить в единой системе:

- числовые признаки ряда (цены, объёмы, доходности, технические индикаторы),

- метки точек перелома и аномалий,
- числовые и текстовые оценки новостного фона,
- графические представления цен (изображения свечных и индикаторных графиков).

Экспериментально показано, что такой подход позволяет повысить точность прогнозов и приблизиться к результатам специализированных моделей.

Таким образом, результаты работы демонстрируют актуальность и эффективность интеграции LLM в задачи финансового прогнозирования и очерчивают перспективы создания интеллектуальных торговых систем нового поколения.

1. ПРОГНОЗИРОВАНИЕ ФИНАНСОВЫХ РЯДОВ

1.1. Определение временного ряда

Временной ряд — это последовательность значений одного и того же показателя, упорядоченная по времени. Каждому уровню ряда (наблюдению) соответствует момент времени или порядковый номер. В отличие от традиционной выборки, где анализируются лишь базовые характеристики данных (среднее, дисперсия, распределение), при работе с временными рядами учитывается зависимость между соседними наблюдениями.

В финансовой сфере классическим примером служит динамика ценовой серии: например, ежедневная цена закрытия акций ПАО «Газпром» на Московской бирже. При анализе таких данных зачастую выделяют долгосрочную тенденцию (тренд) и краткосрочные колебания, чтобы построить инвестиционную стратегию и оценить риск в последующих периодах.

1.2. Требования к исходным данным

Для получения объективного прогноза временной ряд должен удовлетворять следующим базовым требованиям:

1. Равномерный шаг во времени

- a. Все наблюдения фиксируются с одинаковым шагом (например, ежедневно, еженедельно или ежемесячно).
- b. На финансовых рынках часто возникают пропуски: выходные, праздничные дни или приостановки торгов. В таких случаях необходимо решить, оставлять ли ряд «разреженным» (с пропусками) или заполнять отсутствующие даты, например перенося последнюю известную цену на пропущенные дни.

2. Синхронность наблюдений

- a. При использовании дневной цены закрытия важно, чтобы все значения соответствовали моменту закрытия торгов.

Недопустимо смешивать цены открытия, средневзвешенные или внутридневные показатели.

3. Полнота данных

- a. Краткие пропуски (1–2 торговых дня) можно устранить интерполяцией или переносом предыдущего значения.
- b. Длительные «провалы» (например, многодневные приостановки торгов активом) требуют применения специальных методов восстановления либо исключения таких периодов из анализа.

1.3. Классификация временных рядов

1. Одномерный временной ряд (univariate) — содержит значения только одного показателя. Например, ежедневная цена закрытия акции ПАО «Сбербанк».
2. Многомерный временной ряд (multivariate) — фиксируются одновременно несколько взаимосвязанных показателей. Например, цена и объем торгов одной и той же акции.

1.4. Основные свойства временных рядов

При выборе и оценке методов прогнозирования важно учитывать ключевые статистические характеристики временного ряда.

1. Стационарность — свойство ряда, при котором его статистические характеристики не меняются во времени. В слабой (или второстепенной) стационарности сохраняются постоянными лишь два первых статистических момента: математическое ожидание и ковариация.

2. Автокорреляция — зависимость между значениями ряда на разных лагах. Когда текущее значение ряда зависит от предыдущих, говорят о наличии автокорреляции (ACF). Именно это свойство позволяет моделям учитывать «эффект памяти» и строить более точные прогнозы.

1.5. Декомпозиция временного ряда

Для анализа и построения прогноза временной ряд традиционно представляют как сумму четырёх компонент:

1. Тренд (trend) — длинная восходящая или нисходящая динамика ряда без учёта сезонных, циклических и случайных колебаний.
2. Цикличность (cycle) — относительно длительные отклонения от тренда, связанные с макропроцессами. В отличие от сезонности, циклы не обязательно имеют постоянный период, но могут охватывать несколько кварталов или лет.
3. Сезонность (seasonality) — регулярные колебания уровня ряда в пределах фиксированного календарного периода (например, годового или квартального). Сезонные изменения обусловлены повторяющимися факторами (праздники, климат, потребительские предпочтения) и проявляются в виде одинаковых шаблонов через каждый сезонный интервал.
4. Остатки (residuals) — случайные флуктуации и нерегулярные отклонения, не объясняемые ни трендом, ни сезонностью и ни циклическостью. Остатки рассматриваются как «шум», который можно либо учитывать при построении моделей, либо фильтровать для повышения устойчивости прогноза.

1.6. Обработываемый интервал

Поскольку временной ряд — это последовательность взаимосвязанных данных, для прогнозирования требуется определить интервал (окно) обработки новых наблюдений.

Скользящее окно (sliding window) - фиксируется размер окна w (например, w дней). При каждой точке прогноза модель обучается на последних w наблюдениях:

- Исторический ряд y_1, y_2, \dots, y_n .
- Для прогноза в момент $n+1$ используем данные y_{n-w}, \dots, y_n .
- Для прогноза в момент $n+2$ окно сдвигается на единицу: $y_{n-w+1}, \dots, y_{n+1}$ (если значение y_{n+1} уже известно или используется предсказанное).

Метод применяется, если ряд потенциально нестационарен на длительных отрезках или старые данные могут не отражать актуальные закономерности, а также когда требуется фиксированное число наблюдений для обучения.

Расширяющееся окно (expanding window) - окно «растёт» со временем: на шаге k модель обучается на всех доступных наблюдениях $y_1, y_2, \dots, y_{n+(k-1)}$:

- Для прогноза y_{n+k+1} используют полный набор данных y_1, y_2, \dots, y_{n+k} .
- Никакие наблюдения не удаляются: каждое новое значение сразу добавляется в обучающую выборку.

Этот метод уместен, если ранние данные влияют на текущую динамику, ряд относительно стационарен, или объём исторических данных невелик.

1.7. Типы прогнозов

После выбора интервала (окна) обработки данных возникает вопрос: как из этого окна получить прогноз? Существует два основных подхода: прогноз «точка за точкой» (one-step-ahead forecast) и мультишаговый прогноз (multi-step-ahead forecast).

1. Точка за точкой (one-step-ahead forecast) - на каждом временном шаге модель предсказывает только следующее значение ряда. Формально:

$$\hat{y}_{t+1} = f(y_t, y_{t-1}, y_{t-2}, \dots)$$

Когда наблюдение y_{t+1} становится известным (например, после закрытия торгов), оно добавляется в обучающую выборку (скользящее или расширяющееся окно), и строится прогноз \hat{y}_{t+2} . Процесс повторяется: ежедневно окно сдвигается вместе с рядом, а качество оценки измеряется отклонением \hat{y}_{t+1} от реального y_{t+1} . Такой подход позволяет постоянно корректировать модель на основе последних истинных данных, поэтому на каждом шаге прогноз максимально объективен и не накапливает ошибок.

2. Мультишаговый прогноз (multi-step-ahead forecast) - — модель генерирует сразу серию прогнозов $(\hat{y}_{n+1}, \hat{y}_{n+2}, \dots, \hat{y}_{n+h})$, где h — горизонт прогнозирования (например, $h = 20$). Выделяют два способа:

- a. Рекурсивный способ (recursive) - поочерёдно прогнозируется каждая следующая точка. Однако ошибки на ранних шагах передаются дальше, что приводит к экспоненциальному росту общей погрешности.

- b. Прямой способ (direct) - строится либо h отдельных моделей, либо одна модель, выдающая вектор из h прогнозов. У подхода нет «эффекта накопления ошибки», но каждая модель игнорирует промежуточную информацию, что может снижать точность на больших горизонтах.

- i. Модель 1 предсказывает y_{n+1} на основе y_1, \dots, y_n .

- ii. Модель 2 обучается на тех же данных и предсказывает y_{n+2} .
- iii. И так далее до y_{n+h}

Подход применяется, когда требуется получить сразу несколько будущих значений. Однако точность на дальних горизонтах обычно ниже из-за роста ошибок, и иногда уступает качеству one-step-ahead прогноза.

1.8. Метрики качества прогнозирования

Для базовой оценки моделей прогнозирования временных рядов в данной работе используются следующие метрики:

1. Средняя абсолютная процентная ошибка (mean absolute percentage error, MAPE) - показывает среднюю относительную ошибку прогноза в процентах:

$$MAPE = \left(\frac{100\%}{N} \right) \times \sum \left| \frac{y_t - \hat{y}_t}{y_t} \right|$$

Преимущество MAPE — интерпретируемость в процентах, что удобно при сравнении моделей на разных рядах. Недостаток — чувствительность к малым или нулевым значениям y_t .

2. Корень из среднеквадратичной ошибки (root mean squared error, RMSE) - отражает среднеквадратичную величину ошибки:

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (y_t - \hat{y}_t)^2}$$

Благодаря квадратичному штрафу, RMSE сильнее реагирует на крупные отклонения, что важно при оценке риска значительных промахов. Однако метрика зависит от масштаба данных и не даёт нормированной оценки.

3. Точность направления (directional accuracy, DA) - доля случаев, когда знак изменения прогноза совпадает со знаком фактического изменения:

$$DA = \frac{1}{N-1} \cdot \sum_{\{t=1\}}^{\{N-1\}} \mathbb{1}(\text{sign}(\hat{y}_{\{t+1\}} - y_t) = \text{sign}(y_{\{t+1\}} - y_t)) \times 100\%.$$

DA оценивает способность модели предсказывать направление движения ряда, но не учитывает величину ошибки, а только её знак.

Использование всех трёх метрик позволяет комплексно оценить модели по абсолютной ошибке, относительной ошибке и правильности направления прогноза.

1.9. Генерация дополнительных признаков временного ряда

В базовом формате финансовые данные представлены как последовательность свечных баров (OHLCV: open, high, low, close, volume) — цены открытия, максимума, минимума, закрытия и объёма торгов. Однако пять исходных параметров часто недостаточны для учёта сложной динамики рынка: всплесков ликвидности, изменения волатильности, краткосрочных импульсов и долгосрочных трендов. Для обогащения модели применяют процесс «feature engineering» (генерация признаков), превращающий одномерный ряд цен в многомерный набор, фиксирующий нелинейности и эффекты на разных временных масштабах.

1.9.1. Технические индикаторы

Технические индикаторы — заранее заданные функции от цен и объёмов, выявляющие специфические рыночные сигналы:

- Сглаживающие средние - выделяют направление тренда, устраняя высокочастотный шум:

- простая скользящая средняя (simple moving average, SMA)
 - экспоненциальная скользящая средняя (exponential moving average, EMA)
- Индикаторы момента (momentum) - измеряют изменение цены за фиксированный лаг, отражая силу движения.
- Осцилляторы - помогают определить зоны перекупленности и перепроданности:
 - индекс относительной силы (relative strength index, RSI)
 - схождение/расхождение скользящих средних (moving average convergence/divergence, MACD)
- Полосы Боллинджера (Bollinger Bands) строятся вокруг SMA с учётом стандартного отклонения, фиксируя периоды повышенной или пониженной волатильности.
- Средний истинный диапазон (average true range, ATR) оценивает средний диапазон колебаний.
- Объёмные индикаторы - отражают соотношение спроса и предложения через анализ ценовых изменений и объёмов:
 - on-balance volume (OBV)
 - chaikin money flow (CMF)

1.9.2. Статистические признаки

Автоматизированное извлечение статистических признаков (например, библиотекой TSFresh) позволяет получать сотни показателей, описывающих распределение и структуру ряда:

1. Моменты распределения: среднее, дисперсия, асимметрия, эксцесс.
2. Автокорреляционные меры: автокорреляционная функция (ACF) и частичная автокорреляционная функция (PACF) на различных лагах.
3. Энтропийные характеристики: спектральная энтропия после преобразования Фурье, пермутационная энтропия порядка d и лага τ .

4. Экстремумы: число пиков и впадин, амплитуды.
5. Длительность серий: длины монотонных участков (неубывания и невозрастания).
6. Выбросы и другие статистики.

Совместное использование технических и статистических признаков создаёт богатый многомерный вектор для каждой точки ряда. Это повышает точность и устойчивость прогнозов по сравнению с работой исключительно с первоначальными ценовыми данными.

1.10. Базовые методы прогнозирования

В этом разделе рассматриваются базовые методы прогнозирования, служащие эталоном для сравнения более сложных моделей.

1.10.1. Наивное повторение

Наивное повторение — это самый простой метод прогнозирования временного ряда. Прогноз на следующий шаг принимается равным последнему известному значению:

$$\hat{y}_{t+1} = y_t$$

Алгоритм не требует обучения и не имеет настраиваемых параметров: хранится лишь текущее наблюдение y_t , и оно сразу используется в качестве прогноза. Хотя метод очень прост, он часто служит базовым уровнем при сравнении качества более сложных моделей.

1.10.2. Скользящая средняя (simple moving average, SMA)

Скользящая средняя для прогноза следующего значения вычисляет среднее последних w наблюдений, сглаживая краткосрочные флуктуации и устраняя высокочастотный шум. Это позволяет выявлять направление тренда:

если текущая цена выше SMA, говорят о восходящем движении; ниже — о нисходящем.

$$\hat{y}_{t+1} = \frac{1}{w} \sum_{i=0}^{w-1} y_{t-i}$$

Алгоритм не требует предварительного обучения: по мере поступления нового значения y_{t+1} окно сдвигается — включается y_{t+1} , а самое старое наблюдение исключается, после чего прогноз снова вычисляется по той же формуле.

Преимущество:

- простота реализации и наглядность результатов.

Недостаток:

- равномерная весовая нагрузка на все w точек, что ведёт к запаздыванию прогноза и учёту устаревших данных.

1.10.3. Экспоненциальная скользящая средняя (exponential moving average, ЕМА)

Экспоненциальная скользящая средняя придаёт больший вес свежим наблюдениям и быстрее реагирует на изменения ряда, чем простая скользящая средняя (SMA).

