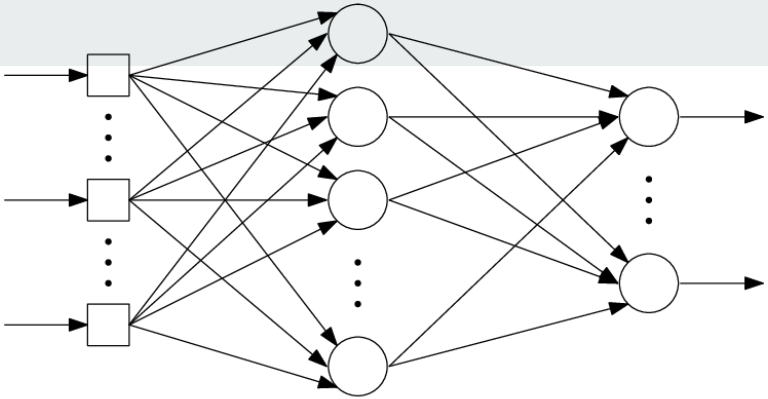
Digital VLSI Design
Final Project

Swapnil Pakhare 20161199
Pooja Srinivas 20171403
Ayush Anand - 2016
Mayank Taluja - 20161102

Model Based Project

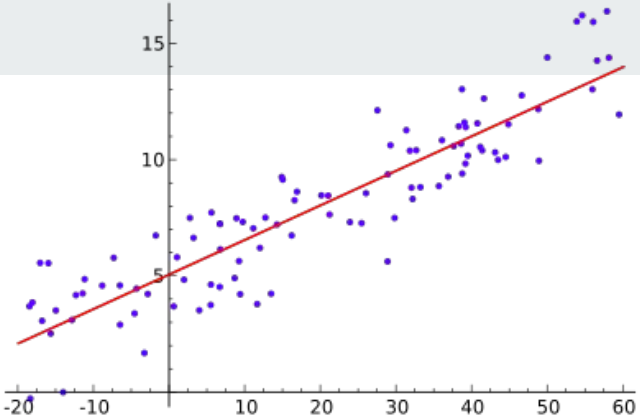
- Reducing the cost and time of any model generation and testing is of essence in research to develop better technologies.
- Training supervised machine learning models to predict the leakages, power and delays.
- These models can be trained once and in turn used multiple times to test the working of a specific technology.
- The motivation of using this method is to speedup the entire process of testing and analyzing the working of a technology.

Neural Networks



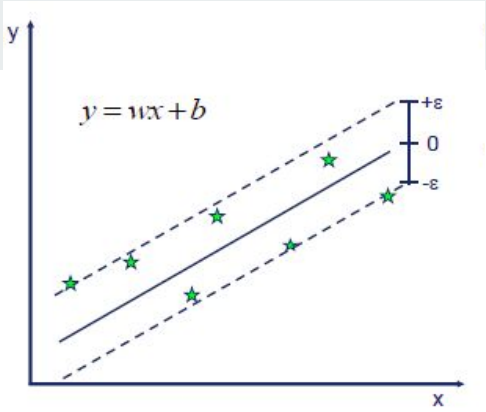
- Artificial Neural Networks are modeled loosely after the human brain, that are designed to recognize patterns.
- For this project we have trained a neural network with one hidden layer to predict the output for a specific input condition for a given circuit.
- The number of outputs can be varied depending on the circuit and output to be evaluated.

Linear Regression

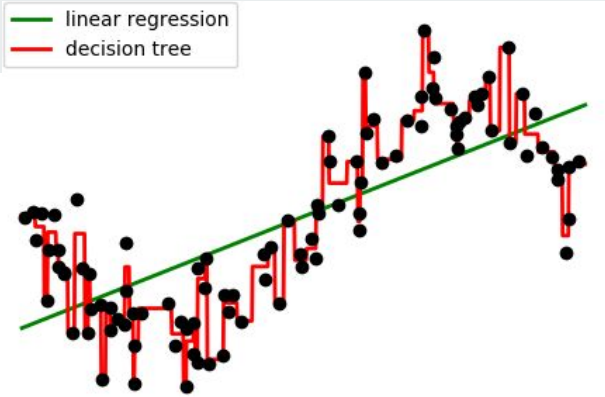


- Linear regression is a linear approach to modelling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables).
- To model multiple response variables simple linear regression was not sufficient. Multiple Output Regression is used to model a simple linear regression to a multi output linear regression.
- Multiple Output Regression fits, the regressor model specified, one per output. Hence the entire regressor behaves as one and enables the prediction of multiple outputs.

