Implementation of Machine Learning sampling techniques for initial geometry prediction in heavy ion collision experiments

Abhisek Saha

School of Physics
University of Hyderabad

Phys. Rev. C 106, 014901 (2022)

ICPAQGP-2023
Puri, India

Introduction

- In heavy-ion collisions, the initial geometry parameters have a significant impact on the final state particle production.
- However, the calculation of such quantities is nearly impossible in experiments as the length scale range in the level of a few fermi.
- We implement various ML-based supervised regression techniques and demonstrate high prediction accuracy of three important properties that determine the initial geometry of the HIC experiments.
- Though ML techniques have been used previously to determine the impact parameter of these collisions, we study **multiple ML algorithms**, **their error spectrum**, and **sampling methods using exhaustive parameter scans** and ablation studies to determine a combination of efficient algorithm, and a tuned training set that gives multi-fold improvement in accuracy for different heavy-ion collision models.

Measurement of centrality in experiment

- Although the impact parameter is an important parameter, but it is difficult to calculate or measure from the experiments as it is one of the initial state parameters.
- The process of getting b is to make a back-calculation using theoretical models like the Glauber model.
- The determination of the impact parameter is related to the charged multiplicity produced during the heavy ion collision.

$$\frac{dN_{ch}}{d\eta} = n_{pp}\left[(1-x)\frac{N_{part}}{2} + xN_{coll}\right] \tag{1}$$

x is the fraction of contribution from hard processes. n_{pp} is the multiplicity per unit rapidity in pp collisions and N_{coll} is the number of binary NN collision.

- In the Glauber model, the participant nuclei are related to the density distribution of the nucleons inside the nuclei.

$$\begin{aligned} N_{part}(b) = & \int T_A(s)(1 - exp[-\sigma_{inel}^{NN} T_B(b-s)])ds \\ & + \int T_B(b-s)(1 - exp[-\sigma_{inel}^{NN} T_A(b)])ds \end{aligned} \tag{2}$$

$T_A(s)$ is the thickness function of nucleus A, b is the impact parameter

- Using Eq.1 and Eq.2, the impact parameter hence centrality, can be estimated by fitting the multiplicity spectra.
- Our goal is to calculate some of the initial state parameters which are difficult to calculate using Machine Learning (ML) Models.

Machine Learning Inputs and outputs

- Machine learning is a method of data analysis where the machine, i.e., the model learns from the input data by tuning the model's own hyperparameters and applying the learning to make predictions on the test data.
  - Supervised ML models.
  - Train data and test data.
- In this study, the transverse momentum(p_T) spectra are taken as features and the impact parameter is taken as the target variable, which the model must predict.
- We have used the AMPT (A Multi-Phase Transport) model to generate the transverse momentum spectra of Au-Au collision events at 200 GeV collision energy.

  Che-Ming Ko et al., PRC 72, 064901 (2005)

Eccentricity and Collective flow

- Eccentricity is also one of such initial state parameters that is difficult to measure but has larger impacts on the HIC experiment outcomes e.g. anisotropic flows.
- Initial fluctuation of eccentricity affects elliptic flow coefficients in both the Glauber model and the CGC model.

T. Hirano and Y. Nara, Phys. Rev. C 79, 064904 (2009)

$$\epsilon_n(b) = \frac{< r^n cos(n\phi - n\psi) >}{r^n} \tag{3}$$

$$\epsilon_{part} = \frac{\sqrt{\sigma_y^2 - \sigma_x^2 + 4\sigma_{xy}^2}}{\sigma_y^2 + \sigma_x^2} \tag{4}$$

- $r = \sqrt{x^2 + y^2}$
  σ's are the variances of the positions of the particles,
  $\sigma_x^2 =< x^2 > - < x >^2$, $\sigma_y^2 =< y^2 > - < y >^2$ and
  $\sigma_{xy} =< xy > - < x >< y >$.

Problems to be addressed

- To find the best ML model which gives optimum accuracy (with all possible hyperparameter combinations) in impact parameter and eccentricity prediction.
- To make a model-independent study. We want to train the ML models with AMPT data and see if they can predict other HIC model data.
- We will analyze the error distribution impact parameter predictions and eccentricity predictions.
- We will find ways to minimize the error through data re-balancing and see what procedure of re-balancing techniques is more effective.