Коэффициент сглаживания α обычно выбирают как:

$$\alpha = \frac{2}{w + 1},$$

где w — размер окна.

Основная формула ЕМА:

$$EMA_t = \alpha y_t + (1 - \alpha) EMA_{t-1}.$$

Преимущество:

- Последнему наблюдению присваивается вес α , а влияние предыдущих значений убывает экспоненциально:

$$\text{вес } y_{t-k} = \alpha (1 - \alpha)^k.$$

Недостаток:

- Хотя старые значения постепенно «затухают», они по-прежнему учитываются, что может вызывать небольшое запаздывание прогноза.

ЕМА сочетает адаптивность к новому сигналу с сохранением исторической информации, делая её популярным инструментом для скользящего анализа временных рядов.

1.10.4. Ансамбль скользящих средних (ensemble of moving averages)

Ансамбль скользящих средних объединяет несколько ЕМА (или SMA), рассчитанных с различными размерами окна, чтобы одновременно учитывать краткосрочные, среднесрочные и долгосрочные колебания. Пусть

- $EMA_t^{(w_{\text{short}})}$ - ЕМА с коротким окном w_{short} ,
- $EMA_t^{(w_{\text{medium}})}$ - ЕМА со средним окном w_{medium} ,
- $EMA_t^{(w_{\text{long}})}$ — ЕМА с длинным окном w_{long} ,

и веса w_{short} , w_{medium} , w_{long} таковы, что $w_{\text{short}} + w_{\text{medium}} + w_{\text{long}} = 1$, и тогда прогноз строится как взвешенная сумма:

$$\hat{y}_{t+1} = w_{\text{short}} EMA_t^{(w_{\text{short}})} + w_{\text{medium}} EMA_t^{(w_{\text{medium}})} + w_{\text{long}} EMA_t^{(w_{\text{long}})}.$$

Преимущества:

- Учитывает динамику на разных временных масштабах;
- Быстро реагирует на резкие изменения (короткое окно), сохраняя при этом сглаженность (длинное окно);
- Снижает шум и запаздывание, характерные для одиночных ЕМА., сглаживая шумы одной компоненты и уменьшая запаздывание другой.

Недостатки:

- Необходимо подбирать оптимальные веса и размеры окон, что усложняет настройку;
- Вычислительная сложность чуть выше по сравнению с одной ЕМА.

1.10.5. Метод дрейфа (drift method)

Метод дрейфа предполагает, что временной ряд движется с постоянным средним темпом изменения. Прогноз для шага $n+1$ вычисляется как последнее наблюдение y_n плюс средний «шаг» изменения за весь период:

$$\hat{y}_{n+1} = y_n + \frac{y_n - y_1}{n - 1}$$

Преимущества:

- Учитывает динамику на разных временных масштабах;
- Быстро реагирует на резкие изменения (короткое окно), сохраняя при этом сглаженность (длинное окно);

Недостатки:

- Плохо работает при резких изменениях или выраженной нелинейности, так как темп изменения остается постоянным, независимо от новых колебаний.

1.11 Статистические модели

Для построения прогнозов временных рядов часто применяют классические статистические методы (ARIMA, SARIMA, TBATS и др.), основанные на предположении о стационарности. Если это условие не выполняется, параметры, оценённые на исторических данных, становятся неактуальными, и прогноз теряет устойчивость.

1.11.1. Проверка стационарности

Перед преобразованиями необходимо проверить стационарность ряда с помощью теста Колмогорова–Дики–Фуллера (augmented Dickey–Fuller, ADF).

$$\Delta y_t = \alpha + \beta t + \gamma y_{t-1} + \sum_{i=1}^p \delta_{i\Delta} y_{t-i} + \varepsilon_t,$$

где $\Delta y_t = y_t - y_{t-1}$. Если оценённый γ статистически значимо меньше нуля (p-value < 0.05), ряд считается стационарным.

1.11.2. Удаление тренда и сезонности

1. Разложить ряд на компоненты (тренд, сезонность, остатки) методом STL или декомпозицией.
2. Исключить детерминированный тренд (полиномиальная регрессия или скользящее среднее).
3. Удалить сезонную составляющую.

1.11.3. Стабилизация дисперсии

Применить преобразование Бокса–Кокса:

$$y_t^{(\lambda)} = \begin{cases} \frac{y_t^\lambda - 1}{\lambda}, & \lambda \neq 0, \\ \ln(y_t), & \lambda = 0. \end{cases}$$

1.11.4. Разностное преобразование

1. Первичная разность: $\Delta y_t = y_t - y_{t-1}$.
2. При необходимости — вторая и далее: $\Delta^2 y_t = \Delta y_t - \Delta y_{t-1}$.
3. Для сезонных рядов — сезонная разность: $\Delta_s y_t = y_t - y_{t-s}$.
4. После каждого шага повторять ADF-тест.

1.11.5. Логарифмические доходности

Использовать вместо цен P_t лог-доходности

$$r_t = \ln \frac{P_t}{P_{t-1}}.$$

Прогнозируем $r_{t+1} \dots, r_{t+k}$ и восстанавливаем цену:

$$\hat{P}_{t+k} = P_t \cdot \exp \sum_{i=1}^k \hat{r}_{t+i}.$$

1.11.6. Алгоритм приведения ряда к стационарному

1. ADF-тест исходного ряда.
2. При нестационарности — удаление тренда и сезонности.
3. Преобразование Бокса–Кокса.
4. Разностное преобразование (включая сезонные разности).
5. Альтернатива: переход к лог-доходностям.

Только после приведения ряда к стационарному виду приступают к оценке параметров ARIMA-подобных моделей. Без этого этапа модели могут не сойтись или выдать нестабильные прогнозы.

1.11.7. ARIMA (autoregressive integrated moving average)

ARIMA объединяет три компонента: авторегрессию (AR), интегрирование разностями (I) и скользящее среднее (MA). При нестационарном ряде y_t сначала выполняют дифференцирование порядка d :

$$z_t = \Delta^d y_t, \Delta y_t = y_t - y_{t-1},$$

где d – порядок разностей, необходимый для достижения стационарности.

Стационарный ряд z_t моделируют как ARMA(p, q):

$$z_t = \sum_{i=1}^p \varphi_i z_{t-i} + \sum_{j=1}^q \theta_j \varepsilon_{t-j} + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma^2),$$

где p и q соответственно — порядки авторегрессии и скользящего среднего.

Эквивалентная компактная запись через оператор B ($By_t = y_{t-1}$):

$$(1 - \phi_1 B - \dots - \phi_p B^p)(1 - B)^d y_t = (1 + \theta_1 B + \dots + \theta_q B^q) \varepsilon_t.$$

Этапы подбора модели:

1. Определение d : выполняют ADF-тесты для $y_t, \Delta y_t, \dots$ до стационарности.
2. Выбор p, q : анализ ACF и PACF стационарного ряда z_t .
3. Сравнение моделей:

$$AIC = -2 \ln L_{\max} + 2(p + q + 1),$$

$$BIC = -2 \ln L_{\max} + (p + q + 1) \ln N.$$

4. Проверка: гарантируют стационарность AR-полинома и обратимость МА-полинома.

Прогнозирование:

- Одношаговый:

$$\hat{z}_{T+1} = \sum_{i=1}^p \phi_i z_{T+1-i} + \sum_{j=1}^q \theta_j \varepsilon_{T+1-j}, \quad \hat{y}_{T+1} = y_T + \hat{z}_{T+1}$$

- Мультишаговый: рекурсивно подставляют \hat{z} и $\hat{\varepsilon}$ для расчёта последующих шагов.

Преимущества:

- Гибкость в настройке параметров (p, d, q);
- Статистически обоснованные доверительные интервалы для прогнозов.

Ограничения:

- Требование стационарности ограничивает способность к моделированию нелинейных эффектов;
- Чувствительность к выбросам;
- При выраженной сезонности предпочтительнее SARIMA.

1.11.8. Auto-ARIMA (automatic ARIMA selection)

Auto-ARIMA — это автоматизированный метод подбора оптимальных порядков модели ARIMA, включая сезонные компоненты SARIMA. Алгоритм избавляет от ручной идентификации параметров по ACF/PACF и состоит из следующих этапов:

1. Определение порядка интегрирования d : последовательно выполняются ADF-тесты для $y_t, \Delta y_t, \dots$ до достижения стационарности.
2. Подбор сезонного порядка D : при наличии регулярных циклов (например, годовая сезонность) проводится ADF-тест сезонных разностей и выбирается минимальный D .
3. Поиск p, q (и P, Q для SARIMA): перебираются комбинации порядков авторегрессии и скользящего среднего, оцениваются информационные критерии AIC, BIC - выбирается модель с наименьшим значением критерия.

Преимущества:

- Существенно ускоряет и упрощает этап предварительного исследования;
- Снижает риск пропуска важной модели при большом объёме гипотез.

Ограничения:

- Комбинаторный поиск может быть вычислительно затратным при очень длинных рядах или множественных сезонностях;
- Оптимизация по AIC/BIC не гарантирует учёт сложных нелинейных связей в данных;
- При необходимости глубокого анализа часто требуется последующая ручная донастройка.

Несмотря на ограничения, Auto-ARIMA остаётся эффективным инструментом для быстрого получения работоспособной ARIMA/SARIMA-модели без ручной переборки параметров.

1.11.9. TBATS (trigonometric seasonality, Box–Cox transformation, ARMA errors, trend, seasonal components)

TBATS разработана для временных рядов с несколькими сезонными циклами и нестабильной дисперсией. Она объединяет пять компонентов:

- **Trigonometric seasonality** — модель сезонности через гармоники,
- **Box–Cox transformation** — стабилизация дисперсии,
- **ARMA errors** — учёт остаточной автокорреляции,
- **Trend** — локальный линейный тренд с адаптивным шагом,
- **Seasonal components** — суммарная сезонная часть.

1. Box–Cox преобразование:

$$y_t^{(\lambda)} = \begin{cases} \frac{y_t^\lambda - 1}{\lambda}, & \lambda \neq 0, \\ \ln(y_t), & \lambda = 0. \end{cases}$$

Параметр λ оценивается методом максимального правдоподобия для выравнивания дисперсии.

2. Сезонная модель через гармоники: Для каждого сезонного периода m рассчитывают

$$S_t(m) = \sum_{j=1}^{K_m} \left[a_j^{(m)} \cos\left(\frac{2\pi jt}{m}\right) + b_j^{(m)} \sin\left(\frac{2\pi jt}{m}\right) \right], K_m = \left\lfloor \frac{m}{2} \right\rfloor,$$

где $\{a_j^{(m)}, b_j^{(m)}\}$ — оценки амплитуд гармоник.

3. Локальный тренд:

$$\begin{aligned} \ell_t &= \ell_{t-1} + b_{t-1} + \alpha \varepsilon_t, \\ b_t &= b_{t-1} + \beta \varepsilon_t, \end{aligned}$$

α, β — коэффициенты сглаживания, — остатки.

4. Остаточная компонента (ARMA errors): ε_t моделируют как ARMA(p,q).

5. Итоговая модель:

$$\hat{y}_t = \text{Box} - \text{Cox}^{-1} \left(\ell_t + b_t + \sum_m S_t(m) + \varepsilon_t \right),$$

Преимущества:

- Поддержка множества (в том числе некрatных) сезонных периодов;
- Гибкая стабилизация дисперсии;
- Эффективное моделирование сложных циклов через тригонометрию.

Ограничения:

- Значительная вычислительная нагрузка при множестве сезонностей;
- Сложность настройки и потребность в большом объёме данных для оценки всех компонент.

1.12. Линейные модели

Линейные модели — базовый инструмент прогнозирования временных рядов благодаря простоте настройки, вычислительной эффективности и прозрачности. Их суть в том, что будущее значение ряда представляется как линейная комбинация заранее выбранных признаков (например, лагов, технических индикаторов).

1.12.1. DLinear (decomposed linear)

Модель DLinear сначала декомпозирует ряд на тренд \hat{T}_t , сезонность \hat{S}_t и остаток \tilde{y}_t :

$$\tilde{y}_t = y_t - \hat{T}_t - \hat{S}_t.$$

Далее остаток прогнозируется линейной регрессией по своим лагам:

$$\tilde{y}_{t+1} = \sum \varphi_i \tilde{y}_{t-i+1}.$$

Итоговый прогноз восстанавливается как:

$$\hat{y}_{t+1} = T_{t+1} + S_{t+1} + \tilde{y}_{t+1}.$$

Преимущества:

- Упрощает задачу линейной регрессии, убирая крупные циклы.
- Улучшает точность при выраженной детерминированной структуре.

Ограничения:

- Зависит от качества декомпозиции тренда и сезонности.
- Сложно моделировать взаимодействие между компонентами.

1.12.2. NLinear (nonlinear linear)

Модель NLinear расширяет линейную регрессию за счёт полиномиальных и взаимодействующих признаков. Для каждого шага строят вектор:

$$x_t = (y_{t-1}, y_{t-2}, \dots, y_{t-p}, y_{t-1}^2, y_{t-1}y_{t-2}, \dots),$$

и прогнозируют:

$$\hat{y}_{t+1} = w_0 + \sum_{k=1}^K w_k x_{t,k},$$

где K — количество признаков.

Преимущества:

- Сохраняет простоту линейной модели, но захватывает сглаженные нелинейности.

Ограничения:

- Быстро растёт число параметров.
- Требуется регуляризации или отбора признаков для предотвращения переобучения.

1.13. Рекуррентные модели

Рекуррентные нейронные сети (recurrent neural networks, RNN) и их улучшенная версия — LSTM (long short-term memory) — предназначены для работы с последовательными данными. В отличие от «классических» моделей,

они сохраняют состояние о предыдущих шагах, что позволяет уловить долгосрочные зависимости во временных рядах.

