

# A Machine Learning based Framework for KPI Maximization in Emerging Networks using Mobility Parameters

Joel Shodamola, Usama Masood, Marvin Manalastas and Ali Imran

AI4Networks Research Center, Dept. of Electrical & Computer Engineering, University of Oklahoma, USA

Email: {joelshodams, usama.masood, marvin, ali.imran}@ou.edu

**Abstract**—Current LTE network is faced with a plethora of Configuration and Optimization Parameters (COPs), both hard and soft, that are adjusted manually to manage the network and provide better Quality of Experience (QoE). With 5G in view, the number of these COPs are expected to reach 2000 per site, making their manual tuning for finding the optimal combination of these parameters, an impossible feat. Alongside these thousands of COPs is the anticipated network densification in emerging networks which exacerbates the burden of the network operators in managing and optimizing the network. Hence, we propose a machine learning-based framework combined with a heuristic technique to discover the optimal combination of two pertinent COPs used in mobility, Cell Individual Offset (CIO) and Handover Margin (HOM), that maximizes a specific Key Performance Indicator (KPI) such as mean Signal to Interference and Noise Ratio (SINR) of all the connected users. The first part of the framework leverages the power of machine learning to predict the KPI of interest given several different combinations of CIO and HOM. The resulting predictions are then fed into Genetic Algorithm (GA) which searches for the best combination of the two mentioned parameters that yield the maximum mean SINR for all users. Performance of the framework is also evaluated using several machine learning techniques, with CatBoost algorithm yielding the best prediction performance. Meanwhile, GA is able to reveal the optimal parameter setting combination more efficiently and with three orders of magnitude faster convergence time in comparison to brute force approach.

**Index Terms**—Machine Learning, SON, Genetic Algorithm, Optimization, Cell Individual Offset (CIO), Handover Margin (HOM).

## I. INTRODUCTION

Since its advent, the wireless network has undergone several phases of technological advancement. From 2G technology which mainly supports voice communication to 4G which brought us high speed internet and now to upcoming 5G which promises ultra-fast connection speed and massive capacity. However, alongside this development is the increase in intricacy of the cellular network. As a rule of thumb, the more services a network supports, the more complex the functionalities and configuration parameters of the network become. This rise in complexity can be mirrored by the fact that the number of COPs per site has risen steadily from 500/site for 2G to 2000/site for 5G.

These COPs directly impact the networks performance and are usually measured through Key Performance Indicators (KPIs). COPs that are sub-optimally tuned usually lead to

poor network performance and degraded KPIs which ultimately result to unsatisfactory user QoE. That is why it is critical to make sure that these parameters are correctly set and adjusted. Current industry practice is to tune these parameters manually based on domain knowledge, intuition and sometimes hit and trial approach. However, manual configuration of these numerous parameters can be time consuming, inefficient and expensive on the side of the operators. As a result, mobile operators tend to deal on tweaking a limited number of COPs to improve certain KPIs, therefore limiting the possibility of deriving the utmost performance that a network can achieve. In addition, the expected densification in emerging network will add to this problem which will make optimal parameter setting discovery unfathomable.

In order to tune these parameters properly, it is essential to learn how KPIs behave with the variations in COPs. However, as operators try very little combinations of configuration parameters in real networks, it is very difficult to interpolate and learn the behavior of the system performance due to changes in COPs. For this reason, the task of modeling COP-KPI relation is almost impossible using conventional interpolation techniques. This is where Machine Learning (ML) comes into play. With machine learning, it is possible to model and map out functions that cannot be directly or mathematically interpreted in the data [1], [2]. This capability makes machine learning a promising tool to accurately capture the network dynamics due to changes in COPs even with very little and sparse experiment data.