ML Model Comparison in b prediction

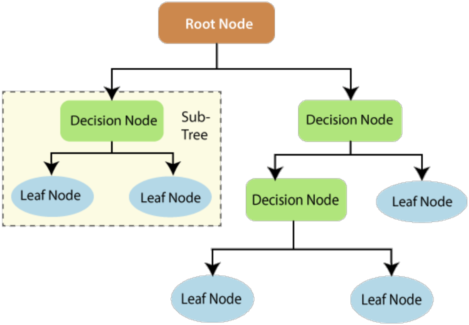
Model	R^2	MAE	RMSE	MSE
Gradient Boosting Regressor	0.9709	0.3834	0.4819	0.2323
Light Gradient Boosting Machine	0.9702	0.3878	0.4876	0.2378
Random Forest Regressor	0.9689	0.3972	0.4984	0.2484
Extra Trees Regressor	0.968	0.4024	0.5048	0.2549
AdaBoost Regressor	0.9676	0.4049	0.5079	0.2581
K Neighbors Regressor	0.9649	0.4226	0.5295	0.2804
Linear Regression	0.9642	0.422	0.5341	0.2855
Ridge Regression	0.9642	0.422	0.5341	0.2855
Least Angle Regression	0.9642	0.422	0.5341	0.2855
Huber Regressor	0.9642	0.4216	0.5346	0.2861
Bayesian Ridge	0.9642	0.422	0.5341	0.2855
Orthogonal Matching Pursuit	0.9635	0.4272	0.5398	0.2916
Decision Tree Regressor	0.9405	0.5503	0.6888	0.4745
Passive Aggressive Regressor	0.8849	0.7482	0.9058	0.9197
Lasso Regression	0.7461	1.1484	1.4246	2.0318
Elastic Nets	0.6253	1.4093	1.7305	2.9977

k-Nearest Neighbors(kNN)

- The kNN algorithm uses feature similarity to predict new data points. It compares similar features between the unknown test data and the known data and predicts a value depending on how closely this resembles the points in the training set.
- if the values in the features of a test data are closer to the value of the same features in the train data, then it is most probable that the target feature of the test data will have a similar value as the target variable of the train data.
- The hyperparameters space:
    - The number of nearest neighbors (1-50)
    - 'distance': Distance between test data and train data (Euclidean, Manhattan and Minkowski distance)
    - 'weights': importance given with distance ('uniform' and 'distance')

Random Forest and ExtraTrees Regression

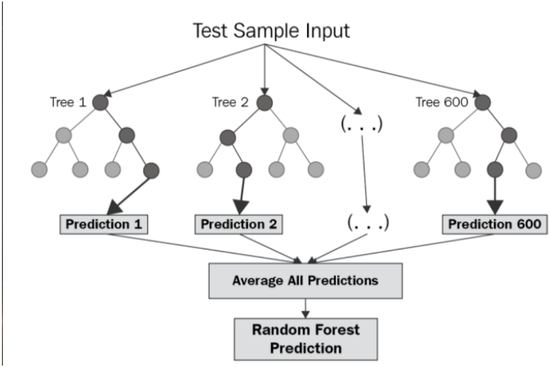
- Decision tree regression models are built in the form of a tree structure. Each node in a tree specifies a condition and for each outcome, there is a branch or a leaf associated with the node. If the data fulfills the condition, it goes to a specific branch or leaf of the node.
- The splitting is done based on variance reduction
  $Var\ Red = Var(parent) - \sum_i w_i Var(child_i)$
  $IG = Entropy(parent) - \sum_i w_i * Entropy(child_i)$



O. Mbaabu, https://www.section.io/engineering-education/introduction-to-random-forest-in-machine-learning/

Random Forest and ExtraTrees Regression

- In the case of ensemble methods like ETR or RF, different subsets of decision trees are taken for one outcome. The final result is the aggregation of all these outcomes.



C.Bakshi https://levelup.gitconnected.com/random-forest-regression-209c0f354c84

Random Forest and ExtraTrees Regression

- Differences:
  - The splitting of nodes is based on random splits(not best split) among a random subset of the features selected at every node.
  - The sampling of data is done without replacement (Bootstrapping=False).
- The hyperparameters space:
  - 'n_estimator': the number of trees in the forest(100 to 1200)
  - 'max_features': the maximum number of features the algorithm considers to split a node
  - 'max_depth': the number of nodes a tree (5-30)
  - 'min_samples_leaf':minimum number of samples required to build a leaf node(1-10)
  - 'min_samples_split': Min. number of sample required to split at an internal node (2-100)

Hyperparameter Settings

Different sets of hyperparameter combinations of a model are used as trials, and the accuracies are checked for each of these combinations.