1.13.1. RNN

В RNN на каждом шаге t вычисляют скрытое состояние h_t и прогноз \hat{y}_t :

$$h_t = \sigma(W_x x_t + W_h h_{t-1} + b_h),$$
$$\hat{y}_t = W_y h_t + b_y,$$

где

- x_t — входной вектор на шаге t ,
- h_{t-1} — скрытое состояние предыдущего шага,
- σ — поэлементная нелинейная активация (\tanh или ReLU),
- W_x, W_h, W_y и b_h, b_y — обучаемые параметры сети.

Выход \hat{y}_t может усредняться по всем шагам или использоваться только для финального состояния h_t , в зависимости от задачи.

Преимущества:

- Учитывает последовательный характер данных.
- Хранит память о предыдущих значениях.

Недостатки:

- Подвержена проблемам затухающего и взрывающегося градиента при длинных последовательностях, что затрудняет обучение долгосрочных зависимостей.

1.13.2. LSTM (long short-term memory)

LSTM решает проблему затухающего градиента в RNN, вводя механизм гейтов и ячейку состояния для селективного запоминания и забывания информации на длительных отрезках.

Компоненты LSTM-блока:

- Гейт забывания f_t - решает, какую часть предыдущего состояния c_{t-1} сохранить.
- Входной гейт i_t - определяет, какую новую информацию добавить в ячейку.
- Гейт кандидатной информации \tilde{c}_t - генерирует вектор новых кандидатов.
- Ячейка состояния c_t - аккумулирует старое и новое содержимое.
- Выходной гейт o_t - выбирает, какую часть c_t передать дальше

$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f),$$

$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i),$$

$$\tilde{c}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c),$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t,$$

$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o),$$

$$h_t = o_t \odot \tanh(c_t)$$

$$\hat{y}_t = W_y h_t + b_y$$

Преимущества:

- Устойчивость к затухающему градиенту за счёт гейтов.
- Гибкое управление памятью и длительными зависимостями.

Недостатки:

- Высокая вычислительная сложность и большое число параметров.
- Требуется регуляризации (dropout, L2) и тщательной настройки гиперпараметров.

1.14. Методы машинного обучения

Методы машинного обучения применяются к временным рядам на основе сформированных признаков, объединяющих исторические значения ряда, технические индикаторы и статистические характеристики в многомерный вектор. В отличие от классических статистических моделей,

ML-алгоритмы автоматически выявляют сложные зависимости и адаптируются к разнообразным данным.

1.14.1. Линейная регрессия (linear regression)

Линейная регрессия прогнозирует y_{t+1} как линейную комбинацию признаков $x_{t,1}, x_{t,2}, \dots, x_{t,p}$:

$$\hat{y}_{t+1} = w_0 + \sum_{i=1}^p w_i x_{t,i}.$$

Параметры w_0, \dots, w_p находятся путём минимизации суммы квадратов ошибок:

$$\min_{w_0, \dots, w_p} \sum_{t=1}^{N-1} \left(y_{t+1} - w_0 - \sum_{i=1}^p w_i x_{t,i} \right)^2.$$

Коэффициент w_i показывает вклад признака $x_{t,i}$ в прогноз.

При большом числе признаков или их коллинеарности применяют регуляризацию:

- Ridge (L2): $\min_w \sum y - Xw^2 + \lambda \sum w_i^2$.
- Lasso (L1): $\min_w \sum y - Xw^2 + \lambda \sum |w_i|$.

Преимущества:

- Простота обучения и интерпретации;
- Вычислительная эффективность.

Ограничения:

- Линейность модели не всегда отражает сложные нелинейные зависимости в финансовых временных рядах.

1.14.2. К ближайших соседей (K-nearest neighbors, K-NN)

K-NN применяет принцип близости в признаковом пространстве к прогнозированию временных рядов.

Для каждого шага t формируется вектор признаков

$$x_t = (x_{t,1}, \dots, x_{t,p}),$$

включающий, например, предыдущие $\backslash(p\backslash)$ значений ряда или технические индикаторы.

При прогнозировании \hat{y}_{t+1} алгоритм находит множество индексов

$$\mathcal{N}_{K(t)} = \arg \min_{\{S \subseteq \{1, \dots, T-1\}, |S|=K\}} \sum_{i \in S} \|x_t - x_i\|^2,$$

где $\|\cdot\|$ — обычно евклидово расстояние.

Далее прогноз вычисляется как среднее ответов соседей:

$$\hat{y}_t^{+1} = (1/K) \times \sum_{i \in \mathcal{N}_{K(t)}} y_i^{+1}.$$

Преимущества:

- Не требует обучения параметров;
- Адаптивен к нелинейным структурам и сезонным повторениям;
- Прогноз формируется на основе реальных исторических точек.

Ограничения:

- Вычислительная сложность сильно растёт при большом объёме данных и высоком размерности;
- Чувствительность к выбору K и метрики расстояния;
- Ограниченная устойчивость к шуму и выбросам, так как вклад каждого соседа одинаков.

1.14.3. XGBoost (eXtreme Gradient Boosting)

XGBoost — высокоэффективная реализация градиентного бустинга деревьев решений, оптимизированная для скорости, распараллеливания и контроля переобучения. В задачах прогноза временных рядов на вход подается

вектор признаков, включающий лаги ряда, технические индикаторы и статистические характеристики.

Ключевые этапы обучения:

1. Инициализация прогноза: $F_0(x)$, часто константа (среднее).
2. Итерации $m = 1, \dots, M$:
 - a. Вычисление остатков $r_i^{(m)} = y_i - F_{m-1}(x_i)$.
 - b. Обучение очередного дерева f_m на остатках:

$$f_m = \arg \min_{\{f \in \mathcal{F}\}} \sum_{i=1}^n r_i^{(m)} - f(x_i)^2 + \Omega(f),$$

где $\Omega(f)$ учитывает глубину, число листьев и веса листовых узлов.

- c. Обновление ансамбля с темпом обучения η :

$$F_m(x) = F_{m-1}(x) + \eta f_m(x).$$

Регуляризация и оптимизации:

- Добавочный штраф $\Omega(f)$ контролирует сложность каждого дерева.
- Встроенное распараллеливание ускоряет построение деревьев.
- Поддержка пропусков в признаках без предварительной обработки.

Преимущества:

- Эффективная обработка больших наборов признаков;
- Автоматическое выявление сложных нелинейных взаимодействий;
- Встроенная регуляризация и параллельная обработка.

Недостатки:

- Требуется тонкая настройка гиперпараметров η (глубина, количество деревьев, регуляризационные коэффициенты);
- Вычислительно и памятьёмко на больших данных;
- Интерпретируемость ниже по сравнению с простыми моделями.

1.14.4. CatBoost (categorical boosting)

CatBoost — реализация градиентного бустинга деревьев, оптимизированная для работы с категориальными признаками и предотвращения смещения целевой переменной. В отличие от XGBoost и LightGBM, CatBoost применяет Ordered Target Statistics (OTS), при котором значения целевой переменной для категорий вычисляются только на записях, предшествующих текущему шагу, что исключает утечку информации и сохраняет хронологию данных.

1. Ordered Target Statistics для категорий - для каждой категориальной переменной и каждого наблюдения t формируется признак:

$$TS_j^{(t)} = \frac{\sum y_i + a \cdot prior_i^{1_{t-1}}}{((t-1) + a)},$$

где

- y_i — целевое значение на шаге $i < t$,
- a — сглаживающий коэффициент,
- $prior$ — априорное значение (например, среднее по всему ряду).

2. Градиентный бустинг деревьев - после генерации OTS-признаков CatBoost обучает последовательность регрессионных деревьев:

$$F_m(x) = F_{m-1}(x) + \eta f_m(x),$$

где

- f_m — дерево, минимизирующее $\sum y_i - F_m^{-1}(x_i) - f(x_i)^2 + \Omega(f)$,
- η — темп обучения,
- $\Omega(f)$ — регуляризационный штраф.

Преимущества:

- Эффективная работа с категориальными признаками без «one-hot encoding»;
- Отсутствие утечки целевой информации благодаря OTS;

- Встроенная регуляризация и распараллеливание;
- Специальные схемы валидации для временных рядов (например, скользящее окно по времени).

Ограничения:

- Повышенные требования к памяти и времени обучения при больших объемах;
- Сложность интерпретации ансамбля деревьев по сравнению с линейными моделями;
- Необходимость тщательной настройки гиперпараметров (глубина деревьев, learning rate, регуляторы).

1.14.5. Random Forest Regression

Random Forest — ансамбль декоррелированных деревьев решений, объединённых по принципу усреднения прогнозов. Для временных рядов на каждом шаге t формируется вектор признаков

$$x_t = (\text{лаг}_1, \dots, \text{лаг}_p, \text{тех. индикаторы}, \dots),$$

после чего обучается M деревьев: каждое дерево f_m строится на случайной подвыборке строк и случайном подмножестве признаков.

Итоговый прогноз:

$$\hat{y}_{t+1} = \frac{1}{M} \sum_{m=1}^M f_m(x_t).$$

Преимущества:

- Не требует масштабирования признаков;
- Автоматически оценивает важность признаков;
- Усреднение снижает дисперсию и устойчиво к выбросам;
- Эффективен при разнообразных типах признаков.

Ограничения:

- Потеря интерпретируемости у отдельных деревьев;
- Вычислительно затратен при большом M и высоком измерении признаков;
- Не учитывает временную структуру — требует тщательного формирования лаговых и агрегированных признаков

1.14.6. Prophet

Prophet — аддитивная модель временных рядов от Facebook, предназначенная для прогнозирования тренда, сезонности и эффектов праздников. Особенно полезна при наличии выбросов и пропусков данных.

Основная формула:

$$y_t = g(t) + s(t) + h(t) + \varepsilon_t, \varepsilon_t \sim \mathcal{N}(0, \sigma^2).$$

1. Тренд $g(t)$ - кусочно-линейная функция с автоматически определяемыми точками перегиба:

$$g(t) = (k + a(t)^T \delta) \cdot t + (m + a(t)^T \gamma),$$

где

- k, m — базовый наклон и уровень,
 - $a(t) \in \{0,1\}^L$ — индикаторы L точек перегиба,
 - δ, γ — приращения в этих точках.
2. Сезонность $s(t)$ - ряд Фурье с периодом P (по умолчанию $P=365.25$ или $P=7$):

$$s(t) = \sum \left[a_n \cdot \cos 2\pi n t / P + b_n \cdot \sin 2\pi n t / P \right]_{n=1}^N$$

где N — число гармоник (обычно 10–20).

3. Праздничные эффекты $h(t)$

$$h(t) = \sum_{j=1}^J \kappa_j \mathbf{1}\{t \in \text{Holiday}_j\},$$

где пользователь задаёт список J праздников, а κ_j — их влияния.

Преимущества:

- Автоматическая детекция точек перегиба;
- Поддержка нескольких сезонностей и праздников;
- Устойчивость к выбросам и пропускам;
- Интуитивно понятная аддитивная декомпозиция.

Ограничения:

- Аддитивная модель не учитывает мультипликативные взаимодействия;
- Для мультипликативных эффектов требуется логарифмирование данных;
- Не использует экзогенные регрессоры без дополнительного задания.

1.15. VAR (vector autoregression)

Vector Autoregression (VAR) — обобщение авторегрессии на многомерный случай, в котором каждое значение вектора зависит от прошлых значений всех компонентов. Это позволяет моделировать перекрёстные влияния между несколькими рядами одновременно. В финансовом контексте VAR полезен там, где цены разных активов, объёмы торгов и макроиндикаторы взаимосвязаны и эволюционируют совместно.

Пусть $\mathbf{y}_t = (y_{1t}, y_{2t}, \dots, y_{Kt})^\top$ — вектор из K временных рядов. Тогда VAR(p) записывается как $\mathbf{y}_t = \sum_{i=1}^p \Phi_i \mathbf{y}_{t-i} + \mathbf{c} + \varepsilon_t$, где

- $\Phi_i \in \mathbb{R}^{K \times K}$ — матрицы коэффициентов автозависимости,
- $\mathbf{c} \in \mathbb{R}^K$ — вектор констант,
- $\varepsilon_t \sim \mathcal{N}(\mathbf{0}, \Sigma)$ — вектор белого шума.

Оценка параметров выполняется как набор K независимых OLS-регрессий при условии стационарности \mathbf{y}_t .

Прогноз и анализ:

- Одношаговый и мультишаговый прогноз: предсказывают y_{t+1}, \dots, y_{t+h} .
- Impulse response function (IRF): показывает, как единичный шок в одной переменной влияет на другие через лаги.

Преимущества:

- Учитывает перекрёстные автозависимости;
- Позволяет анализировать передаточные механизмы (IRF);
- Простота оценки через OLS.

Ограничения:

- Число параметров $K^2p + K$ быстро растёт;
- Требует большого объёма данных и стационарности;
- Может переобучаться, поэтому часто используют регуляризацию или Bayesian VAR.

1.16. Библиотека tsai

tsai — высокоуровневая библиотека на базе PyTorch и FastAI для быстрого прототипирования и обучения современных моделей на временных рядах [6]. Она предоставляет:

- Готовые архитектуры: широкий набор SOTA-решений для классификации, регрессии.
- Утилиты предобработки: масштабирование, разбиение на окна, генерация лаговых признаков.
- Систему колбэков: мониторинг метрик, ранняя остановка, логирование и визуализация.

Некоторые доступные архитектуры доступные в tsai:

- InceptionTime [7] — сверточная сеть с Inception-блоками, эффективно обрабатывает паттерны разной длины.
- XceptionTime [8] — углублённая сверточная модель с разделёнными (depthwise) свёртками для экономии ресурсов и повышения выразительности.
- XCM [9] — гибридная CNN/RNN-модель, объединяющая свёртки по времени и по каналам для извлечения пространственно-временных признаков.
- gMLP (Gated MLP) [10] — полностью линейная архитектура с «шлюзами» (gates), управляющими потоком информации, без внимания и сверточных слоёв.
- TSSequencerPlus [11] — вариация Sequencer: каждый временной шаг проходит через слой нормализации и MLP, что позволяет моделировать длинные зависимости без механизмов внимания.

