


**Machine-learning model-driven prediction of the initial geometry in heavy-ion collision experiments**Abhisek Saha <sup>1</sup>, Debasis Dan,<sup>2</sup> and Soma Sanyal<sup>1</sup><sup>1</sup>*School of Physics, University of Hyderabad, Gachibowli, Hyderabad, 500046 India*<sup>2</sup>*Microsoft India (R & D) Pvt. Ltd. Microsoft Campus, ISB Road, Gachibowli, Hyderabad, 500032 India*

(Received 30 March 2022; accepted 9 June 2022; published 5 July 2022)

We demonstrate high prediction accuracy of three important properties that determine the initial geometry of the heavy-ion collision (HIC) experiments by using supervised machine learning (ML) methods. These properties are the impact parameter, the eccentricity, and the participant eccentricity. Although 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 tuned training set that gives multifold improvement in accuracy for all three different heavy-ion collision models. The three models chosen are a transport model, a hydrodynamic model, and a hybrid model. The motivation of using three different heavy-ion collision models was to show that even if the model is trained using a transport model, it gives accurate results for a hydrodynamic model as well as a hybrid model. We show that the accuracy of the impact-parameter prediction depends on the centrality of the collision. With the standard application of ML training methods, prediction accuracy is considerably low for central collisions. Our method increases this accuracy by multiple folds. We also show that the eccentricity prediction accuracy can be improved by inclusion of the impact parameter as a feature in all these algorithms. We discuss how the errors can be minimized and the accuracy can be improved to a great extent in all the ranges of impact-parameter and eccentricity predictions.

DOI: [10.1103/PhysRevC.106.014901](https://doi.org/10.1103/PhysRevC.106.014901)**I. INTRODUCTION**

Ever since the first run of the heavy-ion collision experiments, a lot of studies have been carried out describing and analyzing the data that we get from these experiments [1–3]. The beam energy scan program of the BNL Relativistic Heavy Ion Collider (RHIC) runs the collider experiments using Au-Au nuclei at collision energies from 7.7 to 200 GeV [4–7]. One of their important aims is to search for the critical point in the QCD phase diagram [8]. The matter created in these experiments has a high baryon density. On the other hand, Pb-Pb collisions are conducted at the LHC at collision energies of 2.76 TeV [9], 5.02 TeV [10], the aim of these experiments is to examine the high-temperature region of the QCD phase diagram. The distribution of particles in the initial stage is different for different collision systems and this affects the final stage particle spectra and the anisotropic flows [11]. The primary data we get from these experiments are the transverse momentum ( $p_T$ ) spectra, the rapidity ( $y$ ) spectra, the pseudorapidity ( $\eta$ ) spectra, the particle-antiparticle ratios, and the multiplicity fluctuations. Some phenomena, e.g., the

anisotropic flows, can be obtained directly from these data. But some parameters are difficult to calculate directly from the experimental data. These are the impact parameter, the initial state geometry parameters, (e.g., eccentricities) event plane angles, etc. The impact parameter is the distance between the centers of the colliding nuclei on the transverse plane of the collision. In experiments, the data are always studied with respect to the centrality of the collision because we get different spectra for different centrality collisions. The high-centrality collisions are those where the impact parameter is close to zero i.e., the head-on collisions. The peripheral collisions refer to the higher values of the impact parameter. The collision centrality plays an important role in determining the final particle spectra. The multiplicity distribution of different species is observed to depend on the centrality of the collision. In Refs. [6,12], the multiplicity fluctuations at different centralities are studied at RHIC energies, and in Refs. [9,10,13], the same has been studied at collision energies 2.76, 5.02, and 5.44 TeV, respectively. The centrality is not a property that can be attained directly from the experiments, but it can be calculated with the help of theoretical modeling by using the Glauber model [14] or some other similar model. The impact parameter as well as the initial geometry of the collision is difficult to determine experimentally. This is true especially for the more central collisions. That is why there are various proposals to determine the impact parameter. Apart from different simulations and algorithms, neural networks have also been proposed to determine the impact parameter from the experimental data [15].

---

*Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI. Funded by SCOAP<sup>3</sup>.*

The determination of the impact parameter is related to the charged multiplicity produced during the heavy-ion collision. The charged particle multiplicity has a contribution from hard and soft collision processes, which in turn depend on the number of participants and also on the number of binary collisions. If  $x$  is the fraction of contribution from hard processes then the charged particle multiplicity per unit pseudorapidity can be expressed as

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

where  $n_{pp}$  is the multiplicity per unit rapidity in  $pp$  collisions and  $N_{coll}$  is the number of binary NN collision. The number  $N_{part}$  of participant nuclei can be expressed as a function of the impact parameter [14,16]. If  $T_A(s)$  is the thickness function of nucleus A, i.e., the probability density function of finding nucleons in A, then the number of participants in A at the transverse position  $s$  can be found by multiplying with the probability of binary nuclei-nuclei collisions with the nucleons of the nucleus B at the same position ( $b-s$ ), where  $b$  is the impact parameter. So the total number of participants can be expressed as

$$N_{part}(b) = \int T_A(s) \{1 - \exp[-\sigma_{inel}^{NN} T_B(b-s)]\} ds + \int T_B(b-s) \{1 - \exp[-\sigma_{inel}^{NN} T_A(b)]\} ds. \quad (2)$$