Support Vector Regression

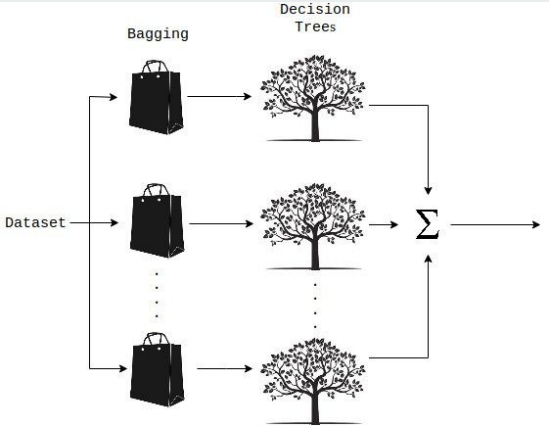


- The Support Vector Regression (SVR) uses the same principles as the SVM for classification
- The main idea is to minimize error, individualizing the hyperplane which maximizes the margin, keeping a check so that, part of the error is tolerated.
- Non Linear SVR was also implemented but not retained because there was no significant decrease in error and hence all the kernel functions were redundant adding to the computation complexity.
- SVR was performed over Linear, polynomial and rbf.



Decision Tree Regression

- Decision tree builds regression or classification models in the form of a tree structure.
- THis model learns local linear regression with a tree structure based on the depth to fit the training data.
- The model was tested varying the depth but no significant change was observed for the given data set.
- The variation in depth was tested for other data with noise and increasing the depth by a large amount in those cases led to overfitting.

