defAI: An Autonomous AI Agent for DeFi/Trading with Aptos Integration

Your Name

March 7, 2025

Abstract

This paper presents defAI, an autonomous AI agent designed for decentralized finance (DeFi) and trading applications. The system optimizes yield, rebalances portfolios, and manages risk by combining advanced statistical metrics with on-chain operations via the Aptos blockchain. Novel algorithms such as a risk-adjusted yield score, risk index, and a dynamic rebalancing trigger are introduced. In addition, we detail the system design, pseudo code, and a flowchart illustrating the operational flow of defAI. Finally, the complete backend code implementation is provided.

1 Introduction

Decentralized Finance (DeFi) enables novel yield opportunities and risk management methods across lending protocols and decentralized exchanges (DEXs). However, the rapidly changing market dynamics require an autonomous solution to continuously optimize and rebalance portfolios. defAI addresses these challenges by:

- Optimizing yield using a novel risk-adjusted yield score.
- Monitoring risk through a rolling volatility-based risk index.
- Triggering portfolio rebalancing when significant deviations occur.
- Integrating with the Aptos blockchain via a Python SDK to execute on-chain transactions.

2 System Design and Architecture

The defAI system is structured into several modules:

1. **Data Acquisition:** Fetching historical market data via CCXT or local CSV.

2. **Metrics Calculation:** Computing the yield score and risk index over a rolling window.
3. **Rebalancing Signal:** Determining when the portfolio allocation deviates from a target.
4. **On-chain Integration:** Using the Aptos Python SDK to submit transactions (e.g., yield harvesting or portfolio rebalancing).
5. **Visualization:** Generating interactive charts (using Plotly) to display the evolution of metrics.

3 Algorithms and Pseudo Code

3.1 Yield Score Calculation

For a window of N periods, the return at time t, $r(t)$, is computed and then averaged:

$$\mu_r(t) = \frac{1}{N} \sum_{i=0}^{N-1} r(t-i)$$

The rolling standard deviation (volatility) is:

$$\sigma_r(t) = \sqrt{\frac{1}{N} \sum_{i=0}^{N-1} \left(r(t-i) - \mu_r(t)\right)^2}$$

The yield score is given by:

$$\text{Yield Score}(t) = \frac{\mu_r(t)}{\sigma_r(t) + \epsilon}$$

where ϵ is a small constant.

3.2 Risk Index Calculation

The risk index is based on volatility:

$$\text{Risk Index}(t) = \sigma_r(t) \times 100$$

3.3 Rebalancing Signal and Aptos Integration

Let the target allocation be a_0 (typically 0.5). Compute a new allocation:

$$a(t) = \min\left(1, \max\left(0, a_0 + \frac{\text{Yield Score}(t) - \text{Risk Index}(t)}{1000}\right)\right)$$

A rebalancing signal is triggered if:

$$|a(t) - a_0| > \delta,$$

with $\delta = 0.1$. Upon a trigger, the system submits an on-chain transaction via the Aptos SDK.

3.4 Pseudo Code

Algorithm 1 defAI Main Algorithm

procedure DEFAI_PROCESS(symbol, use_csv)
    data ← FETCHDATA(symbol, use_csv)
    if data is empty **then**
        return error
    end if
    yield_score ← CALCULATEYIELDSCORE(data, window=30)
    risk_index ← CALCULATERISKINDEX(data, window=30)
    allocation ← target_allocation + $\frac{yield\_score - risk\_index}{1000}$
    allocation ← clamp(allocation, 0, 1)
    deviation ← |*allocation*− target_allocation|
    if *deviation* > 0.1 **then**
        transaction ← PERFORMAPTOSTRANSACTION("rebalance")
        log trade with transaction details
    end if
    plot ← GENERATEINTERACTIVEPLOT(data, yield_score, risk_index)
    return trade details, plot, metrics summary
end procedure

4 Flowchart

Below is a flowchart illustrating the operational flow of defAI:

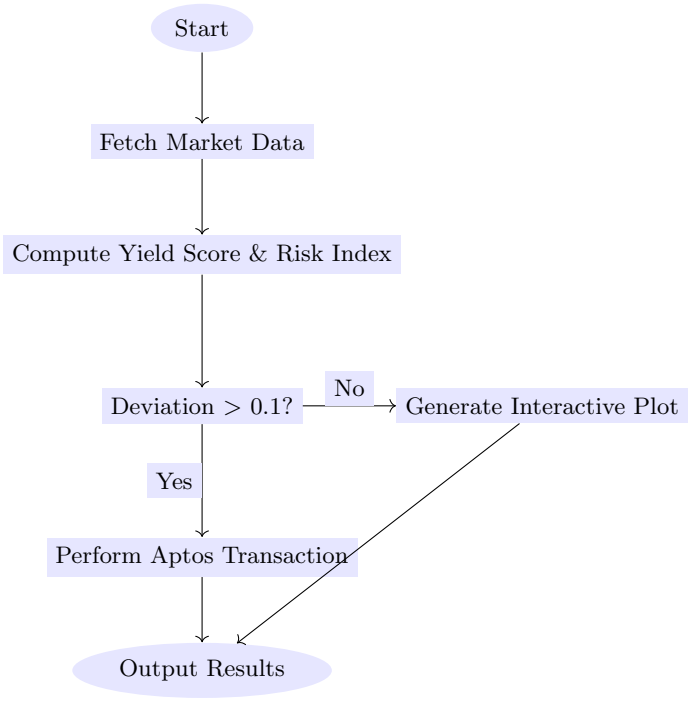

Figure 1: System Flowchart of defAI Module

5 Backend Codebase Overview

The backend code implements the defAI module as a new class (`DefAIAgent`) integrated within an existing Flask server. It includes:

- Data acquisition functions (via CCXT or CSV).
- Computation functions for yield score and risk index.
- A simulated rebalancing signal.
- A placeholder function for Aptos transactions using the Aptos Python SDK.
- Plot generation (using Plotly for interactive visualization).

6 Conclusion

defAI combines advanced algorithmic trading metrics with on-chain Aptos integration to provide an autonomous solution for yield optimization and risk management in DeFi. The presented system design, pseudo code, and flowchart

offer an overview of the implementation strategy, while the complete backend code (appended below) demonstrates a production-oriented design ready for further development and live testing.

Appendix: Backend Code Listing

```
% Insert the complete backend code here (see separate code listing)
```