Here the total number of participating nuclei is found out by summing over the contribution from nucleus A and nucleus B. Using Eqs. (1) and (2), the impact parameter, hence centrality, can be estimated by fitting the multiplicity spectra. In this method, the multiplicity fitting must be done for every event to obtain its centrality. An easier way of getting the centrality is to use machine-learning models. Machine learning has been invoked to determine the impact parameter from the experimental data in several papers [17–19]. Using machine learning, we can automate the whole process and the impact parameter can be calculated in an efficient way. The advantage of using machine learning (ML) is that it requires less computational power and computational time, which makes the process more agile. Most of the work in this field is related to the deep neural network algorithms. The convoluted neural network has also been used to make predictions about the impact parameter [20]. The first paper to demonstrate the importance of neural network analysis for improving the accuracy of the determination of the impact parameter is Ref. [21]. Using an artificial neural networks (ANNs) or convolutional neural networks (CNNs) they have effectively determined the impact parameter, but these networks require the tuning of hundreds of parameters. This makes the process computationally expensive. On the other hand, different non-neuronal ML models like state vector machine (SVM), random forest, kNN, etc. require fewer parameters to produce results with an accuracy similar to that of the ANN or CNN models. Thereafter, many papers explored various machine-learning algorithms to obtain more accurate results for the impact parameter.

In this study, we analyze various machine-learning algorithms and provide a rigorous comparison of the accuracy

and efficiency of these algorithms by using well-defined techniques of machine learning to show a critical gap in their prediction accuracy for central collisions. We have mainly focused on three properties, impact parameter, eccentricity, and participant eccentricity. We analyze the errors in the predictions and discuss the causes that lead to these errors. We find that the accuracy is less for the low impact parameters. This is an already-known problem in the determination of the impact parameter. We provide a custom sampling method that shows significant improvement in accuracy over commonly used sampling methods in the ML community. We have also used a particular HIC model for training, while the data from two different HIC models have been used to make the predictions. This indicates that, for a well-defined training data set, the predictions for the impact parameter using the ML model are model independent.

In this study, the transverse momentum ( $p_T$ ) spectra are taken as features and the impact parameter, eccentricity, and participant eccentricity are taken as the target variable which the model must predict. We have used a multiphase transport (AMPT) model to generate the transverse momentum spectra of Au-Au collision events at 200 GeV collision energy [22]. The charged particle multiplicity has been studied previously using the AMPT model [23]. As the target variables are known for fitting, we use supervised machine-learning algorithms. Also, the target is a continuous variable so it can have any real value, hence we use regression algorithms.

The focus of our study is predicting the impact parameter and the eccentricity. Eccentricity is one of the parameters which gives us the initial geometrical shape of the collision region. This also affects the elliptic flow of produced particles, which is one of the important observables used to study collective behavior in heavy-ion collisions. In Ref. [24], the effects of eccentricity fluctuation on the elliptic flow is studied at  $\sqrt{s} = 200$  GeV for Au-Au and Cu-Cu collisions. In a recent study [25], the flow-harmonics are studied as a function of different components of initial anisotropy using the AMPT model. In our study, the learnings and experiences gathered by the ML models from the impact-parameter prediction are passed on to predict the eccentricity of the initial stage of the heavy-ion collision system. We have also looked at how the inclusion of the impact parameter as a feature affects the prediction accuracy. We have made predictions of the initial-state anisotropy, using the impact parameter. The initial-state anisotropy is given by [16],

$$\epsilon_n(b) = \frac{\langle r^n \cos(n\phi - n\psi) \rangle}{r^n}, \quad (3)$$

where  $r = (x^2 + y^2)^{1/2}$ ,  $\psi = \tan^{-1}(y/x)$ ,  $n = 2$  gives the eccentricity, and  $n = 3$  gives the triangularity. The above eccentricities are with respect to the reaction plane. We have also trained the model to predict participant plane eccentricity, which is given by [24]

$$\epsilon_{part} = \frac{\sqrt{\sigma_y^2 - \sigma_x^2 + 4\sigma_{xy}^2}}{\sigma_y^2 + \sigma_x^2}, \quad (4)$$

where the  $\sigma$  are the variances of the positions of the particles,  $\sigma_x^2 = \langle x^2 \rangle - \langle x \rangle^2$ ,  $\sigma_y^2 = \langle y^2 \rangle - \langle y \rangle^2$ , and  $\sigma_{xy} = \langle xy \rangle - \langle x \rangle \langle y \rangle$ . Here  $\langle \dots \rangle$  is the average over the transverse plane.

The AMPT is a transport model which has been used extensively to model the different stages of the heavy-ion collision from the initial collision dynamics to the final stage hadron dynamics. However, like all models, it has certain drawbacks. There are alternate simulations based on hydrodynamics which also give reliable outputs which match well with the data. In this study, we have taken the results from other models, too. This is to test if the predictions of the ML algorithms depend crucially upon the nature of the model used. Our results show that, as long as the models accurately reflect the experimental data, the ML algorithms do not distinguish between the different models. A ML algorithm trained on a specific model gives pretty accurate results when tested with the data generated by a different model.

The two other heavy-ion collision models used in this study are VISH2 + 1 [viscous Israel Stewart hydrodynamics in (2 + 1) dimensions] [26] and a hybrid model made of a hydro evolution model and a hadronic cascade model [27]. These two models are different from the AMPT model which is used to train the ML algorithms. So the ML models train from the  $p_T$  spectra and the impact-parameter data of AMPT events and predicts impact parameters by using test data of  $p_T$  spectra from the VISH2 + 1 and the hybrid models. We choose the initial conditions of different models such that they generate the  $p_T$  spectra close to the one that is obtained in the actual experiments. In this way, we are examining the efficiency of the ML algorithms in a model-independent manner. However, the model-independency is limited only to those models which generate the  $p_T$  spectra close to the experimental  $p_T$  spectrum. The  $p_T$  spectra of the hydro and the hybrid model are fit to the experimental  $p_T$  spectra to measure the effectiveness of ML models to reproduce the experimental data.