To overcome the current challenges of learning COP-KPI relations and finding optimal Configuration and Optimization Parameter (COP) combination to maximize KPIs, we have proposed a framework that involves: first, leveraging the power of machine learning to learn and predict the network performance given a set of COP using data gathered from a realistic industry-grade simulator and second, finding the optimal COP combination values that will yield the maximum KPI using a heuristic search technique in the form of GA. In this paper, two pertinent COPs related to mobility, CIO and HOM are chosen, with mean SINR as the optimized KPI.

## A. Relevant Work

3GPP [3] introduced the CIO parameter as an LTE standard that controls handover and ensures proper load balancing from one cell to another. Since then, CIO has been a parameter of interest to the research community to find out the optimal values of CIO that lead to a desired KPI. However, sole optimization of CIO also generated further challenges in cell management resulting to a drop in SINR [4]. It is therefore pertinent that CIO is combined with other mobility parameters like hysteresis and Time-to-Trigger (TTT) to taper off mobility related problems such as handover failure cases like too early, too late and handover to wrong cells.

Authors in [5] observe the effect of HOM and TTT on call drop ratio (CDR) and handover ratio and conclude that the effect of TTT is inconsequential. However, the underlying task is not only to discover what particular set of parameters has more influence on a desired KPI but the set of values of these parameters that produce the optimal KPI. Authors in [6] determine optimal values of TTT and CIO from initial suboptimal configurations based on weighting factors with respect to handover failure cases. In [7], authors discuss the optimal values of HOM with respect to user mobility and handover failure rates while authors in [8] use TTT and HOM to address the same issue. However, none of these aforementioned studies leveraged on machine learning to derive optimal values of parameters.

Most of the relevant studies either used machine learning (ML) techniques to tune hard parameters or used a single ML algorithm for KPI optimization. Authors in [9] used ML to optimize the same KPI in our study but with reference to hard parameters like antenna azimuth. Although authors in [10] used machine learning, support vector machine (SVM) in particular, they tuned one handover parameter, CIO to enhance user throughput. In [11], authors use GA without any application from ML to determine optimal values of just HOM to attain reduction in power consumption.

In this paper, ML is utilized to capture the KPI behavior with changes in COPs before applying GA to maximize the objective function. In that light, the main contributions of this paper are:

- A novel combination of mobility parameters consisting of CIO and HOM as COPs for COP-KPI modeling using machine learning
- Performance comparison of different ML algorithms for COP-KPI modeling
- Use of genetic algorithm on the outputs of ML model for faster convergence to achieve optimal COP combination

## B. Parameters Definition

- 1) Cell Individual Offset (CIO): This is a measurement quantity used to determine cell association, regulate cell coverage and usually incorporated in power measurement to control handover. As shown in Fig. 1, handover can be made earlier (or later) by changing the CIO values. CIO is

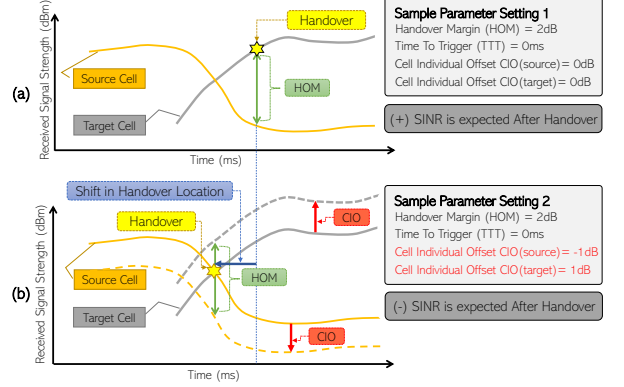


Fig. 1. Effect of CIO on Handover and SINR

usually given positive or negative values; where a positive value augments the RSRP of the cell and a negative value diminishes the RSRP of intended cells. Fig. 1 also shows the effect of suboptimal CIO tuning in terms of SINR. As the figure shows, wrongly tuned CIO can cause too early handover where the source cell is still better than the target cell resulting to high interference and thus poor SINR.

