

Application of a machine learning algorithm (XGBoost) to offline RHIC luminosity optimization

X. Gu

April 2021

Collider Accelerator Department
Brookhaven National Laboratory

U.S. Department of Energy

USDOE Office of Science (SC), Nuclear Physics (NP) (SC-26)

Notice: This technical note has been authored by employees of Brookhaven Science Associates, LLC under Contract No. DE-SC0012704 with the U.S. Department of Energy. The publisher by accepting the technical note for publication acknowledges that the United States Government retains a non-exclusive, paid-up, irrevocable, world-wide license to publish or reproduce the published form of this technical note, or allow others to do so, for United States Government purposes.

DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or any third party's use or the results of such use of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof or its contractors or subcontractors. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.

Application of a machine learning algorithm (XGBoost) to offline RHIC luminosity optimization

X. Gu, A. Fedotov, D. Kayran

April 9, 2021

The operation parameter optimization in 2020 RHIC low energy run is difficult. First, the RHIC luminosity is affected by many RHIC operation parameters, as well as affected by many Low Energy RHIC electron Cooling (LEReC) operation parameters. Second, the luminosity signal in this run is noisy and not sensitive to these parameter changes, especially when these parameters are very close to their optimized values. It is not easy to distinguish the effects of one parameter from all other operation parameters separately. Therefore, it is difficult to optimize the luminosity by varying these parameters one by one. To find a way for luminosity optimization, we analyze some operation parameters via a machine learning algorithm - XGBoost. After constructing a black-box surrogate model from XGBoost and plotting their partial dependency plots (PDF) and SHAP value plots for different operation parameters, we can find the effects of an individual parameter on the RHIC luminosity and optimize it accordingly.

1. Motivation

For a circular collider like RHIC, we can change some machine parameters and beam parameters to maximize its luminosity. The luminosity formula for round Gaussian and equal beams at the interaction point (IP), as is the case in RHIC, can be expressed as

$$L = n_b \frac{f_c N^2}{4\pi\sigma^2} H \quad (1)$$

where n_b is the collision bunch number, f_c is the collision frequency, σ is the transverse RMS beam size at the IP, and N is the particle number per bunch. H is a geometric factor that accounts for the hourglass and crossing angle effect. Therefore, we can optimize the luminosity according to the transverse ion beam size, the bunch length, the ion beam intensity, and the revolution frequency.

During RHIC operation, several machine parameters can affect the ion beam size and the bunch length, such as the beta*, the RF voltage, and the injection

emittance. The machine tune, the chromaticity, the collimator position, and the RF voltage are also critical for a higher integrated luminosity. We can optimize these parameters for a longer beam lifetime by minimizing the beam loss, which can be caused by all kinds of effects.

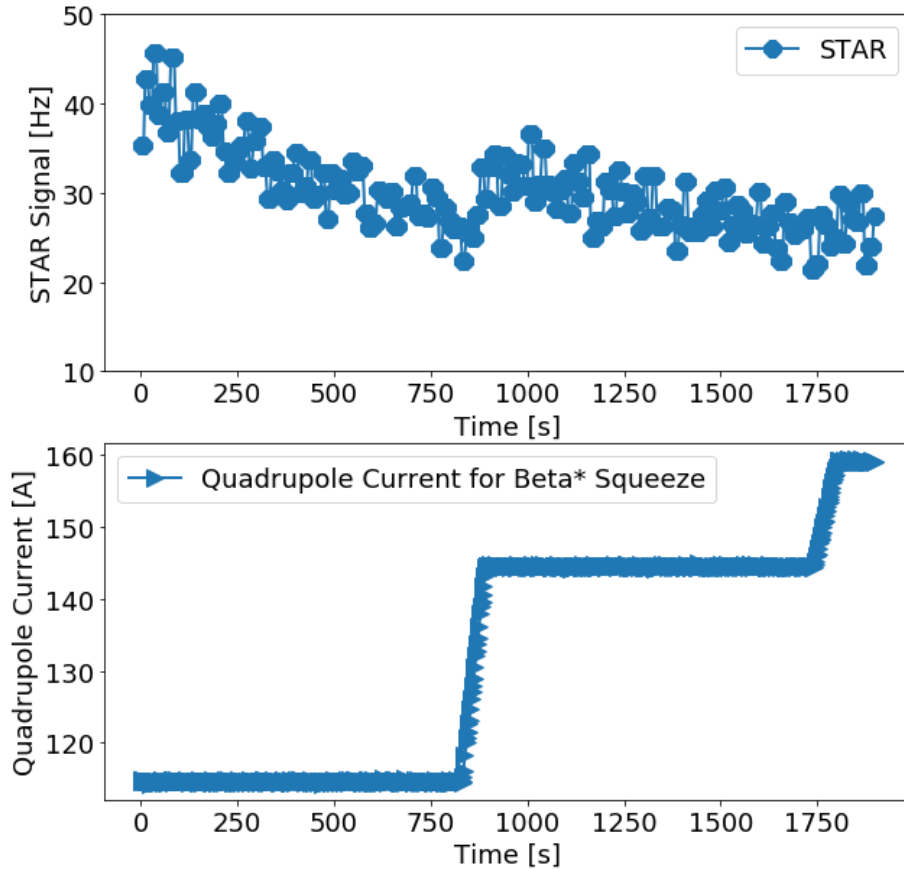


Fig. 1: The luminosity and Q1 quadrupole current for beta* squeeze.

In the 2020 RHIC low energy run, electron cooler LEReC operated to improve the integrated luminosity. The parameters of the LEReC electron accelerator can affect the cooling of ion bunches and the luminosity. These parameters are the electron beam current, the electron energy, the current of solenoidal magnets in the cooling section (which provides focusing for the electron beam), and the beam position (transverse alignment of electrons with respect to the ion beam).

All the above RHIC and LEReC operation parameters can affect the luminosity. The effects of some parameters are straightforward, such as the ion beam intensity, the injection beam emittance, the machine beta*, and the revolution

frequency. On the other hand, the impact of other parameters is less straightforward or indirect. These parameters are the LEReC magnets current, the electron beam current, the electron beam energy, the RHIC machine tune, the chromaticity, the collimator position, and others.

It is hard to differentiate the impact of these effects on the luminosity. They could also change simultaneously store by store, for example, the ion injection intensity and the ion beam emittance, the electron beam position, and the electron energy. Furthermore, the luminosity signal is noisy in the 2020 RHIC low energy run. Fig. 1 shows the luminosity signal as well as a quadrupole magnet current for the beta* squeeze. The luminosity signal is changed from 35 Hz to 45 Hz within several seconds. Operationally, it becomes impractical to optimize these parameters one by one according to the luminosity. This optimization is more impractical when we have moved some parameters very close to their optimized values and would like to do some fine optimizations.

To address the above issues, we can use a machine-learning algorithm. Machine-learning is a powerful tool for solving complex problems that can not be defined by some clear rules or equations. Machine-learning is more useful if some optimization problems have too many input parameters, or they correlate in an unknown way. Therefore, we implemented a machine-learning algorithm (XGBoost) [1] [2] to the 2020 RHIC and LEReC operation data analysis. After acquiring some data for these parameters and constructing a model from XGBoost, the luminosity as a function of an individual parameter can be plotted separately for different parameters. Then we can find some optimized operation parameter values for a higher integrated luminosity.

2. Operation Data and Pre-Processing

2.1. Data Acquisition

For a machine-learning project, the data is the first, and the machine-learning algorithm is the second. It is critical and essential to get more data with a wide variety and high quality. Therefore, some RHIC and LEReC operation parameter data is acquired before constructing an XGBoost model. Table 1 lists these parameters and Table 2 lists the acquired data.

The store length is about 2400 seconds for a nominal store in the RHIC 2020 low energy run. To have more data sets or to include some early aborted store data, these data are obtained within the first 1900 seconds of a nominal physics store. In total, we used 791 useful stores (or data sets, measurements, samples, examples) with 24 inputs (or variables, parameters, features, attributes).

The ion beam sizes (emittance) are from the H-jet emittance measurement without further calculation. We averaged them for the first 20 seconds. We also averaged the blue and the yellow ion beam intensity for the first 10 seconds. If there was a beta* squeeze during a store, we ramp up the currents of some

Table 1: Parameters and Their Abbreviations

RHIC		LReC	
Parameters	Abbreviations	Parameters	Abbreviations
Intensity B	IntenB	Electron BPM B Cooling	ebpmB
Intensity Y	IntenY	Electron BPM Y Cooling	ebpmY
Emittance B	SizeB	Ion BPM B	ibpmB
Emittance Y	SizeY	Ion BPM Y	ibpmY
Tune B H	TuneBH	Solenoid 1 B	Bsol1
Tune B V	TuneBV	Solenoid 1 B	Bsol1
Tune Y H	TuneYH	Electron Beam Current	Current
Tune Y V	TuneYV	Electron Beam Energy	Energy
Chrom B H	ChromBH	B Y Quadrupole Current	BYquad
Chrom B V	ChromBV		
Chrom Y H	ChromYH		
Chrom Y V	ChromYV		
Collimator BH	CollBH		
Collimator BV	CollBV		
Collimator YH	CollYH		
Collimator YV	CollYV		
Beta* Squeeze Ramp	Ramp		
Luminosity	Lumi		

quadrupole magnets. Other parameters are all the average value of the total 1900 second store data.