In Sec. II, we give a brief description of the heavy-ion collision models used in this study. In Sec. III, we talk about the ML models used in this study. We also describe the parameters used to check the accuracy of different ML models. The learning process of various algorithms as well as the tuning of the hyperparameters are given in this section. We have also used rebalancing techniques to improve the accuracy of the results. These rebalancing techniques are discussed in this section. Section IV discusses the results and the predictions made by the ML models of the eccentricity and the participant eccentricity. It also discusses the ranges of eccentricity where optimum accuracy has been observed. The efficiency of predicting the impact parameter using unknown data of different HIC models and experimental data is discussed in this section. In the end, we show how the accuracy can be improved by rebalancing the dataset. We then summarize the paper in Sec. V.

## II. EVENT GENERATION

### A. The AMPT model

The AMPT model is a publicly available heavy-ion collision model which generates heavy-ion collision events. It is often

used to understand the results obtained from experiments and it has successfully given the results which match well with the experimental observations [22]. There are two versions of AMPT. In both, the initial condition is generated by the HIJING model [28–30]. Here the initial configuration of nucleons is determined by the Glauber model with a Woods-Saxon nuclear distribution. Particle production from two colliding nuclei is given in terms of two processes. In hard processes, the momentum transfer is larger and they are described by pQCD and they produce minijets. The soft processes are those where the momentum transfer is lower and described by the nonperturbative process by the formation of strings. Of the two models, in the default version, the partons recombine with their parent strings after the end of interaction in the partonic state and forms hadrons using the Lund String fragmentation model [31]. In the string melting (SM) version of AMPT, the strings are converted to their valence quarks and antiquarks. The partonic stage interactions are described by Zhang’s parton cascade (ZPC) where the interactions are described by the Boltzmann equations [32]. The scattering cross section of the parton interactions is calculated using pQCD. The simplified relation between total parton elastic-scattering cross-section and the medium induced screening mass is taken as

$$\sigma \approx \frac{9\pi\alpha_s^2}{2\mu^2}, \quad (5)$$

where  $\alpha_s$ , the strong-coupling constant and  $\mu$ , the screening mass, taken as 0.33 and 3 fm<sup>-1</sup>, respectively, for a total cross section of 3 mb. When the partons stop interacting, they are hadronized by a quark coalescence model. Here the nearest quark-antiquark pair is converted into a meson and the three nearest quarks or antiquarks are converted into a baryon or an antibaryon. The hadronic dynamics are described by a relativistic transport (ART) model [33]. We have used both the versions of the AMPT model. In all the collision setup, we use Au-Au collision at 200 GeV collision energy. Different centralities are considered for different purposes. The other settings are the same as the parameters taken in Refs. [23,34].

### B. The VISH2 + 1 model

VISH2 + 1 is a publicly available code where the evolution of the system created in heavy-ion collisions is described by relativistic causal viscous hydrodynamics [26,35,36]. The code has been tested extensively and successfully reproduces the results from experiments [37]. The initial distribution is taken from the Glauber model in terms of energy-momentum tensor  $T^{mn}$ . Then it solves the local energy-momentum conservation equation  $d_m T^{mn} = 0$ , where

$$T^{mn} = eu^m u^n - p\Delta^{mn} + \pi^{mn}. \quad (6)$$

Here  $\Delta^{mn} = g^{mn} - u^m u^n$ ,  $u^m$  and  $u^n$  are the velocity components, and  $p$  is the pressure.  $\pi^{mn}$  is the viscous shear pressure which follows the evolution equation,

$$D\pi^{mn} = \frac{1}{\tau_\pi}(2\eta\sigma^{mn} - \pi^{mn}) - (u^m \pi^{nk} - u^n \pi^{mk})Du_k \quad (7)$$

$D = u^m d_m$  and the symmetric and traceless shear tensor is given by,  $\sigma^{mn} = \frac{1}{2}(\nabla^m u^n + \nabla^n u^m) - \frac{1}{3}\Delta^{mn} d_k u^k$ . The pressure  $p$  and the energy density  $e$  are related by the equation of state (EoS), which is used to solve the hydrodynamic equations. There are three different EoSs used in this study, EoS-L, SM-EoS Q, and s95p-PCE. The EoS-L is based on lattice QCD data where a smooth crossover transition connects the QGP state to the chemically equilibrated hadron resonance gas (HRG) state [37]. SM-EoS Q is the smoothed version of the EOS Q where a first-order phase transition with a vacuum energy (bag constant) connects the noninteracting QGP state to the chemically equilibrated HRG state [38]. The s95p-PCE equation of state is obtained from fits to lattice QCD data for crossover transition at high temperatures and to a partial chemical equilibrium system of the hadrons at low temperatures [39].

In the Israel-Stewart [40,41] framework, the generalized hydrodynamic equation of an energy-momentum tensor  $T^{mn}$ , together with viscous pressure contributions  $\pi^{mn}$  is solved with a collision timescale  $\tau_\pi$  (relaxation time). The longitudinal boost-invariance is implemented and seven equations are solved, three for the  $T^{\tau\tau}$ ,  $T^{\tau x}$ , and  $T^{\tau y}$  and four for the  $\pi^{mn}$ . Here a flux-corrected transport (FCT) algorithm called a sharp and smooth transport algorithm (SHASTA) [42] is used to solve the hydrodynamic and kinetic equations. It has two stages. In the transport stage, the multidimensional calculations are simplified in terms of geometric interpretation, which is followed by an antidiffusive or corrective stage. This technique is also applied in codes like AZHYDRO [43]. The final spectra are obtained on a freeze-out hypersurface where the fluid stops interacting. The freeze-out is computed using the Cooper-Frye procedure [44] at a decoupling temperature  $T_{\text{dec}}$ . Here the ISPECTRA (iS) code is used, which is a fast Cooper-Frye particle momentum distribution technique that gives discrete momentum distribution of the desired hadron species [45]. We get events of emitted hadrons similar to the events generated in experiments which are then used for ML model predictions.

