ML Algorithm Implementation

Nahian Siddique
Doga Ozgulbas
Shahrukh Alam Khan

1 Motivation

The current challenge in big data is the efficient analysis of large-scale data. In recent years, there has been an overwhelming increase in the size of data collected in all industries. While the size of this data has grown exponentially, the processing power of computers has not. We must therefore rely on optimization algorithms to improve our data processing speeds. One such novel class of optimization algorithms is swarm intelligence which rely on the collective behavior of many decentralized systems to process data. In this project we explore two of these swarm intelligence algorithms and find their utility in big data.

2 Background

2.1 Swarm Intelligence

Optimization is a field of computer science and mathematics which deals with finding the best solution from a set of possible solutions.

Optimization problems can be divided into two categories:

- Classical or Deterministic solutions
- Nature Inspired or Stochastic solutions

Deterministic solutions involve the use of mathematical models to find the exact solution. However, a certain category of problems, called NP-Hard problems, cannot be solved in a reasonable time using deterministic solutions. This led to the exploration of nature inspired solutions.

As the name implies, nature inspired solutions make use of models inspired from the behavior of natural phenomena. These algorithms use stochastic search to find near optimal solutions and by being heuristic, perform significantly faster than deterministic solutions.

Nature inspired algorithms can be further divided into two categories:

- Evolutionary algorithms
  - Genetic algorithms
  - Differential evolution
- Swarm intelligence
  - Particle swarm optimization
  - Chemical reaction optimization
  - Ant colony optimization
  - Gravitational search optimization
  - Binary bat optimization

Swarm intelligence is a class of population-based metaheuristics. According to swarm intelligence, the collective behavior of many disconnected systems with defined rules can result in predictable and controlled outcomes. The rules to define the behavior of the population has been modeled after various natural systems.

All swarm intelligence based metaheuristics have three basic stages:

1. Population initialization – The population or the swarm is created and each member of the swarm represents one given solution to the problem. Usually this stage is randomized.
2. Iteration – For a certain number of repetitions, the members of population undergo certain predefined operations to search for a solution. The search space may be:
   - Local, also known as Intensification or Exploitation – The members of the swarm search within their neighborhood to find the local optima.
   - Global, also known as Diversification or Exploration – The swarm as a whole searcher for the global optima.
3. Termination – The algorithm is terminated by some stopping criteria.

It is well known that according to the No-Free-Lunch-Theorem, any metaheuristic will perform well only on a certain set of problems. This leads us to continually finding newer and more novel algorithms that can help us solve different classes of problems. In this project, we look at particle swarm optimization, a well established and highly used metaheuristic, and compare it to the up and coming chemical reaction optimization.

2.2 Clustering

Clustering is one of the most popular exploratory data analysis technique in which we used to get intuition about data structure. Clustering is defined as:

"Identification of sub groups in the data such that the same group have data points that have similar features.". The criteria we used for the clustering of data point is that we search for the subgroup of samples based on same features or subgroups of features based on samples. One of the applications of clustering is in market segmentation; in which we try to find the behavior of customers that how many people have same behaviors or attributes. There is a lot of other examples as well.

Clustering is form of unsupervised learning method, unlike supervised learning. In this method, the model is not supervised and it is used to find the unknown the unknown patterns in the data, the output of the clustering algorithm is being compared to true labels.

3 Algorithms

3.1 K-Means

K-means algorithm is simple to implement among all the clustering algorithms so it is determined as one the most popular technique. K-means algorithm is defined as: "It is an iterative algorithm in which we try to divide out dataset into K per-defined distinct non-overlapping clusters and this group have the data point similar in their properties."

K-means clusters the data points having similar features closest to each other while keeping the clusters different. Initially, the random values of centroid is being picked up and the distance between all the data points and centroids is calculated and the minimum distance centroid dominates the cluster. The less variant data points we have in the clusters, the more homogeneous properties they have.

How it works?

The distance formula for calculating the distance between the data points and centroid is:

$$d = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

The procedure of k means algorithm involves the following steps:

1. Firstly, we specify the number of clusters.
2. Secondly, we shuffle all the dataset and then initializing the centroid by picking the random centroids, from the data points, called as **k** data points.
3. Thirdly we take the mean of same cluster data points and then the iteration is being started and it continues until there will be no difference between the centroids.

- The sum of squared distance between the data points and centroids is calculated.
- The minimum distance centroid is assigned to the cluster.
- The next centroid is calculated by taking the average of all the data points having same clusters.

So, this is the whole procedure of implementation of k means algorithm.