- 2) Handover Margin (HOM): HOM is a mobility related parameter that administers handover. As shown in Fig. 1(a) and(b), it is the assigned threshold that must be met, when the power of a target cell begins to exceed the power of the serving cell. It is usually within the range of 0 to 10dB.
- 3) Capacity: We can define cell capacity in terms of SINR. SINR is mostly used for radio link quality and throughput measurements. With throughput being the maximum data rate within a channel and from shannon capacity in equation (2), increase in SINR will lead to a proportional increase in capacity. [4] described this relationship in expression (2). The mean SINR of a user associated with a cell  $s$  for a given time slot  $t$  is:

$$\gamma_x^s = \frac{P_t^s d_x^s}{\sum_1^i w_i P_t^i d_x^i + N_p} \quad (1)$$

$$Capacity = N_s - \frac{1}{w_i} \sum_X \frac{\tau_x}{\log_2(1 + \gamma_x^s)} \quad (2)$$

where  $\gamma_x^s$  is the average SINR of user  $x$  and cell  $s$ , with  $P_t^s$  and  $P_t^i$  as transmission powers of cell  $s$  and interfering cell  $i$ ,  $d_x^s$  and  $d_x^i$  as the pathloss components of associated cell  $x$  and interfering cells  $i$  respectively.  $N$  is the noise,  $w_i$  the load of the interfering cell and  $N_s$  the total physical resource blocks (PRB).

The rest of this paper is organized as follows: Section II highlights the proposed framework, Section III discusses the simulation setup. In Section IV, results and analysis are presented and finally Section V concludes the paper.

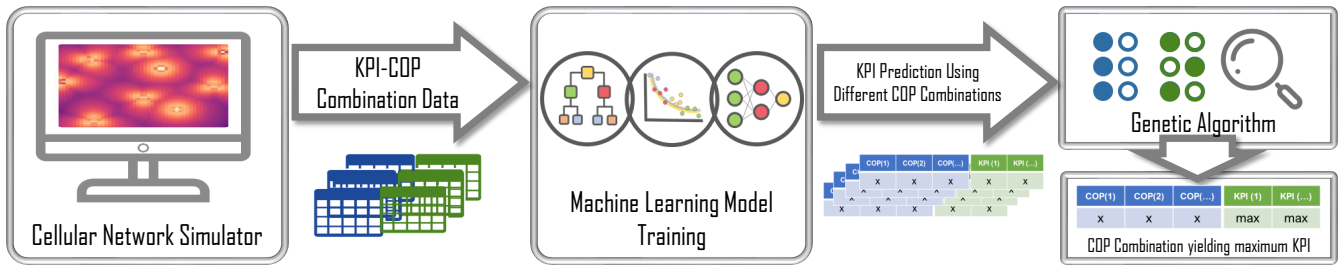


Fig. 2. Proposed Framework for KPI Maximization

## II. SYSTEM FRAMEWORK

As shown in Fig. 2, the proposed system framework takes into consideration three processes to achieve the objective function. First is the data generation part. For this, an industry grade system level simulator is used. Simulators have the advantage of generating large amount of data such as COP-KPI relations which are otherwise impossible to gather from a real network. The second part of the framework is the KPI prediction using machine learning techniques. Here, COP combination and KPI data gathered from the simulator are used to train the machine learning models and learn how changes in COP combinations affect the KPIs. This prediction model enables network operators to envisage network behaviours and extrapolate important information. The last part of the framework involves discovery of the optimal parameter values using genetic algorithm. In this stage, prediction model of best performing machine learning model is used to initialize parameters. Detailed discussion of the data generation, machine learning techniques used, and how genetic algorithm is leveraged is shown below.

### A. Data Generation

In order to learn how the cellular network behaves with changes in COP values, one approach is to try several combinations of these parameters and then observe the resulting KPI trends. However, this approach is not practical especially in live networks as there are hundreds of thousands of parameter combinations possible. Aside from that, one badly tuned parameter might impose a huge risk in the network's performance. For this reason, simulators are a viable alternative to gather data which cannot be done on a live network due to the above-mentioned restrictions.