### C. Hybrid model

We have used the IEBE-VISHNU code package [27], which is a hybrid model made by combining a  $(2 + 1)$ -dimensional viscous hydrodynamic model and a hadronic cascade model. Instead of using the whole package, we have used the modules separately for better handling of inputs and outputs. The output of the iS particle sampler obtained at the end of hydro evolution is used for hadronic rescatterings. Here, the UrQMD after-burner package is used to serve this purpose. After the particlization, the hadrons are produced on the hypersurface with individual production time and location. The position and momenta along with the ids are then written in a standard OSCAR1997A format which is suitable for hadronic rescattering [46]. This is done using the OSCAR to UrQMD converter routine. This also propagates all the hadrons backward in time so that all of them have a fixed initial time and the Boltzmann collision integral can be performed in the UrQMD model.

Ultrarelativistic quantum molecular dynamics (UrQMD) is a transport model where the dynamics of the hadrons are mod-

eled [47,48]. UrQMD can generate a whole collision system starting from the nuclear collision to the hadronic spectra but here we have only used it to get the hadronic evolution. The interaction among the hadrons is evaluated using the Boltzmann equation for the distribution of all hadrons. The system evolves through binary collisions or by  $2-N$ -body decays. Fifty-three baryon species and 24 different meson species, along with their resonances, antiparticle states, and isospin-projected states are considered in the UrQMD interactions. The interaction among the hadrons and their resonances in this model are described in Ref. [48].

## III. MACHINE LEARNING METHODS

### A. Machine-learning algorithms and tuning of hyperparameters

As mentioned in the introduction, there are various ML algorithms that we have tested for this study e.g.,  $k$ -nearest-neighbors, gradient boosting regression, decision trees, etc. Details of these ML algorithms are available in Ref. [49]. The accuracy of these models has been tested using standard measures such as  $R$ -squared, the root mean square error (RMSE), the mean squared error (MSE), and the mean absolute error (MAE). After running various ML algorithms, we find that although all the algorithms give very good predictions for the impact parameter, only three of them perform well for the eccentricity prediction. Hence, we concentrate only on these three algorithms. They are the  $k$ -nearest-neighbors (kNN), extra-trees regressor (ET), and the random forest regressor (RF) model. In kNN model, the target is predicted by doing a local interpolation of the target associated with the  $k$  nearest neighbors of the training dataset [50]. ET and RF are types of ensemble methods. In RF, the decision trees are made during the training and a mean of the ensemble is calculated [51]. In ET, randomized decision trees are considered that are made of subsamples of the training dataset [52]. We have used a 10-fold cross-validation (CV) to obtain a robust estimate of the parameters [53]. This also gives a bias-variance trade-off.

Since we have used these ML algorithms for studying the data from three different HIC models, we have standardized the data before processing them. In this study, we are using the  $p_T$  spectra of charged particles as features in the dataset. The  $p_T$  spectra is obtained for the midrapidity region with a rapidity window of  $-0.5$  to  $0.5$ . All the  $p_T$  bins have a different range of values. The difference is more significant when we compare a lower- $p_T$  bin with a higher- $p_T$  bin. Thus, it is important to make them standardized. This makes the model compatible with a new dataset coming from a different HIC model. Two types of scaling are used in this study, (i) the standard scaler or  $Z$ -score normalization and (ii) the min-max scaler [54]. To use both of these scaling techniques, we have used python sklearn.preprocessing library [55]. In most of the cases discussed in this study, we observe that the  $Z$ -score method provides an accuracy greater than the min-max scaling by 4% to 6%. So, in all the cases, we have used the  $Z$ -score standardization.

After standardization, the  $p_T$  spectra serve as the features in the dataset and the target variables are the impact parameter, the eccentricity, and the participant eccentricity. When the

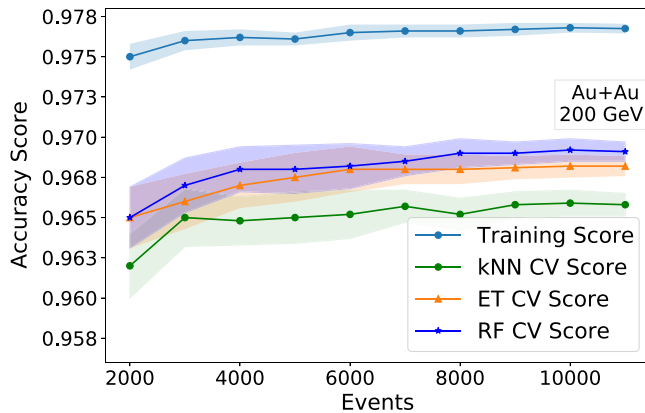


FIG. 1. The learning curve of the kNN (green dots), ET (orange triangles), and RF (blue stars) models. Accuracy scores as a function of the number of events are shown and attain saturation after 3000 events for the training set (sky line), and after 6000–8000 events for the test sets. The shaded region is the standard deviation.

impact parameter is used as the target variable, only the  $p_T$  spectra is used as feature variables. For the other targets, the predicted impact parameter is included in the dataset as a feature variable as all the other targets have a dependency on the impact parameter. In this way, all the dependent variables

can be measured just by giving the  $p_T$  spectra as inputs. A standard training and test set separation was done for model evaluation.

