

Во время посещения сайта вы соглашаетесь с использованием файлов [cookie](#)

Хорошо



Михаил Шардин ★

личный блог



11 марта 2025, 04:45

+ Подписаться

## Тестирование торговой стратегии с использованием нового индикатора Джона Ф. Элерса на Python для дневных данных Московской биржи

Торговля акциями требует гибкости, особенно когда речь идет о тестировании стратегий технического анализа на прошлых данных. Я выбрал Python и библиотеки `backtesting.py` и `aiomex`, потому что они позволяют анализировать рынок без сложных платформ и ограничений. Python дает свободу автоматизации, `backtesting.py` обеспечивает удобный и быстрый механизм тестирования стратегий, а `aiomex` позволяет скачивать данные напрямую с Московской биржи без привязки к брокеру.

Важно, что `backtesting.py` получил обновление после четырех лет без обновлений, что делает его актуальным инструментом. И в отличие от `MetaTrader`, `StockSharp`, `TSLab` и `Quik`, которые работают с Московской биржей, но требуют Windows, если брокер имеет API, то можно запускать скрипт на любом сервере, включая облачные решения и Raspberry Pi.

В этой статье я протестирую самую свежую стратегию теханализа Джона Ф. Элерса (John Ehlers), направленную на устранение запаздывания скользящей средней. Разберемся, как её адаптировать к акциям Московской биржи и протестировать с помощью Python.

### Новый индикатор Джона Элерса «устранение запаздывания скользящей средней»

Одна из главных проблем стандартных скользящих средних (SMA) — это запаздывание. Поскольку SMA рассчитывается как среднее за определенный период, её значение всегда отстает от реальной цены, что мешает своевременному входу в сделку.

Джон Элерс предложил решение — прогнозируемая скользящая средняя (PMA, Projected Moving Average). В отличие от обычных скользящих, PMA использует линейную регрессию для прогнозирования будущих значений, уменьшая лаг.

Откройте счёт в ВТБ Мои Инвестиции

Введите текст комментария



SBER

Формула

PMA: `PMA = SMA + Slope * Length / 2,`

где Slope — наклон линии регрессии.

Дополнительно Элерс предложил прогнозировать саму PMA: `PredictPMA = PMA + 0.5 (Slope - Slope[2]) Length`

и наклон: `PredictSlope = 1.5 Slope - 0.5 Slope[4].`

Пересечения PredictPMA и PMA помогают находить точки входа и выхода, делая стратегию более адаптивной к изменениям рынка.



S&P 500 E-Mini Futures

Откройте счёт в ВТБ Мои Инвестиции

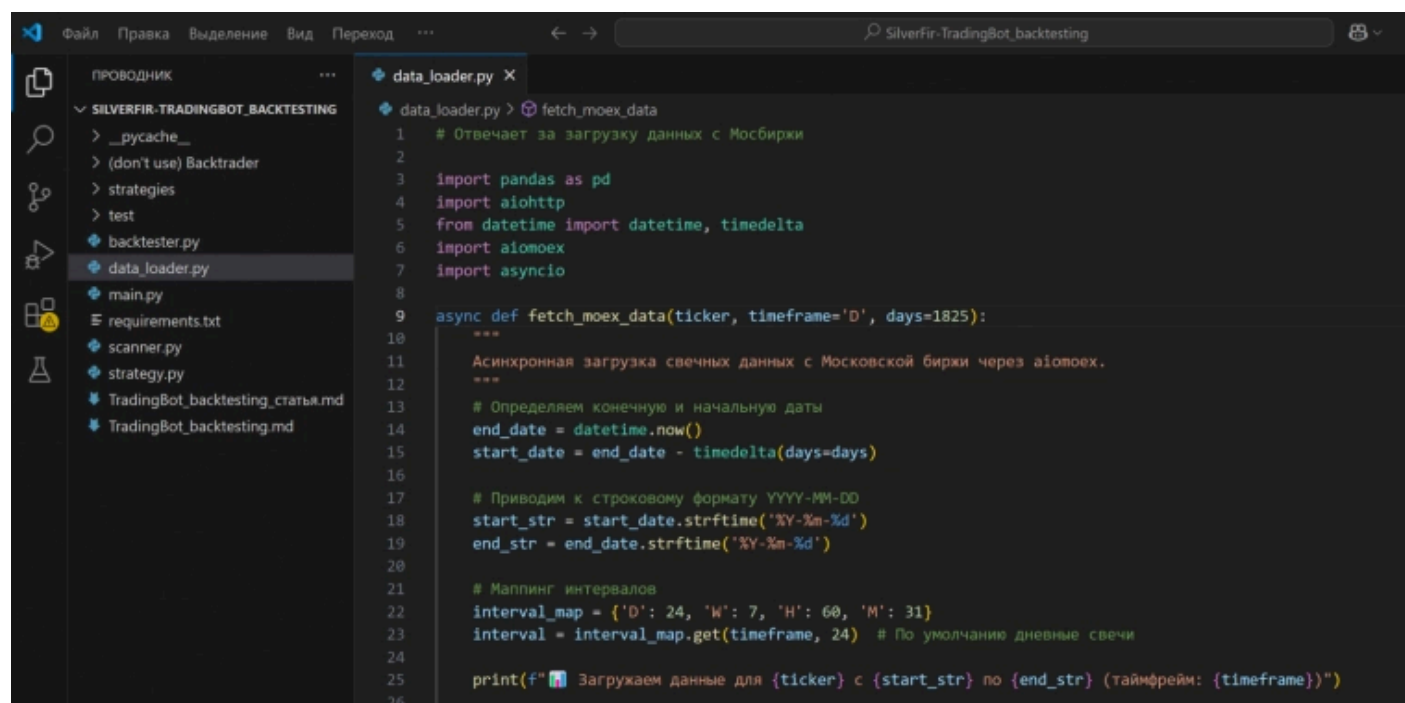
## Риск-менеджмент:

- Первоначальный стоп-лосс устанавливается на 10% ниже цены входа.
- Выход из позиции осуществляется по скользящему стопу на основе ATR.

## Реализация бэктестинга через backtesting.py. Определение топ-20 акций по объему

Весь код [представлен на GitHub](#).

### Модуль data\_loader.py



```
1 # Отвечает за загрузку данных с Мосбиржи
2
3 import pandas as pd
4 import aiohttp
5 from datetime import datetime, timedelta
6 import aiomoe
7 import asyncio
8
9 async def fetch_moex_data(ticker, timeframe='D', days=1825):
10     """
11     Асинхронная загрузка свечных данных с Московской биржи через aiomoe.
12     """
13     # Определяем конечную и начальную даты
14     end_date = datetime.now()
15     start_date = end_date - timedelta(days=days)
16
17     # Приводим к строковому формату YYYY-MM-DD
18     start_str = start_date.strftime('%Y-%m-%d')
19     end_str = end_date.strftime('%Y-%m-%d')
20
21     # Мappings интервалов
22     interval_map = {'D': 24, 'W': 7, 'H': 60, 'M': 31}
23     interval = interval_map.get(timeframe, 24) # По умолчанию дневные свечи
24
25     print(f"Загружаем данные для {ticker} с {start_str} по {end_str} (таймфрейм: {timeframe})")
26
```

Для тестирования стратегии необходимо загружать актуальные данные о торгах.

В этом помогает библиотека aiomoe, которая предоставляет API-доступ к Московской бирже. В модуле data\_loader.py реализована функция fetch\_moex\_data, позволяющая асинхронно получать исторические данные по свечам.

Функция запрашивает данные за последние 1825 дней (примерно 5 лет) и конвертирует их в формат Pandas DataFrame. Особенность реализации — использование асинхронного HTTP-клиента aiohttp, что ускоряет загрузку. Данные приводятся к удобному формату: преобразуются даты, устанавливается индекс, а названия колонок заменяются на стандартные для анализа.

Откройте счёт в ВТБ Мои Инвестиции

```

1  import asyncio
2  import pandas as pd
3  from datetime import datetime, timedelta
4  import aiohttp
5  import aiomoe
6  from data_loader import fetch_moe_data
7
8  async def get_tqbr_securities():
9      """Получает список всех бумаг из TQBR."""
10     url = "https://iss.moex.com/iss/engines/stock/markets/shares/boards/TQBR/securities.json"
11     query = {"iss.meta": "off", "iss.only": "marketdata", "marketdata.columns": "SECID"}
12
13     try:
14         timeout = aiohttp.ClientTimeout(total=20)
15         async with aiohttp.ClientSession(timeout=timeout) as session:
16             iss_client = aiomoe.ISSClient(session, url, query)
17             data = await iss_client.get()
18
19             if not data or "marketdata" not in data:
20                 return []
21
22             df = pd.DataFrame(data["marketdata"])
23             return df["SECID"].tolist() if "SECID" in df.columns else []
24
25
26

```

После загрузки данных важно отобрать ликвидные бумаги. Для этого в модуле `scanner.py` реализована функция `get_top_20_stocks`, которая анализирует объем торгов за последние 14 дней и выделяет 20 наиболее ликвидных акций.