We wrote a script to automate the data generation process. This script automatically changes the values of COPs and records the resulting KPI for each combination. The output of the simulator coupled with the automation script is a table of different CIO and HOM combinations with the expected SINR for each combination. The data is utilized to train several machine learning models and also to predict the average SINR for each CIO-HOM combination. Models are trained in two types of training data, one using all possible combinations (100 percent) and one using only (10 percent) of the training data.

This is done to test the robustness of the machine learning model to sparse data which is usually the case in a real cellular network.

### B. Machine Learning Algorithms for KPI Prediction

Machine Learning can be categorized as supervised and unsupervised learning. The former requires a set of training data to learn from and maps to a corresponding output based on observed relationship from the trained data while the latter takes unlabelled data and understands its distribution to map into several labels. Supervised learning can be further categorised as classification and regression problems. The linearity or non-linearity of these relationship are usually represented and this differs from model to model. We apply various ML algorithms and observe their prediction performance. In this paper, five machine learning models are evaluated.

1) *Linear Regression:* This method is an example of a solution to a regression based problem which describes the functional relationship between dependent ( $y_i$ ) and independent attributes ( $x_i$ ) by ascribing weights,  $w$  to the function through a method of linearity as shown in equation (4). The target attribute (dependent variable), is extracted from a set of  $N$  input features with coordinates  $(x_i, y_i)_{i=1}^N$  and  $f(x_i)$  being a function of the weights  $\phi(x_i, w)$  [12]. This can be summarized in equation (3) where  $\epsilon$  is the error.

$$y_i = f(x_i) + \epsilon \quad (3)$$

$$y_{w,b} = wx + b \quad (4)$$

2) *K-Nearest Neighbor (KNN):* KNN makes no assumption for distribution of vectors by using distance calculation measures to categorize a set of input features in vector space based on a variable,  $k$ , which symbolizes the number of neighbors or surrounding features. Essentially, as batches of new independent features are introduced, it locates the  $k$  neighbors closest to the new features and ascribes them to a target majority. The major distance metrics are cosine similarity and euclidean distance.

3) *Extreme Gradient Boosting (XGBoost):* This is an instance of a category of gradient boosting called ensemble learning which aggregates multiple weak models (trees), iterates through them by ascribing weights to each model to give a final

model. Rather than some random selection, each subsequent model is a correction of the previous. Two objective parameters are introduced here namely training loss and regularization function as shown in equation (5).

$$obj(\theta) = L(\theta) + \omega(\theta). \quad (5)$$

Here  $L$  represents training loss and  $\omega$ , regularization. XGBoost adds some extra functions to reduce computational complexity and improve on scalability and prediction accuracy.

4) *Categorical Boosting (CatBoost)*: This is another category of gradient boosting which falls under ensemble methods. CatBoost was developed to give a better performance than its contemporaries like XGBoost and LightGBM in terms of efficiency, reducing the time wasted on tuning of parameters through excellent default functions and reduces overfitting on training samples. It uses an ordered mechanism to get training samples per time.

5) *Deep Neural Network (DNN)*: DNN is a class of Artificial Neural Network (ANN) inspired by the biological brain that consists of hidden layers between input and output, represented by multi-perceptrons to discover hidden features and layers of inputs. Each hidden layer is supported by activation functions among which are TanH, Sigmoid and ReLU depending on the nature of the problem. The outputs also are supported by optimizers namely Stochastic Gradient Descent (SGD), AdaDelta, AdaGrad and Adam with the latter being the most efficient. DNN is able to learn intricate characteristics of input vectors and categorize them more efficiently.