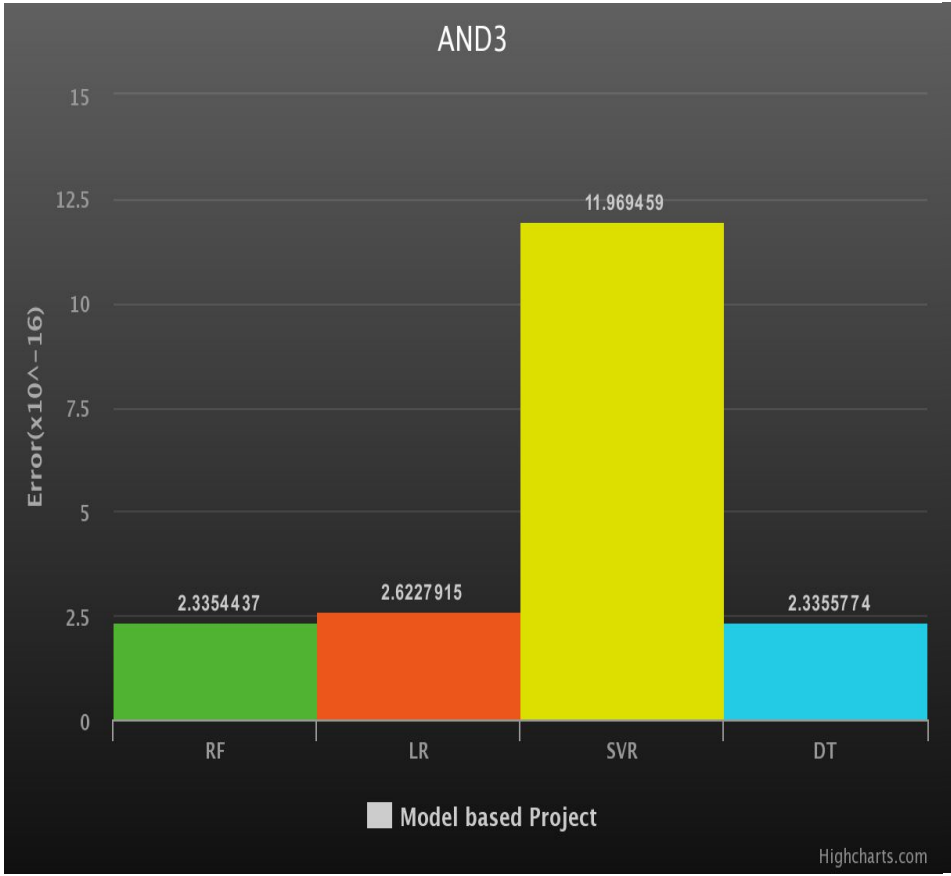


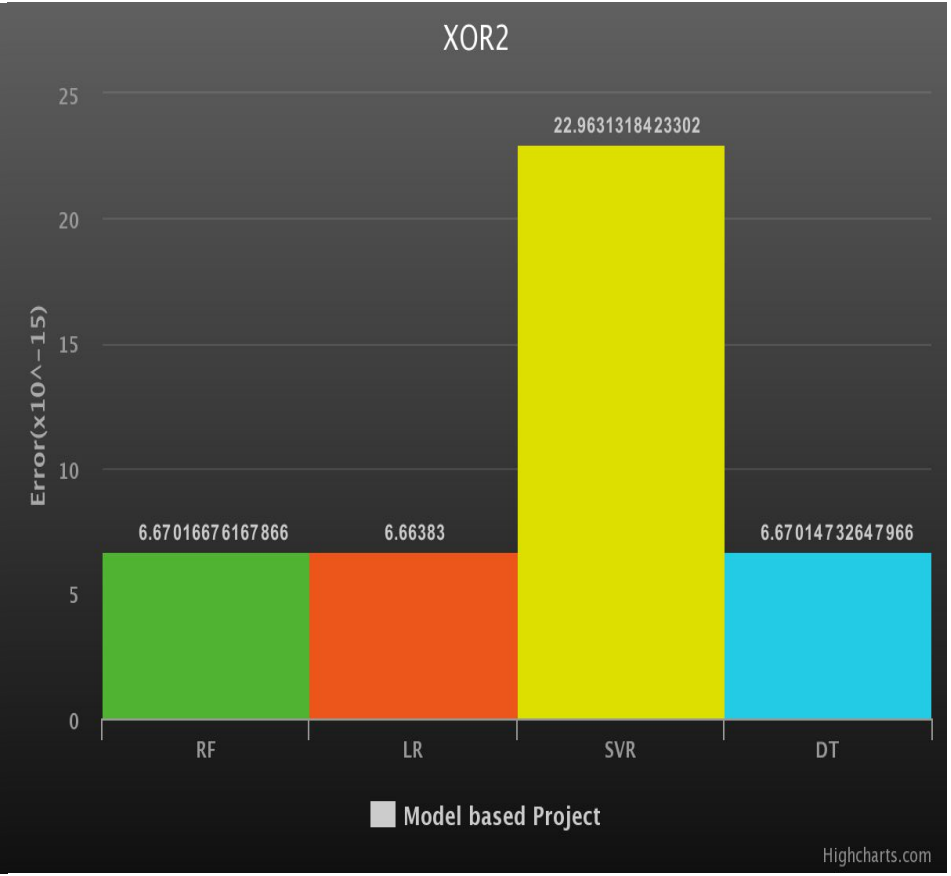
Random Forest Regression

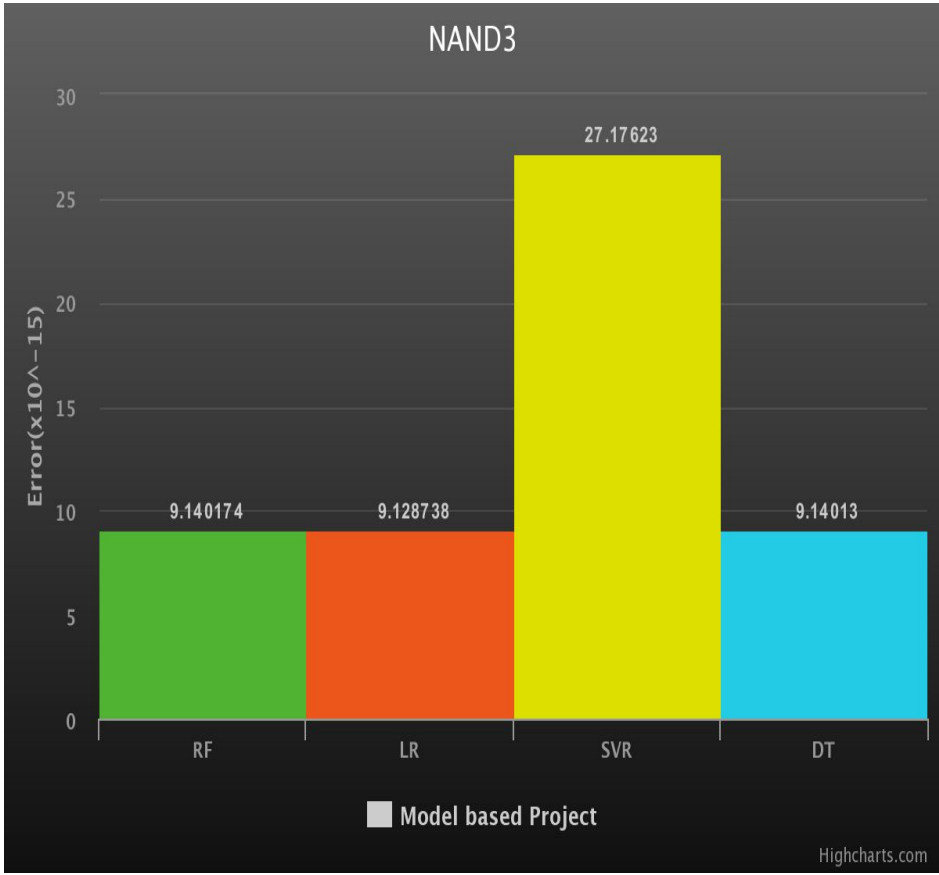
- A random forest is a meta estimator that fits a number of classifying or regression decision trees, to improve the predictive accuracy and control over-fitting.
- Bagging, in the Random Forest method, involves training each decision tree on a different data sample where sampling is done with replacement.
- In our model, we varied the n_estimator parameter which decides the number of decision tree models used. The errors remain almost constant even with large variations.