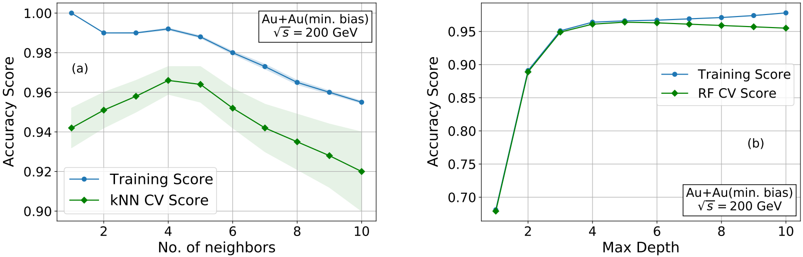


Figure: Change in accuracy as a function of hyperparameters. a) kNN model with the number of nearest neighbors hyperparameter(left), b)Random Forest with max depth hyperparameter (right)

Principal Component Analysis

- We used the PCA method to reduce the colinearity and compared the outcomes to the already achieved accuracy using all the features.
- The features are reduced by creating new sets of uncorrelated variables by maximizing the variance of the input features.
- The covariance matrix is formed using the covariance of the features. Then the eigenvectors and eigenvalues of the covariance matrix are obtained, and some top eigen values are chosen.
  $var(x) = \frac{\sum_{1}^{n}(x_i - \mu)^2}{n-1}$,
  $cov(x, y) = \frac{\sum_{1}^{n}(x_i - \mu_x)(y_i - \mu_y)}{n-1}$
- A matrix W is constructed using the eigenvectors corresponding to the selected eigenvalues.
- The original dataset is then transformed via matrix W, and a new k-dimensional feature subspace is obtained.

Principal Component Analysis

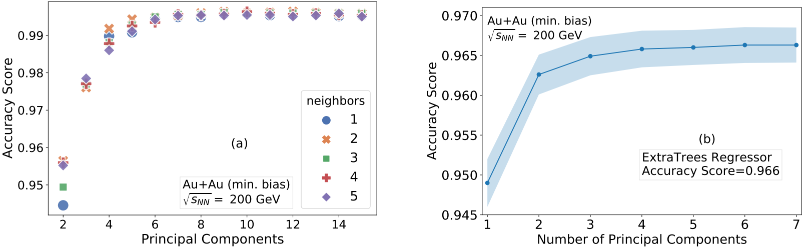


Figure: a) Accuracy of a a) kNN model(left) and b) an ETR model(right) as a function of the number of principal components used

Conclusion:

- The saturation in the accuracy score is achieved for the use of 7 or more principal components in both cases.
- At least 10 features or 10 principal components are needed to obtain an accurate result for the eccentricity and ϵ_{part}.

Impact parameter as a feature

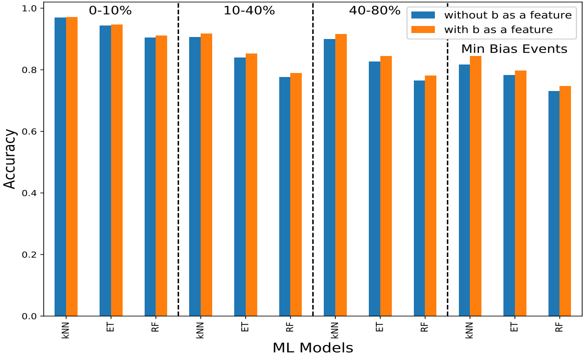


Figure: Effect on the eccentricity prediction accuracy by the inclusion of impact parameter as a feature. The orange bar represents accuracy with impact parameter and blue bars represent accuracy without impact parameter

Impact Parameter Prediction plot

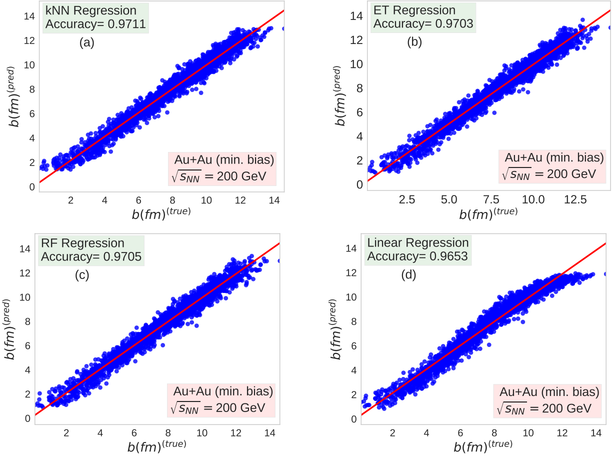


Figure: Impact parameter prediction using kNN(a), ET(b), RF(c) and LR(d) model. These plots are obtained for a random train and test set split of input