There are eight beam position monitors (BPMs, 704 MHz) in both the blue and the yellow LReC cooling sections for the horizontal and the vertical planes. The BPM data in Table 1 are the average of these 16 BPMs readings.

We excluded some parameters from constructing an XGBoos model. These parameters are the ion beam loss rate (or ion beam decay) and the electron beam angles. First, they are not the controllable parameters that we can directly tune during our operation. Second, they have fewer effects on the machine learning model, and they have some correlations with other parameters that will affect later analysis. For many machine learning algorithms, we avoid using the correlated input parameters. We sometimes can't interpret a machine learning model correctly because of this.

2.2. Parameter Correlations

To check the correlation between all input parameters before implementing a machine learning algorithm, we can use the efficiency of correlation to evaluate

Table 2: The acquired data sets used in XGBoost. The store fill number is from 27341 to 28785.

	ibpmB	ibpmY	ebpmB	ebpmY	Current	Energy	TuneBH	TuneBV	TuneYH	TuneYV	...	CollBV	CollYH
0	74.69	156.60	292.76	105.77	18.07	2.05	-0.84	-1.24	-0.80	-1.20	...	53,077.00	53,560.23
1	100.30	148.61	412.65	147.46	18.08	2.05	-0.84	-1.24	-0.80	-1.20	...	53,077.00	53,567.89
2	80.27	147.55	346.11	79.46	18.10	2.05	-0.84	-1.24	-0.80	-1.20	...	53,077.00	52,243.90
3	108.98	147.82	420.70	103.77	18.15	2.05	-0.84	-1.24	-0.80	-1.20	...	53,077.00	53,529.17
4	82.45	142.04	374.23	131.54	18.23	2.05	-0.84	-1.24	-0.80	-1.20	...	53,077.00	53,564.48
...
786	66.21	81.83	126.15	66.04	16.93	2.05	-0.84	-1.26	-0.82	-1.18	...	56,069.15	55,637.70
787	79.62	86.64	141.64	75.94	16.93	2.05	-0.84	-1.26	-0.82	-1.18	...	56,079.01	55,667.98
788	74.77	81.33	124.42	75.59	16.93	2.05	-0.84	-1.26	-0.82	-1.18	...	56,065.25	55,646.96
789	76.34	84.84	136.14	87.43	16.93	2.05	-0.84	-1.26	-0.82	-1.18	...	56,087.38	55,622.72
790	66.29	96.22	121.77	77.83	16.93	2.05	-0.84	-1.26	-0.82	-1.18	...	56,083.35	55,664.86

791 rows × 24 columns

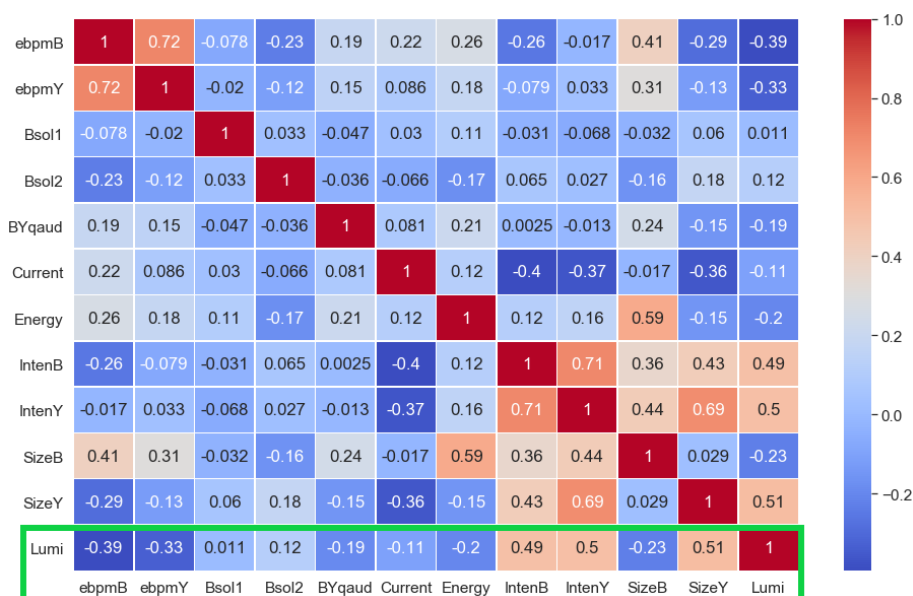


Fig. 2: The correlations between some variables.

the degree to which two input parameters are linearly related. When we explain a black box model results from some machine learning algorithms, to avoid an inaccurate interpretation, we should pay more attention to the input parameters with a high correlation coefficient between themselves.

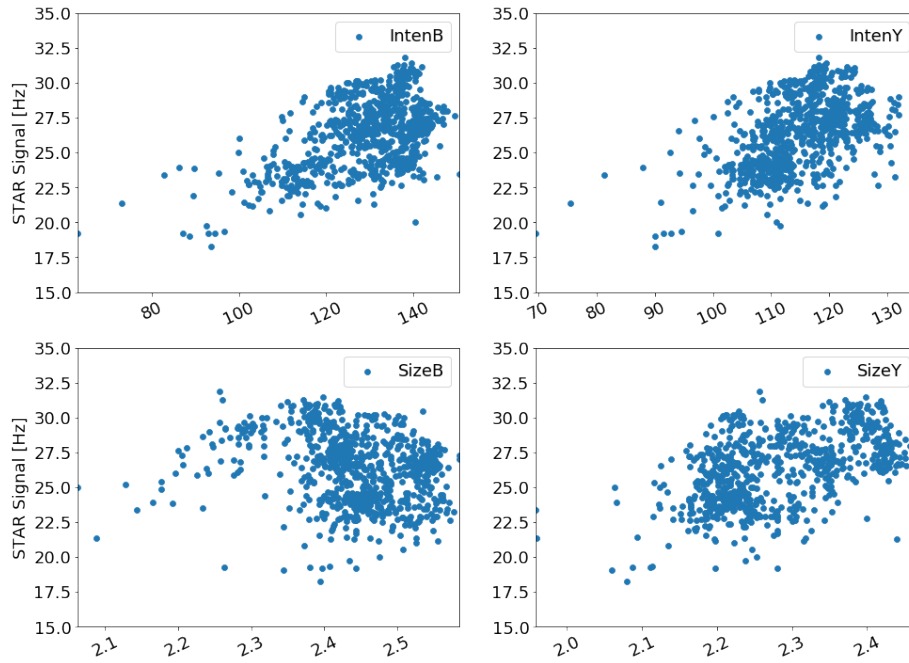


Fig. 3: The top plots are the luminosity scattering plot as a function of the blue and yellow ion beam intensity. The bottom figures are the luminosity scattering plots as a function of the blue and yellow ion beam size. The unit of the ion intensity is 1E9; the RMS beam size unit is μm (emittance unit)

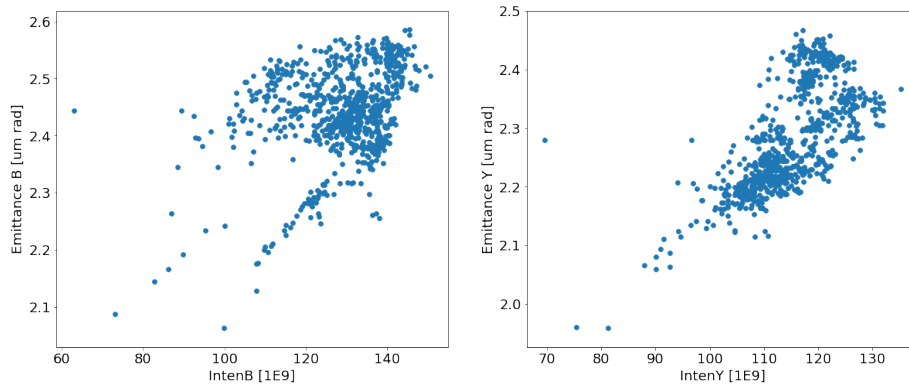


Fig. 4: The scattering plot shows the relationship between the ion beam intensity and the beam emittance or beam size.

The correlation coefficient equals the covariance of the two variables divided by the product of their standard deviations. We define the correlation coefficient as:

$$\text{corr}(X, Y) = \frac{\text{Cov}(X, Y)}{\sigma_X \sigma_Y} \quad (2)$$

while $\text{Cov}(X, Y) = E[(X - \mu_X)(Y - \mu_Y)]$ is the covariance of two random variables X and Y. And μ_X and μ_Y are their expected value. σ_X and σ_Y are their standard deviations. $\text{Cov}(X, Y) = 1$ means the two variables X and Y have a perfect positive linear relationship, and one variable increases with another variable. $\text{Cov}(X, Y) = -1$ also means X and Y have perfect linearity between them, but one variable decrease when another variable increase. The $\text{Cov}(X, Y) = 0$ means there is no correlation between them, and the change of one parameter doesn't affect another parameter at all.