It is important to have enough events to achieve the best accuracy without consuming too many computing resources. The learning curve of a machine-learning model tells us how effectively a model is learning throughout its running time. We present the learning experience as a function of events. In Fig. 1, the learning curves of the kNN (green circles), ET (orange triangles), and RF (blue stars) models are shown, where the changes in the cross-validation (CV) accuracy are represented with the number of event iterations. The training score curve is shown only for the kNN model (sky color circles) which shows the accuracy while fitting the training data to the model. In the training case, the accuracy comes to a saturation very early around 3000 events, while the cases of test data accuracy shown by the other curves saturate around 6000 to 8000 events. All the learning in this study are performed over 10000 events. The shaded area represents the standard deviations in the accuracy score.

In Fig. 2, the accuracy plots for impact-parameter predictions using kNN [Fig. 2(a)], ET [Fig. 2(b)], RF [Fig. 2(c)], and linear regression (LR) [Fig. 2(d)] models are shown. The ML models are trained by using the charged particle  $p_T$  spectra data of the AMPT-SM model. The linear regression algorithm finds a linear relationship between a dependent and one or

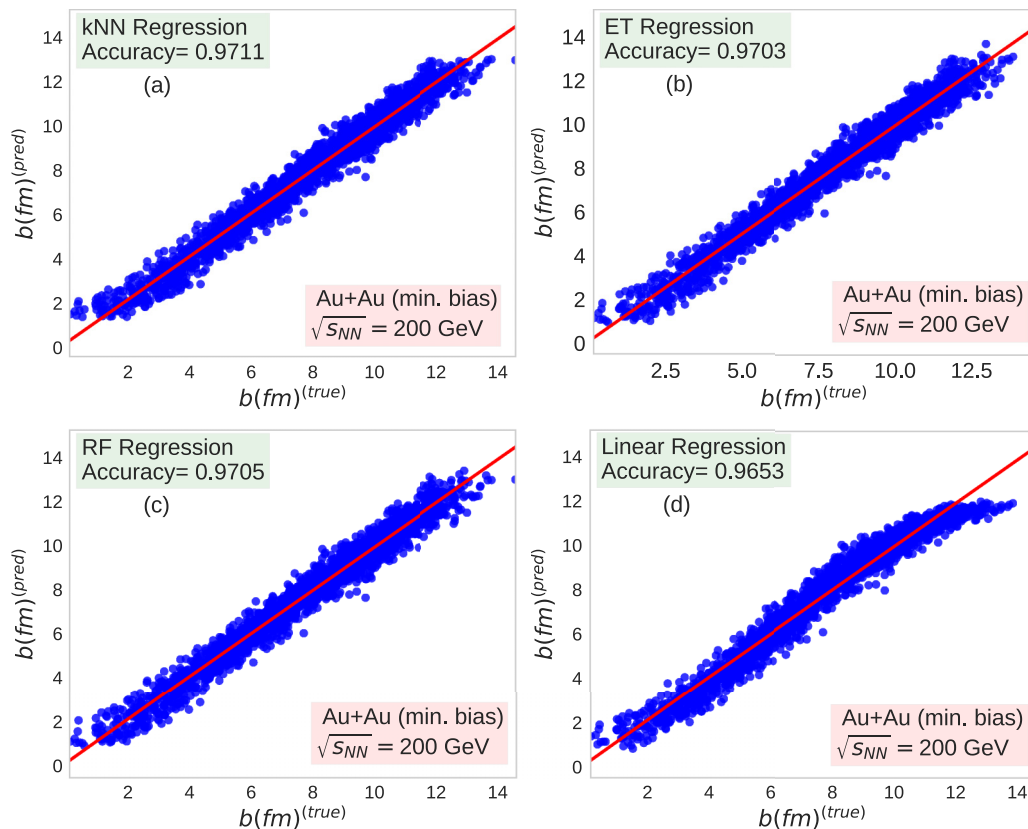


FIG. 2. Impact-parameter prediction using the (a) kNN, (b) ET, (c) RF, and (d) LR models with accuracy scores 97.11%, 97.03%, 97.05%, and 96.53%, respectively, for events of Au + Au system at collision energy 200 GeV. These plots are obtained for a random train and test set split of input events.

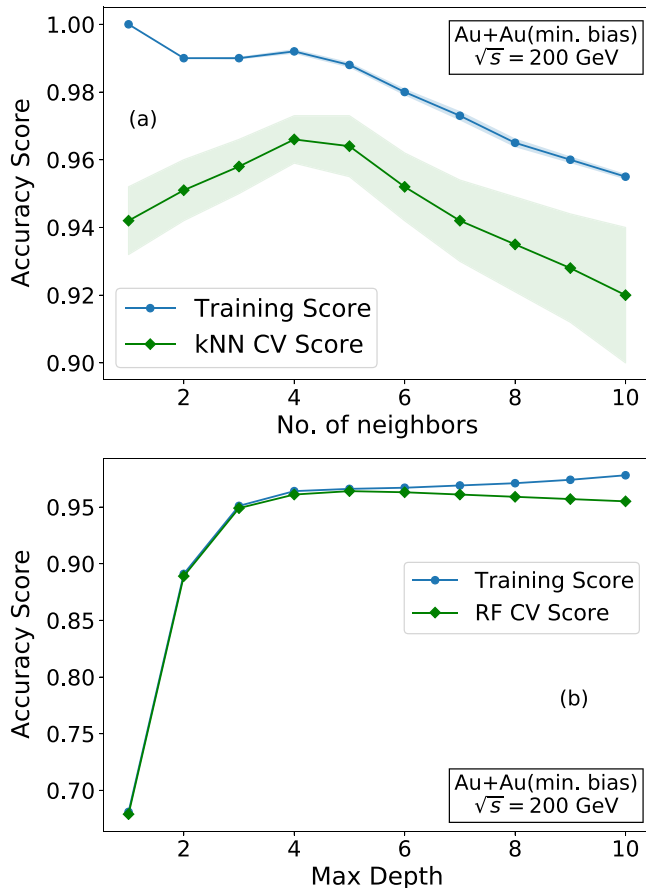


FIG. 3. Change in accuracy as a function of hyperparameters. (a) kNN model with the number of nearest neighbors hyperparameter, (b) random forest with max depth hyperparameter.