Error in b predictions

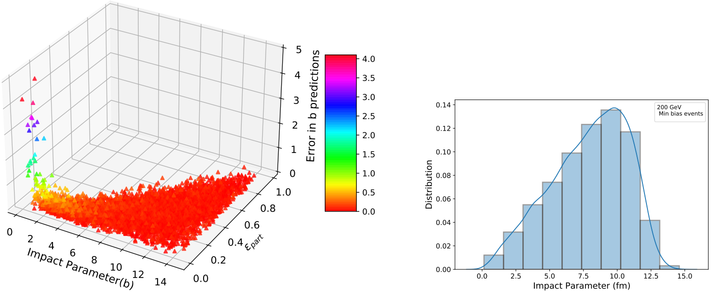


Figure: Error in the prediction of impact parameter as a function of impact parameter and eccentricity distribution. This is for 200 GeV Au-Au collisions and the prediction is obtained using a kNN model

ϵ and ϵ_{part} prediction accuracy

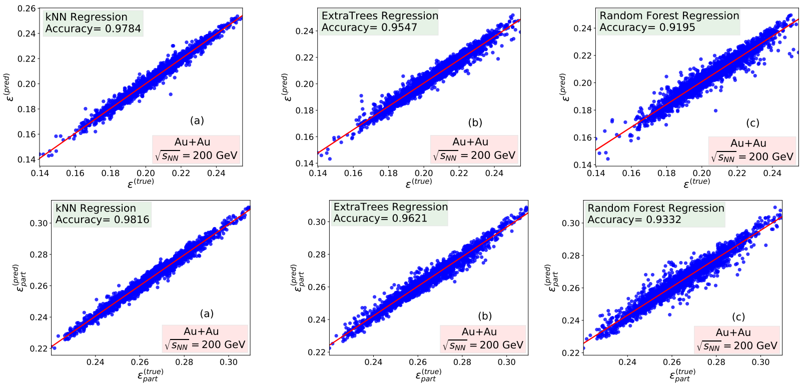


Figure: ϵ_2(top) and ϵ_{part}(bottom) prediction using kNN(a), ET(b) and RF(c) model. These plots are obtained for a random train and test set split of input events

ϵ_{part} prediction for higher range

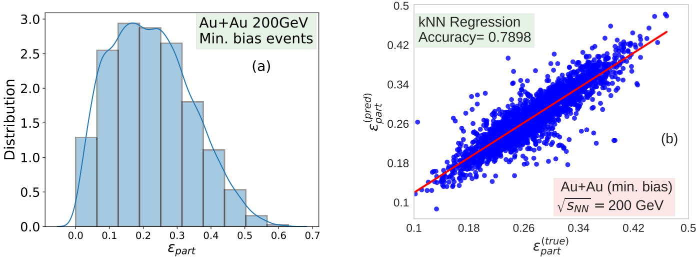


Figure: a) Histogram plot of participant eccentricity distribution(left) b) Prediction plot of ϵ_{part} using kNN model(right) of minimum bias Au-Au dataset at $\sqrt{s}$=200 GeV given by the AMPT model

Conclusion:

- For a larger range of ϵ_{part}, the accuracy is lowered to 78.98% from its previous value of 98.16%.
- The event distribution of ϵ is skewed.

Checking Model dependency

- We have taken test data from other HIC models. This is to test if the predictions of the ML algorithms depend crucially upon the nature of the model used.
- VISH2+1 model: The evolution of the system created in heavy ion collisions is described by relativistic causal viscous hydrodynamics.

  H. Song and U. Heinz, Phys. Rev. C 77, 064901 (2008).
- Hybrid Model: We have used the iEBE-VISHNU code package, which is a hybrid model made by combining a (2+1)-dimensional viscous hydrodynamic model and a hadronic cascade model (UrQMD).

  C. Shen et al., Computer Physics Communications 199, (2016), 61-85.

b predictions for hydro and hybrid model events

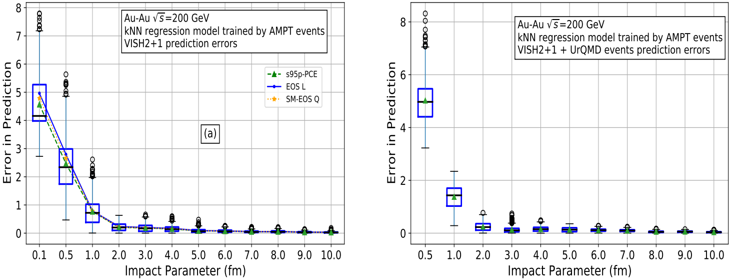


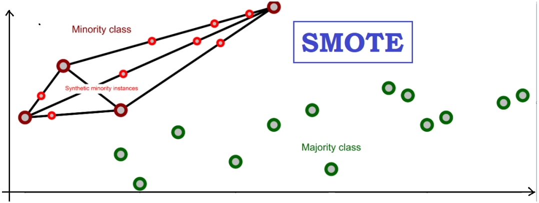
Figure: Error plot of impact parameter predictions by kNN model of different centrality events of a) vish2+1 and b)urqmd simulations