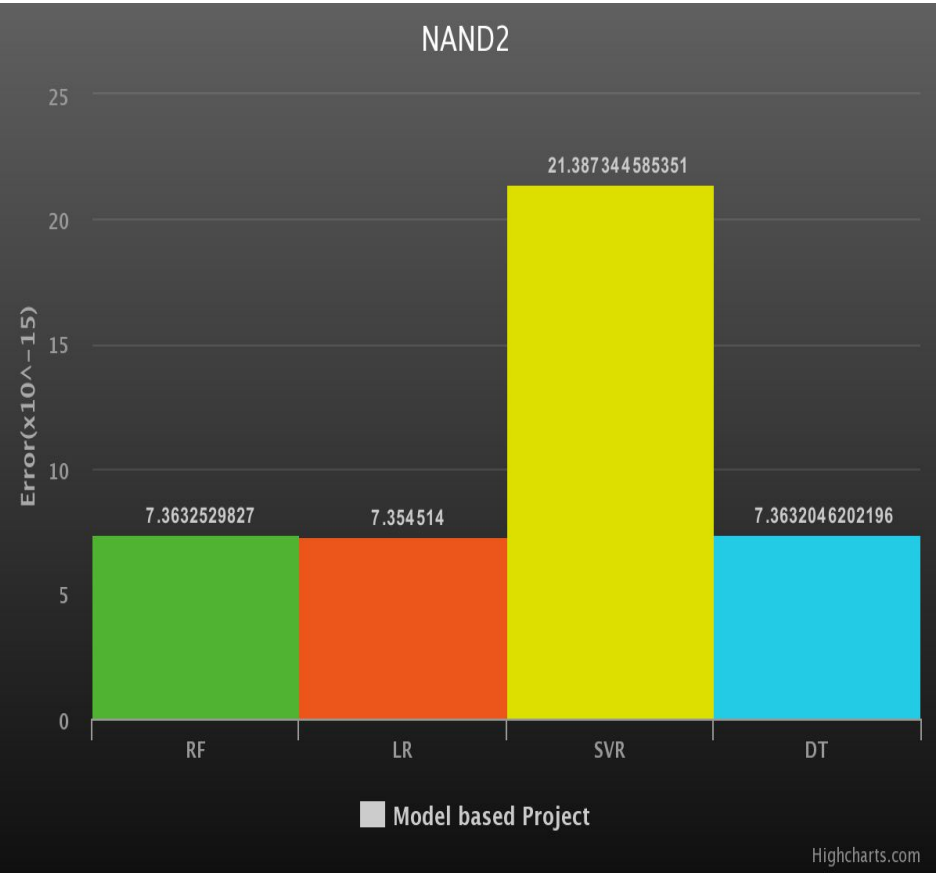
Results

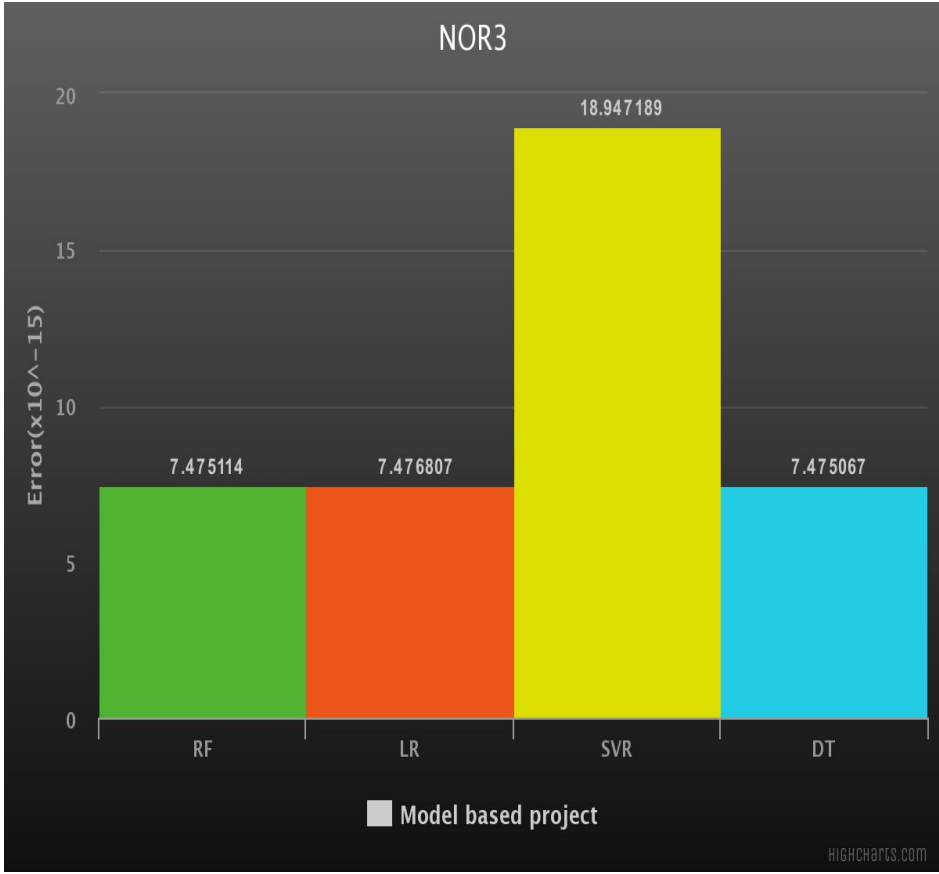
- Mean Square Error has been calculated individually for every gate or circuit with different input conditions.
- For example, AND2 has 00, 01, 10, 11 input conditions. Each model has been train separately for all the conditions to give delays and leakages.
- Final results were calculated by taking a sum of the all the input conditions for both delays and leakages, to obtain a final error value for a specific circuit from each model.