Fig. 2 is the matrix with correction coefficient values that shows the correlations between some input parameters. The correction coefficient between the luminosity and other parameters show in the green box at the bottom of the figure. The IntenY and the IntenB have a positive 0.5 correction coefficient between the luminosity. That means that the luminosity will likely increase if the ion intensity increase. That is consistent with the theory result (Eq. 1).

The correction coefficient between the luminosity and the SizeY is abnormal, and it has a positive 0.51 instead of a negative value. Presently, we can explain it by the high correction coefficient of 0.69 between the IntenY and the SizeY. We will further explain it in the next section with a scattering plot between them.

Other parameters have less correlation with the luminosity, such as the electron beam current and energy. But this doesn't mean these parameters are not critical to the luminosity. If we have optimized the electron beam current

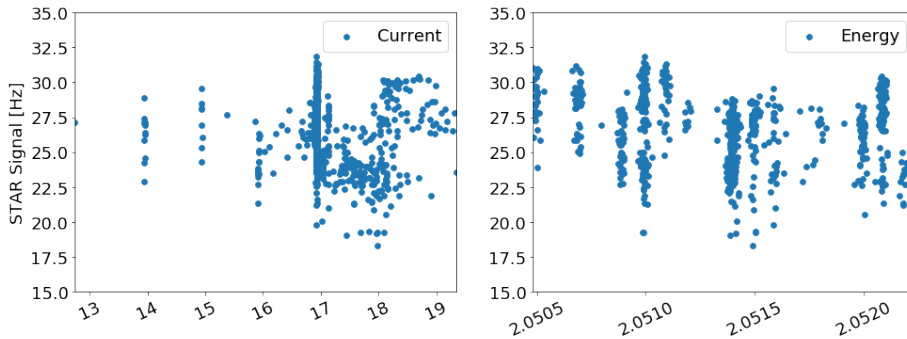


Fig. 5: The plots are the luminosity plots as a function of electron beam current and energy. The unit of the electron beam current mA, and the energy unit is MeV.

and the electron beam energy for cooling, and there are not many changes during the routine operation. Therefore their correlation coefficient with the luminosity is small.

In Fig. 2, there are some other highly correlated items. In LEReC, the electron beam passes through both the blue and the yellow cooling sections. The beam position changes upstream will affect the downstream beam position. Therefore, the correlation coefficient of the blue and the yellow beam position is 0.72. In RHIC, because both the blue and yellow injection ion beam intensity are affected by the same source (AGS), the correlation coefficient of IntenY and IntenB is 0.71.

Some correlations cannot be explained directly, such as the correlation between the blue ion beam size and the electron beam energy, as well as the correlation between the blue ion beam size and the blue electron beam position. Therefore, we can't use these correlation coefficients to determine whether one variable causes another one except there is a physical relationship between them.

2.3. Scattering Plots

To investigate the effects of some parameters, we plot the luminosity scattering plot as a function of these parameters. The top two plots in Fig. 3 are the relationship between the luminosity and the ion beam intensity. From these plots, one can find that a higher ion intensity tends to have a higher luminosity, which is predicted by Eq. 1. It is also consistent with the correlation results.

The bottom two plots are the relationship between the luminosity and the RMS beam size (or emittance). The plot of the SizeY (or emittance) shows that a larger beam size results in a higher luminosity. It is not consistent with the Eq. 1. We can explain it by Fig. 4. From Fig. 4, we can find that a higher beam intensity will result in a higher emittance during injection for the blue ion beam and the yellow ion beam. There are some correlations between the ion beam intensity and the initial beam size as shows in Fig. 2.

The left plot in Fig. 5 shows the relationship between the luminosity and the LEReC electron beam current. The right plot in Fig. 5 is the relationship between the luminosity and the electron beam energy. From these plots, it is not easy to find a clear trend between the luminosity and these parameters. Therefore, there is limited guidance on optimizing the luminosity by tuning these parameters store by store.

Other parameters, such as the machine tune, the chromaticity, the electron beam position, and the ion beam position, also have similar behaviors. Their scattering plots are in Appendix A. We also can't optimize the operation parameters only according to these plots.

3. XGBoost: A supervised machine learning algorithm

3.1. Extreme Gradient Boosting

To distinguish the effects on the luminosity from an individual input parameter, we would like to have a function with multivariance for the luminosity. It may have a similar function as below:

$$Lumi = f(x_1, x_2, x_3 \dots x_n) \quad (3)$$

while $x_1, x_2, x_3 \dots x_n$ are all input parameters that can affect the luminosity directly or indirectly.

With Eq. 3, we could find a function or partial dependence plot for each input parameter. Therefore we can distinguish the effects from the individual input parameter and optimize the luminosity according to these plots.

We have implemented a machine-learning algorithm named Extreme Gradient Boosting (XGBoost) to create a black-box model between the luminosity and all input parameters. This model can have a similar function of Eq. 3. XGBoost is a supervised machine learning algorithm for either a classification problem or a regression problem via a gradient descent method to optimize its loss function. It was used by many winning teams of machine learning competitions, including 'CERN LHCb experiment Flavour of Physics' [3]. It becomes well-known after 'Higgs Machine Learning Challenge' [4] and the 'HEP meets ML award' has been given to its authors [5].

XGBoost tries to construct the relationship or model with the best predictive power by using some previous logged input and output data. To evaluate how well the algorithm models the input data sets, the algorithm can use a loss function as its criterion. The loss function describes the difference between the prediction and the actual output. Mean squared error (MSE) is one of the loss function for a regression algorithm.

$$MSE = \frac{1}{m} \sum_{i=1}^m (y_i - y_i^p)^2 \quad (4)$$

where $m = 791$ is the total number of data sets for the case in this note.

The goal of a machine learning method is to minimize its loss function by calculating its error term. For a machine learning algorithm, if the error term is the gradient of the loss function, the method is called 'gradient'. If it also uses an ensemble learning method [6] such as Boosting to sample the training data and combine several multiple trained models to one model, it is called 'gradient boosting'. If it includes the first and second-order Taylor approximation (dependency on the 1st and 2nd order derivative) of its loss function into its error calculation, it becomes 'extreme gradient boosting'.

In this note, we constructed an appropriate unknown functional relationship (or black-box model) between the luminosity and some input parameters via

Table 3: Parameters Importance

Abbreviations	R^2 Reduction	Abbreviations	R^2 Reduction
IntenB	0.2446	TuneBV	0.0188
CollYV	0.1768	CollYH	0.0158
IntenY	0.1109	ibpmB	0.0152
Energy	0.0782	ChromBH	0.0139
CollBH	0.0551	CollBV	0.0110
SizeB	0.0418	TuneYH	0.0108
SizeY	0.0390	ChromYV	0.0050
ibpmY	0.0378	BYquad	0.0040
Current	0.0296	Ramp	0.0035
ebpmY	0.0196	ChromBV	0.0023

XGBoost. The function is denoted by a hypothesis function h close to the output (luminosity). It is defined as below:

$$y \approx h(x_1, x_2, x_3, \dots, x_n) \quad (5)$$

where $n = 24$ is the total number of the input parameters from Table.1.

3.2. XGBoost Model R^2 Score and Parameters Importance

To have an XGBoost model, we use 85% of the 791 data sets as the training data to construct (train) the model. To evaluate the model performance, we use the other 15% data sets to compare the model prediction and its actual value. Before using this data, we also did some data pre-processing, such as filling the empty data points with an average value.

After constructing an XGBoost model, a R^2 score function (the coefficient of determination) is used to evaluate the model performance. It is 0.87 for the best XGBoost model achieved in this note with MSE of 0.5. Normally, $R^2 = 1$ or $MSE = 0$ means the prediction of the regression model matches the actual value perfectly. The 15% test data points and their predictions agree very well and are plotted in Fig. 6

We could evaluate the effects of an individual parameter on the model performance via a permutation method [7]. The method shuffles the values in a single column (ibpmB in Table. 2 for example) while keeping other columns unchanged, then check how much the R^2 score decreases. We list the model score deduction in Table. 3 for each input parameter after random shuffling. That is the parameter importance of constructing this model.

But this parameter importance is only valid for this machine learning model. They could have slightly different results for different models with the same

algorithm. Furthermore, as we mentioned before, we should also distinguish this importance from the actual parameter importance during the operation. For example, we have already optimized the electron beam current and the electron beam energy, and their parameter importance cannot be captured by the model with the existing data.