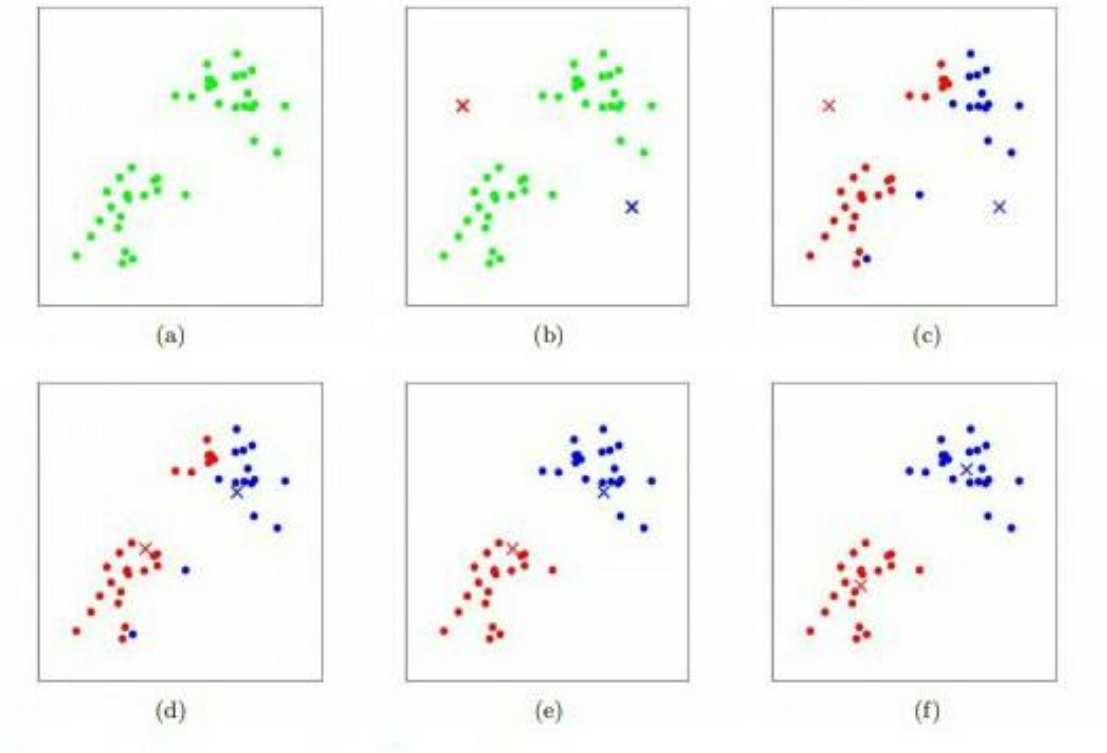

Figure 1. (a) The initial unclustered data. (b) Initialization of random centroids. (c) Initial clustering using first set of centroids. (d)(e) Continued iterations leading to defined clusters. (f) Final centroids established with well-defined clusters.

K means, due to its popularity, has a variety of applications in each and every field. Some of the more popular application of k means are: market segmentation, image compression, document clustering, image segmentation etc. Our basic goal in implementing k means is to get a meaningful intuition of structure of data we are considering. Also, the prediction to build different models is done when cluster predict different models of our data.

3.2 Particle Swarm Optimization

Particle Swarm Optimization (PSO) is inspired from the be refers to the group of the birds of swarms. We will try to understand this algorithm with the help of an example: Let suppose there is group of swarms and

they are hungry so they are searching for their foods. These birds can be corelated with the jobs in the computation system which are hungry in the same way for the resources just as swarms. There is only one food particle in the area of these birds. This food particle is equal to a resource. There are many computation jobs but limited resources. Now, suppose that the food is hidden somewhere and the birds try to find the food so the algorithm is designed to find that food. If every bird is trying to find the food in the area at the same time so it may cause lot of panic and it would be a great wastage of time. The birds don't know the exact location of particle but they do know their distance from that particular particle and the best approach to find the particle is to follow the bird that is closest to that particle. In the computation environment this behavior of birds is simulated and the algorithm is thus called as particle swarm optimization. This algorithm is also termed as population based stochastic algorithm and was developed by Dr. Russel C. Eberhart and Dr. James Kennedy in the year 1995. This is what the particle swarm optimization actually is.

