Проценты

План исследования и реализации системы внутридневных торговых решений на основе микроструктурных данных и Transformer-модели

1) Сбор и формализация датасета

1.1. **Источники данных.** Сформировать единый набор временных рядов на основе данных биржи (например, Binance) с фиксированным временным шагом (\Delta t) (рекомендуемо: 1–5 минут) для выбранного набора инструментов (spot/perp). 1.2. **Обязательные компоненты датасета (минимальный состав).** Для каждого интервала времени (t) сохранить:

- **OHLCV**: open, high, low, close, volume;
- **временные признаки**: час суток, день недели, время до закрытия/открытия сессии (если применимо), циклическое кодирование времени (sin/cos);
- **производные от цены**: лог-доходности на нескольких лагах (например, 1, 3, 5, 15, 60 минут), кумулятивная доходность на окнах;
- **оценки волатильности**: rolling-std/ATR/realized volatility на окнах (например, 15, 60, 240 минут);
- **объёмные статистики**: rolling volume, z-score объёма, доля “аномального” объёма относительно исторического окна.

1.3. **Дополнительные компоненты для деривативов (при доступности).** Добавить:

- **funding rate** и **время до следующего funding-события**;
- **open interest** (OI) и (\Delta OI);
- **basis/premium** (разница perp vs spot) и её динамика. Эти переменные рассматриваются как факторы режима (risk appetite, перекос позиционирования), релевантные на горизонтах десятков минут–часов.

1.4. **Синхронизация и контроль качества.** Выполнить:

- выравнивание рядов по времени (resampling на (\Delta t));

- обработку пропусков (явная маркировка/импутация с сохранением индикаторов пропусков);
- контроль корректности цен/объёмов (фильтрация выбросов, проверка монотонности timestamp);
- документирование частоты обновления и доли пропусков как характеристик датасета.

2) Постановка задачи и формирование целевых переменных

2.1. **Определение горизонта предсказания.** Зафиксировать прогнозный горизонт (H) в диапазоне внутридневных решений (например, 30–240 минут), совместимый с выбранным шагом (\Delta t). 2.2. **Базовый таргет.** Рассмотреть регрессию ожидаемой доходности (\Delta p_{t \to t+H}) или классификацию направления (up/down/flat). 2.3. **Торгуемый таргет (рекомендуемо как основной).** Определить метку “торгового действия” с учётом издержек:

- оценить транзакционные издержки (c) (комиссии + эффективная часть спрэда + прокси проскальзывания);
- сформировать классы:
  - long, если (E[\Delta p_{t \to t+H}] > c),
  - short, если (E[\Delta p_{t \to t+H}] < -c),
  - flat иначе. Такой таргет обеспечивает согласование ML-оптимизации с практической торгуемостью сигнала.

3) Протокол валидации и предотвращение утечек информации

3.1. **Временная схема валидации.** Использовать исключительно **walk-forward / rolling window validation**: обучение на прошлом интервале, тестирование на последующем, без случайного перемешивания. 3.2. **Исключение leakage.** Гарантировать, что все признаки в момент (t) вычислены только из данных (\le t), а нормализация/стандартизация проводится:

- либо по обучающему окну,

- либо по “expanding window” без использования будущих данных. 3.3. **Стабильность и переносимость.** Оценивать метрики по периодам и (при наличии) по нескольким инструментам, фиксируя дисперсию качества во времени.

4) Бейзлайны и базовый уровень качества

4.1. **Табличные бейзлайны.** Обучить модели на агрегированных признаках:

- Logistic Regression / Linear models (как нижняя граница),
- LightGBM/XGBoost (как сильный табличный baseline). 4.2. **Метрики.** Помимо стандартных ML-метрик (balanced accuracy, macro-F1, precision/recall по классам) обязательно считать:
- **coverage** (доля времени, когда модель предлагает long/short),
- **калибровку вероятностей** (например, ECE или reliability-анализ),
- **PNL-proxy** на исторических данных с учётом costs (см. п. 6). Цель этапа — получить воспроизводимую опорную точку, относительно которой будет оцениваться вклад Transformer-архитектуры.

5) Отбор и структурирование признаков

5.1. **Контролируемая автоматизация.** Использовать автоматизированный отбор признаков как инструмент снижения размерности и выявления информативных групп, но избегать неконтролируемого “генерирования тысяч фичей”. 5.2. **Методы.**

- feature importance (gain/split) в градиентном бустинге,
- permutation importance на валидационных окнах,
- **групповые абляции**: исключение целых семейств признаков (volume-группа, volatility-группа, derivatives-группа и т.д.) с повторной оценкой качества. Результат этапа — компактный и интерпретируемый набор признаков и понимание их вклада.

6) Преобразование предсказаний в торговые решения и оценка “вне рынка”

6.1. **Сигнальный слой.** Интерпретировать выход модели как вероятностную оценку и вводить детерминированную политику принятия решений (пороговые правила, фильтры ликвидности/волатильности, ограничения частоты сделок). 6.2. **Бэктест с фрикциями.** Выполнить оценку стратегии в офлайн-режиме с реалистичными допущениями:

- комиссии, спред, проскальзывание (как минимум константное, далее сценарное),
- задержка исполнения (latency penalty) как смещение входа/выхода на (\Delta t) или на фиксированное время,
- ограничения по риску (лимиты дневного убытка, max position, cooldown). 6.3. **Отчётные метрики стратегии.** PnL, max drawdown, turnover, число сделок, средний трейд, распределение сделок по времени суток, устойчивость по walk-forward периодам. Цель — зафиксировать разрыв “ML-качество ↔ торговый результат” и формализовать условия, при которых сигнал торгуем.

7) Transformer-модель для интеграции микроструктурной информации во внутридневные решения

7.1. **Формат входа.** Представить рынок как последовательность признаков: [X_t = [x_{t-L+1}, \dots, x_t], \quad x_t \in \mathbb{R}^d] где (L) — длина окна наблюдения (например, 240–720 шагов для 1–3 суток по минутам или несколько часов по 1–5 минутам), (d) — число признаков. 7.2. **Архитектура.** Использовать encoder-only Transformer (или Patch-based Transformer, если окно длинное) с позиционным кодированием и механизмом внимания, ориентированным на выделение значимых периодов и режимов. 7.3. **Целевая функция.**

- для классификации: cross-entropy с учётом дисбаланса классов (weights/focal loss при необходимости);
- для регрессии: robust loss (Huber/quantile) и последующий перевод в торговые классы через порог costs. 7.4. **Обучение и регуляризация.** Early stopping по walk-forward валидации, dropout, weight decay, контроль переобучения через

сравнение устойчивости по периодам. 7.5. **Сравнение с бейзлайнами.** Проводить прямое сравнение с LightGBM/XGBoost по одинаковому протоколу разметки и walk-forward, включая торговые метрики (п. 6), а также анализ абляций (удаление отдельных групп входных признаков и/или компонентов архитектуры).

8) Итоговая интерпретация и выводы

8.1. **Анализ вклада Transformer-архитектуры.** Показать, улучшает ли последовательная модель качество и/или устойчивость относительно табличных бейзлайнов, и за счёт каких признаков/периодов. 8.2. **Ограничения и угрозы валидности.** Явно описать влияние нестационарности, издержек, задержек и ограничений исполнения, а также сценарии, в которых сигнал деградирует. 8.3. **Практическая применимость.** Сформулировать условия, при которых модель может использоваться как компонент сигнального слоя внутридневной торговой системы (в т.ч. требования к мониторингу и периодическому переобучению).