Это лишь небольшой набор из постоянно расширяющегося каталога моделей tsai, доступного для гибкого экспериментирования и быстрой валидации идей на временных рядах.

1.17. FEDOT (framework for evolutionary design of optimization techniques)

FEDOT — платформа для автоматического конструирования гибридных конвейеров машинного обучения и прогнозирования временных рядов на основе эволюционных алгоритмов. Вместо ручной сборки фиксированной архитектуры, FEDOT эволюционно комбинирует операции предобработки, модели и постобработку в едином пайплайне [12].

Ключевые шаги для работы с временными рядами:

- Генерация начальной популяции: создаются разные конвейеры со случайным набором операторов:

- трансформации признаков (лагирование, дифференцирование, нормализация),
 - базовые модели (ARIMA, Prophet, деревья, линейная регрессия),
 - ансамблевые методы.
- Эволюционный отбор: используют операторы, которые замещают, вставляют или удаляют компоненты пайплайнов:
 - скрещивания (crossover),
 - мутации (mutation),
- Многоцелевой отбор: оценивают конвейеры по точности MAPE, RMSE и по сложности, что позволяет находить баланс между качеством и вычислительными затратами.
- Формирование финального пайплайна: отобранная лучшая модель адаптируется к особенностям ряда (мультивариантность, пропуски, сезонность).

Преимущества:

- Полная автоматизация конструирования конвейеров;
- Гибкость в работе с любыми типами данных;
- Прозрачность итоговой структуры пайплайна и возможность ручной донастройки.

Ограничения:

- Высокие вычислительные затраты при больших объёмах данных;
- Зависимость качества от правильно заданного пространства операторов и параметров эволюции;
- Медленная сходимость при сложных задачах без предварительного сужения поиска.

1.18. Chronos (chronos-forecasting)

Chronos — это набор предварительно обученных трансформерных моделей для вероятностного прогнозирования временных рядов,

разработанный в Amazon Science и доступный под лицензией Apache 2.0 [13]. В отличие от классических подходов, где ряд кодируется в виде признаков и подаётся на вход RNN/LSTM, Chronos обрабатывает временные серии как текст: значения квантуются и превращаются в токены, а затем автокорректирующая T5-архитектура учится восстанавливать следующие «слова» (будущие значения).

Chronos рассматривает временной ряд как последовательность дискретных токенов. Модель обучается на задаче автозаполнения: по префиксу-«предыстории» генерируется несколько возможных продолжений, что позволяет получить не одиночный прогноз, а распределение будущих траекторий.

Ключевые компоненты:

- Квантизация и токенизация: преобразование непрерывных значений в дискретный словарь с учётом динамического диапазона ряда.
- T5-архитектура: укороченный трансформер с механизмом автогенерации, оптимизированный для работы с короткими и длинными контекстами временных рядов.
- Вероятностный вывод: метод автогенерации (autoregressive sampling) выдаёт сразу несколько траекторий, что важно для оценки доверительных интервалов.
- Zero-shot и дообучение: Предварительно обученные модели демонстрируют сильные zero-shot результаты на десятках бенчмарков и могут тонко дообучаться на конкретных наборах данных.

Таким образом, Chronos представляет собой современный инструмент для исследовательских проектов по прогнозированию временных рядов, особенно когда требуется быстро получить вероятностные предсказания без ручного конструирования сложных архитектур.

1.19. Прогнозирование временных рядов с помощью больших языковых моделей

Исследования показывают, что большие языковые модели (large language models, LLM), первоначально предназначенные для обработки текстов, могут применяться к задачам анализа и прогнозирования временных рядов, в том числе финансовых [14, 15].

1.19.1. Прямой подход (one-shot prompting)

В простейшей схеме LLM получают на вход запрос (prompt), содержащий историческую последовательность значений и инструкцию. Например:

«Ты - эксперт по прогнозированию финансовых временных рядов. У нас есть следующий ряд: 102.5, 103.0, ..., 102.0, 101.8.

Спрогнозируй одношаговый прогноз и верни только число.»

Несмотря на свою простоту, этот метод позволяет LLM генерировать адекватные прогнозы на следующий временной шаг. Однако точность такого подхода сильно зависит от качества prompt'a и ограничена возможностями модели без дополнительной донастройки.

1.19.2. Подход StockTime

В работе [16] предложена мультимодальная архитектура StockTime, в которой ценовые ряды кодируются как «патчи» фиксированной длины l , а затем обрабатываются совместно с текстовыми подсказками.

1. Формирование патчей - пусть ценовой ряд $\{p_1, p_2, \dots, p_D\}$ имеет длину D . Он разбивается на $n = \lfloor D/l \rfloor$ непересекающихся сегментов, где каждый «патч» h_i трактуется как один токен:

$$h_i = \{p_{(i-1)l+1}, \dots, p_{il}\}, i = 1, \dots, n.$$

2. Авторегрессионный энкодер - сегмент h_i подаётся на LSTM-энкодер, выдающий скрытое представление

$$pe_i \in \mathbb{R}^{d_{llm}}.$$

3. Текстовые подсказки - из каждого сегмента извлекают дополнительные признаки:

- a. корреляции между акциями,
- b. статистические тренды (минимум, максимум, среднее, изменение),
- c. временные метки.

Они формируют текстовый prompt, который кодируется замороженным LLM (например, LLaMA3-8B) в эмбединг ce_i .

4. Объединение эмбедингов - числовые и текстовые представления объединяются:

$$e_i = pe_i + ce_i.$$

5. Прогнозирование и оптимизация - вектор $\{e_1, \dots, e_n\}$ подаётся на вход замороженного LLM. Обучается лишь верхний линейный слой, который предсказывает токен h_{n+1} . Функция потерь:

$$MSE = \left(\frac{1}{(n-l)} \right) \sum \left\| \hat{h}_i - h_i \right\|_2^2 \Big|_{i=2}^{n+1}.$$

На наборах S&P 100 (дневные и почасовые) и четырёх публичных датасетах StockTime снизил MSE и повысил Information Coefficient по сравнению с general LLM, FinLLM и специализированными time-series LLM (Times-LLM, AutoTimes, FPT). При этом модель остаётся ресурсно эффективной (замороженный бэкбон LLM) и гибкой (произвольная длина lookback без изменения архитектуры).

StockTime демонстрирует эффективность интеграции временных и текстовых признаков в LLM для прогноза финансовых рядов и открывает перспективы развития мультимодальных подходов в алгоритмической торговле и риск-менеджменте.

2. ДЕТЕКТИРОВАНИЕ АНОМАЛИЙ И ПОИСК ТОЧЕК ПЕРЕЛОМА В ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДАХ

Обнаружение аномалий (anomaly detection) и точек перелома (change-point detection) являются важнейшими задачами анализа временных рядов. В финансовом контексте они помогают фиксировать смену трендов, начало кризисов и возможности для входа и выхода из позиций.

2.1. Аномалии

Аномалия — отдельная точка или группа точек, существенно отличающиеся от ожидаемого поведения ряда. Выделяют три основных типа:

- Точечные аномалии (point anomalies). Отдельные выбросы, не вписывающиеся в статистические характеристики.
- Контекстуальные аномалии (contextual anomalies). Значения аномальны только в определённом контексте (например, объём торгов необычен в конце сессии).
- Коллективные аномалии (collective anomalies). Набор последовательных наблюдений, которые вместе образуют аномалию (например, серия резких падений цены).

2.2. Точки перелома

Точка перелома — момент, при котором меняются параметры ряда (среднее, дисперсия, автокорреляция или параметры модели). В финансовых временных рядах это может быть:

1. Смена тренда. Переход от восходящего к нисходящему движению или наоборот.
2. Изменение волатильности. Переход в фазу повышенной активности торгов.

3. Сдвиг корреляций. Изменение взаимосвязей между активами или секторами.

2.3. Классификация методов

Методы обнаружения аномалий и точек перелома делятся на три группы:

1. Статистические методы

- a. Контрольные карты (Shewhart, CUSUM, EWMA). Отслеживают отклонения остатков модели (например, ARIMA).
- b. Тесты на стационарность и разрывы:
 - i. ADF (augmented Dickey–Fuller) и KPSS — проверка на стационарность для выявления возможных структурных сдвигов.
 - ii. Pettitt, Bai–Perron — прямое выявление точек изменения параметров ряда.

2. Непараметрические методы

- a. Сегментация с оптимизацией (binary segmentation, PELT). Минимизируют суммарную стоимость разбиения (например, сумма квадратов ошибок).
- b. Методы ядра (kernel change-point detection). Оценивают изменения распределения через RKHS-меры.

3. Методы машинного обучения и глубокого обучения

- a. Кластеризация и k-NN. Обнаружение выбросов по расстоянию до «облака» нормальных точек.
- b. Автоэнкодеры (autoencoders, VAE). Ошибки восстановления указывают на аномалии.
- c. Рекуррентные автоэнкодеры (LSTM-AE). Сильно отклоняющийся прогноз означает аномалию или смену режима.

2.4. Генерация аномалий

Для объективной оценки точности алгоритмов обнаружения аномалий в финансовых временных рядах часто внедряются синтетические выбросы (synthetic anomalies). Ниже описан один из воспроизводимых подходов.

2.4.1. Оценка базовой плотности доходностей

Пусть y_t ($t = 1, \dots, T$) — финансовый ряд без выраженных аномалий на подмножестве индексов $\mathcal{T}_0 \subset \{1, \dots, T\}$.

Для получения стационарного ряда вычисляют логарифмические доходности:

$$r_t = \ln\left(\frac{y_t}{y_{t-1}}\right), t = 2, \dots, T$$

На участке \mathcal{T}_0 оценивают эмпирическое распределение F_r доходностей двумя способами:

- Параметрическая оценка (Gaussian):

$$\hat{\mu} = \left(\frac{1}{(|\mathcal{T}_0| - 1)}\right) \sum_{t \in \mathcal{T}_0} r_t,$$
$$\hat{\sigma}^2 = \left(\frac{1}{(|\mathcal{T}_0| - 1)}\right) \sum_{t \in \mathcal{T}_0} (r_t - \hat{\mu})^2.$$

- Непараметрическая оценка (kernel density): оценка плотности с помощью ядровой функции (например, Гауссово ядро).

Полученная функция распределения F_r далее используется для генерации синтетических аномалий с известными метками.

2.4.2 Генерация базового синтетического ряда

Пусть F_r — распределение лог-доходностей, оценённое на «чистой» от аномалий части ряда. Тогда синтетический ряд $\{\tilde{y}_t\}$ строится так:

1. Инициализация: $\tilde{y}_1 = y_1$.
2. Рекурсивная генерация:

$$\begin{aligned}\tilde{r}_t &\sim F_r, \\ \tilde{y}_t &= \tilde{y}_{t-1} \cdot \exp(\tilde{r}_t), \\ t &= 2, \dots, T.\end{aligned}$$

В результате получаем модельный ряд $\{\tilde{y}_t\}$, который воспроизводит статистику доходностей исходных данных, но лишён исходных выбросов.

2.4.3 Внедрение аномалий

Для внедрения N синтетических аномалий выбирают индексы $\{\tau_1, \dots, \tau_N\} \subset \{2, \dots, T\}$ обычно равномерно или случайно.

На каждом τ_i заменяют сгенерированную доходность \tilde{r}_{τ_i} на «экстремальное» значение Δ_i . Например:

- Случайная аномалия: $\Delta_i \sim \mathcal{N}(0, \hat{\sigma})$ — моделирует выброс из «толстохвостого» распределения.
- Фиксированный скачок: $\Delta_i = \ln(1.1)$ — соответствует однократному росту на 10 %.

После задания $\tilde{r}_{\tau_i} = \Delta_i$ пересчитывают все последующие цены по рекурсии:

$$\begin{aligned}\tilde{y}_t &= \tilde{y}_{t-1} \cdot \exp(\tilde{r}_t), \\ t &= \tau_i + 1, \dots, T.\end{aligned}$$

В результате в синтетическом ряде получаются N заранее известных точек аномалий, что позволяет объективно оценить точность алгоритмов обнаружения.

2.4.4. Метрики оценки качества

Пусть $A = \{\tau_i\}$ — истинные метки аномалий, $\hat{A} = \{\hat{\tau}_j\}$ — предсказанные метки. Классические метрики:

$$Precision = \frac{|\hat{A} \cap A|}{|\hat{A}|},$$

$$Recall = \frac{|\hat{A} \cap A|}{|A|},$$

$$F_1 = 2 \frac{Precision \times Recall}{Precision + Recall}.$$

Более продвинутую оценку даёт NAB score (Numenta Anomaly Benchmark), который учитывает не только факт обнаружения, но и временную близость предсказания к истинной метке и принимает значение в диапазоне $[-1, +1]$. В дальнейшем NAB score не применяется, но служит примером метрики, чувствительной к своевременности детекции.

Таким образом, комбинация Precision, Recall, F_1 и (при необходимости) NAB score обеспечивает всестороннюю оценку качества детекторов аномалий в финансовых временных рядах.

2.5. Статистические методы детектирования аномалий

Рассмотрим четыре базовых статистических детектора для одномерных временных рядов. Во всех «скользящих» методах используется окно длины w ; глобальные методы анализируют весь доступный ряд.

2.5.1. IQR-детектор (OutlierDetector)

Идея: выбросы определяются по межквартильному размаху. Формулы:

Q_1, Q_3 — 25% и 75% перцентили ряда x_t ,

$$IQR = Q_3 - Q_1,$$

$$Lower = Q_1 - k \cdot IQR,$$

$$Upper = Q_3 + k \cdot IQR$$

Аномалия, если $x_t < Lower$ или $x_t > Upper$.