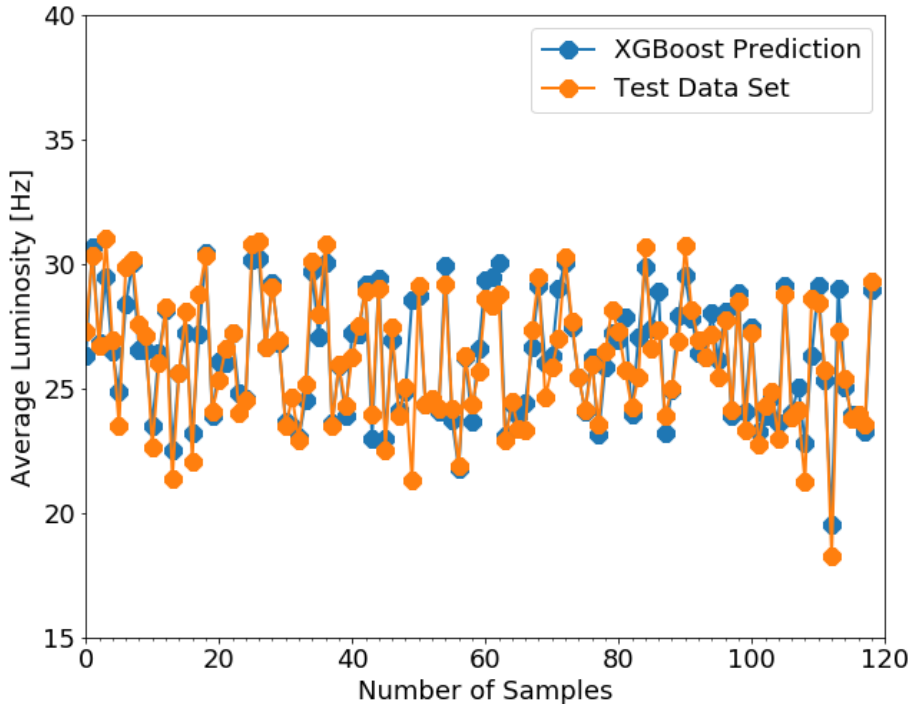


Fig. 6: The comparison plot between the test data points and their prediction from the model

4. Model Interpretation with Partial Dependence Plot

As discussed in the previous section, we cannot optimize the luminosity only according to their scattering plots, their correlations, or their parameter importance.

To resolve this issue, we can use a partial dependence plot (PDP) [8] [9] [10]. After constructing a machine learning model, a PDP can plot the marginal effects of one or two input parameters on the model prediction. The partial function $\hat{h}_{x_S}(x_S)$ is defined as below:

$$\hat{h}_{x_S}(x_S) = \frac{1}{m} \sum_{i=1}^m h_{x_S}(x_1^{(i)}, x_2^{(i)}, \dots, x_S, \dots, x_n^{(i)}) \quad (6)$$

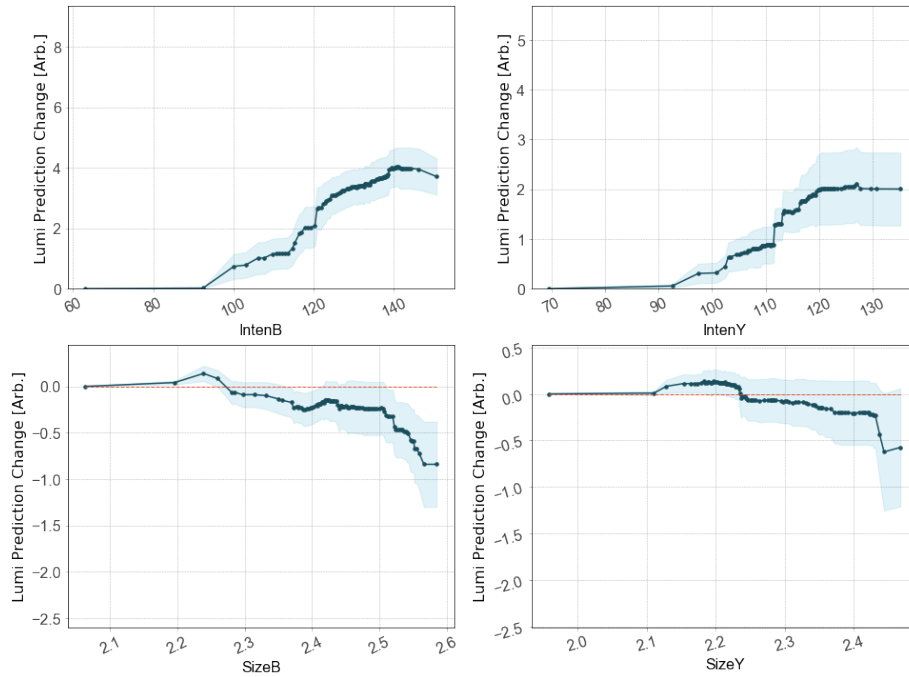


Fig. 7: The PDP plot of the ion beam intensity [1E9] and the ion beam size (emittance) [μm]

where h is the XGBoost hypothesis model (Eq5), while $m = 791$ is the total number of the data sets and $n = 24$ is the total number of input parameters. x_S

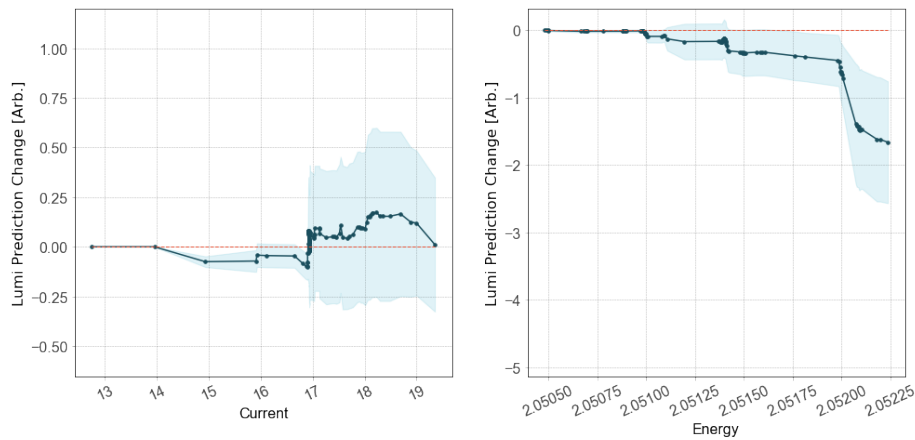


Fig. 8: The PDP plot of the electron beam current [mA] and energy [MeV].

is the interested input parameter.

To get a partial dependence plot for the $ibpmB$ parameter in Table. 2, we can replace all other 790 values in that column with the first value $ibpmB = 74.69$. This value will be one horizontal data point (x-axis) of the PDP plot. Meanwhile, we keep other columns unchanged and calculate the average and standard deviation of all model predictions with these 791 data sets. Thus, we get a corresponded vertical data point (y-axis) of the PDP plot. That is the first data point for the PDP plot. After that, we can repeat this procedure with the second value of $ibpmB = 100.30$ for the second data point until the last value in this column.

Fig. 7 shows the PDP plots of the ion beam intensity and the beam size. The light blue area is the deviation of the model predictions. The predicted luminosity has a clear trend for both the blue and the yellow intensity. It also has a clear trend for both the blue and yellow beam size. The PDP plots of the beam size also agree with the theory (Eq.1) qualitatively. While from their scattering plots in Fig. 3, it is not clear for the SizeB. It even doesn't agree with the theory for the SizeY. This PDP plot demonstrated that the constructed XGBoost model and their PDP plots could distinguish the effects of one parameter from other parameters and predict the luminosity correctly.

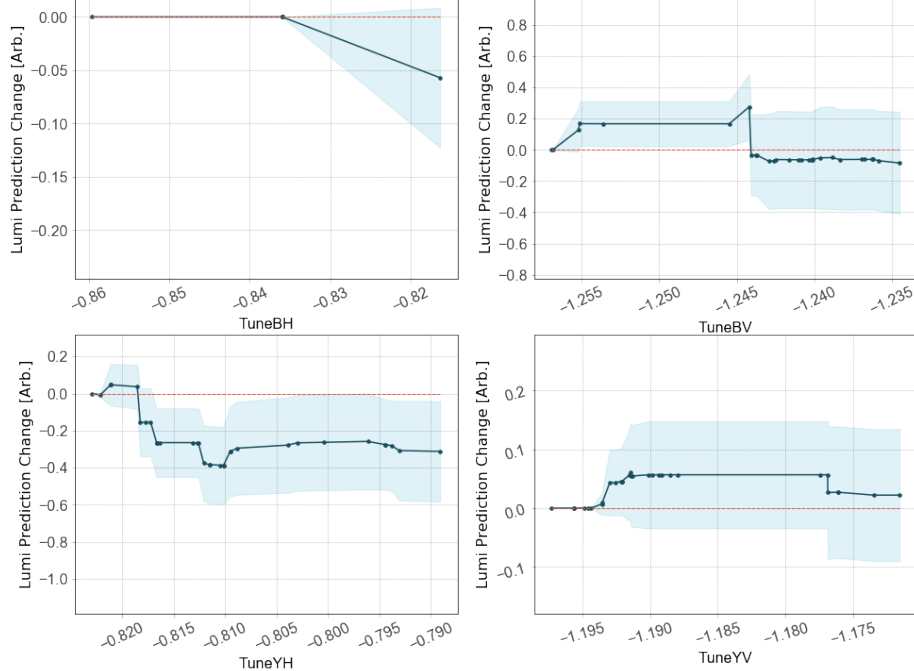


Fig. 9: The PDP plot of the machine tune (the setpoint values).