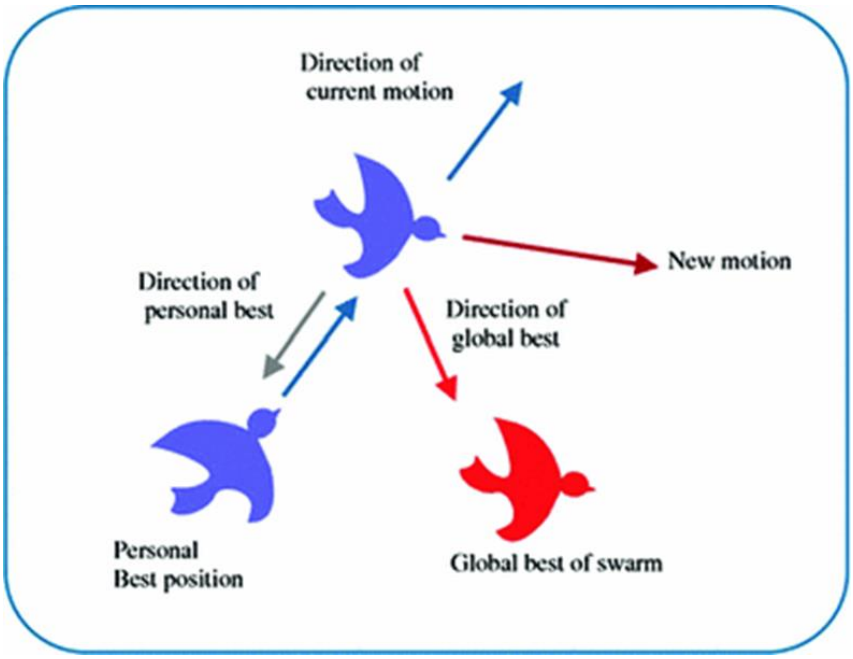

Figure 2. Representation of PSO using bird flocking.

Particle Swarm Optimization finds swarms of particle that is updating from iteration to iteration. The best way to seek the optimal condition is that each particle follows the direction to its previously best position

$$pbest\,(i,t) = \arg\min_{k=1,\ldots,t} \left[f\left(P_i\left(k\right)\right)\right], \quad i \in \left\{1,2,\ldots,N_P\right\},$$

$$gbest\,(t) = \arg\min_{\substack{i=1,\ldots,N_P \\ k=1,\ldots,t}} \left[f\left(P_i\left(k\right)\right)\right],$$

and the global best position in the swarm.

Where i = Particle index,

Np = Total number of particles

t = he current iteration number

f = fitness function

P = Position

The following equation can be used to find the velocity V and position P of the particle:

$$V_i(t+1) = \omega V_i(t) + c_1 r_1 \left(pbest(i,t) - P_i(t)\right) + c_2 r_2 \left(gbest(t) - P_i(t)\right),$$
$$P_i(t+1) = P_i(t) + V_i(t+1),$$

Where V = velocity,
w = inertia weight
r1 = global exploration
r2 = local exploitation
c1, c2 = positive constant parameters (acceleration coefficients)

Inertia weight is used to balance r1 and r2 which are uniformly distributed random variables within range [0,1].

The upper bound for velocity parameter is usually set. "Velocity clamping" is the method to limit particles flying out of the search space. There is another strategy proposed by Clerc and Kennedy called "constriction coefficient", when he did theoretical analysis of swarm dynamic, in which the velocities are constricted too. The previous velocity is represented in the first part of the formula, which is known as inertia. The individual particle thinking of each particle is represented in the second part of the formula which is known as "cognitive". It supports the particles to follow the direction towards their own best position found so far. The collaborative effect of the particles to search for the global optimal solution is represented in the third part of the formula which is known as "cooperation".

The application of Particle Swarm Optimization varies widely in academic and industrial fields so far. Based on the analytical tool which is provided by the "Web of Science Core Collection", the hottest applications characterized are in following domains:

1. Electrical and Electronics Engineering
2. Automation Control Systems
3. Communication Theory
4. Operations Research
5. Mechanical engineering
6. Civil Engineering
7. Fuel and Energy
8. Medicine Engineering
9. Chemical Engineering
10. Biological Engineering

Just as all other swarm intelligence based optimization techniques, there are many drawbacks of PSO technique as well, which includes: premature, high computational complexity, slow convergence, sensitivity to the parameters and so on. One of the main causes of all these problems is that the cross over operator is not utilized by PSO as employed in GA or DE. Hence the distribution of good info among candidate is not at the required level. Another reason is that the relationship between r1 and r2 is not handled properly by PSO so the local minimum is normally converged by it easily.