Плюс: простота, нет требований к распределению. Минус: чувствителен к несимметрии и длинным хвостам.

2.5.2. Скользящее среднее (MeanAnomalyDetector)

Идея: сравнение с локальным средним и дисперсией. Формулы (окно w):

$$\mu_t = \frac{1}{w} \sum_{i=t-w+1}^t x_i, \sigma_t = \sqrt{\frac{1}{w} \sum_{i=t-w+1}^t (x_i - \mu_t)^2}.$$

Аномалия, если $x_t > \mu_t + \alpha \cdot \sigma_t$ или $x_t < \mu_t - \alpha \cdot \sigma_t$, где α (обычно 3) — порог.

Плюс: учитывает локальные тренды. Минусы: требует надёжной оценки σ_t , чувствителен к размеру окна.

2.5.3. Параметрический детектор (DistributionBasedAnomalyDetector)

Идея: аппроксимация всего ряда нормальным распределением. Формула p-value:

$$\hat{\mu}, \hat{\sigma}^2 - \text{оценки по всему ряду}$$
$$p_t = 2 \cdot \left[1 - \Phi \left(|x_t - \hat{\mu}| / \hat{\sigma} \right) \right].$$

Аномалия, если $p_t < \beta$, (обычно $\beta=0.05$).

Плюс: эффективен при нормальных данных. Минус: некорректен при значительных отклонениях от нормальности.

2.5.4. Хвостовой детектор (RareDistributionDetector)

Идея: на каждом шаге выявлять n самых экстремальных точек в окне.

Алгоритм:

1. Берут последние w значений $\{x_{t-w+1}, \dots, x_t\}$.
2. Сортируют по возрастанию и присваивают ранг $r \in [1, w]$.

3. Аномалия, если $r \leq n$ или $r \geq w - n + 1$.

Параметры: n — глубина хвоста; w — размер окна. Плюс: фокус на строго экстремальных событиях; Минусы: игнорирует умеренные выбросы и зависит от выбора n .

Эти методы образуют базовый инструментарий для статистического анализа аномалий перед применением более сложных ML-подходов.

2.6. Прогнозный подход

Метод основан на анализе ошибки одноступенчатого прогноза: точки, где прогноз существенно отклоняется, считаются аномальными.

1. Построение прогноза - выбирается любой метод предсказания. Например: Экспоненциальная скользящая средняя (ЕМА) с коэффициентом α .
2. Вычисление ошибки (MAPE) - для каждого момента t считаем

$$\hat{y}_t = \text{прогноз на шаг } t,$$
$$MAPE_t = |y_t - \hat{y}_t| / y_t \times 100\%.$$

3. Критерий аномалии - точка t считается аномалией, если $MAPE_t > \delta$, где δ — заранее заданный порог (подбирается на валидационной выборке или как k -кратное среднее/стандартное отклонение ошибки).

Преимущества:

- Простота реализации;
- Автоматическая адаптация к локальным изменениям динамики.

Ограничения:

- Необходимость выбора порога δ ;
- Возможность «пропуска» аномалий при плавном нарастании ошибки;
- Зависимость качества от выбранного метода прогноза.

2.7. Isolation Forest

Isolation Forest — ансамблевый метод обнаружения аномалий, основанный на идее, что редкие и выбивающиеся объекты легче «изолировать» при случайных разбиениях пространства признаков.

Идея метода:

- Строится множество деревьев, где на каждом узле выбирается случайный признак и случайный порог разбиения.
- Аномальные объекты попадают в верхние (близкие к корню) листья, так как для их изоляции требуется меньшее число разбиений.

Для объекта x определяют среднюю длину пути $h(x)$ от корня до листа в ансамбле из T деревьев. Аномальная оценка рассчитывается как:

$$s(x, n) = 2^{-\frac{E[h(x)]}{c(n)}},$$
$$c(n) \approx 2H_{n-1} - \frac{2(n-1)}{n},$$
$$H_{n-1} = \sum_{k=1}^{n-1} \frac{1}{k},$$

$n = \{\text{число объектов в каждом дереве}\}.$

Чем ближе $s(x, n)$ к 1, тем более аномален объект.

Преимущества:

- Не требует предположений о распределении данных;
- Линейная сложность по числу объектов и признаков;
- Не требует ручного задания порогов — аномалии выявляются по высокой оценке.

Ограничения:

- Чувствительность к числу деревьев T и максимальной глубине;

- Не учитывает временную структуру ряда без явного добавления лаговых признаков;
- Возможны ложные срабатывания при высокой размерности, если аномалии не сильно удалены.

2.8. Кластерный подход

Кластерные методы обнаруживают аномалии как точки, плохо вписывающиеся в образованные группы. Важно заранее выбрать пространство признаков (лаги, индикаторы) и размер окна/понижение размерности.

2.8.1. K-Means

Идея: после снижения размерности PCA объекты группируют в K кластеров, а удалённые точки считаются аномалиями.

PCA-преобразование: $z_t = W^T(x_t - \bar{x})$, где $W \in \mathbb{R}^{p \times d}$ — первые d собственных векторов, \bar{x} — средний вектор.

Кластеризация: минимизация внутрикластерного SSE.

Критерий аномалии: $d_t = \|z_t - \mu_{\{c(t)\}}\|$, аномалия, если $d_t > \delta$.

Плюс: простота, детектирование глобальных структур. Минусы: требует задания K, чувствителен к выбору d и центров.

2.8.2. K-NN

Идея: объекты с большой средней дистанцией до своих K ближайших соседей считаются аномалиями.

Расчёт дистанции: $\bar{d}_t = \left(\frac{1}{K}\right) \times \sum \|x_t - x_i\|_{i=1}^K$, где x_i — i-й ближайший сосед по евклидову (или другой) метрике.

Критерий аномалии: $\bar{d}_t > \delta$.

Плюсы: не требует кластеров, адаптивен к локальной плотности.
Минусы: выбор K и меры расстояния влияет на результаты; вычислительно затратен при больших данных.

Оба метода выявляют объекты «вне» основной массы и подходят для временных рядов, приведённых к многомерному признаковому представлению.

2.9. Автоэнкодеры

Автоэнкодер — это нейросетевая модель, состоящая из двух частей:

- Энкодер f_θ , сжимающий вход x_t в латентное представление $z_t = f_\theta(x_t) \in \mathbb{R}^d$.
- Декодер g_ϕ , восстанавливающий аппроксимацию $\hat{x} = g_\phi(z_t)$.

Функция потерь: $\min_{\{\theta, \phi\}} \sum \|x_t - g_\phi(f_\theta(x_t))\|^2_{t=1}^T$.

Критерий аномалии: пусть реконструкционная ошибка $\epsilon_t = \|x_t - \hat{x}_t\|$.
Точка t считается аномалией, если $\epsilon_t > \delta$, где δ — заранее выбранный порог.

Преимущества:

- Автоматическое извлечение информативных нелинейных признаков.
- Гибкость в выборе глубины и ширины латентного пространства.

Недостатки:

- Необходим достаточный объём «чистых» данных для обучения.
- Чувствительны к архитектуре сети и выбору порога δ .
- Требуют подбора гиперпараметров (размер латентного слоя, регуляризация и т. п.).

2.10. Детектирование многомерных аномалий (PyOD)

Для многомерного анализа аномалий удобно использовать библиотеку PyOD (Python Outlier Detection), которая реализует разнообразные алгоритмы без предположений о распределении данных.

- HBOS (Histogram-based Outlier Score): Строит отдельные гистограммы для каждого признака, оценивает вероятность нахождения в каждом бине и объединяет их (произведение или сумма), предполагая независимость признаков.
- ABOD (Angle-based Outlier Detection): Вычисляет дисперсию углов между векторами от рассматриваемой точки к другим объектам; аномалии характеризуются низкой вариативностью углов. Метод точен, но ресурсоёмок при больших выборках.
- LOF (Local Outlier Factor): Оценивает локальную плотность точки относительно плотности её k -ближайших соседей; объекты в менее плотных областях получают высокие LOF-значения.
- CBLOF (Cluster-based Local Outlier Factor): Делит данные на «большие» и «малые» кластеры, аномалии — это объекты из небольших или отдалённых кластеров, оценка основана на размере и дистанции до крупных кластеров.
- MCD (Minimum Covariance Determinant): Находит робастную оценку ковариационной матрицы на подмножестве точек с минимальным детерминантом; точки с высоким Махалобисовым расстоянием считаются выбросами.
- PCA (Principal Component Analysis): Реконструирует объекты по первым d главным компонентам и измеряет норму остатка $\|x - \hat{x}\|$; аномалии дают большие ошибки восстановления.
- LSCP (Locally Selective Combination in Parallel): Строит ансамбль детекторов и для каждой точки выбирает ограниченное число наиболее

надёжных моделей в её локальном окружении, объединяя их прогнозы для финальной оценки.

Выбор оптимального метода зависит от размера и структуры данных, размерности признакового пространства, а также требований к вычислительным ресурсам и интерпретируемости.

2.11. Ruptures

Ruptures — это Python-библиотека для офлайн-обнаружения точек перелома (change-point detection) в одномерных и многомерных временных рядах [17]. Она предоставляет высокоуровневый API и набор оптимизированных алгоритмов, позволяющих быстро экспериментировать с разными методами и функциями стоимости.

Ruptures предназначена для «последующего» (offline) анализа рядов, когда весь сигнал известен заранее. Цель — найти моменты времени, в которых статистические свойства ряда изменяются. Библиотека подходит для задач финансовой аналитики, мониторинга промышленных процессов и других областей, где важны структурные сдвиги.

Ruptures реализует несколько популярных подходов к поиску точек перелома:

- Pelt (Pruned Exact Linear Time) – точное решение с линейной сложностью в среднем;
- Binseg (Binary Segmentation) – рекурсивная двоичная сегментация, быстрый приближённый метод;
- Bottom-up – объединение сегментов по наименьшему росту стоимости;
- Window – скользящее окно фиксированной длины с оценкой изменения внутри;
- Kernel – непараметрический метод, оценивающий разрыв распределения через ядровую функцию.

Для оценки «качества» сегмента используются различные cost-функции, например:

- CostL2 — сумма квадратов ошибок (Gaussian likelihood);
- CostAR — автогрегрессия порядка p ;
- CostRbf — RBF-ядерный метод;
- CostRank — непараметрический ранговый критерий.

Преимущества:

- Большой выбор алгоритмов и cost-функций;
- Простота экспериментирования с параметрами;
- Оптимизированная производительность и масштабируемость.

Ограничения:

- Оффлайн-режим не подходит для стриминговых данных;
- Чувствителен к выбору параметров (penalty, number of breakpoints, window size);
- Для многомерных данных требуется тщательный подбор cost-функции или объединение сигналов.

2.12. Детектирование аномалий с помощью LLM

Большие языковые модели (large language models, LLM) могут обрабатывать числовые последовательности, представленные в текстовом виде, и использоваться для обнаружения аномалий во временных рядах [18, 19]. Существует два основных сценария:

2.12.1 Пакетный анализ

В этом режиме одной передаётся вся история ряда и инструкция обнаружить аномалии. Пример prompt'a:

«Ты — эксперт по детектированию аномалий. Последовательность ежедневных цен закрытия за последние 10 дней: «102.5, 103.0, 102.8, 105.2, 104.9, 150.0, 104.7, 104.3, 104.0, 103.8». Определи индексы аномалий (начиная с 1) и верни только номера дней.»

Модель, обнаружив выброс 150.0, вернёт “6”. Плюс: один запрос — вся серия. Минус: теряется точность при длинных списках и ограничении контекста.

2.12.2. Детектирование с использованием прогноза

Здесь LLM получают короткий контекст, прогноз и фактическое значение, чтобы судить об аномалии:

«Ты — эксперт по детектированию аномалий. Предыдущий значения ряда: «100.4, 102.5, 102.3, 101.8, 101.0, 100.8». Прогноз для следующего значения 100.0, фактическое значение — 108.0. Является ли это аномалией? Ответь “Да” или “Нет”».

При необходимости в prompt можно добавить допустимый порог расхождения, например “аномалия, если $|\text{факт} - \text{прогноз}| > 5\%$ ”. Плюсы: учитывает прогноз; более лаконичен. Минусы: требует вычисления прогноза; вычисляет значение одной точки.

Результат LLM-детектора (1/0 или индекс) можно использовать как дополнительный категориальный признак в последующих моделях, что помогает учитывать резкие изменения в динамике финансовых показателей.

3. ОЦЕНКА ВЛИЯНИЯ НОВОСТЕЙ НА СТОИМОСТИ АКТИВОВ С ПОМОЩЬЮ LLM

В работе продемонстрирована эффективность LLM при ежедневном отборе активов по новостным заголовкам: портфель показал более чем пятикратный рост за 1.5 года на рынке США без учёта комиссий [20].

Собираются новостные заголовки, привязанные к конкретным компаниям, секторам и рынку, очищаются от дубликатов.

Для оценки используется системный промпт:

«Ты — финансовый эксперт.

Оцени влияние новости `{заголовок}` на цену акций компании N:

- ПОЛОЖИТЕЛЬНОЕ — цена вырастет
- НЕЙТРАЛЬНОЕ — влияние слабое или неочевидное
- НЕГАТИВНОЕ — цена упадет

Отвечай только одним словом».

Выполняется преобразование ответов: «ПОЛОЖИТЕЛЬНОЕ» → 1, «НЕЙТРАЛЬНОЕ» → 0, «НЕГАТИВНОЕ» → -1.

Все значения оценок новостей для конкретной компании за день складываются и усредняются, таким образом получается оценка новостного фона компании. Для секторов и рынка оценка вычисляется аналогично.