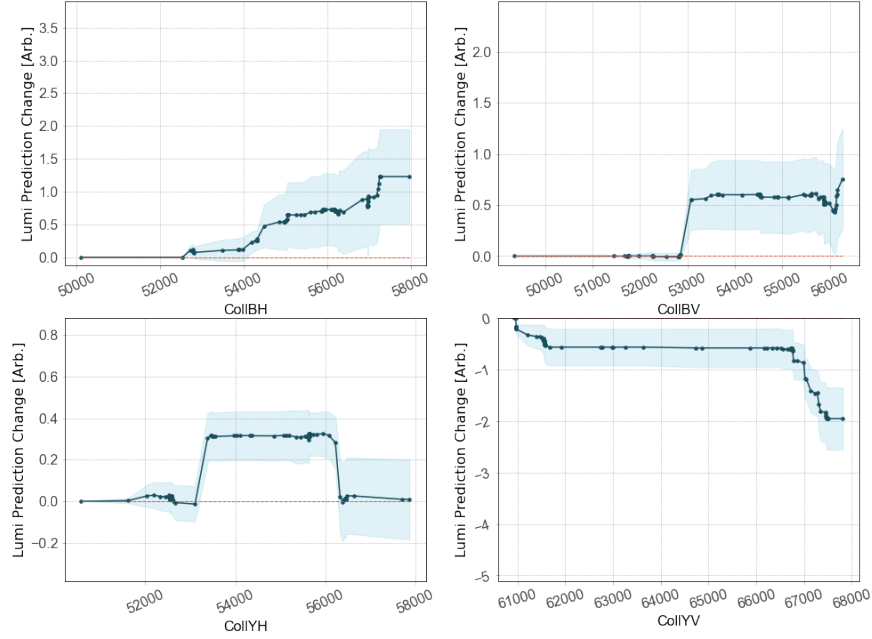


Fig. 10: The PDP plot of the machine collimators [steps]. (where 0 corresponds to collimators fully retracted out and 2000 steps equal to 1 mm.)

Fig. 8 is the PDP plot for the electron beam current and the electron energy. This plot shows that the 17 mA operation electron current is very close to the optimized region (18 mA). Meanwhile, according to the plot, we can further optimize the luminosity by decreasing the electron beam energy.

Fig. 9 is the machine tune PDP plot. These tunes and chromaticities are the setpoint values to the machine, not the measurement values. According to the plot, we can optimize the luminosity by reducing the blue vertical and yellow horizontal tune. The blue horizontal plan is less clear because of fewer data points, while the yellow vertical tune of 1.185 is very close to its maximum.

Fig. 10 is the collimator position PDP plots. The unit of the collimator position is steps, and 2000 steps equal to 1 mm. During operation, the collimator positions are a compromise between the STAR luminosity and its background. The greater number of the collimator position (steps) means the collimator closer to the ion beam, while 0 corresponds to collimators fully retracted out. To have a cleaner background, it could be that the yellow horizontal collimator and vertical collimator too close to the ion beam, and they affected the luminosity. The yellow horizontal collimator position of 56000 μm could be an optimized position for both luminosity and background. The plot also suggests that it is possible

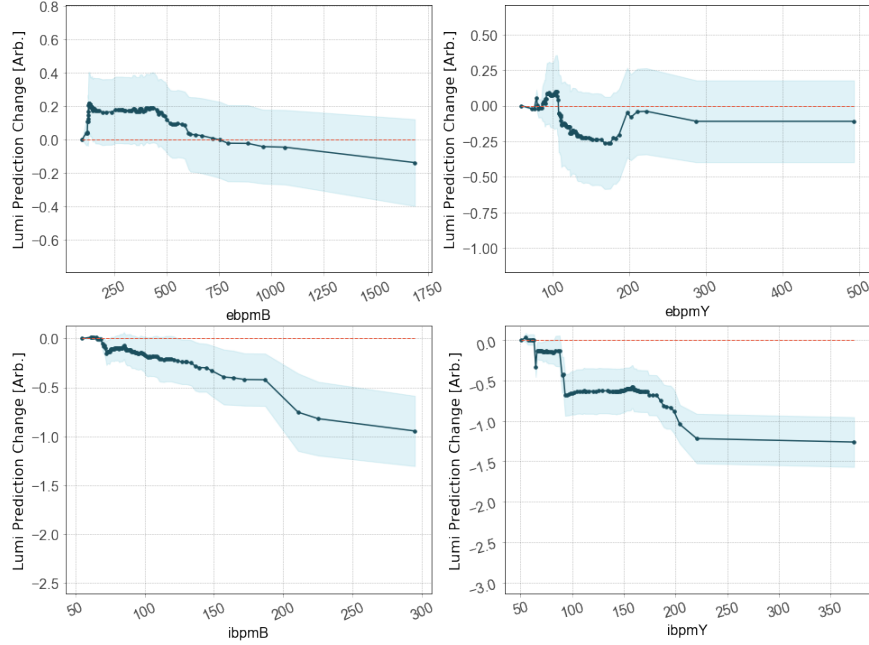


Fig. 11: The PDP plot for the ion beam and the electron beam positions [um].

to optimize the luminosity with the blue horizontal and vertical collimators.

Fig. 11 are the PDP plots for the ion beam position and the electron beam position. We can find that a smaller ion beam and electron beam position tends to have a higher luminosity, which agrees with the LEReC cooling theory [11]. The PDP plot for the machine chromaticity is in Appendix A.

From the above plots, we can see that a PDP can show the effects of individual parameters. To have a clear interpretation of a PDP plot, we should make sure there is no significant correlation between these input parameters. We have used the correlation heatmap Fig. 2 to choose the parameters for the XGBoost algorithm.

5. Model Interpretation with Shapley Value and SHAP

To overcome the potential correlation issue from a PDP plot, we can also use so-called Shapley values (SHAP, Shapley Additive exPlanation) [12] [13] to interpret the effect of an individual input parameter on the XGBoost model prediction. The Shapley value calculates the marginal contribution of one input parameter to the model prediction, as it does in the PDP plot. The formula to calculate the Shapley value of an input parameter is [9]

$$\phi_j = \sum_{S \subseteq (x_1, \dots, x_n) \setminus (x_j)} \frac{|S|!(n - |S| - 1)!}{n!} (h_x(S \cup x_j) - h_x(S)) \quad (7)$$

where (x_1, \dots, x_n) includes all input parameter values ($n=24$) in one data set (one row in Table.2). x_j is the interested parameter, and S is one of all possible parameter subsets without x_j . And j is the j^{th} parameter in this data set (this row) and it is less than n^{th} . Because the order in which a machine learning model uses an input parameter can affect its Shapley value calculation (the contribution to the model prediction), we need all possible subsets. Each subset S can be any possible parameter combination, and they can include 0, or 1, \dots ,

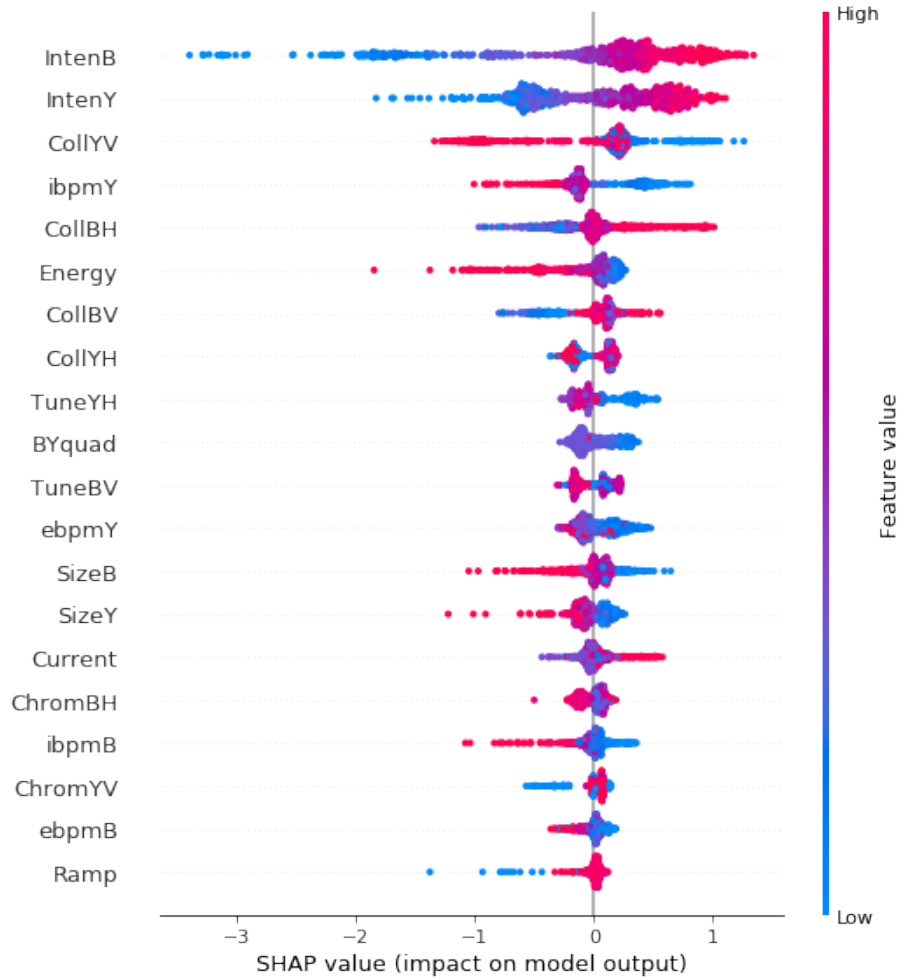


Fig. 12: The overview of all SHAP values for all input values in Fig.2

or $n - 1$ input parameters.