### C. Genetic Algorithm

To show the intrinsic nature of the relationship between objective function and parameters, a plot of SINR as a function of CIO and HOM for one base station (BS) is shown in Fig. 3. From this figure, the problem is seen as a non-convex function which can also be termed as a NP-hard optimization problem with several local maxima and a global maxima. We use genetic algorithm (GA) in conjunction with machine learning model as the heuristic optimization technique based on its induction from natural and biological evolution.

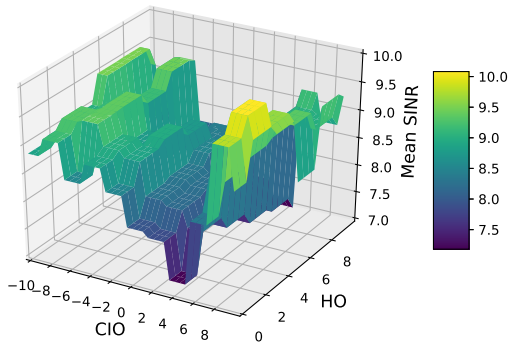


Fig. 3. Non-convexity of Objective Function.

Essentially, GA generates a population of a specified size that consists of several individuals (or solutions), each consists

---

### Algorithm 1 Genetic Algorithm for Objective Function

---

**Input:**

Parameters  $X(CIO_1, CIO_2, CIO_3, HOM_1, HOM_2, HOM_3)$   
 Machine learning model (**F**)  
 sample/iteration **P**  
 Maximum Generation **G**  
 Population **POP**  
 Crossover **Cr**  
 Mutation **M**

**Output:**

Solution  $X_{sol} = X(CIO_1, CIO_2, CIO_3, HOM_1, HOM_2, HOM_3)$   
 $\text{argMax}[X, f(X)]$

- 1: Randomly generate parameters from  $X$  with size **P**;
  - 2: Assign fitness function from **F** for each sample **P**;
  - 3: open an empty set **POP** and save **P** inside;
  - 4: **for**  $i=1$  to **G** **do**
  - 5:   evaluate **POP** for each **P** sample using **F**;
  - 6:   create an empty set called Parent **Par**;
  - 7:   select best elites from **POP** and save in **Par**;
  - 8:   **for** two best elites in **Par** **do**
  - 9:     perform crossover **Cr** (**Par**<sub>1</sub>, **Par**<sub>2</sub>);
  - 10:    get two children from **Cr** and mutate each;  
       **M**(**Cr**<sub>1</sub>, **Cr**<sub>2</sub>);
  - 11:    **end for**
  - 12:   Store each child solution and update in **POP**;
  - 13: **end for**
  - 14: **Return:** The last set of **POP** and that becomes  $X_{sol}$
- 

of variables that are randomly initialized. In our framework, these initial values are pulled from the ML prediction function. A criteria is set for maximum iteration, where for each iteration, solutions are generated and passed to the subsequent generation. The algorithm keeps iterating until it converges to an optimal solution. Each of the solutions in these generations are evaluated by a fitness function derived from the machine learning model to select elites considered as the best solutions or parents that are further iterated for crossover and mutation. Algorithm 1 shows the pseudo code with the detailed break down on how this GA process occurs. The crossover method used was Simulated Binary crossover (SBX) which converts each values to binary values to initiate the single point crossovers. SBX incorporates a spread factor and makes a random selection from a probability distribution.

### III. SIMULATION SETUP

Using the ray-tracing based industry grade system-level simulator, we have created a cellular network scenario composed of 12 macro base stations with 3 sectors each. However, changes in parameters are concentrated only to 3 sectors as shown in Fig. 4. The rest of the base stations are used to add interference and thus capture realistic SINR. A total of 356 users are created which are randomly distributed around the simulation area. To make it more realistic, these users have



different speeds as well as different services used such as VoIP and high-speed internet. Table 1 shows the detailed parameter settings for the simulation. Each base station is equipped with two mobility parameters, CIO and HOM. For the base stations of interest, the value of CIO and HOM ranges from [-10 to 10] dB and [0 to 10] dB respectively. For the rest of the base stations, the value of these parameters remains constant. With CIO having 21 possible values, 11 for HOM and considering three BSs, we would have  $21^3 \times 11^3$  (12,326,391) possible COP combinations. However, we have observed that using all this possible combination is not necessary as SINR values change minutely. Therefore, we have decided to use CIO and HOM values with step size of 2 to reduce the data size, that is ( $11^3 \times 6^3$ ) possible combinations.

