
Algorithm 1: Backtesting Profit and Loss based on Prediction Model

Data: DataLoader *Loader*, selected prediction model *model*, number of sliding window *N*, extent of lagedPrice *laged*

Result: An array describing changes in cash *CA*;
An array describing changes in sumValue *SUM*;
An array describing results of prediction model *RA*

```
1 Initial MyAccount with 10000 cash;
2 round  $\leftarrow$  0 ;
3 CA  $\leftarrow$  [10000] ;
4 RA  $\leftarrow$  [ ];
5 An array describing values of stocks held SV  $\leftarrow$  [0];
6 foreach X in Loader do
7     round ++;
8     midprice  $\leftarrow$  (bestAsk + bestBid)/2;
9     prediction  $\tilde{y} \leftarrow$  model(X);
10    if  $\tilde{y} == \text{up}$  then
11        ask1laged  $\leftarrow$  best ask price at the next laged timestep;
12        myAccount.buy(ask1laged, ask1Volumu);
13        /* update 3 array */
14        RA append 1;
15        CA append current cash;
16        SV append midprice * #Stocks;
17    else if  $\tilde{y} == \text{down}$  then
18        bid1laged  $\leftarrow$  best bid price at the next laged timestep;
19        myAccount.sell(bid1laged, bid1Volumu);
20        RA append -1;
21        CA append current cash;
22        SV append midprice * #Stocks;
23    else
24        RA append 0;
25        CA append current cash;
26        SV append midprice * #Stocks;
27    if round == N then
28        sell all stocks in myAccount using a market order;
29        RA append -1;
30        CA append current cash;
31        SV append 0;
32        break;
33 end
34 SUM  $\leftarrow$  CA + SV;
35 return CA, SUM, RA;
```