3.3 KMPSO

KMPSO algorithm is a combination of K-Means and Particle Swarm Optimization algorithms. This method is created to get better results than just an implementation of PSO or K-Means algorithms. K-Means clustering is one of the best-known and widely used algorithms for data clustering. Although K-Means algorithms are easy to understand, implement and operate, they are not suitable for many types of data and are not ideal for parallel implementation. One of the main disadvantages of cluster efficiency of K-Means algorithms is their sensitivity to the first centroids set to random data points. This problem is avoided by combining the K-Means algorithm with Particle Swarm Optimization. K-means is first run for a fixed number of iterations independently and the obtained solution becomes a particle for PSO, while other particles are randomly generated as before. In this stage PSO starts running independently. The diversity of the swarm ensures that a global search is conducted. In our clustering implementation by using KMPSO, resulting performance is significantly better than just PSO and K-Means [1].

3.4 Chemical Reaction Optimization

Chemical Reaction Optimization (CRO) is a recently established metaheuristics for optimization, inspired by the nature of chemical reactions. A chemical reaction is a natural process of transforming unstable molecules to more stable ones. In CRO, different solutions are modelled as different molecular structures, and interaction between different molecule, i.e. a chemical reaction, creates new molecules with different structures. In other words, a molecular structure is one solution and by finding the molecule with the highest stability, we in turn find the optimum solution [2] [3].

When initializing the population, each molecule is given certain set of attributes. They are as follows:

1. **Molecular structure** is a vector or matrix that describes one complete solution for the given problem space.
2. **Potential energy** is determined by its structure, i.e., stability. Meaning, a molecule with less PE has a more stable structure and is therefore a better solution. Thus, the objective function value corresponds to the PE of a molecule. The objective function can be defined as $PE_{\chi} = f(\chi)$.
3. **Kinetic energy** is the energy described by the motion of the molecule.
4. **Number of hits (NumHit)**, which is simply the number of collisions a molecule has gone through.
5. **Minimum structure** is the structure with the least PE, i.e. the most stable molecule. Each molecule keeps a record of its minimum structure.
6. **Minimum potential energy** the lowest PE achieved by the molecule and corresponds to the minimum structure. When a molecule achieves a new minimum potential energy, it updates is minimum structure.
7. **Minimum hit number (MinHit)** is the number of collisions it took for the molecule to achieve minimum structure.

Every reaction will cause some change in these attributes, and eventually lead us to the most stable configuration, where we will find the optimal solution.

The model of chemical reactions in CRO is classified into two categories, each with two possible outcomes [2] [3]:

- Unimolecular reaction
    1. On-wall ineffective collision: When a molecule hits the wall of the container and bounces off of it with some molecular attribute changes. This reaction is only allowed if the PE of the new molecular structure, $\chi`$, is less than the total energy of the old molecular structure, $\chi$. That is, $PE_{\chi} + KE_{\chi} \geq PE_{\chi`}$.

2. Decomposition: This occurs when the molecule hits the wall of the container and splits into two new molecules with very different molecular structures. Decomposition is allowed when $NumHit - MinHits > \alpha$, where α is a threshold value for decomposition.

- Intermolecular reaction
  3. Intermolecular ineffective collision: When two molecules collide and bounce off of each other with subtle changes in molecular structure. This reaction is only allowed under the following condition: $PE_{\chi 1} + KE_{\chi 1} + PE_{\chi 2} + KE_{\chi 2} \geq PE_{\chi 1`} + PE_{\chi 2`}$.
  4. Synthesis: When two molecules collide and fuse into a new molecule with a completely different molecular structure. The condition for synthesis is $KE_{\chi 1} \leq \beta$ and $KE_{\chi 2} \leq \beta$, where β is a threshold value for synthesis.

On-wall ineffective collision and intermolecular ineffective collision implement intensification. These two reactions change the structure of the molecules slightly based on the KE and PE. This allows the system to perform local search. Decomposition and synthesis implement diversification. Either one molecule splits into two or two molecules combine into one. In both cases there is a drastic change in the molecular structure and the system the system can perform global search. As a result, CRO can perform an overall search to find the best solution.