Портфель инвестиций строится из компаний с лучшими оценками. Для учёта влияния всех новостей вводится взвешенная итоговая оценка $score = w_1 * company_{score} + w_2 * sector_{score} + w_3 * market_{score}, w_1 + w_2 + w_3 = 1$.

Оценка новостей может служить дополнительным числовым признаком для предсказательных моделей, так как расширяет «область видимости» моделей.

4. РАЗРАБОТКА КОМБИНИРОВАННОГО ПОДХОДА ПРОГНОЗИРОВАНИЯ ФИНАНСОВЫХ ВРЕМЕННЫХ РЯДОВ С ПОМОЩЬЮ LLM

Для повышения точности мы предлагаем комбинированный подход, в котором LLM не ограничиваются только историей цен, а получают дополнительные источники информации. В частности, модель будет учитывать технические и статистические признаки, сигналы аномалий, финансовые новости — то есть те же данные, на которые ориентируются инвесторы при принятии решений. Это расширение «прямого метода» (one-shot prompting), применяемого к одномерным рядам (раздел 1.19.1), позволит LLM глубже улавливать скрытые зависимости и учитывать более широкий контекст при прогнозировании следующих значений временного ряда.

4.1. Переход к многомерным рядам

Для расширения «прямого метода» на многомерные входы сначала формируется набор кандидатов-признаков, описанных в разделе 1.9 (технические индикаторы, статистические характеристики, lag-признаки и т. п.). Из этого пула отбираются наиболее информативные переменные двумя способами:

1. Экспертный отбор. Руководствуясь финансовой логикой, вручную выбираются ключевые индикаторы.
2. Автоматический отбор. С помощью моделей (RandomForest, LightGBM, SHAP) вычисляются важности признаков и остается топ k.

На финальном этапе в список признаков добавляют ранее рассчитанный бинарный флаг «Аномалия» (раздел 2.1), что помогает LLM учитывать смену рыночного режима и заметнее реагировать на структурные изменения в ряде.

Далее обновлённый prompt включает одновременно несколько модальностей:

«Ты эксперт по прогнозированию финансовых временных рядов. Даны последние w шагов по следующим признакам:

- Цена закрытия (руб.): 102.5, 103.0, 102.8, ... , 101.8.
- Объём торгов (тыс. акций): 10, 11, 15, ..., 25.
- Доходности (в %): +0.48, -0.19, ... +6.04
- ...
- Аномалия (0 - нет, 1 – да): 0, 0, 0, ..., 1

Спрогнозируй одно следующее значение цены и верни только число».

При формировании запроса необходимо контролировать общий объём токенов: если передаётся слишком длинный список признаков или слишком большое окно w , модель может не уложиться в лимит контекста и потерять часть данных. Для оптимизации рекомендуется:

- Уменьшение w в скользящем окне (раздел 1.6).
- Сокращение числа признаков до наиболее релевантных.
- Сжатие текстового контекста через предварительное агрегирование или summarization.

4.2. Учёт новостного фона

Новости отражают изменение настроений рынка. Чтобы LLM учитывала этот фактор, добавим новостной контекст двумя способами.

4.2.1. Интеграция оценок новостного фона

Сначала получаем числовую оценку влияния новостей (раздел 3), например в шкале $-100...+100$. Эти значения добавляются в prompt как дополнительный признак:

«Новостной фон (-100 – крайне отрицательный / 100 крайне положительный): 50, 70, ..., -10, -50».

4.2.2. Прямое включение заголовков

Если модель поддерживает большой контекст, можно сразу передавать отобранные последние заголовки:

«Ты эксперт по прогнозированию финансовых временных рядов. Даны последние w шагов:

- Цены: 102.5, ... , 101.8.
- Объёмы торгов: 10, ..., 25.
- Доходности (в %): +0.48, ... +6.04
- ...
- Аномалия (0 - нет, 1 – да): 0, ..., 1

Последние новостные заголовки (сначала самые свежие):

- Компания планирует выплатить дивиденды.
- В компании заявили о смене генерального директора.

Спрогнозируй следующее значение ряда цена и верни только одно число».

Преимущества и недостатки подходов:

- Оценки: компактно, но без деталей.
- Заголовки: детализация, но риск превысить лимит токенов.

Выбор зависит от максимальной длины контекста модели и важности текстовых нюансов.

4.3. Передача изображения ряда

Новые версии LLM поддерживают мультимодальный ввод — они могут одновременно обрабатывать текст и изображения. В рамках комбинированного подхода в качестве дополнительного «канала» можно передать график цены (или несколько графиков с техническими индикаторами) за последние w дней.

1. Подготовка изображения:

- a. Строится линейный график цены закрытия за окно w .
 - b. При необходимости накладываются скользящие средние, Bollinger Bands или другие индикаторы.
 - c. Экспортируется в формате PNG/BASE64 в зависимости от модели.
2. Выбор модели с поддержкой обработки изображения.
3. Формирование финального prompt'а. Вместе с текстовыми признаками (цены, объёмы, доходности, аномалии, новостной фон) к модели добавляется изображение графика:

«Ты эксперт по прогнозированию финансовых временных рядов. Мы имеем следующий характеристики:

- Цены: 102.5, ... , 101.8.
- Объёмы торгов: 10, ..., 25.
- Доходности (в %): +0.48, ... +6.04
- ...
- Аномалия (0 - нет, 1 – да): 0, ..., 1
- Новостной фон (-100 – крайне отрицательный / 100 крайне положительный): 20, ..., 90
- [Изображение графика цен и индикаторов]

Спрогнозируй следующее значение ряда цена и верни только одно число.»

Преимущества мультимодальности:

- Захват визуальных паттернов (тренды, уровни поддержки/сопротивления).
- Улучшенное понимание сложных отношений между числовыми и графическими данными.
- Снижение неопределённости за счёт объединения нескольких источников информации.

В результате комбинированный подход — текст + изображение — позволяет существенно расширить пространство входных данных.

5. ЭКСПЕРИМЕНТЫ

Для экспериментов использованы дневные данные о торгах на Московской бирже с 2016 по 2025 год для 10 компаний входящих в индекс МосБиржи: Аэрофлот (AFLT), Лукойл (LKOH), Московская Биржа (MOEX), Новатек (NVTK), Пик (PIKK), Сбербанк (SBER), Вконтакте (VKCO), ВТБ (VRBR), X5 Group (X5), Яндекс (YDEX).

5.1. Прогнозирование одномерных рядов

В ходе экспериментов с одномерными рядами цен закрытия было показано, что несмотря на разнообразие применённых моделей — от простейших «наивных» повторений и статистических сглаживаний до современных нейросетевых архитектур — все они оказываются подвержены высокой степени неопределённости в прогнозе на следующий день. Разделение данных на 95% обучения и 5% тестирования, а также использование библиотек ETNA и DARTS позволили унифицировать оценку моделей по метрикам MAPE, RMSE и Directional Accuracy (DA), а также по времени вычисления одного предсказания.

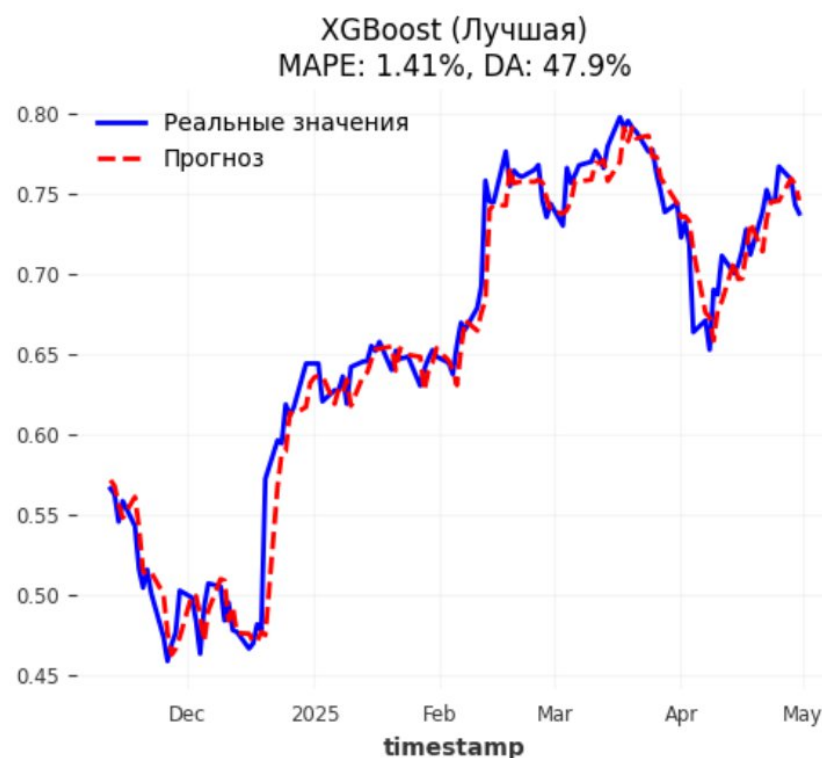


Рисунок 1. Визуализации прогнозов на одномерном ряде

Лучшие результаты в терминах среднеквадратичной и процентной ошибки продемонстрировали модели TCN, TBATS и Chronos, приблизившись к уровню «наивного» метода, однако даже они не смогли существенно обойти его по MAPE ($\approx 2\%$) (Приложение 1, Таблица 1). При этом ни одна из моделей не значительно не отличается от отметки в 50% по правильности направления движения цены, что свидетельствует о доминировании случайных флуктуаций на однодневных интервалах. Визуализация работы XGBoost на Рисунке 1 подтверждает, что алгоритмы в большинстве случаев лишь слегка корректируют экстраполяцию предыдущей динамики, а не прогнозируют значение с учётом глубоких закономерностей.

Низкая Directional Accuracy и близкие к «наивному» уровни MAPE подчёркивают, что одноступенчатое прогнозирование финансовых временных рядов остаётся крайне сложной задачей, в которой даже мощные ML- и DL-методы демонстрируют лишь незначительный прирост точности. Вместе с тем модели на базе больших языковых моделей продемонстрировали вполне сопоставимые значения MAPE — лучшим среди них оказался DeepSeek-Chat с MAPE $\approx 2.32\%$ (Приложение 1, Таблица 2).

5.2. Детектирование точечных аномалий

Было проведено сравнение нескольких алгоритмов детектирования аномалий на синтетических рядах, сгенерированных по распределению доходностей «чистого» отрезка данных «Аэрофлота». Наибольшие значения Precision и F1-score продемонстрировала библиотека Ruptures (0.94 и 0.86 соответственно) (Приложение 1, Таблица 3). Это говорит о том, что Ruptures одновременно точно выделяет реальных выбросы и допускает лишь небольшое число ложноотмеченных точек. Именно по этой причине она была выбрана в качестве основного инструмента для автоматической разметки аномалий на реальных временных рядах.

Модели на базе LLM показали высокий Recall (0.84), но существенно более низкий Precision (0.47), что свидетельствует о склонности обнаруживать выбросы там, где их на самом деле нет. Такая характеристика может быть полезна, если критично не пропускать ни одной потенциальной аномалии, но требует последующего строгого фильтра или ручной верификации, чтобы убрать ложные срабатывания.

Классические статистические и кластерные методы (EMA-Forecast, HBOS, LOF, K-Means, K-NN, Isolation Forest и др.) в целом показали сбалансированные, но низкие значения Precision и Recall (F1-score от 0.16 до 0.53), что указывает на их ограниченную способность работать с искусственно созданными резкими выбросами в финансовых рядах.

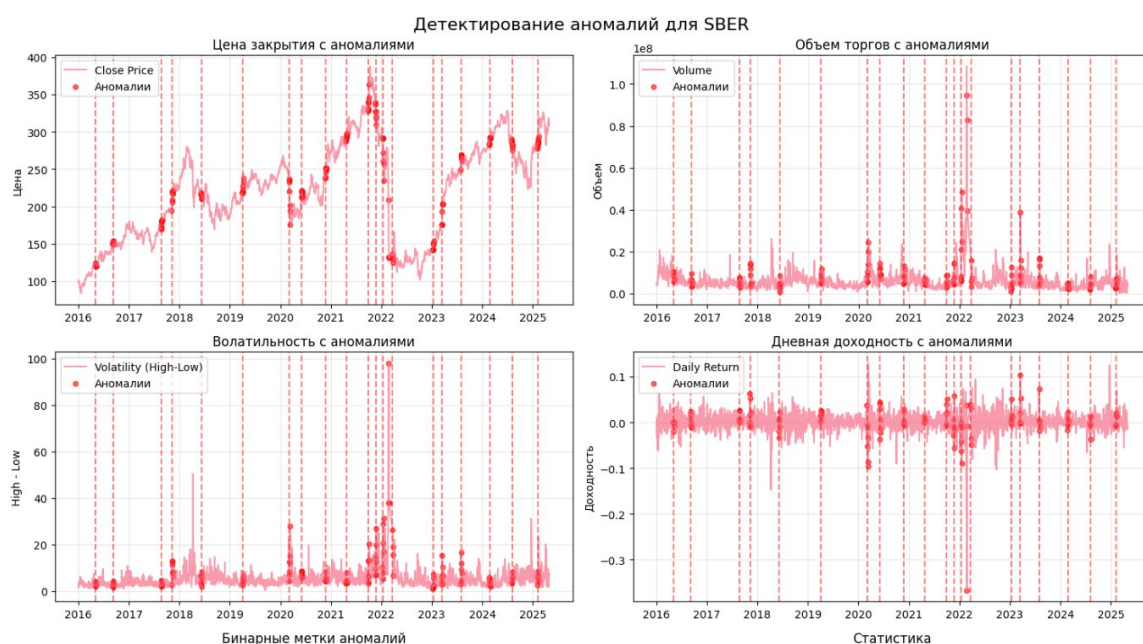


Рисунок 2. Обнаруженные аномалии библиотекой Ruptures на временных рядах

5.3. Оценка новостей

В качестве дополнительного источника сигналов был использован корпус новостных заголовков, собранный из открытых источников и привязанный к компаниям из исследуемой выборки. Для классификации текстов и оценки их влияния на цену акций применялась модель GigaChat в соответствии с алгоритмом, описанным в разделе 3: каждой новости

присваивалась метка «ПОЛОЖИТЕЛЬНОЕ» / «НЕЙТРАЛЬНОЕ» / «НЕГАТИВНОЕ», а затем эти метки преобразовывались в числовые оценки и агрегировались по дням.

