DESIGN DOCUMENT

Objective:

To get the trading position for the market.

Description:

The trading algorithm is based in the following formulation: -

$$\min\left\{\frac{1}{2}(\|\omega\|^2 + A\theta^2) + \alpha\sum_{t=1}^{T} a_t - \beta\sum_{t=1}^{T} b_t + \gamma\sum_{t=1}^{T}(c_t + d_t)\right\} \quad (1)$$

Subject to:-

$$r_t\,[\omega^T \emptyset(x^{t-1}) + \theta)] + (a_t - b_t) = 0 \qquad \text{such that } a_t\,,\, b_t \geq 0$$

$$\delta\omega^T\,[\emptyset(x^t) - \emptyset(x^{t-1})] + (c_t - d_t) = 0 \qquad \text{such that } c_t\,,\, d_t \geq 0$$

The formulation named as “OMEGA RATIO FORMULATION” aims at minimizing the losses (contained in a_t), transaction costs (contained in c_t and d_t) and maximizing the profits (contained in b_t).

The Ø’s represents the samples. Each sample is BarsBack dimensional i.e. each sample has past BarsBack features which can simply be the closing prices or some complex Indicator or a combination of both.

The T is the TradingWindowSize (T>=1).

α, β, γ are the weighing factors for the cumulative losses, profits and transaction costs respectively. The condition α>β ensures that minimizing losses is more important than maximizing profits. So we carefully implement this condition while selecting the parameters.

r is the return at time t given by r_t =price$_t$ -price$_{t-1}$

δ is the transaction cost incurred for the scrip being traded.

The above formulation is our primal formulation. After we apply KKT conditions and convert the above into dual formulation, it becomes:-

$$Min\frac{1}{2}\sum_{t=1}^{T}\sum_{u=1}^{T}\Big\{\lambda_t r_t \lambda_u r_u \phi(x^{t-1})^T\phi(x^{u-1}) + \lambda_t r_t g_u \delta[\phi(x^{t-1})^T\phi(x^u) - \phi(x^{t-1})^T\phi(x^{u-1})] + g_t g_u \delta^2[\phi(x^t)^T\phi(x^u) - \phi(x^t)^T\phi(x^{u-1}) - \phi(x^u)^T\phi(x^{t-1}) + \phi(x^{t-1})^T\phi(x^{u-1})] + \frac{1}{A}\lambda_t r_t \lambda_u r_u + \lambda_u r_u g_t \delta[\phi(x^{u-1})^T\phi(x^t) - \phi(x^{u-1})^T\phi(x^{t-1})]\Big\} \quad (2)$$

Constrained to the following conditions: -

$\beta \leq \lambda_t \leq \alpha$

$-\gamma \leq g_t \leq \gamma$

Minimizing equation (2) gives set of λ's and g's which gives us ω and θ.

$\omega=\sum_{t=1}^{T}\lambda_t r_t \Phi(x^{t-1}) + \sum_{t=1}^{T} g_t \delta(\Phi(x^t) - \Phi(x^{t-1}))$

$\theta=\frac{1}{A}\sum_{t=1}^{T}\lambda_t r_t$

Ideally, the current position in the market is given by sgn($\omega^T\emptyset(x^t) + \theta$) .But to inculcate the confidence of trader into our strategy, we compute the current position as: -