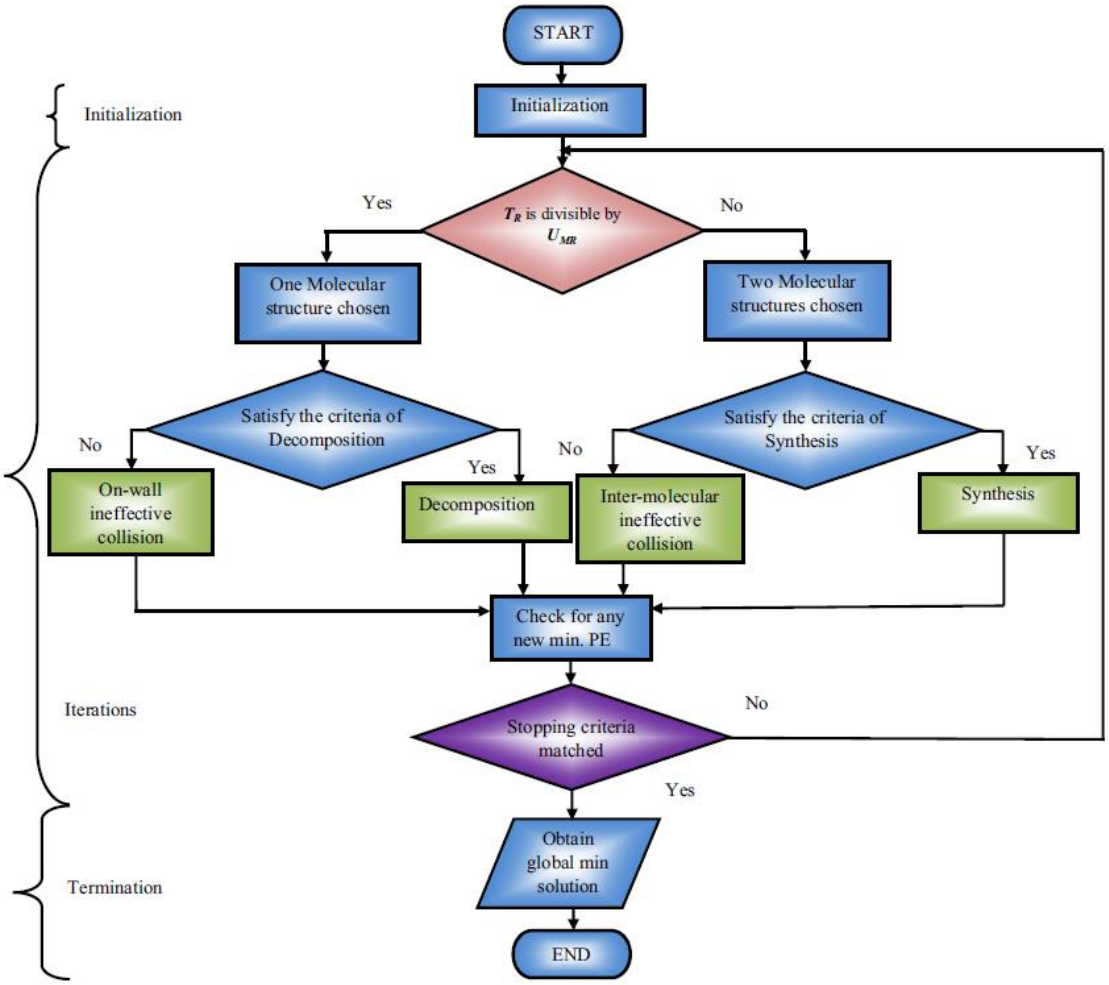

Figure 3. Flow diagram of CRO

4 Methodology

4.1 Cloud computing

Cloud computing is used to store and access data over the Internet. It does not store any data on the hard disk. In cloud computing, you can access data from a remote server. For our project, we have made use of Amazon Web Services (AWS). AWS platform provides flexible, reliable, scalable, easy to use and cost-effective cloud computing solutions. It offers a wide range of global cloud-based products for different business purposes. Products include storage, databases, analytics, networking, mobile, development tools.

AWS provides the following compute services:

1. **EC2** (Elastic Compute Cloud): EC2 is a virtual machine in the cloud on which you have OS level control.
2. **LightSail**: This cloud computing tool automatically deploys and manages the computer, storage, and networking capabilities required to run your applications.
3. **Elastic Beanstalk**: The tool offers automated deployment and provisioning of resources like a highly scalable production website.
4. **EKS** (Elastic Container Service for Kubernetes):The tool allows you to Kubernetes on Amazon cloud environment without installation.
5. **AWS Lambda**:This AWS service allows you to run functions in the cloud. The tool is a big cost saver for you as you to pay only when your functions execute.

In our project we have created an Apache Hadoop cluster on Amazon EC2 compute service. We worked on Ubuntu Server 16.04 LTS to create our instances. For the instance type, we choose **t2.micro** and requested four instances on the instances page, we setup the names of the instances with one namenode and three datanode. One of the datanode is also set as secondary namenode.