Алгоритм работы следующий:

1. Получение списка всех торгуемых акций на основном рынке (TQBR) через API Московской биржи.
2. Асинхронная загрузка дневных данных по каждому инструменту с помощью `fetch_moe_data`.
3. Расчет суммарного объема торгов за 14 дней.
4. Формирование списка из 20 акций с наибольшим объемом.

Таким образом, отбираются бумаги с высоким оборотом, что повышает надежность тестирования стратегии и снижает риск торговли неликвидными активами.

## Реализация бэктестинга через `backtesting.py`. Тестирование стратегии на исторических данных

### Зачем всё разделил на модули?

Разделение кода на модули делает его более удобным для сопровождения, масштабирования и переиспользования. В нашем случае:

- `data_loader.py` отвечает за загрузку данных с Московской биржи.
- `scanner.py` фильтрует ликвидные бумаги.
- `backtester.py` выполняет бэктестинг.

- `strategy.py` содержит описание стратегии.

Откройте счёт в ВТБ Мои Инвестиции

**Пример кода для бэктестинга с использованием backtesting.py**

```

import asyncio
import pandas as pd
from backtesting import Backtest
from data_loader import fetch_moex_data
from strategy import LongOnlyPMAMultiTimeframeATRTrailingStop

async def run_backtest(ticker):
    print(f"\n{'='*50}")
    print(f"🚀 Запуск бэктеста для {ticker}")
    print(f"{'='*50}\n")

    # Получаем данные
    df, start_str, end_str = await fetch_moex_data(ticker) # Получаем start_str

    # Запускаем бэктест
    print("⌚ Запуск бэктеста...")
    strategy_class = LongOnlyPMAMultiTimeframeATRTrailingStop # Класс стратегии
    strategy_name = f"{ticker}_{start_str}_{end_str}_LongOnlyPMAMultiTimeframeAT
    DynamicStrategyClass = type(strategy_name, (strategy_class,), {}) # Создаем
    bt = Backtest(df, DynamicStrategyClass, cash=100_000, commission=0.002) # Ис
    stats = bt.run()

    # Вывод результатов
    print("\n📊 Результаты бэктеста:")
    print(f"⚙️ Стратегия: {strategy_name}") # Выводим динамическое имя стратегии
    print(f"📅 Период тестирования: с {stats['Start']} по {stats['End']}")
    print(f"💰 Начальный капитал: 100,000 руб.")
    print(f"💵 Конечный капитал: {stats['Equity Final ($)']:.2f} руб.")
    print(f"📈 Общая доходность: {stats['Return (%)']:.2f}%")
    print(f"📊 Годовая доходность: {stats['Return (Ann.) (%)']:.2f}%")
    print(f"📉 Коэффициент Шарпа: {stats['Sharpe Ratio']:.2f}")
    print(f"📉 Максимальная просадка: {stats['Max. Drawdown (%)']:.2f}%")
    print(f"🔄 Количество сделок: {stats['# Trades']}")
    print(f"✅ Процент выигрышных сделок: {stats['Win Rate (%)']:.2f}%")

```

Откройте счёт в ВТБ Мои Инвестиции

```
try:
    bt.plot()
    print("✅ График успешно построен!")
except ValueError as e:
    print(f"❌ Ошибка при построении графика: {e}")
print(f"\n{'='*50}")
print(f"🎯 Бэктест для {ticker} завершен")
print(f"{'='*50}\n")
return stats
```

### Основные метрики оценки

Для оценки стратегии используются ключевые метрики:

- Общая доходность (%) — показывает, сколько стратегия заработала за весь период тестирования.
- Годовая доходность (%) — усреднённый годовой прирост капитала.
- Коэффициент Шарпа — измеряет соотношение доходности к риску.
- Максимальная просадка (%) — определяет максимальную потерю капитала.
- Процент выигрышных сделок.
- Средняя продолжительность сделки.