The XGBoost model or function is h_x (Eq.5) and $h_x(S \cup x_j) - h_x(S)$ represents the marginal contribution of x_j for one possible subset (S) of parameters. It

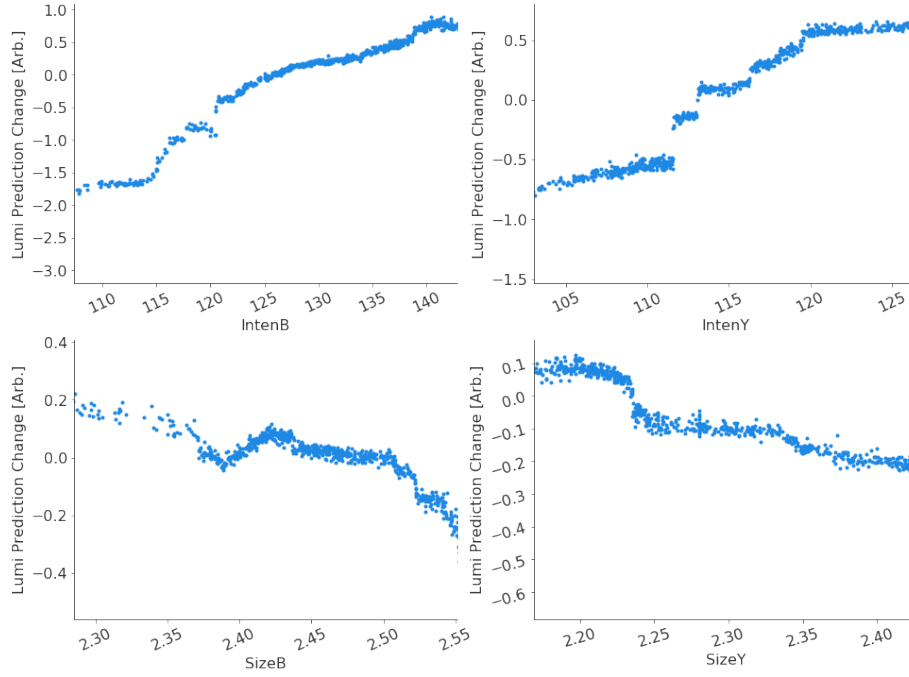


Fig. 13: The SHAP values plots for the ion beam intensity [1E9] and beam size [μm] (emittance).

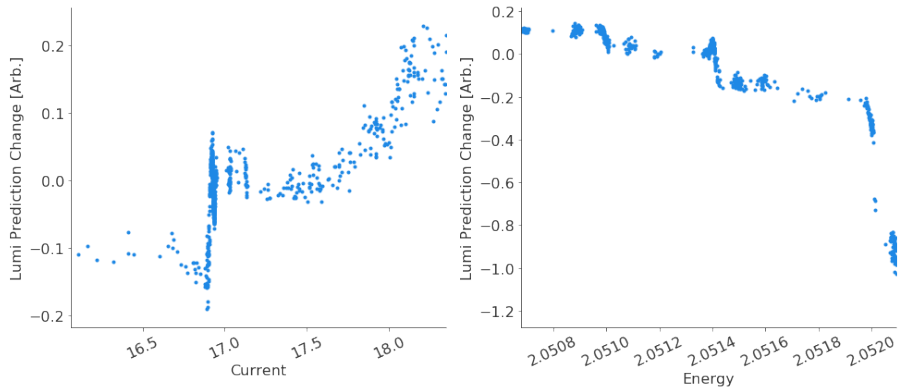


Fig. 14: The SHAP values plots for the electron beam current [mA] and energy [MeV].

calculates the difference of the parameter x_j on the model prediction with and without this parameter. The weighted total value of the marginal contributions for all possible subsets is the Shapley value of the j parameter value in this data set (this row).

With Eq.7, we can calculate the Shapley value for an individual parameter value that represents its contribution to the model prediction. After calculating the contribution (Shapley value) of the $ibpmB = 74.69$ value in the first row of Table.2, we can repeat this calculation for the second input parameter values $ibpmY = 156.60$ until the last parameter value in the row. Thus, we calculate the contribution (Shapley values) of all input parameter values in this data set. After we repeat the calculation for all other rows of data sets (791 rows in Table.2), we will get a 791×24 Shapley value table.

Then we can plot these SHAP values as well as all parameter values in Fig. 12. Each dot has three characteristics. First, the horizontal axis shows its Shapley value that indicates whether the effect of that value causes a higher or lower prediction. Second, the most left vertical location shows what parameter it is depicting, such as IntenB, IntenY. We rank these parameters in descending order according to their total contributions to the model prediction. They are similar to the result in Table.3. We can find that the yellow vertical collimator, the blue horizontal collimator, and the electron beam energy are critical in the model, besides the blue and yellow ion beam intensity.

Last, the color shows the high or low value for that parameter. The maximum value of one parameter (one column in Table.2) is red, while the minimum value in that column shows blue. The colors between blue and red represent the values between the maximum and the minimum. From Fig. 12, we can find a high IntenB or a high IntenY value (red) has a positive (a positive SHAP value) effect on the luminosity. But a lower ColLYV (blue) value has a positive value (prediction) on the luminosity. Therefore, we can optimize the luminosity by increasing or decreasing a parameter if we assume a linear relationship between them.

Fig. 12 is the overview of all parameters and their Shapley values. We can also plot the SHAP value for an individual input parameter like the PDP plot. Fig. 13 is the plot for the ion beam intensity and the ion beam size, and Fig. 14 is the plots for the electron beam current and the electron energy. They all have a similar trend to their PDP plots.

We list more SHAP plots in Appendix A for comparison. Both the PDP plots and the SHAP plots show a similar trend between the luminosity and their input parameters.

6. Summary and Discussion

To find the optimum operation parameters for RHIC low energy run, we implemented a machine learning regression algorithm (XGBoost) using RHIC and LEReC operation data.

After constructing an XGBoost model offline, we can distinguish the effects of each operation parameter and optimize the luminosity according to their PDP and SHAP plots. According to these plots, we can further optimize the luminosity by decreasing the electron beam energy and increasing the electron beam current to 18 mA. Other possible optimizations may include reduction of the blue vertical and yellow horizontal tunes, moving out the yellow collimators (horizontal and the vertical), move in the blue horizontal collimators, etc.

To get a good XGBoost model and find the optimized operation parameter values, one needs to change all input parameters within a wide range. If we optimize the luminosity by changing only one or two machine operation parameters during one store, it requires lots of stores. Due to many parameter changes within one store, one cannot easily identify parameters that improve the luminosity. But if we optimize the luminosity with XGBoost and their partial dependence plots (or SHAP plot), there is no such limitation. We can change all input operation parameters during one store and distinguish their effects later with XGBoost and their PDP plots. As a result, we can optimize the luminosity faster and using fewer stores.

But machine-learning algorithms are impartial, and these trained models are based on input data. Therefore if the input data has changed significantly, it will most likely affect the model prediction. Meanwhile, different machine learning algorithms will result in different models and predictions.

7. Acknowledgments

The author appreciated the help from Vincent Schoefer and Aljosa Marusic for providing the data acquirement script. The author also thanks Angelika Dress for the collimator information.