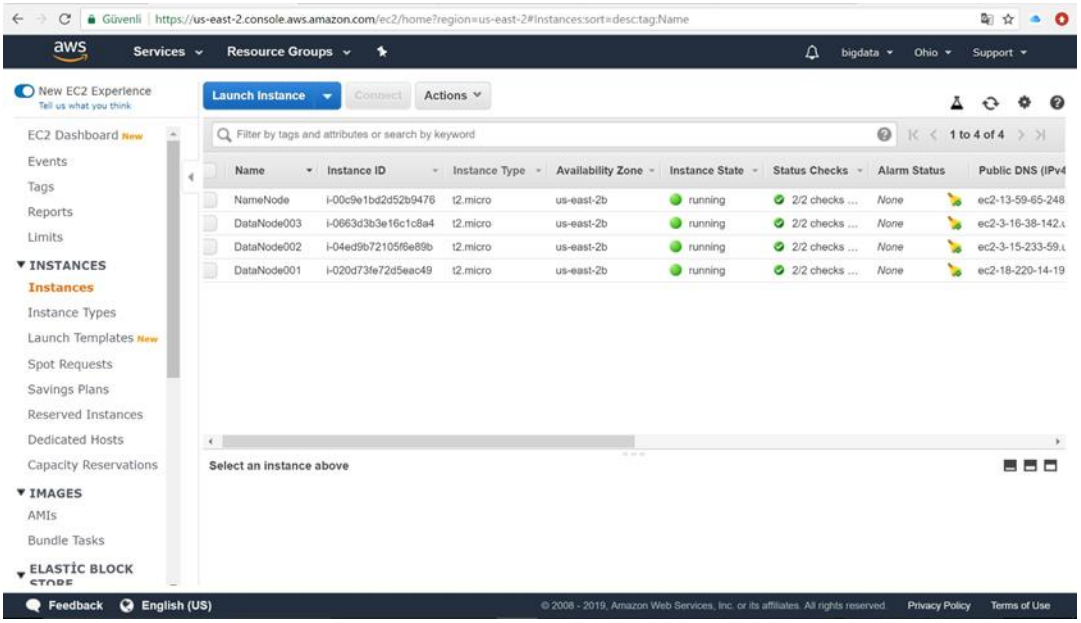

Figure 4. AWS EC2 Instances

In the second part, we were able to work on the nodes by collecting DNS and IP numbers. We have uploaded Hadoop system for every node by using PuTTY software. Also, we have used the WinSCP software to display Ubuntu service and see the files inside every node.