Эти показатели позволяют оценить эффективность стратегии и принять решение о её использовании в реальной торговле.

### Результаты тестирования на акциях Московской биржи

Как положительные, так и отрицательные.

Примеры в [html файлах на GitHub'e](#).

**СПБ Биржа (тикер SPBE):**

Откройте счёт в ВТБ Мои Инвестиции



Результаты бэктеста:

Стратегия: SPBE\_2020-03-03\_2025-03-02\_LongOnlyPMAMultiTimeframeATRTrailingStop

Период тестирования: с 2021-11-19 00:00:00 по 2025-03-01 00:00:00

Начальный капитал: 100,000 руб.

Конечный капитал: 349138.97 руб.

Общая доходность: 249.14%

Годовая доходность: 47.34%

Коэффициент Шарпа: 0.80

Максимальная просадка: -27.41%

Количество сделок: 11

Процент выигрышных сделок: 54.55%

Лучшая сделка: +36.56%

Худшая сделка: -8.31%

Средняя продолжительность сделки: 25 days 00:00:00

**Новатэк ао (тикер NVTK):**

Откройте счёт в ВТБ Мои Инвестиции





Результаты бэктеста:

Стратегия: NVTK\_2020-03-03\_2025-03-02\_LongOnlyPMAMultiTimeframeATRTrailingStop

Период тестирования: с 2020-03-03 00:00:00 по 2025-03-01 00:00:00

Начальный капитал: 100,000 руб.

Конечный капитал: 94443.56 руб.

Общая доходность: -5.56%

Годовая доходность: -1.15%

Коэффициент Шарпа: -0.07

Максимальная просадка: -38.70%

Количество сделок: 22

Процент выигрышных сделок: 27.27%

Лучшая сделка: +18.55%

Худшая сделка: -10.78%

Средняя продолжительность сделки: 23 days 00:00:00

## Проблема учета смены лидеров по объему

Мой код отдельно тестирует каждую акцию из топ-20 на момент отбора (на сегодня).

Однако он не учитывает смену лидеров по объему и не позволяет работать с единой

Откройте счёт в ВТБ Мои Инвестиции



## Следующие шаги

Фиксированный список топ-акций по объему устаревает. Использование динамического реестра позволит оперативно учитывать смену лидеров, корректируя состав активных позиций в стратегии.

Библиотека `ta-lib` мне не очень понравилась из-за сложностей с установкой — проще переписать индикатор вручную в будущем.

Получится ли реализовать это через `backtesting.py`? Скорее всего вряд ли.

Скорее всего придётся вернуться к `Backtrader`.

## Заключение

Тестирование стратегии на акциях Мосбиржи показало её стабильную эффективность при использовании индикатора РМА на дневных свечах.

Python доказал свою ценность в алгоритмической торговле, обеспечивая гибкость и автоматизацию. Однако `backtesting.py` имеет ограничения.

Автор: Михаил Шардин

 [Моя онлайн-визитка](#)

 [Telegram «Умный Дом Инвестора»](#)

11 марта 2025 г.

Python

торговые роботы

4.3K 

 16

 29

 24



**Михаил Шардин**

 Пермь

 257  2 502

 с 23 января 2019

Откройте счёт в ВТБ Мои Инвестиции

11 марта 2025, 06:48

Не работают  
backtesting.py => backtesting.ru?

— Показать 1 ответ



Михаил Шардин ★

11 марта 2025, 08:35



Сергей Сергаев, например в каком направлении? Одним словом



Noname

11 марта 2025, 08:57



Решение запаздывания скользящей средней решается уменьшением таймфрейма, у Элиота в то время этой возможности не было, у нас есть.

15та на часовом таймфрейме равна 900та на минутном таймфрейме. Проблема решена)

— Показать 3 ответа



Михаил Шардин ★

11 марта 2025, 09:11



Сергей Сергаев, вы имеете ввиду что если стратегия показывала хорошие результаты на исторических данных, это не означает, что она будет работать в будущем.

Но я-то думал что единственное слово будет что-то вроде «фундаментал».



Ещё 11 комментариев

Напишите комментарий...