more independent variables [56]. The prediction is performed by using a test dataset containing  $p_T$  spectra of more than 4000 events of minimum bias Au + Au collision at 200 GeV. The red line drawn here is the optimum accuracy line and the blue points are the predictions made by the model. The accuracies achieved are 97.11%, 97.03%, 97.05%, and 96.53% for the kNN, ET, RF, and LR models, respectively. All of these accuracies are observed for a random train-test dataset split. The tenfold cross-validation scores of these models are 97.04%, 97%, 97.01%, and 96.56%, respectively. We get an accuracy of more than 95% for the kNN, ET, and RF models when the ML models are trained using the default AMPT model data. In the case of impact-parameter predictions, most of the machine-learning algorithms give a fair level of accuracy without tuning any of the hyperparameters except in certain critical impact-parameter regimes.

It is known that the choice of parameters can affect the accuracy of a model. Hyperparameter tuning was done to fix the parameters with minimum error validation set. In Fig. 3(a), the change in the accuracy of a kNN model is shown as a function of the number of nearest neighbors hyperparameter. For every configuration, the model is trained using 12 000 events of minimum bias Au + Au collision, and the impact parameter is taken as the target variable. The highest accuracy

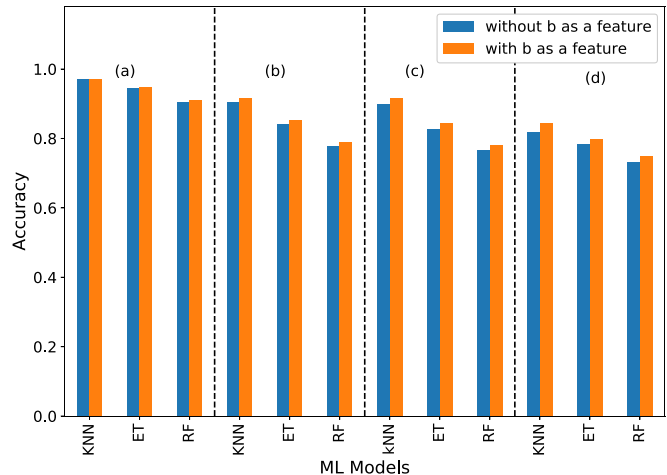


FIG. 4. Effect on the eccentricity prediction accuracy by the inclusion of impact parameter as a feature for different centrality (%), (a) 0%–10%, (b) 10%–40%, (c) 40%–80%, (d) minimum-bias events. The orange bar represents accuracy with impact parameter as a feature and blue bars represent accuracy without impact parameter as a feature.

is attained by the model when the number of nearest neighbors is four or five. This is shown by the green curve which gives the tenfold cross-validation score and the shaded region is the standard deviation. The training score shown by the blue line has a score of 1.00 when the number of nearest neighbors is 1. This is a case of overfitting. For the random forest (RF) model [see Fig. 3(b)], the choice of hyperparameter is the maximum number of levels of the tree. We find that the accuracy saturates for the hyperparameter value of four or five. Like the RF model, we get the maximum CV score of the ET model when the max-depth hyperparameter is four or five. Although the above-mentioned parameters are those that most hamper the accuracy, we fix the other hyperparameters by running the RandomSearchCV function of the sklearn library and checking the accuracy for a different combination of hyperparameters.

As discussed earlier, in Fig. 4, we see how the inclusion of the impact parameter as a feature affects the accuracy of eccentricity prediction. As is seen in earlier studies, the eccentricity depends on the centrality of the collision. Here we found that, by the inclusion of the impact parameter as a feature, the accuracy increased in all the centrality ranges.

The errors in ML can be reduced by determining the highly correlated features in the data. The principal components analysis (PCA) is the most popular technique used for feature reduction of a large dataset [57]. In this work, we tried the “SelectFromFeature” function from the sklearn library and the PCA method to reduce the colinearity and compared the outcomes with the already-achieved accuracy using all the features. We have only shown the result of the PCA method. In Fig. 5(a), the accuracy score of a kNN model, and in Fig. 5(b), the accuracy score of an ET model are shown as a function of the number of principal components is used. Here the accuracy is observed for the impact-parameter predictions by using the  $p_T$  distribution dataset of 12 000 minimum-bias

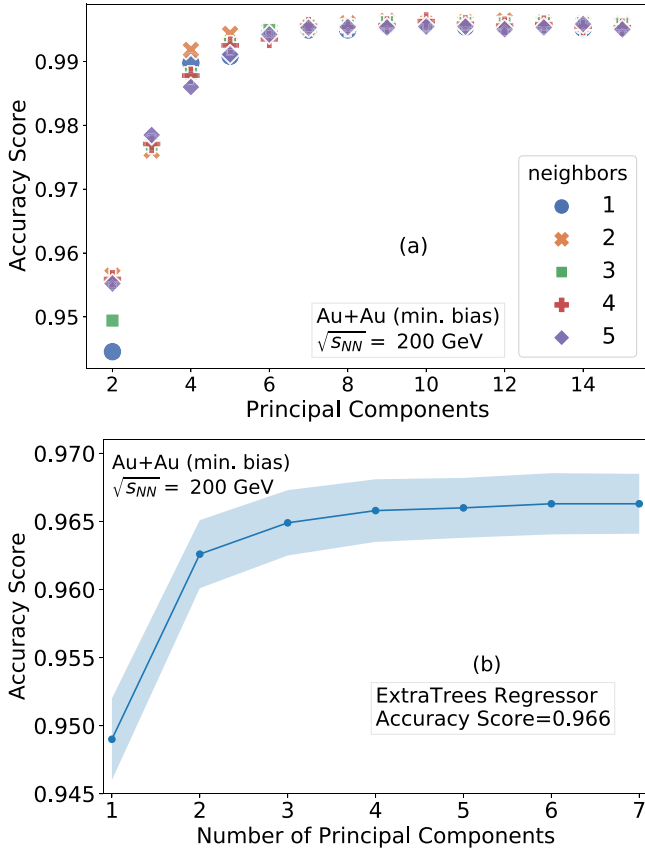


FIG. 5. Accuracy of a (a) kNN model and (b) an ET model as a function of the number of principal components used.