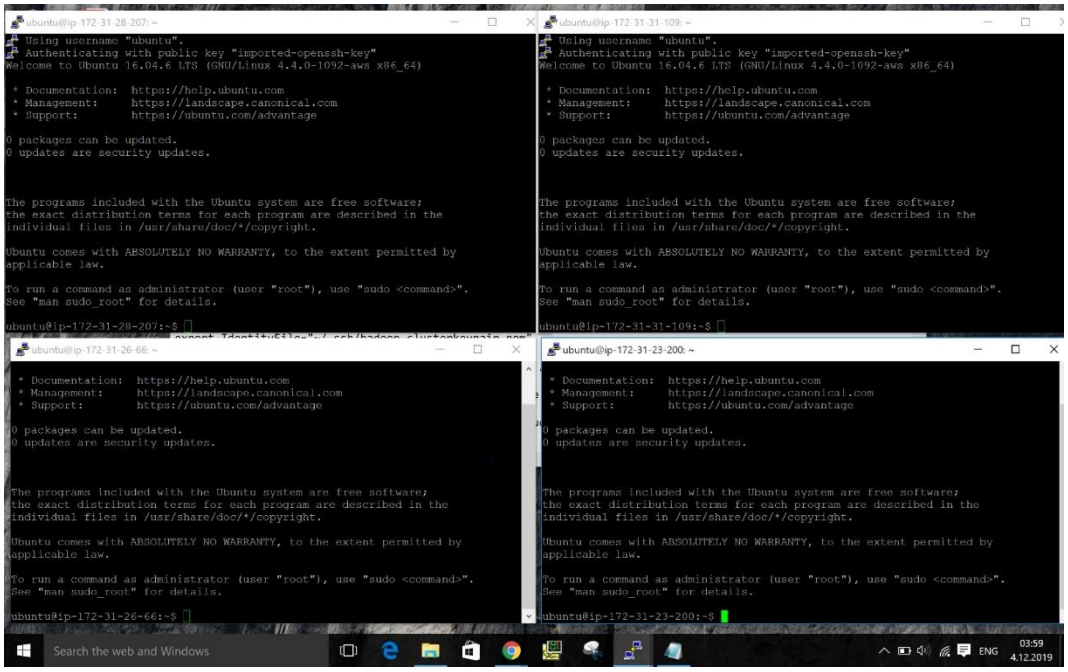

Figure 5. Terminal pages for every node

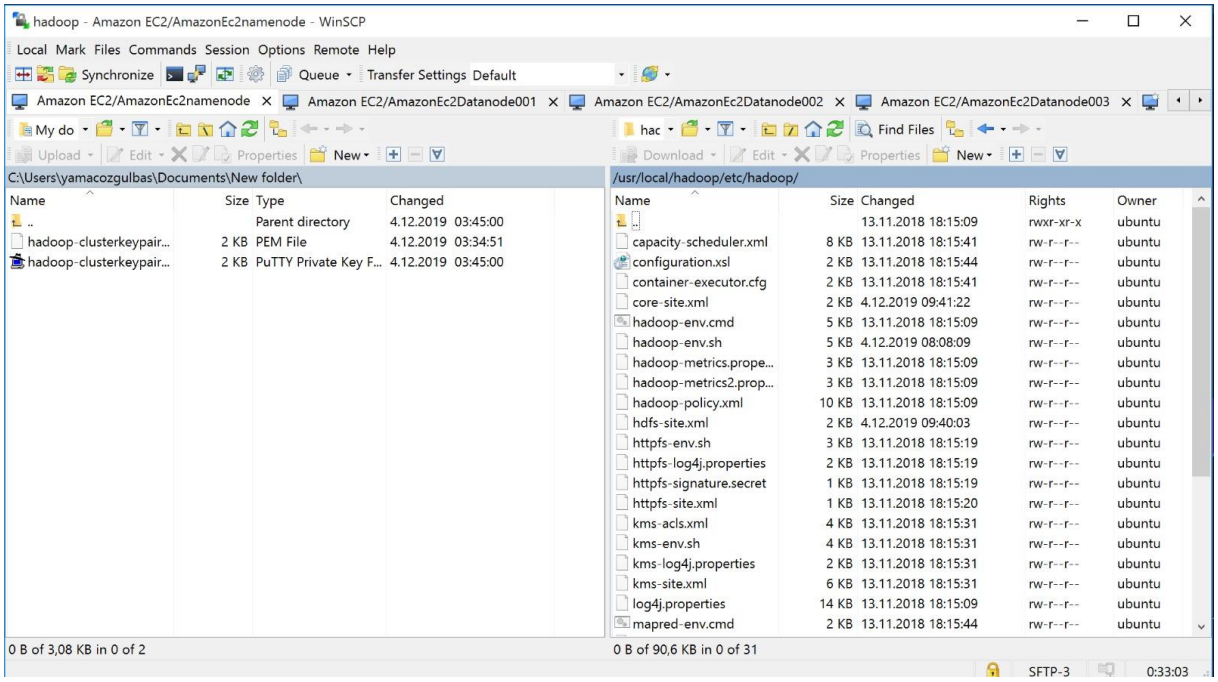

Figure 6. Displaying Ubuntu Service and the files inside the nodes

By using two PuTTY and WinSCP we have completed we have completed setting up AWS clustering. All information can be seen by going into the DNS addresses of nodes.

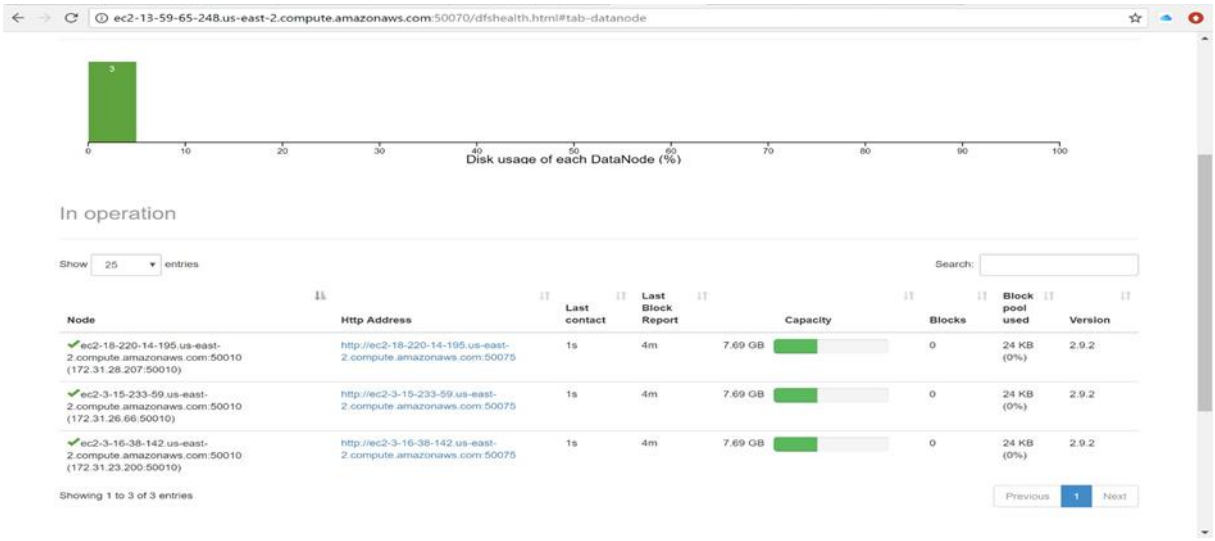

Figure 7. AWS that we have created and the information about three datanode

4.2 Dataset

For this project, two common clustering benchmarks were tested on using both regular PSO and KMPSO.

