

Adaptive Trading Strategies for Cryptocurrencies

Marek Filip

Supervisor: Ing. Ivan Homoliak, Ph.D.

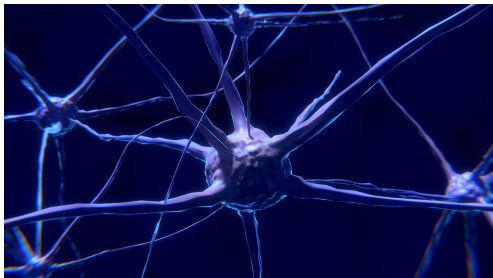


April 28, 2022

- Bear & bull markets
- Adaptive trading strategy
- Stablecoins



- Neural network
- Heuristics & metrics
- Backtesting
- Validation





- Adaptive strategy proposals
- Implementation & evaluation

```

120 use std::collections::HashMap;
121 use std::time::Duration;
122
123 struct TradingEngine {
124     orders: Vec<Order>,
125     executed_orders: HashMap<OrderId, Order>,
126 }
127
128 impl TradingEngine {
129     fn new() -> TradingEngine {
130         TradingEngine {
131             orders: Vec::new(),
132             executed_orders: HashMap::new(),
133         }
134     }
135
136     fn add_order(&mut self, order: Order) {
137         self.orders.push(order);
138     }
139
140     fn execute_order(&mut self, order_id: OrderId) {
141         if let Some(order) = self.orders.iter().find(|o| o.id == order_id) {
142             self.executed_orders.insert(order_id, order.clone());
143             self.orders.remove(order);
144         }
145     }
146
147     fn calculate_profit(&self) -> f64 {
148         let mut profit = 0.0;
149         for order in self.orders.iter() {
150             profit += order.price * order.quantity;
151         }
152         for order in self.executed_orders.iter().values() {
153             profit -= order.price * order.quantity;
154         }
155         profit
156     }
157 }
158
159 fn main() {
160     let engine = TradingEngine::new();
161     engine.add_order(Order {
162         id: 1,
163         price: 100,
164         quantity: 10,
165     });
166     engine.add_order(Order {
167         id: 2,
168         price: 105,
169         quantity: 5,
170     });
171     engine.execute_order(1);
172     engine.execute_order(2);
173     let profit = engine.calculate_profit();
174     println!("Profit: {}", profit);
175 }

```

- 31. 1.: Current State. Data Analyzation.
- 14. 2.: Adaptive strategy inspirations.
- 28. 1.: Proposal of several adaptive strategies.
- 14. 3.: Backtesting implementation.
- 28. 3.: First testing of adaptive strategies results.
- 11. 4.: Further backtesting results.
- 25. 4.: Further Improvements and practical deployment limitations – discussion.
- 1. 5.: Final State meeting.
- 11. 5.: Thesis Submission.