Откройте счёт в ВТБ Мои Инвестиции

## Читайте на SMART-LAB

ЗА ПОЛУГОДИЕ, ЗАКОНЧИВШЕЕСЯ 30 ИЮНЯ 2025 ГОДА (НЕАУДИРОВАННЫЙ)

(в миллионах российских рублей, за исключением прибыли на акции)

	Примечание	За полугодие, закончившееся	
		30 июня 2025 года	30 июня 2024 года
ВЫРУЧКА	16	171 233	201 884
СЕБЕСТОИМОСТЬ РЕАЛИЗАЦИИ	17	(144 244)	(160 257)
<b>ВАЛОВАЯ ПРИБЫЛЬ</b>		<b>26 989</b>	<b>41 627</b>
Коммерческие, общезаявительные и административные расходы	18	(15 722)	(14 177)
Расходы на операционные доходы	19	1 363	585
Расходы на операционные расходы		(137)	(115)
Изменение справедливой стоимости инвестиционной недвижимости		128	—
<b>ОПЕРАЦИОННЫЙ УБЫТОК (ПРИБЫЛЬ)</b>		<b>(7 599)</b>	<b>9 720</b>
Финансовые доходы	20	585	443
Финансовые расходы	20	(26 632)	(16 537)
<b>УБЫТОК ДО НАЛОГА НА ПРИБЫЛЬ</b>		<b>(33 646)</b>	<b>(12 364)</b>
Доходы по налогу на прибыль		8 527	2 636
Изменения в налоговом учете и прочих налоговых вычетах на прибыль		174 148	174 319

### М.Видео - есть ли в мультивселенной компания, где миноритариям что-то достанется?

М.Видео меняет формат допэмиссии. Ранее акционеры одобрили допэмиссию до 500 млн новых акций по закрытой подписке — о...



Mozgovik

03:51

### ДАЙДЖЕСТ ПО РЕЙТИНГОВЫМ ДЕЙСТВИЯМ В ВЫСОКОДОХОДНОМ СЕГМЕНТЕ, ПОРТФЕЛЕ PROBONDS ВДО И РОЗНИЧНЫХ ИНВЕСТИЦИОННЫХ ОБЛИГАЦИЙ ЗА...

● ПАО «ЕВРОТРАНС» АКРА подтвердило кредитный рейтинг на уровне A-(RU) ПАО «ЕвроТранс» — один из крупнейших независимых топливных операторов на рынке Московского региона....



Иволга Капитал

13.09.2025

### Банк России снизил ключевую ставку до 17% — что делать? Мнение аналитиков

Сегодня на заседании Банк России снизил ключевую ставку на 1 п.п., до 17%. Теоретически, для рынков акций и облигаций — это определенно благоприятное событие, но на практике рынок...



БКС Мир инвестиций

13.09.2025

Установите приложение Смартлаба:



RuStore



AppGallery



App Store



О смартлабе

Реклама

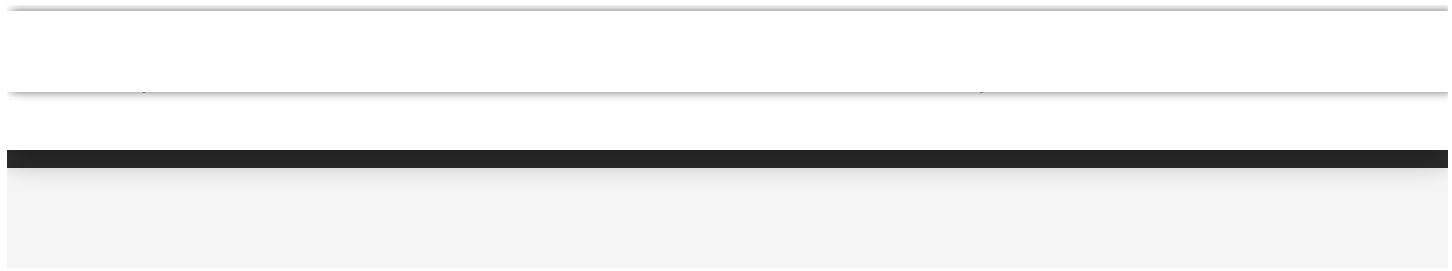
Полная версия



Московская Биржа является спонсором ресурса smart-lab.ru

Источник: ПАО Московская Биржа

Откройте счёт в ВТБ Мои Инвестиции



Откройте счёт в ВТБ Мои Инвестиции