Au-Au collision events at 200 GeV collision energy. The saturation in the accuracy score is achieved for the use of 7 or more principal components in both cases. Also, using seven components, a variance coverage of 95% can be achieved in the case of impact-parameter predictions. So, it is safe to use seven to eight principal components to get good accuracy without losing any major information. We used seven to eight components for the impact-parameter determination. We also found that at least ten features or ten principal components are needed to obtain an accurate result for the eccentricity and the participant eccentricity. This is expected as the data that we are using is the transverse momentum data. Since the impact parameter is known to be correlated with the transverse momentum data hence, we need a smaller number of features to obtain a high accuracy of prediction [58] for the impact parameter as compared with the eccentricity. In all the eccentricity predictions we use the PCA function to transform the features.

### B. Custom resampling for unbalanced training set

The  $p_T$  spectra we used as a feature are comprised of imbalanced datasets. As we have considered the  $p_T$  spectra of minimum bias events, there are a smaller number of events for lower impact-parameter values. Thus, the event distribution of  $p_T$  spectra is left-skewed. The imbalance in the data affects the prediction accuracy of the impact parameter and eccentricity

in the lower- $b$  region ( $b \leq 1$  fm). As is well known in the literature, the impact parameter is not directly accessible to the experiments. Bass *et al.* [59] have pointed out that, although most of the experimental observables depend on the impact parameter, the different methods of impact-parameter estimation are usually optimized for the larger impact-parameter range. This means that the experimental results for head-on collisions pertaining to the lower impact-parameter range will have higher errors due to the inaccuracy of the impact-parameter calculations. Currently, efforts are being made to improve the prediction of the impact parameter in the lower impact-parameter range. This is very important because there are considerable experimental results from head-on collisions which can be better analyzed with an improved prediction of the impact parameter in the lower range. So our aim is to improve the accuracy of the impact parameter in the lower range by balancing the data set appropriately.

There are a few sampling techniques in machine learning for rebalancing datasets, e.g., SmoteR and ADASYN [60,61]. These are python packages that increase (over-sampling) or decrease (under-sampling) the minority and majority data classes, respectively, by using the neighboring data. We evaluated both techniques with all possible hyperparameter combinations. The results discussed in the next section (Sec. IV C) indicate that we do not have a sufficient increase in the accuracy of the predictions and there is a high chance of central events being predicted as noncentral ones.

We then adopt a method of rebalancing the data set using class weights, where different classes are the different impact-parameter regimes. The various combinations of distribution region and weights were evaluated through an exhaustive grid search. Based on test set minimum error, we selected events with impact parameter  $\leq 1.0$  fm to be in category 1 and the rest in category 2. The weights assigned to the two classes are in the ratio 4 : 1. This technique has helped us to reduce the errors further and the results are discussed in detail in Sec. IV C.

## IV. RESULTS AND DISCUSSIONS

### A. Impact parameter and eccentricity prediction

As discussed earlier, the eccentricity is one of the key parameters in heavy-ion collisions. It gives information about initial-state geometry and also affects the final-state particle flows. But like the impact parameter, it is difficult to measure eccentricity directly from experiment. Here in this study, the models which are used to get the impact-parameter prediction are also used in eccentricity prediction. In fact, ET, kNN, and RF are the three best-performing algorithms in the case of eccentricity prediction.

Figure 6 shows the prediction plot of eccentricity using the kNN [Fig. 6(a)] and ET [Fig. 6(b)] models. The accuracies obtained are 97.84% and 95.47%, respectively. This is observed for a randomly split train-test dataset of minimum-bias Au + Au events. The models are trained using 10 000 randomly selected events and the testing is performed over 2000 events, which are shown in Fig. 6. The tenfold cross-validation score is also closer to the accuracy obtained using the random