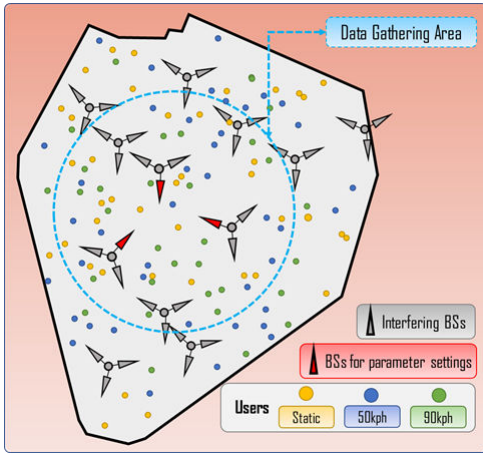


Fig. 4. Simulation Environment

In the simulator we defined the serving cell selection process with respect to 3GPP standard specification in terms of qualification, pre-selection and final selection.

1) *Qualification*: For cells to first be qualified as potential serving cells the RSRP (Reference Signal Received Power) received from the UE must be greater than or equal to the cells' minimum RSRP plus a threshold.

$$R_s^{Tx} \geq T_s^{Tx} + \text{Max}(0, T_{threshold}^{Tx}) \quad (6)$$

where  $R_s^{Tx}$  and  $T_s^{Tx}$  are the RSRP from the UE and cell's minimum RSRP respectively while  $T_{threshold}^{Tx}$  is the cell selection threshold.

2) *Pre-Selection*: Considering all the potential cells that qualify with the requirements above, only the cell that has the highest RSRP received by the UE is preselected as the serving cell ( $S_0$ ).

3) *Final Selection*: Aside from the pre-selected serving cell, any cell among the qualified cells which has the following condition is considered as the best serving cell, that is the highest RSRP plus Cell Individual Offset:

$$R^{Tx(c)} + O_{individual}^{Tx(c)} \geq R^{S_0} + O_{individual}^{S_0} + M_{HO}^{S_0} \quad (7)$$

Here  $R^{Tx(c)}$  and  $O_{individual}^{Tx(c)}$  are the received power and cell individual offset of the candidate or target cell respectively,

$O_{individual}^{S_0}$  is the cell individual offset of the serving cell,  $R^{S_0}$  is the received power of the serving cell and  $M_{HO}^{S_0}$  is the handover margin. The pre-selected cell is considered the best cell if candidate cell does not fulfill this criterion.

During the course of the simulation, the UEs will use the cell association process described above in deciding which cell to camp on. Each time the values of CIO and HOM changes for the 3 BS, the SINR from all the users camp on these BS as well as the users within the data gathering area are collected and averaged.

TABLE I  
PARAMETER SETTINGS FOR SIMULATION

System Parameters	Value
No. of Macro Base Stations	12
No. of sectors per Base station	3
No. of BS for parameter settings	3
Carrier Frequency	2100MHz
Transmission Power	43dBm
Minimum RSRP	-140dBm
Antenna Gain	18.5dBi
No. of users in simulation region	356
Path Loss model	Ray Tracing
CIO range	Max:10dB, Min:-10dB
Handover Margin range	Max: 10dB, Min 0dB

#### IV. RESULTS AND ANALYSIS

Machine learning prediction performances are evaluated in terms of Root Mean Square Error (RMSE) as the validation function. Results shown in Fig. 5 indicate that RMSE ranges from 1.1439 dB to 1.4742 dB which shows a good response to the ML models. As expected, Linear Regression performed the least because its limitations lies in matching data that is not linear and predicting data that is not within the range of training sets. CatBoost performs the best because it reduces any form of overfitting and categorizes each attribute accurately. However, it is worth noting that DNN will outperform its counterparts if the entire 12 million plus data set was used. Another interesting thing to note is the good performance of the ML models even with 10 percent of training data. The average RMSE of all model is 1.2994 dB showing that even with limited data, ML predictions are still able to perform with good level of accuracy.