References

- [1] Tianqi Chen, Carlos Guestrin, XGBoost A Scalable Tree Boosting System, arXiv preprint arXiv:1603.02754
- [2] <https://github.com/dmlc/xgboost>
- [3] <https://home.cern/news/news/computing/flavours-physics-join-lhcb-machine-learning-contest>

- [4] Claire Adam-Bourdarios, Glen Cowan, Cécile Germain, Isabelle Guyon, Balázs Kégl, et al., The Higgs Machine Learning Challenge, doi:10.1088/1742-6596/664/7/072015
- [5] <https://cds.cern.ch/record/2017394?ln=en>
- [6] <https://www.sciencedirect.com/topics/computer-science/ensemble-learning>
- [7] <https://pypi.org/project/eli5/>
- [8] Ray Wright, Interpreting Black-Box Machine Learning Models Using Partial Dependence and Individual Conditional Expectation Plots, Paper SAS1950-2018, <https://www.sas.com/content/dam/SAS/support/en/sas-global-forum-proceedings/2018/1950-2018.pdf>
- [9] Christoph Molnar, Interpretable Machine Learning - A Guide for Making Black Box Models Explainable, <https://christophm.github.io/interpretable-ml-book/pdp.html>
- [10] <https://pdpbox.readthedocs.io/en/latest/>
- [11] S. Seletskiy, M. Blaskiewicz, A. Drees, etc, Accurate setting of electron energy for demonstration of first hadron beam cooling with rf-accelerated electron bunches, Phys. Rev. Accel. Beams 22, 111004 – Published 27 November 2019,
- [12] Scott Lundberg, Su-In Lee. A Unified Approach to Interpreting Model Predictions. <https://arxiv.org/abs/1705.07874>.
- [13] <https://shap.readthedocs.io/en/latest/>

8. Appendix A: More Scattering Plots, PDP Plots and SHAP plots

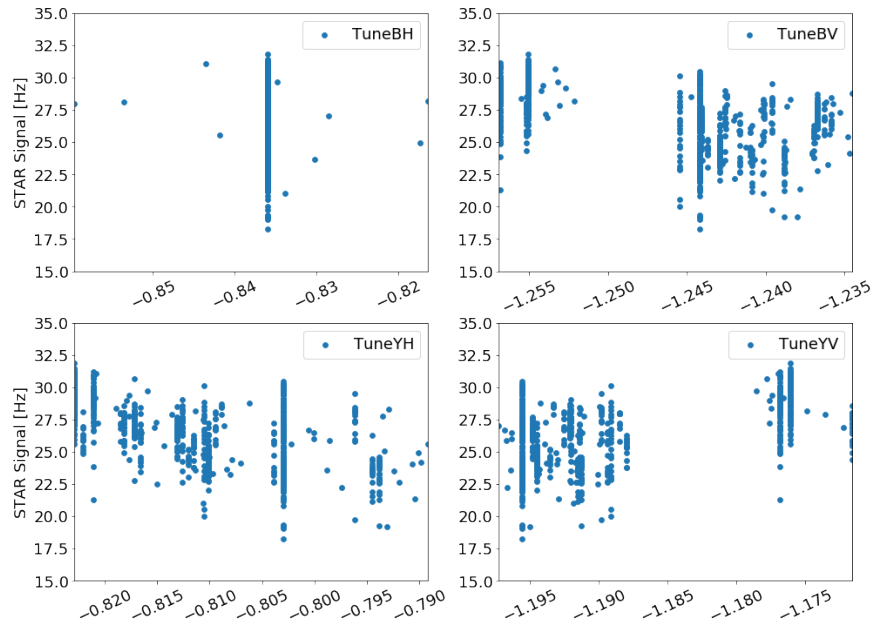


Fig. 15: The scattering plot for the machine tune.

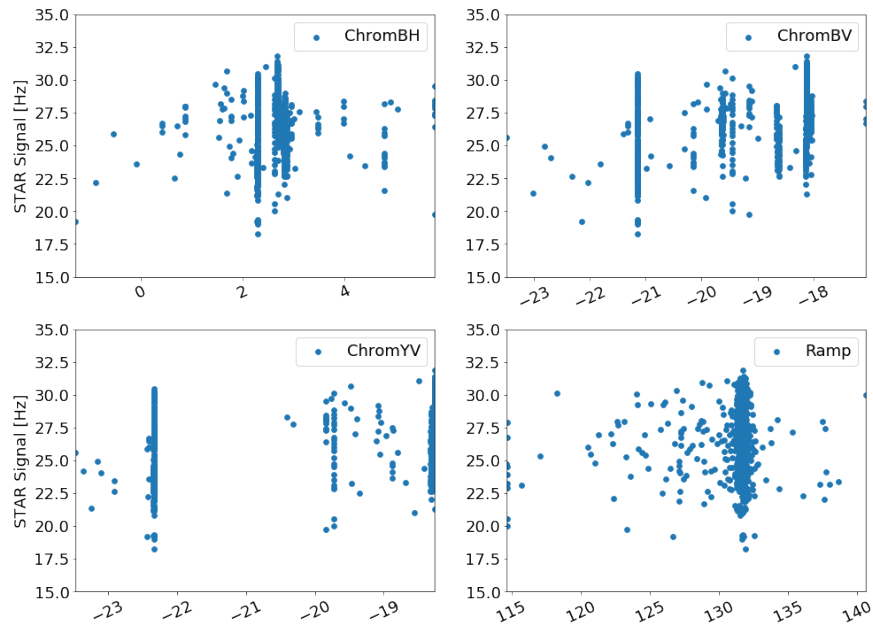


Fig. 16: The scattering plot for the machine chromaticity.

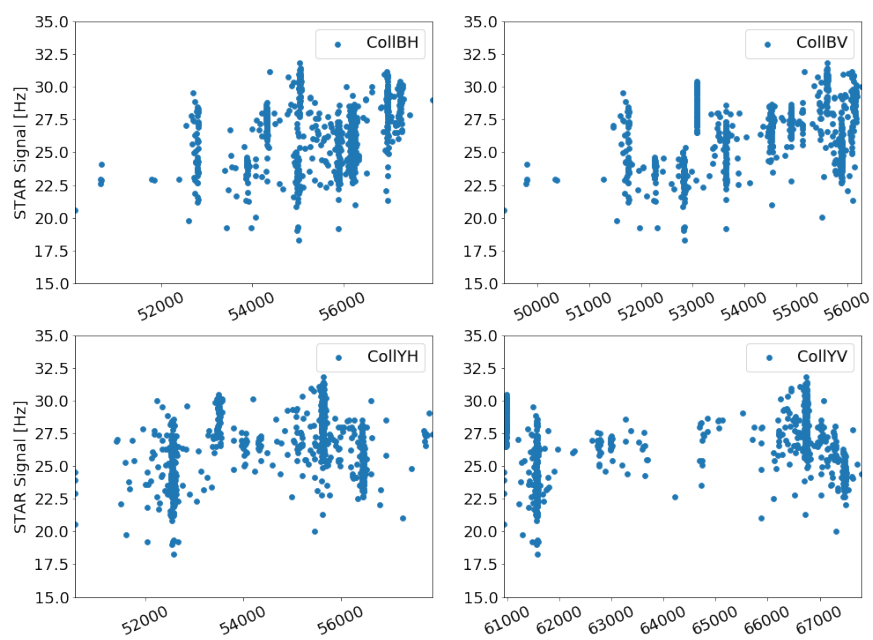


Fig. 17: The scattering plot for the machine collimator position [steps]. 2000 steps equal to 1 mm.

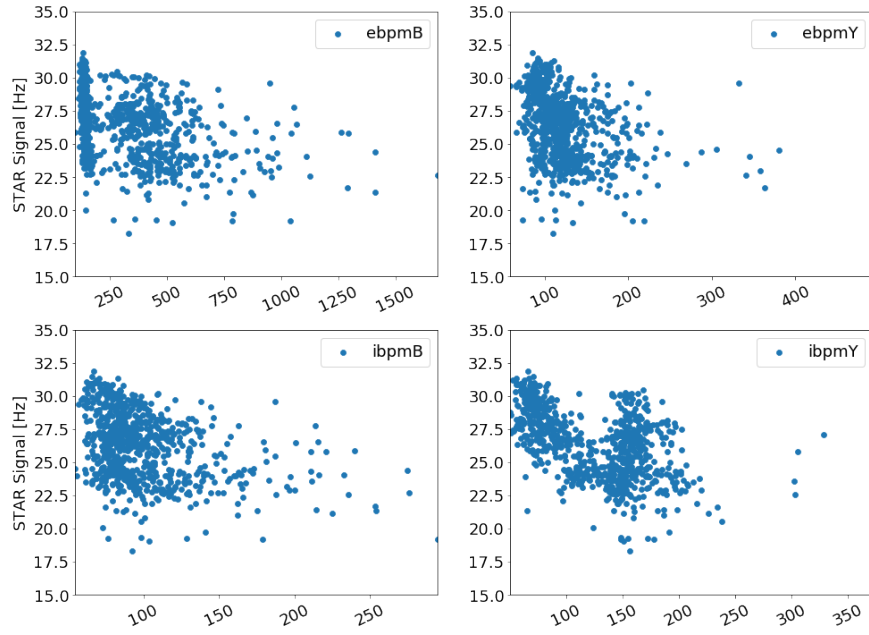


Fig. 18: The scattering plot for the ion beam and electron beam positions [um].