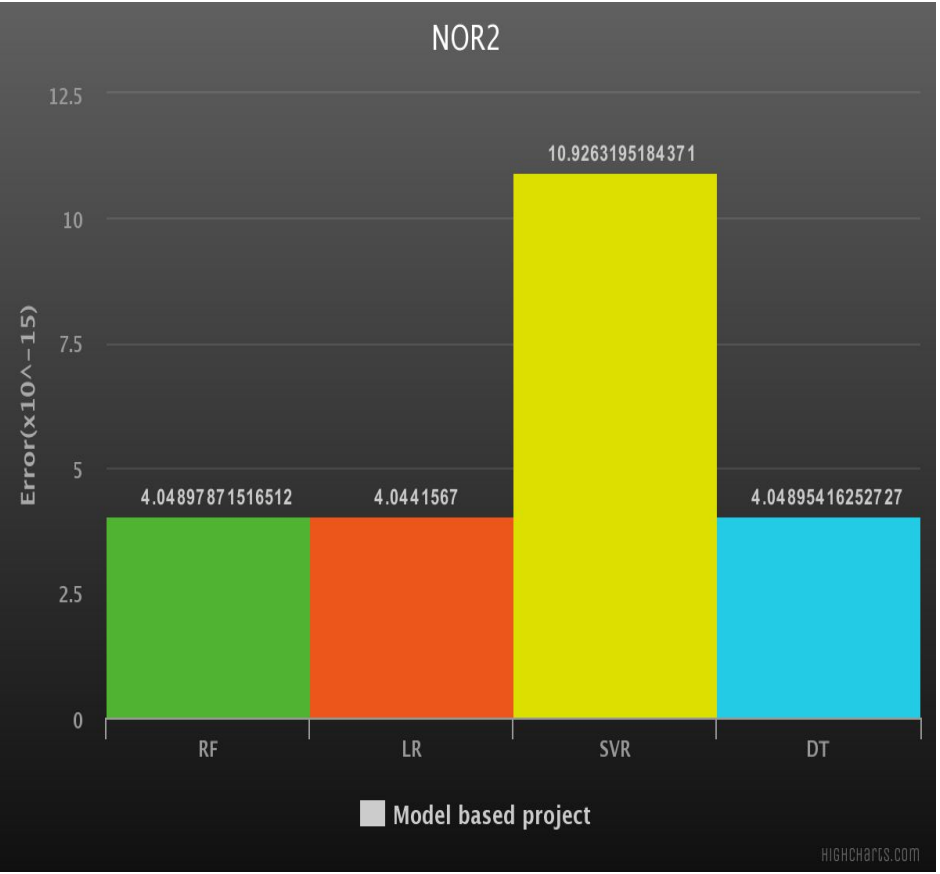

AND3
Error(x10^-16)
15
12.5
10
7.5
5
2.5
0
2.3354437
2.6227915
11.969459
2.3355774
RF
LR
SVR
DT
Model based Project
Highcharts.com


XOR2
Error(x10^-15)
25
20
15
10
5
0
6.67016676167866
6.66383
22.9631318423302
6.67014732647966
RF
LR
SVR
DT
Model based Project
Highcharts.com


NAND3
Error(x10^-15)
30
25
20
15
10
5
0
9.140174
9.128738
27.17623
9.14013
RF
LR
SVR
DT
Model based Project
Highcharts.com


NAND2
Error(x10^-15)
25
20
15
10
5
0
7.3632529827
7.354514
21.387344585351
7.3632046202196
RF
LR
SVR
DT
Model based Project
Highcharts.com


NOR3
Error(x10^-15)
20
15
10
5
0
18.947189
7.475114
7.476807
7.475067
RF
LR
SVR
DT
Model based project
Highcharts.com


NOR2
Error(x10^-15)
12.5
10
7.5
5
2.5
0
10.9263195184371
4.04897871516512
4.0441567
4.04895416252727
RF
LR
SVR
DT
Model based project
Highcharts.com

Algorithm based Project

- Problem Statement - Transistor parameter setting for power and delay optimisations.
- Transistor parameters can be Widths and Lengths of each transistor, Supply Voltage, and Temperature among others.
- We have considered only widths and lengths of transistors in the current algorithm but this can be extended very easily to Voltage and Temperature as well.
- This problem is essentially a multi-dimensional multi-objective minimisation problem.
- Circuit chosen is **Full Adder** and algorithm implemented is **Krill Herd**.