If $\omega^T\emptyset(x^t) + \theta >$ threshold, CurrentPosition=1

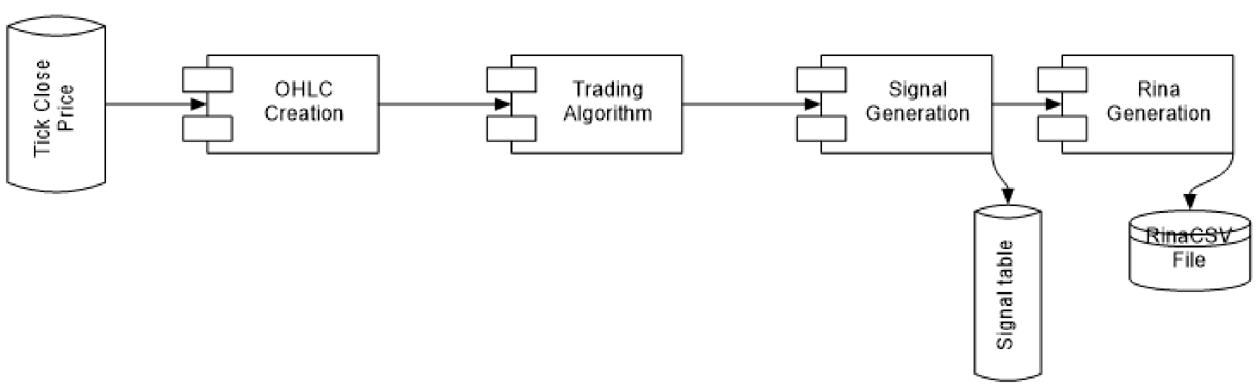
And If $\omega^T\emptyset(x^t) + \theta <$ -threshold, CurrentPosition=-1

So the algorithm requires past T closing prices to compute the current position. The output of the of the trading algorithm is ternary with 1, -1, 0 as the 3 possible states representing the long, short, and neutral position in the market respectively.

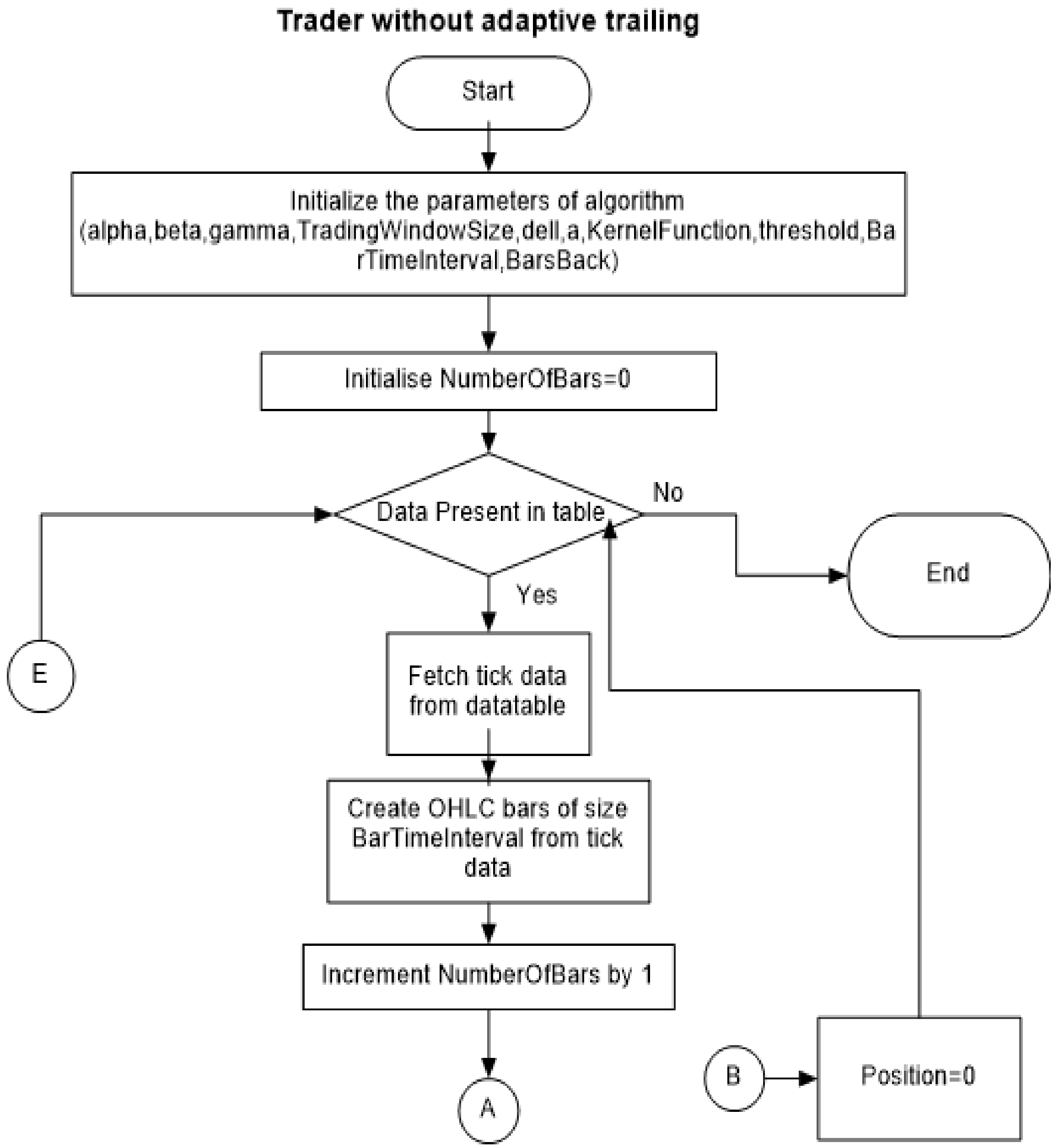
The preprocessing and post processing can be done can be done in an outside module. Preprocessing involves reading the data prices for each tick and preparing the OHLC bars for the time interval specified by BarTimeInterval. Sequence of states can be converted into Trading Signals post processing. We stand Long as long as the output is +1, Short as long as the output is -1 and neutral whenever the output is 0. Static Trailing and Stop Loss can be applied after we take positions in the market.