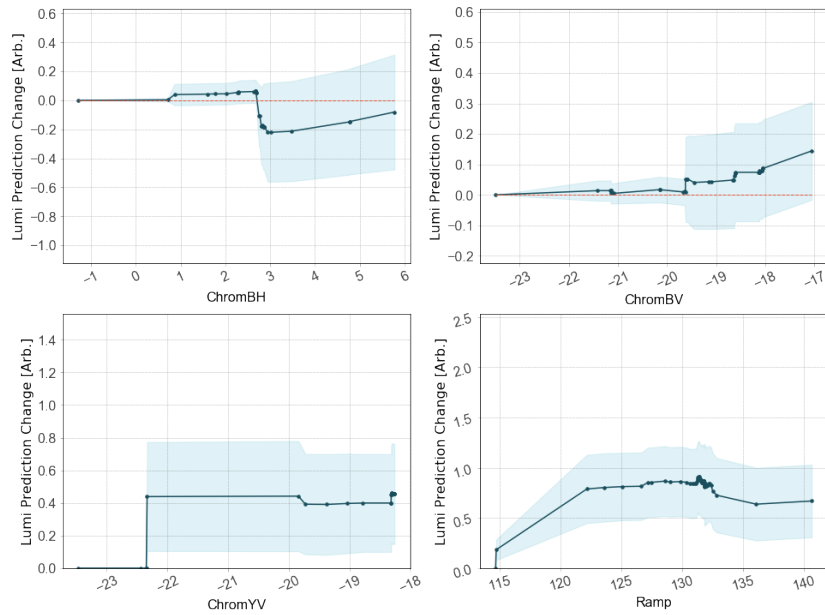


Fig. 19: The PDP plot for the machine chromaticity.

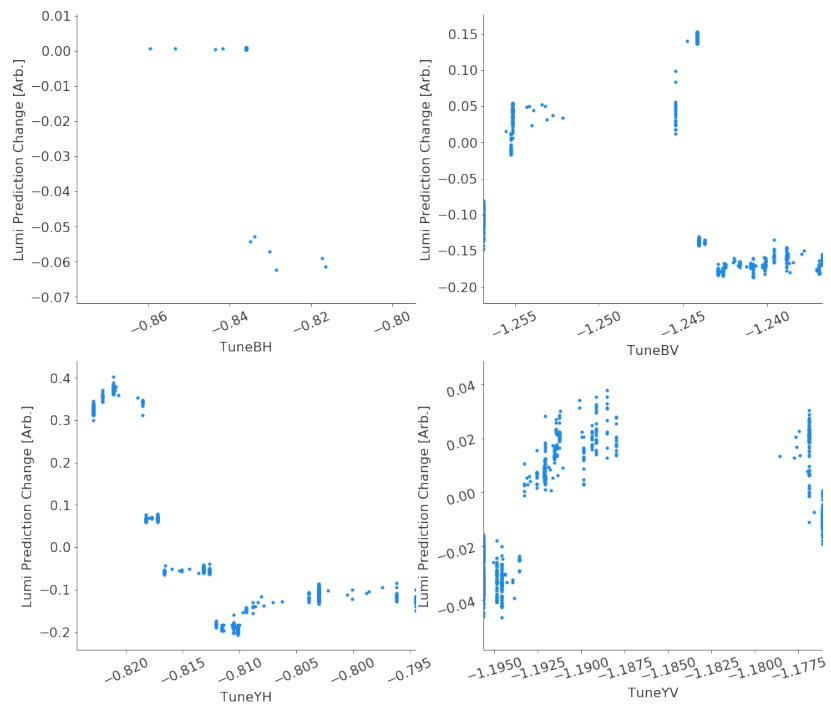


Fig. 20: The SHAP values plots for the machine tune.

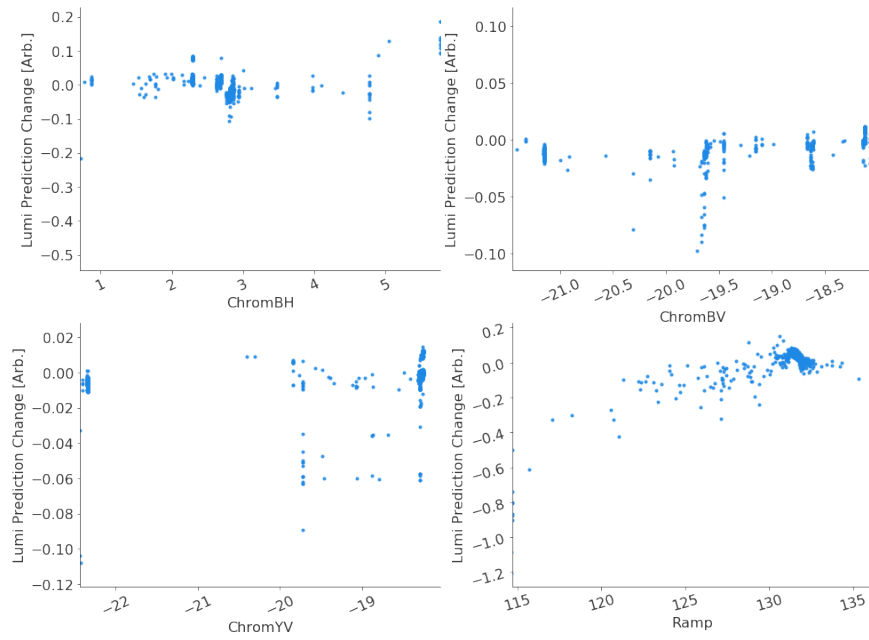


Fig. 21: The SHAP values plots for the machine chromaticity.

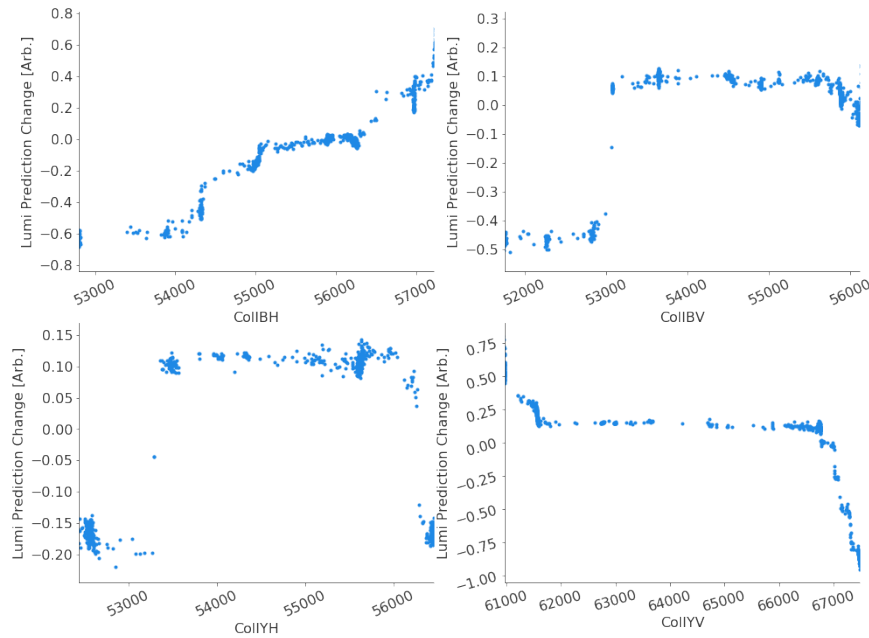


Fig. 22: The SHAP values plots for the machine collimator position [steps]. 2000 steps equal to 1 mm.

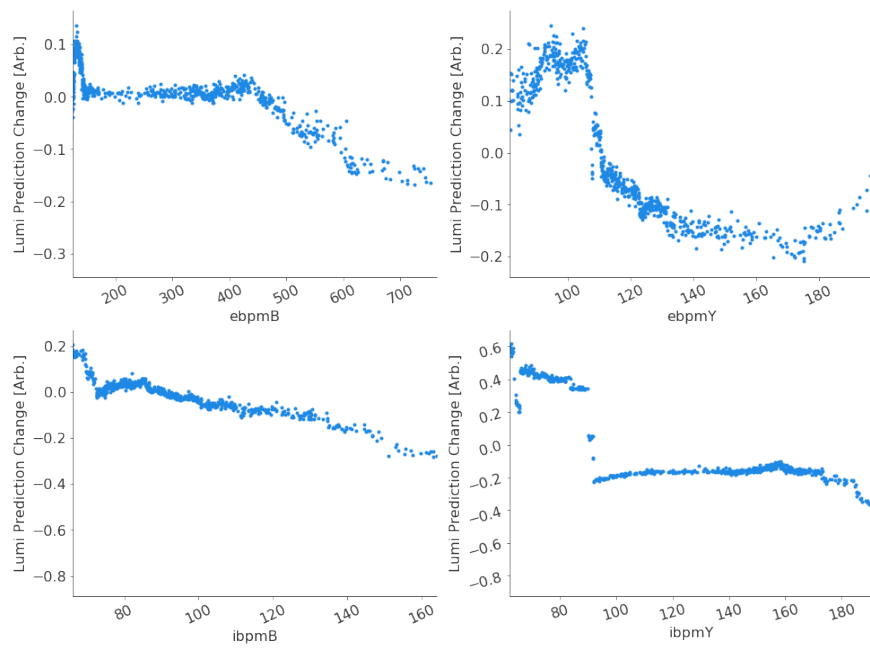


Fig. 23: The SHAP values plots for the ion beam and the electron beam bpms [um].