Из-за большого объёма новостного потока был выполнен анализ только на уровне отдельных компаний — оценка для секторов и рынка в целом не проводилась, что, возможно, ограничило выразительность полученного новостного признака. Тем не менее, на Рисунке 3 видно, что пики «взвешенного» новостного фона часто совпадают с крупными ценовыми движениями: положительный выброс новостей предвещает краткосрочное повышение цен, а негативные всплески коррелируют со спадом.



Рисунок 3. Визуализация оценки новостного фона

Таким образом, новостной фон оказался информативным «каналом» для детектирования аномальных изменений в динамике цен, однако для более надёжных выводов требуется расширить охват оценок на уровень секторов и всего рынка, а также оптимизировать стратегию агрегирования оценок для снижения шума.

5.4. Генерация признаков

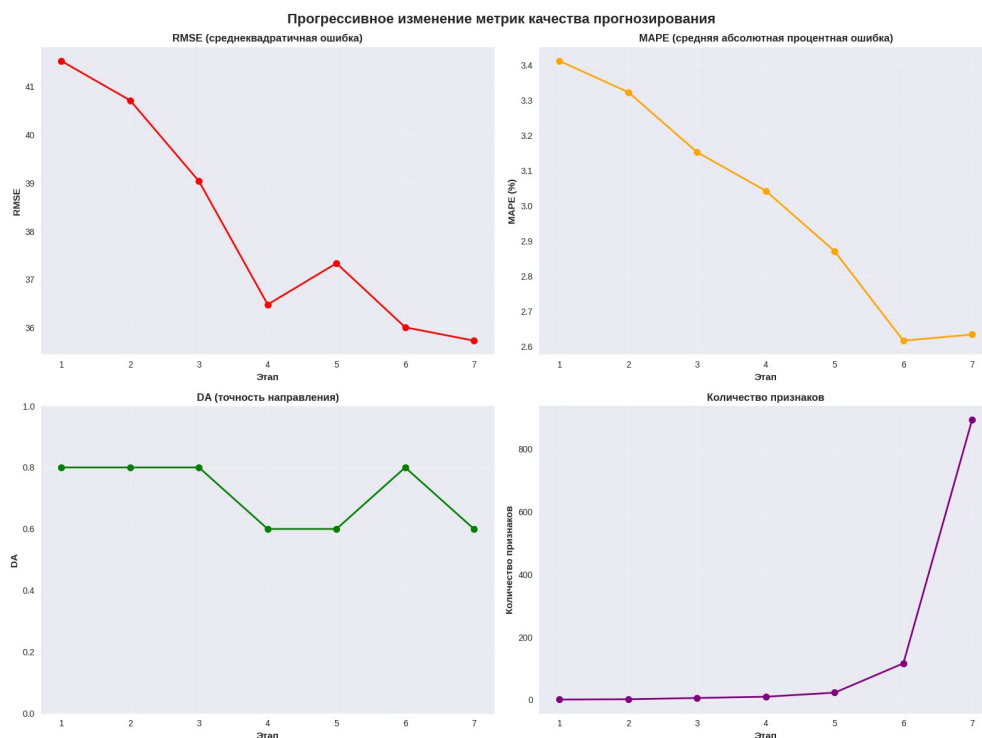


Рисунок 4. Прогрессивное изменение метрик качества прогнозирования

Включение дополнительных признаков (раздел 1.9) к историческим ценам приводит к заметному улучшению качества прогноза модели RandomForest на сокращённом ряде: простое добавление бинарного сигнала «Аномалия» снизило MAPE с 3,41% до 3,32%, а агрегированная оценка новостного фона — до 3,15% (Рисунок 4). Введение данных «свечей» и технических индикаторов продолжило эту тенденцию, опустив MAPE соответственно до 3,04% и 2,87%. При переходе к компонентам PCA (116 переменных) модель достигла MAPE 2,62%, а использование полного набора из почти 800 признаков TSFresh не повлияло на результат (MAPE = 2,63%), что указывает на избыточность части автоматически сгенерированных фичей.

Для LLM-методов проведена серия экспериментов с постепенно растущим числом признаков (Приложение 1, Таблица 5). Интересно, что оптимальной оказалась конфигурация из 21 параметра (включая графическое представление ряда) — при ней MAPE снизился на 5% до 2,32%. Однако при добавлении фичей до этого качество падало, что, вероятно, связано с

ограничениями не «думающих» моделей и затруднением обработки большого объёма входной информации.

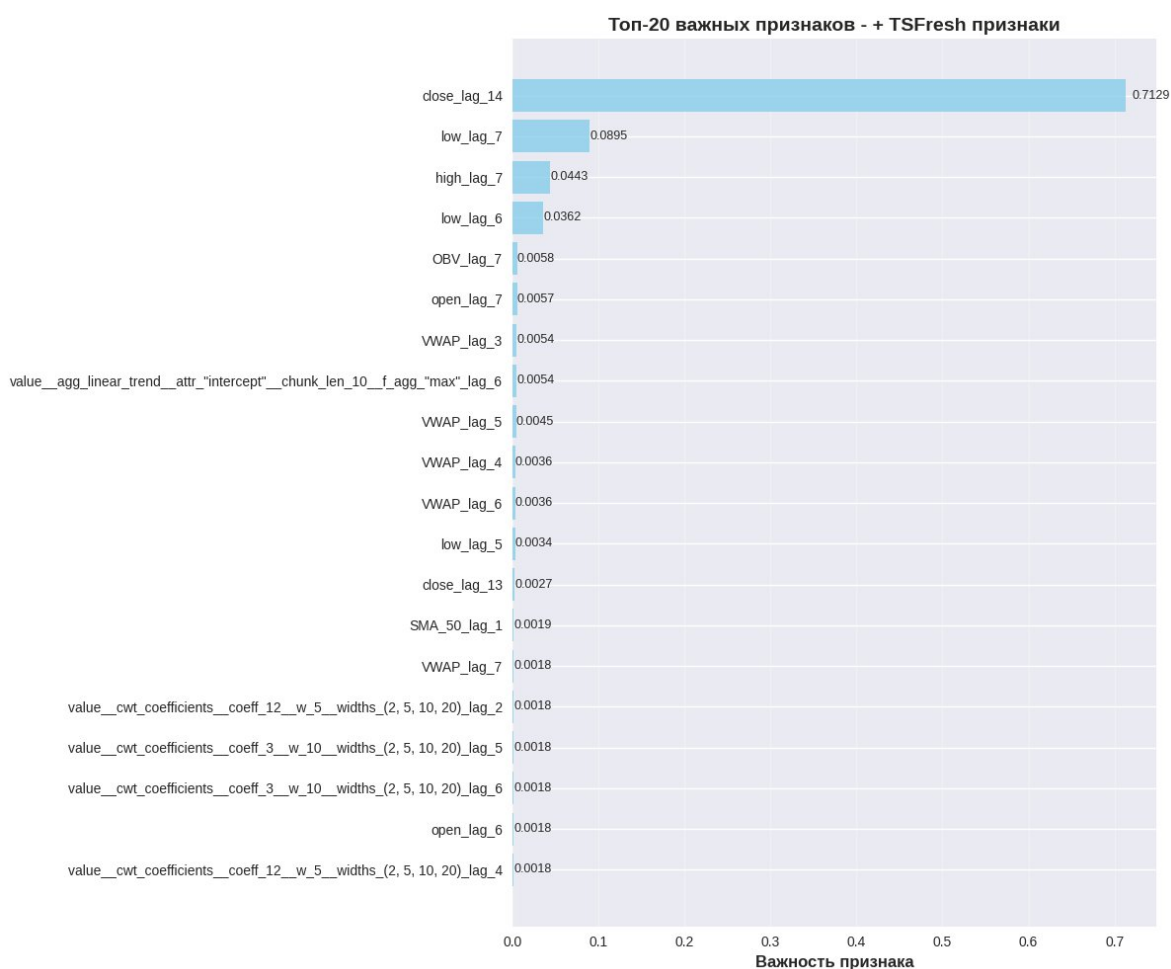


Рисунок 5. Важность признаков для RandomForest

Анализ важности признаков подтвердил, что ключевую роль играют лаги цены — а также минимальные и максимальные значения свечей за недавние периоды. Эти выводы помогают сфокусироваться на наиболее информативных сигналах и избежать «проклятия размерности», когда слишком большой набор признаков приводит к ухудшению обобщающей способности модели.

5.4. Прогнозирование многомерных рядов

В рамках оценки многомерных рядов протестировано двадцать алгоритмов на сокращённых выборках, включающих цену закрытия, сигналы

аномалий и ключевые технические индикаторы. Несмотря на увеличение затрат вычислительных ресурсов — особенно при использовании LLM с передачей графических признаков — ни одна модель не смогла существенно превзойти «наивное» повторение последнего значения по среднему абсолютному процентному отклонению (MAPE), оставив его на уровне 1,9 %. Вместе с тем включение дополнительных признаков позволило большинству алгоритмов заметно сократить отставание: разброс MAPE среди лидеров сузился до узкого диапазона 1,9–2,0% (Приложение 1, Таблица 6).

По среднеквадратичной ошибке (RMSE) наибольшей точности добился RandomForest (27,5), а максимальной точности по направлению движения цены ($DA \approx 60\%$) — CatBoost, что свидетельствует о его способности улавливать скрытые закономерности даже в условиях сильного шума.

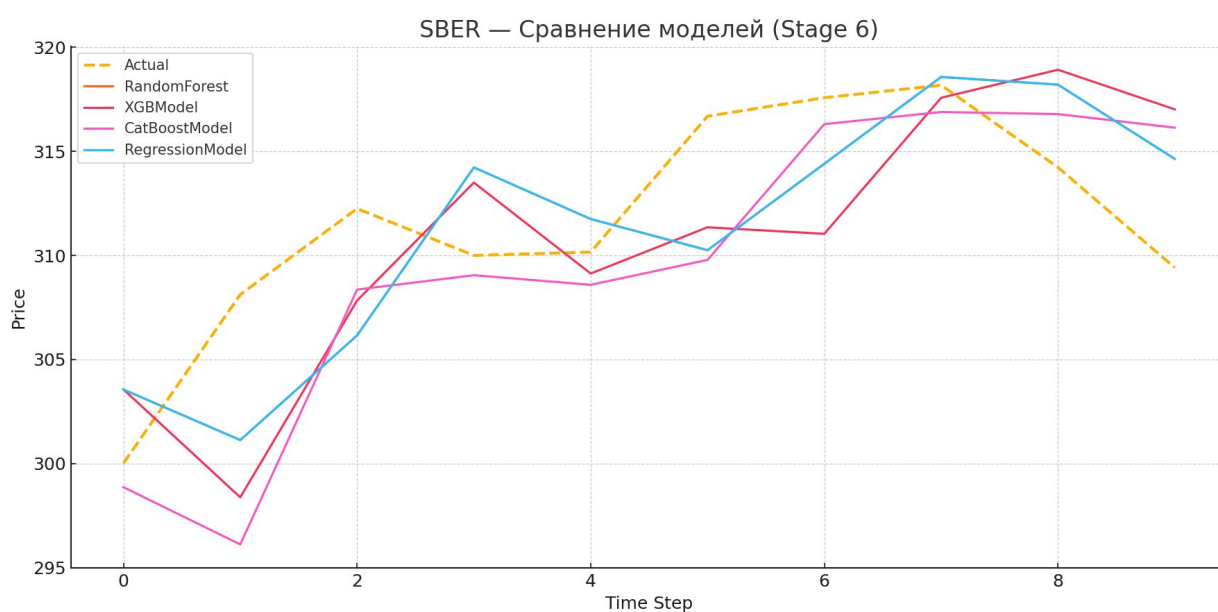


Рисунок 6. Визуализация прогнозов нескольких моделей на многомерных данных

На Рисунке 6 видно, что все модели продолжают следовать за недавним трендом, но их предсказания становятся более точными.

Модели на базе LLM с дополнительными признаками (без дообучения) достигли MAPE около 2%, сравнимого со специализированным Chronos, но не превзошли классические алгоритмы. При этом «обычные» LLM, использующие полный набор признаков, демонстрируют сопоставимые результаты с «рассуждающими» моделями, выигрывая в стоимости и

скорости, что делает их более пригодными для практического применения. В то же время «думающие» LLM остаются крайне затратными (примерно 6\$ за 100 запросов с графиками в Gemini 2.5 Pro) и медленными, что ограничивает их внедрение в продуктивных системах.