Algorithm Project - Introduction

- The Krill Herd Algorithm belongs to the category of Bio-inspired Optimisation Algorithms.
- Such algorithms are chosen because they do extremely well on multi-dimensional optimisation problems. They are then extended to work for multiple objectives.
- As multi-objective Krill Herd does not exist, we reduced all Leakages to a single objective and kept all delays in a strict bound.
- For Full Adder, we have 8 leakages and 6 delays => 14 objectives. To reduce this to a single objective problem, we take mean of leakages and keep delays in bound.

The Krill Herd Algorithm

- Is modelled after “Krills” which are a kind of fish that try to move towards 2 things:
  - Areas of higher food
  - Areas of higher krill density
- It includes Lagrangian for motion. The step update of Krill position is:

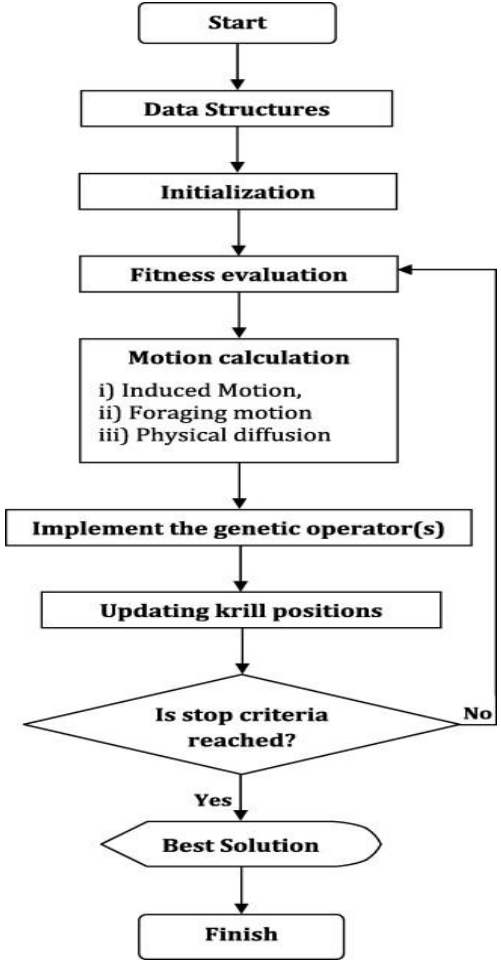
$$X(t+\Delta t) = X(t) + \Delta t * \frac{\partial X}{\partial t}$$

The Krill Herd Algorithm

- For a particle “i”, its step is given by

$$\frac{\partial X_i}{\partial t} = N_i + F_i + D_i$$

- Here, N_i is the Neighbor induced motion, F_i is the Foraging motion and D_i is the random Diffusion motion of the i’th particle.
- Equations of individual motions are given in the research paper (link).


Start
Data Structures
Initialization
Fitness evaluation
Motion calculation
i) Induced Motion,
ii) Foraging motion
iii) Physical diffusion
Implement the genetic operator(s)
Updating krill positions
Is stop criteria reached?
No
Yes
Best Solution
Finish

Results - Delays

Delay	Old	New
delay_lh_nodeaco	1.00558e-11	8.69043e-12
delay_hl_nodeaco	1.17463e-11	2.7143e-11
delay_lh_nodebco	8.47425e-12	7.14982e-12
delay_hl_nodebco	1.18857e-11	2.89149e-11
delay_lh_nodecco	1.03265e-11	1.21012e-11
delay_hl_nodecco	1.13746e-11	1.10815e-11

Results - Leakages

Leakages	Old	New
000	3.65835e-06	2.14053e-06
001	3.69198e-06	2.00234e-06
010	3.04658e-06	1.77634e-06
011	3.00985e-06	1.84428e-06
100	2.98732e-06	1.79932e-06
101	2.69092e-06	1.57589e-06
110	2.99784e-06	1.88214e-06
111	2.66278e-06	1.68353e-06

Thank You