Conclusion:

- As long as the models reflect the experimental data accurately, the ML algorithms do not distinguish between the different models.
- For the lower b range, the errors are higher, similar to AMPT predictions.

Re-Balancing Techniques

- There are a few sampling techniques in machine learning for rebalancing datasets, e.g., SmoteR, ADASYN.
- These are python packages that increase(over-sampling) or decrease(under-sampling) the minority and majority data class respectively with the use of the neighboring data.

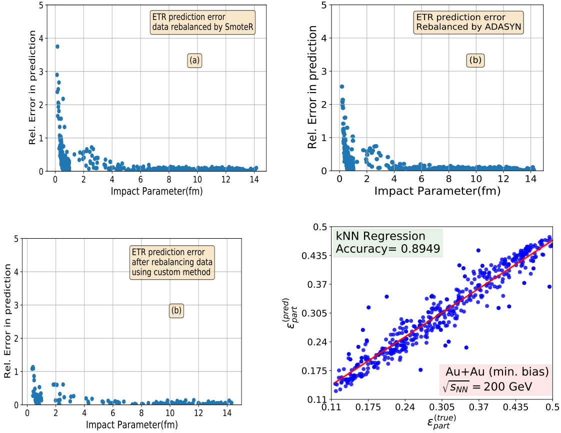


https://iq.opengenus.org/smote-for-imbalanced-dataset/

- We also provide a custom sampling method that shows significant improvement in accuracy over commonly used sampling methods in the ML community.
- In this, we divided the impact parameter range into two parts at b=2fm and gave more weightage to the lower b events.

Results of re-balanced data

Error distribution when the training set is re-balanced using a) SmoteR method, b) ADASYN method, and c)& d) giving weights to input data



Summary

- We find that the accuracy of the impact parameter prediction depends on the centrality of the collision for the different models that are studied.
- The accuracy in eccentricity prediction is found to be dependent on the range of eccentricity considered.
- We find that the eccentricity prediction accuracy improves by the inclusion of the impact parameter as a feature in all these algorithms.
- We discuss how the errors can be minimized, and accuracy can be improved to a great extent in all ranges of impact parameter prediction and eccentricity.
- We also show that the ML algorithms trained by a transport model give accurate predictions for these quantities for both the hydrodynamic and the hybrid models.
- The values of the impact parameter, the eccentricity, and the participant eccentricity can be directly determined from the transverse momentum data using ML models with high accuracy.

Acknowledgements

Thank You!

I would like to acknowledge Department of Science and Technology (DST) Govt. of India for supporting my research through INSPIRE Fellowship (Grant no: IF170627). I would also like to acknowledge Institute of Eminence (IoE), University of Hyderabad for supporting my research.

Eccentricity prediction using vish2+1 data

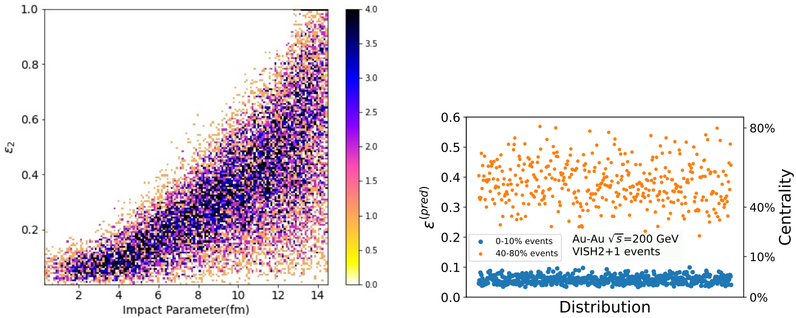


Figure: a) Distribution of impact parameter and ϵ_2 of $\sqrt{s}$= 200 GeV Au-Au collision events, b) Distribution of kNN model eccentricity predictions of 0-10% and 40-80% centrality events of Au-Au collisions at $\sqrt{s}$= 200 GeV from the hybrid (VISH2+1 + URQMD) model.