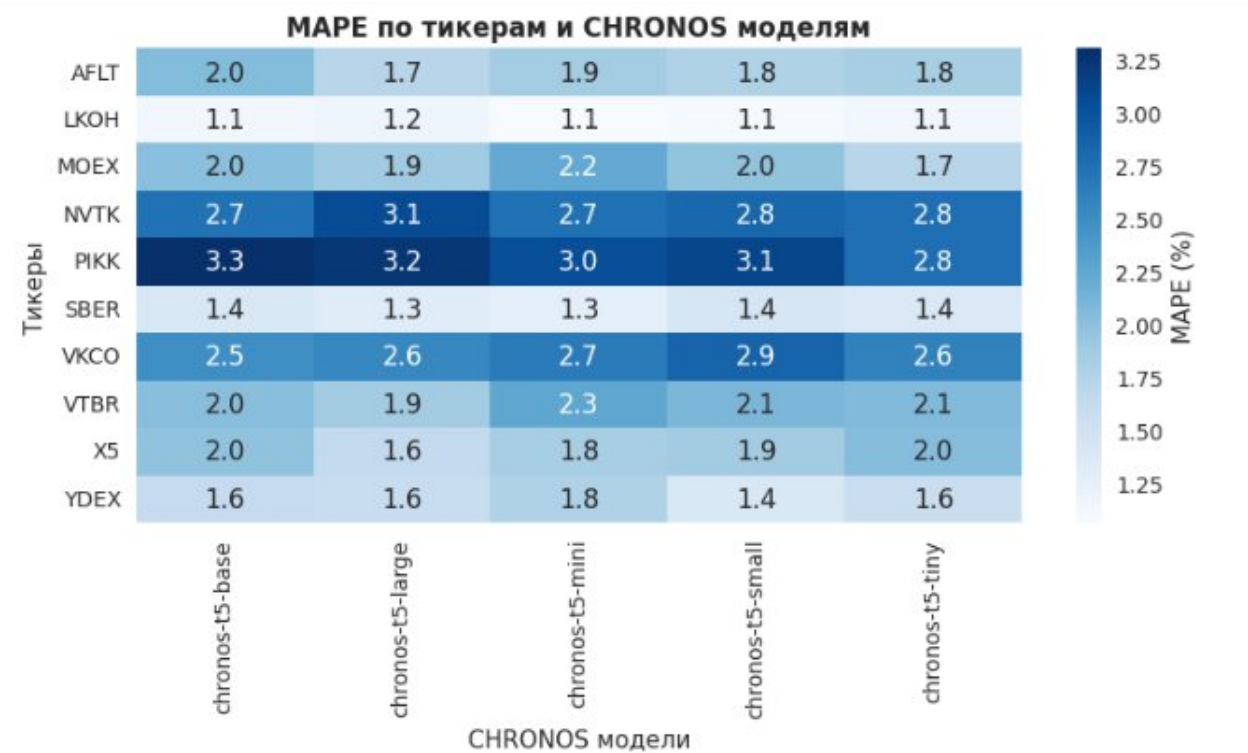


Рисунок 7. MAPE по тикерам для Chronos

В условиях резкой волатильности российского рынка однодневные колебания цен на десятки процентов под влиянием политических и макроэкономических событий остаются практически непредсказуемыми: ни одна модель не смогла значительно превысить уровень «наивного» прогноза. При этом для наиболее ликвидных и «устойчивых» бумаг (SBER, LKOH) ошибка прогнозов ниже, что подчёркивает важность отбора инструментов. Ограничение горизонта в один день не позволяет заранее учесть внезапные новости, поэтому в краткосрочном прогнозировании финансовых временных рядов уровень ошибок, как и раньше, будет оставаться высоким.

ЗАКЛЮЧЕНИЕ

В данной работе проведён всесторонний анализ методов прогнозирования финансовых временных рядов, детектирования аномалий и оценки новостного фона с использованием как классических статистических и машинных алгоритмов, так и современных подходов на базе больших языковых моделей (LLM). Были рассмотрены простые и скользящие методы (наивное повторение, SMA, EMA, ансамбль скользящих средних), классические статистические модели (ARIMA, SARIMA, TBATS), линейные архитектуры (DLinear, NLinear), рекуррентные сети (RNN, LSTM), AutoML-решения (Auto-ARIMA, FEDOT) и специализированные библиотеки (tsai, Chronos). Для детектирования аномалий исследованы статистические детекторы (IQR, Z-Score, distribution-based), прогнозные подходы (EMA+MAPE), многомерные и глубокие алгоритмы (Isolation Forest, кластеризация, автоэнкодеры, PyOD, Ruptures), а также LLM-методы. Оценка влияния новостного фона выполнена через LLM-промпты и последующее агрегирование сентимент-индексов.

На основе синергии этих подходов предложен комбинированный мультимодальный метод, в котором LLM получают расширенный контекст: числовые признаки ряда, метки точек перелома и аномалий, а также как числовые, так и текстовые описания новостного фона. Экспериментально показано, что за счёт расширенного контекста MAPE ниже, чем при использовании однострочного запроса, передающего только историю цен.

Для дальнейшего повышения точности прогнозирования необходимо расширить спектр входных данных, интегрируя макро- и микрофинансовые индикаторы, социально-медиа сигналы, фундаментальные показатели. Также следует исследовать горизонты прогнозирования, чтобы определить оптимальные временные рамки для различных стратегий.

Ссылка на код работы: <https://github.com/SodaWay1617/Financial-Time-Series-Forectsting>

СПИСОК ЛИТЕРАТУРЫ

- [1] Jing Su, Chufeng Jiang, Xin Jin, Yuxin Qiao, Tingsong Xiao, Hongda Ma, Rong Wei, Zhi Jing, Jiajun Xu, Junhong Lin (2024). Large Language Models for Forecasting and Anomaly Detection: A Systematic Literature Review. arXiv:2402.10350
- [2] Udit Gupta. (2023). GPT-InvestAR: Enhancing Stock Investment Strategies through Annual Report Analysis with Large Language Models. arXiv:2309.03079
- [3] Tian Guo, Emmanuel Hauptmann (2024). Fine-Tuning Large Language Models for Stock Return Prediction Using Newsflow. arXiv:2407.18103
- [4] Manqing Dong, Hao Huang, Longbing Cao (2024). “Can LLMs Serve As Time Series Anomaly Detectors?”. arXiv:2408.03475
- [5] Mingtian Tan, Mike A. Merrill, Vinayak Gupta, Tim Althoff, Thomas Hartvigsen (2024). “Are Language Models Actually Useful for Time Series Forecasting?”. arXiv:2406.16964
- [6] Ignacio Oguiza (2023). tsai - A state-of-the-art deep learning library for time series and sequential data. Github
- [7] Hassan Ismail Fawaz, Benjamin Lucas, e.t.c. (2020). InceptionTime: Finding AlexNet for Time Series Classification. arXiv:1909.04939
- [8] Elahe Rahimian, Soheil Zabihi, Seyed Farokh Atashzar, Amir Asif, Arash Mohammadi (2019). XceptionTime: A Novel Deep Architecture based on Depthwise Separable Convolutions for Hand Gesture Classification. arXiv:1911.03803
- [9] Fauvel K., Lin T., Masson V., Fromont É., Termier A. (2021). . XCM: An Explainable Convolutional Neural Network for Multivariate Time Series Classification. Mathematics 2021, 9, 3137.
- [10] Hanxiao Liu, Zihang Dai, David R. So, Quoc V. Le (2021). Pay Attention to MLPs. arXiv:2105.08050
- [11] Yuki Tatsunami, Masato Taki (2022). Sequencer: Deep LSTM for Image Classification. arXiv:2205.01972
- [12] Nikitin N., Vychuzhanin P., Sarafanov M., e.t.c. (2021). Automated evolutionary approach for the design of composite machine learning pipelines. Future Generation Computer Systems

[13] Ansari, Abdul Fatir, Stella, Lorenzo, Turkmen e.t.c. (2024). Chronos: Learning the Language of Time Series. Elsevier: Transactions on Machine Learning Research

[14] Nate Gruver, Marc Finzi, Shikai Qiu, Andrew Gordon Wilson (2023). Large Language Models Are Zero-Shot Time Series Forecasters. arXiv:2310.07820

[15] Mingyu Jin, Hua Tang, Chong Zhang, Qinkai Yu, Chengzhi Liu, Suiyuan Zhu, Yongfeng Zhang, Mengnan Du (2024). Time Series Forecasting with LLMs: Understanding and Enhancing Model Capabilities. arXiv:2402.10835

[16] Shengkun Wang, Taoran Ji, Linhan Wang, Yanshen Sun, Shang-Ching Liu, Amit Kumar, Chang-Tien Lu (2024). StockTime: A Time Series Specialized Large Language Model Architecture for Stock Price Prediction. arXiv:2409.08281v1

[17] C. Truong, L. Oudre, N. Vayatis. (2020). Selective review of offline change point detection methods. Elsevier: Signal Processing

[18] Xiong Xiao Xu, Haoran Wang, Yueqing Liang, Philip S. Yu, Yue Zhao, Kai Shu (2025). “Can Multimodal LLMs Perform Time Series Anomaly Detection?”. arXiv:2502.17812

[19] Zihao Zhou, Rose Yu (2024). “Can LLMs Understand Time Series Anomalies?”. arXiv:2410.05440

[20] Alejandro Lopez-Lira, Yuehua Tang. (2023). “Can ChatGPT Forecast Stock Price Movements?” Return Predictability and Large Language Models. arXiv:2304.07619

ПРИЛОЖЕНИЕ 1

Метод	RMSE	MAPE	DA	TIME (s)
Native Repeat	32.51	1.92%	43.11%	0.00016
SMA	72.80	4.75%	50.79%	0.003
EMA	73.04	5.07%	48.54%	0.00095
SMA Ensemble	64.75	4.48%	48.54%	0.001
ARIMA	110.0	5.89%	49.09%	0.20
AutoARIMA	44.49	2.62%	48.42%	20.00
TBATS	34.82	2,08%	43,56%	29.74
HoltWinters	44.62	2.61%	49.10%	0.34
Theta	44.68	2.62%	49.42%	0.89
DLineral	47.81	2.94%	51.33%	0.001
NLineral	65.84	4.33%	49.14%	0.001
RNN	33.85	2.21%	42.76%	92.01
LSTM	37.99	2.21%	46.63%	117.6
Lineral Regression	35.52	2.06%	47.61%	0.002
K-NN	35.85	2.09%	46.36%	0.021
CatBoost	37.18	2.29%	46.65%	1.716
XGBoost	33.90	2.09%	48.30%	0.058
RandomForest	33.67	2.08%	44.56%	1.555
Prophet	210.63	16.35%	45.92%	1.020
TCN	32.88	2.01%	42.29%	76.00
NBEATS	51.68	4.09%	47.70%	428.95
TFT	69.55	6.42%	51.94%	204.61
TSAI	32.08	2.19%	47.36%	26.01
FEDOT	64.32	6.08%	48.31%	719.48
CHRONOS	33.63	1.99%	46.67%	0.017
LLM	41.14	2.32%	45.32%	5.44

Таблица 1. Усреднённые результаты по 10 тикерам для одномерных временных рядов.

Модель	RMSE	MAPE	DA	TIME (s)
GPT-4o-mini	49.91	3.30%	41.85%	1.05
LLaMA-4	44.19	2.61%	46.04%	1.38
Claude-Haiku	54.92	3.15%	47.87%	2.44
DeepSeek-Chat	41.14	2.32%	45.32%	5.44
GigaChat	96.19	5.99%	47.12%	0.19

Таблица 2. Результаты предсказаний с помощью LLM моделей для одномерных рядов

Метод	Precision	Recall	F1
Ruptures	0.94	0.82	0.86
LLM	0.47	0.84	0.60
EMA-Forecast	0.38	0.96	0.53
HBOS	0.51	0.46	0.47
LocalOutlierFactor	0.43	0.51	0.46
LSCP	0.41	0.54	0.46
KMeans	0.40	0.54	0.45
KNN	0.40	0.52	0.45
IsolationForest	0.40	0.52	0.45
ABOD	0.38	0.49	0.43
CBLOF	0.38	0.50	0.43
PCA	0.28	0.39	0.32
IQR	0.41	0.24	0.28
Z-Score	0.09	0.67	0.16

Таблица 3. Эффективность детектирования точечных аномалий

Этап	Описание	RMSE	MAPE	Признаки
1	Цена закрытия	41.52	3.41%	1
2	Аномалии	40.71	3.32%	2
3	Новости	39.03	3.15%	6
4	Свечи	36.48	3.04%	10
5	Технические индикаторы	37.33	2.87%	23
6	РСА компоненты	36.01	2.62%	116
7	Признаки TSFresh	35.73	2.63%	893

Таблица 4. Эффективность прогнозов RandomForest при разном количестве признаков

Этап	Описание	RMSE	MAPE	Признаки
1	Цена закрытия	34.44	2.46%	1
2	Аномалии	36.38	2.90%	2
3	Новости (оценка)	39.06	2.91%	3
4	Новости (текст)	38.47	2.90%	3
5	Свечи	44.09	3.10%	7
6	Технические индикаторы	40.39	3.06%	13
7	Значимые признаки TSFresh	43.81	3.42%	20
8	График	35.78	2.32%	21

Таблица 5. Эффективность прогнозов ChatGPT-4o-mini при разном количестве признаков

Метод	RMSE	MAPE	DA
Native Repeat	29.01	1.88%	45.5%
CatBoost	30.11	1.95%	60.0%
RandomForest	27.52	1.91%	48.8%
XGBoost	29.19	1.93%	53.3%
LSTM	32.37	2.02%	51.1%
TCN	32.42	2.19%	48.8%
FEDOT	83.62	5.44%	40.7%
TSAI	29.29	2.53%	48.8%
CHRONOS	30.86	1.98%	52.2%
LLM	31.92	2.01%	41.1%

Таблица 6. Усреднённые результаты по 10 тикерам для многомерных временных рядов.

Модель	RMSE	MAPE	DA
GPT-4o-mini	28.69	2.06%	44.6%
GPT-o4-mini	31.14	2.11%	45.5%
Gemini-2.5-Pro	32.63	2.12%	44.4%
MiniMax-M1	34.46	2.34%	48.1%
LLaMA-4	33.23	2.13%	48.8%
DeepSeek-Chat	31.92	2.01%	41.1%
DeepSeek-R1	32.15	2.20%	48.6%
Claude-Haiku	32.09	2.18%	42.2%
GigaChat	52.29	3.52%	44.4%

Таблица 7. Результаты предсказаний с помощью LLM моделей при комбинированном подходе