- Wisconsin Breast Cancer Diagnosis: A labelled dataset classifying breast cancer tumors as malignant or benign with 30 features
- Iris: A labelled dataset classifying species of iris flowers, with 3 classes and 4 features.

5 Results

The results obtained from running the algorithms were quite encouraging. Firstly, on both accounts the KMPSO algorithm performed better in term of F-measure. Additionally, the F-measure obtained for the Wisconsin breast cancer data set with KMPSO is identical to the test done by Sherar and Zulkernine [1]. This give us hope in our implementation of KMPSO. The results clearly demonstrate the superiority of the hybrid KMPSO over regular PSO. We can thus infer that hybrid algorithms can perform better than the regular metaheuristics.

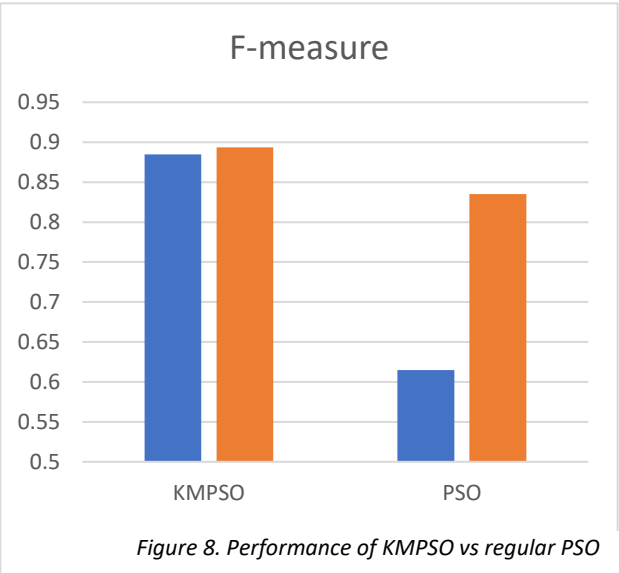

Figure 8. Performance of KMPSO vs regular PSO

6 Conclusion and Future Work

Swarm intelligence algorithms, particularly PSO and its variation KMPSO, are premier tools in big data analysis. These nature inspired algorithms help us to overcome the limitations of computation and perform operations on large datasets. In this project, KMPSO was successfully implemented and tested with promising results. Both datasets yielded high accuracy and f-measure scores. PSO is already a widely used algorithm and the success of its variation, KMPSO, further strengthen its reliability as a metaheuristic tool.

While the work done so far has been satisfactory, there still remains much room for improvement. Our future priority is to expand the scope of our work. The immediate goal is to implement KMCRO and compare the performance metrics of KMCRO against KMPSO using the same datasets. Furthermore, it is also necessary to use much larger and complex datasets for evaluating these algorithms. One pitfall of many swarm intelligence algorithm, including PSO, is that they are often unable to properly cluster skewed datasets. It must be tested whether KMPSO can overcome this weakness and whether CRO and KMCRO are susceptible to this as well. And finally, we want to extend KMPSO and KMCRO to apply to streaming data. Apache Spark has a powerful streaming capability called Spark Streaming. Most of today's data processing requirements include streaming data. The ability to effectively cluster and label steaming data, such as tweets, news or error logs, will be very useful in many scenarios.

7 References

[1] M. Sherar and F. Zulkernine, "Particle swarm optimization for large-scale clustering on apache spark," *IEEE Symposium Series on Computational Intelligence,* pp. 1-8, 2017.

[2] P. S. Rao and H. Banka, "Novel chemical reaction optimization based unequal clustering," *Wireless Networks,* vol. 23, no. 3, pp. 759-778, 2017.

[3] A. Y. Lam and V. O. Li, "Chemical reaction optimization: A tutorial," *Memetic Computing,* vol. 4, no. 1, pp. 3-17, 2012.

[4] Y. Wang and Q. Qian, "A Spark-Based Artificial Bee Colony Algorithm for Large-Scale Data Clustering," *IEEE 20th International Conference on High Performance Computing and Communications; IEEE 16th International Conference on Smart City; IEEE 4th International Conference on Data Science and Systems (HPCC/SmartCity/DSS),* pp. 1213-1218, 2018.