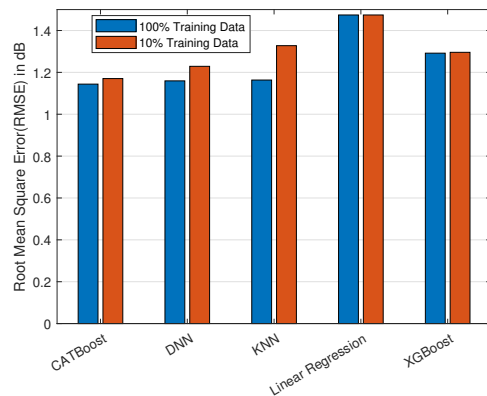


Fig. 5. Evaluation of Different ML Techniques on SINR Prediction.

## CONCLUSION

In this paper, we present a machine learning based framework to capture KPI behavior with changes in COPs and to find the optimal combination of these COPs to maximize the KPI. Using several machine learning techniques, we have predicted the mean SINR of all users with different combinations of CIO and HOM. Analysis shows that CatBoost performs the best among all other techniques evaluated. Meanwhile, GA demonstrates efficiency in finding the optimal CIO-HOM combination that yields maximum mean SINR.

For future studies, we will consider the optimization of more COPs with respect to multiple KPIs in a larger network scale involving more base stations.

## ACKNOWLEDGMENT

This material is based upon work supported by the National Science Foundation under Grant Numbers 1619346, 1559483, 1718956 and 1730650. For more enquires about these projects please visit [www.AI4networks.com](http://www.AI4networks.com)

## REFERENCES

- [1] P. V. Klaine, M. A. Imran, O. Onireti, and R. D. Souza, "A survey of machine learning techniques applied to self-organizing cellular networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2392–2431, 2017.
- [2] U. Masood, H. Farooq, and A. Imran, "A machine learning based 3d propagation model for intelligent future cellular networks," in *2019 IEEE Global Communications Conference (GLOBECOM)*, pp. 1–6, IEEE, 2019.
- [3] G. T. 36.331., "Evolved universal terrestrial radio access (e-utra); radio resource control (rrc); protocol specification," 2011.
- [4] A. Asghar, H. Farooq, and A. Imran, "Concurrent optimization of coverage, capacity, and load balance in hetnets through soft and hard cell association parameters," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8781–8795, 2018.
- [5] P. Muñoz, R. Barco, and I. de la Bandera, "On the potential of handover parameter optimization for self-organizing networks," *IEEE Transactions on Vehicular Technology*, vol. 62, no. 5, pp. 1895–1905, 2013.
- [6] L. Ewe and H. Bakker, "Base station distributed handover optimization in lte self-organizing networks," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 243–247, IEEE, 2011.
- [7] K. Kitagawa, T. Komine, T. Yamamoto, and S. Konishi, "A handover optimization algorithm with mobility robustness for lte systems," in *2011 IEEE 22nd International Symposium on Personal, Indoor and Mobile Radio Communications*, pp. 1647–1651, IEEE, 2011.
- [8] R. D. Hegazy and O. A. Nasr, "A user behavior based handover optimization algorithm for lte networks," in *2015 IEEE Wireless Communications and Networking Conference (WCNC)*, pp. 1255–1260, IEEE, 2015.
- [9] Z. Lin, Y. Ouyang, L. Su, W. Lu, and Z. Li, "A machine learning assisted method of coverage and capacity optimization (cco) in 4g lte self organizing networks (son)," in *2019 Wireless Telecommunications Symposium (WTS)*, pp. 1–9, IEEE, 2019.
- [10] A. Roshdy, A. Gaber, F. Hantera, and M. ElSebai, "Mobility load balancing using machine learning with case study in live network," in *2018 International Conference on Innovative Trends in Computer Engineering (ITCE)*, pp. 145–150, IEEE, 2018.
- [11] R. F. Ahmed, T. Ismail, L. F. Abdelal, and N. H. Sweilam, "Optimization of power consumption and handover margin of sleep/active cells in dynamic h-cran," in *2018 11th International Symposium on Communication Systems, Networks & Digital Signal Processing (CSNDSP)*, pp. 1–6, IEEE, 2018.
- [12] A. Burkov, "The hundred-page learning book," 2019.