After we get the position for current time instant or current bar, we repeat the procedure for next bar by shifting the entire T sized Trading Window to the right exactly by 1 bar.

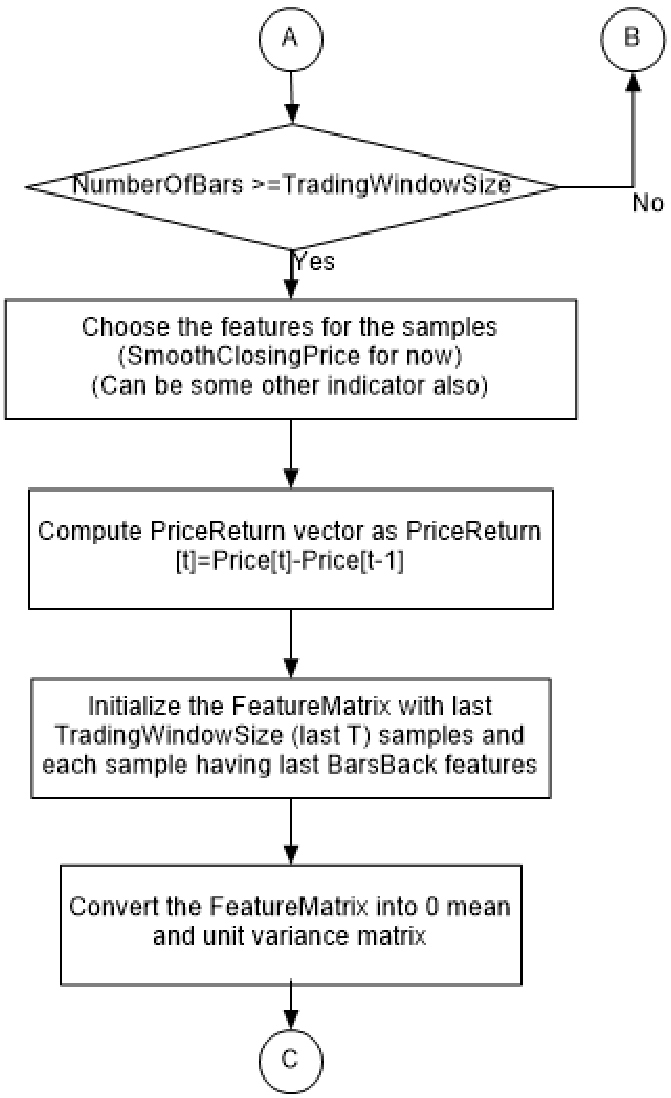
Component Relationship Diagram:

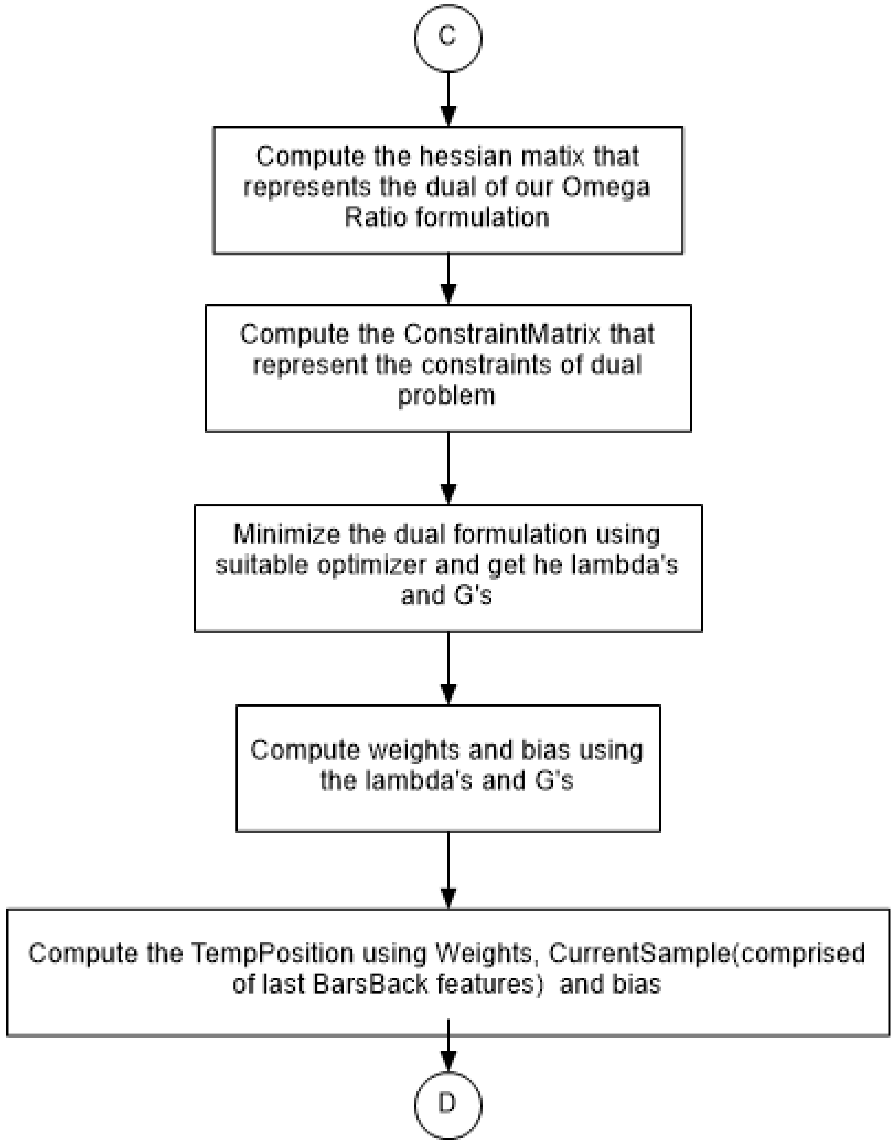


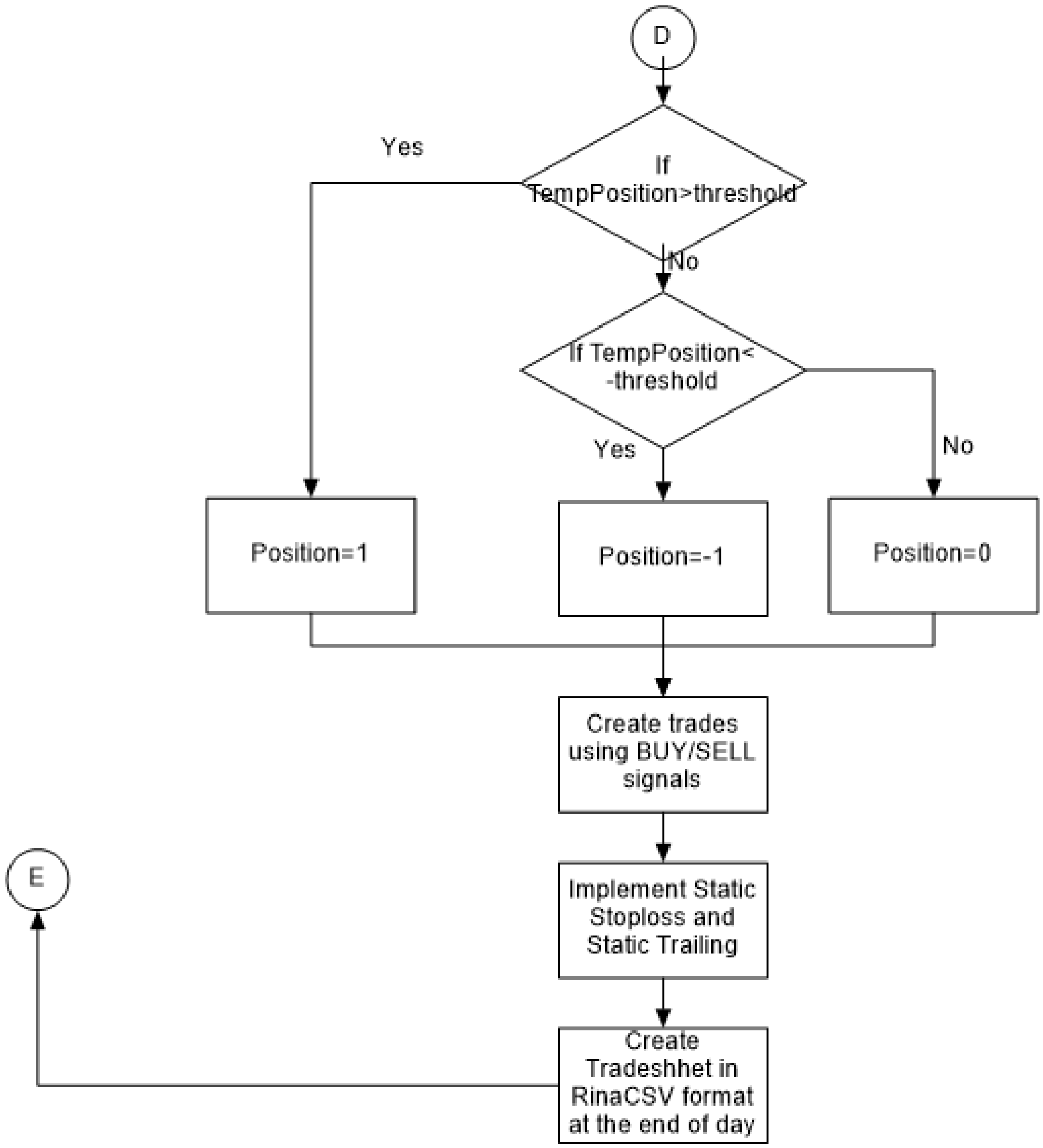
Flowchart:



Trader without adaptive trailing
Start
Initialize the parameters of algorithm (alpha,beta,gamma,TradingWindowSize,dell,a,KernelFunction,threshold,BarTimeInterval,BarsBack)
Initialise NumberOfBars=0
Data Present in table
No
End
Yes
E
Fetch tick data from datatable
Create OHLC bars of size BarTimeInterval from tick data
Increment NumberOfBars by 1
A
B
Position=0


A
B
NumberOfBars >=TradingWindowSize
No
Yes
Choose the features for the samples
(SmoothClosingPrice for now)
(Can be some other indicator also)
Compute PriceReturn vector as PriceReturn
[t]=Price[t]-Price[t-1]
Initialize the FeatureMatrix with last
TradingWindowSize (last T) samples and
each sample having last BarsBack features
Convert the FeatureMatrix into 0 mean
and unit variance matrix
C


C
Compute the hessian matix that represents the dual of our Omega Ratio formulation
Compute the ConstraintMatrix that represent the constraints of dual problem
Minimize the dual formulation using suitable optimizer and get he lambda's and G's
Compute weights and bias using the lambda's and G's
Compute the TempPosition using Weights, CurrentSample(comprised of last BarsBack features) and bias
D


D
Yes
If TempPosition>threshold
No
If TempPosition< -threshold
Yes
No
Position=1
Position=-1
Position=0
Create trades using BUY/SELL signals
E
Implement Static Stoploss and Static Trailing
Create Tradeshhet in RinaCSV format at the end of day