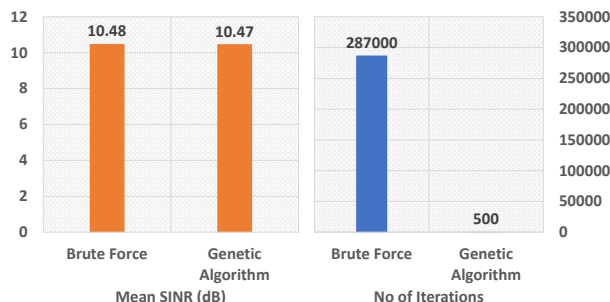


Fig. 6. Optimization Techniques and Evaluation

In finding the optimal value of CIO-HOM combination that maximizes SINR, we have evaluated two search techniques, brute force and genetic algorithm. As shown in Fig. 6, both techniques gave close values of maximum SINR. However, it takes brute force 287,000 iterations, which is the same number of samples, to get to an optimum solution. This shows that for large data size, brute force method is computationally inefficient considering the time it takes to search through the entire data set. Meanwhile, GA takes considerably less amount of iterations, about 500, to find the optimal solution. This fast convergence of GA shows how it can be used efficiently on these kind of problems. GA shows that for the three BSs, CIO values of [-10, -8, 4] dB combined with HOM values of [10, 5, 9] dB will yield the maximum SINR.

Results in Fig. 7 show the efficiency of GA. Based from the results, initial configuration of [0, 0, 0] dB for CIO and [0, 0, 0] dB for HOM for the three BSs give SINR of 7.96 dB. However, after 1 generation, which is equivalent to 100 iterations, SINR of 9.95 dB is achieved while after 300 iterations, it almost found the optimal combination. This shows that even with less amount of iterations, GA can find SINR values much faster than brute force approach. This efficiency depends on the generation and population size which involves another optimization operation that is out of scope of this work.

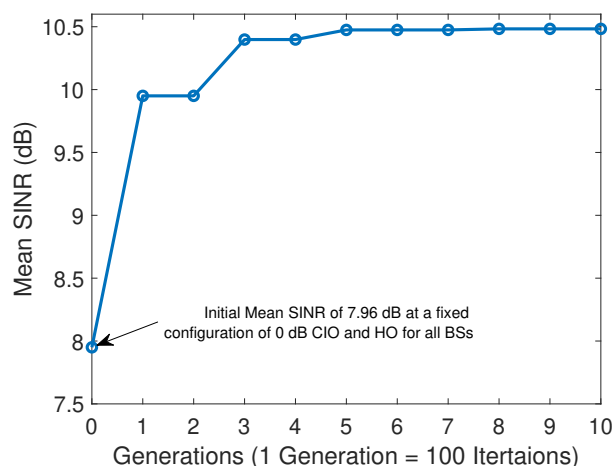


Fig. 7. Maximum mean SINR vs. Iteration for Genetic Algorithm