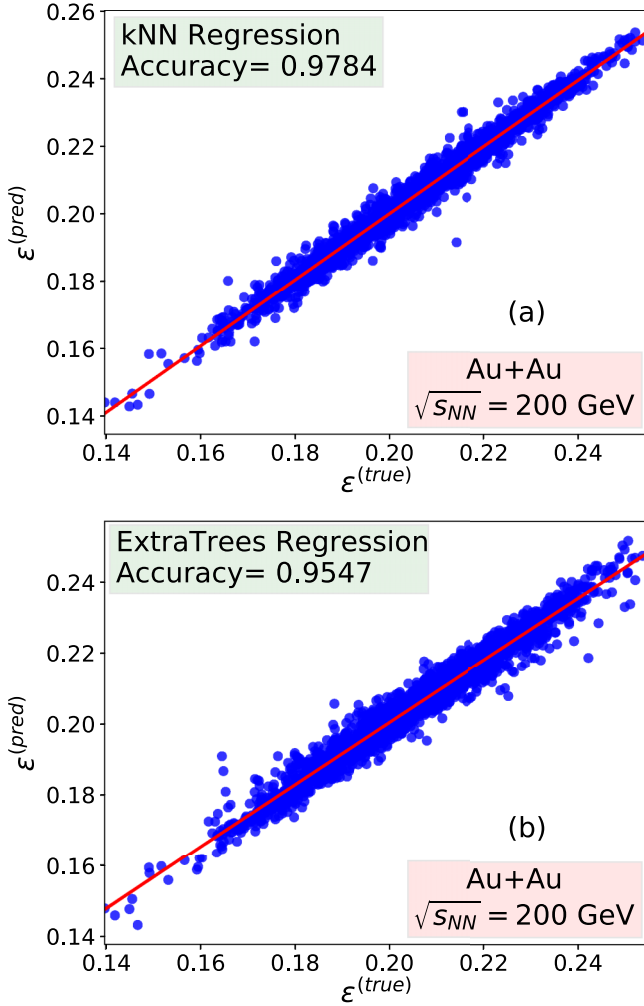


FIG. 6. Eccentricity prediction using the (a) kNN and (b) ET models with their accuracy score 97.84% and 95.47% for events of the Au + Au system at a collision energy of 200 GeV. These plots are obtained for a random train and test set split of input events.

train-test split dataset, 97.52% for the kNN model and 95.18% for the ET model. The tenfold CV score of RF model is 91.95%. We get accuracy between 87% to 93% when the ML models are trained by using the default AMPT model data.

Figure 7 shows the prediction plot of participant eccentricity using the kNN [Fig. 7(a)] and ET [Fig. 7(b)] models. The accuracies obtained are 98.16% and 96.21%, respectively. In this case, the tenfold cross-validation scores are 97.58% and 95.25% for the kNN and ET models, respectively, and 93.78% for the RF model. In Table I, a comparison of accuracy for  $\epsilon_3$  (triangularity) is shown. We have used Eq. (3) to obtain  $\epsilon_3$ . Here also the kNN, ET, and RF models perform better than the other ML models. All of the three have an accuracy of over 90%. The light gradient boosting machine (LGBM) model also has an accuracy of over 88% after a tenfold cross validation. This is a tree-based machine-learning model where the tree grows vertically (leaf-wise) [62].

In the eccentricity prediction figures (Fig. 7), a small range of eccentricity (0.22–0.32) is taken for the model fitting and predictions. It is specifically the range where the maximum

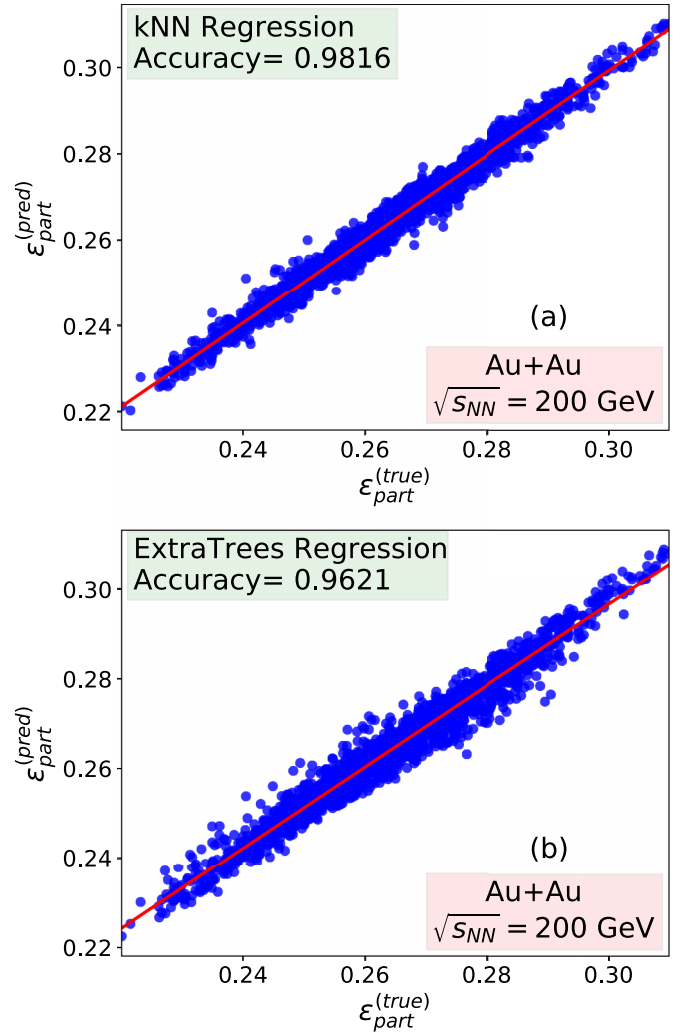


FIG. 7. Participant eccentricity prediction using the (a) kNN and (b) ET models with their accuracy score 98.16% and 96.21% for events of the Au + Au system at a collision energy of 200 GeV. These plots are obtained for a random train and test set split of input events.

prediction accuracy is obtained for all the models. One of the reasons behind this is that the distribution of eccentricity over the events is not isotropic. In Fig. 8(a), the distribution of participant eccentricity is shown. Here the vertical axis represents the normalized number of events, and the

TABLE I. Tenfold cross-validation accuracy of ML models for  $\epsilon_3$  predictions of minimum-bias Au-Au events at  $\sqrt{s} = 200$  GeV.

Model	$R^2$	MAE	RMSE
$k$ -nearest-neighbor regressor	0.9762	0.001	0.0013
Extra trees regressor	0.9574	0.0013	0.0017
Random forest regressor	0.9216	0.0017	0.0023
Light gradient boosting machine	0.8807	0.0022	0.0029
Decision tree regressor	0.7581	0.0024	0.0041
Gradient boosting regressor	0.6309	0.004	0.005



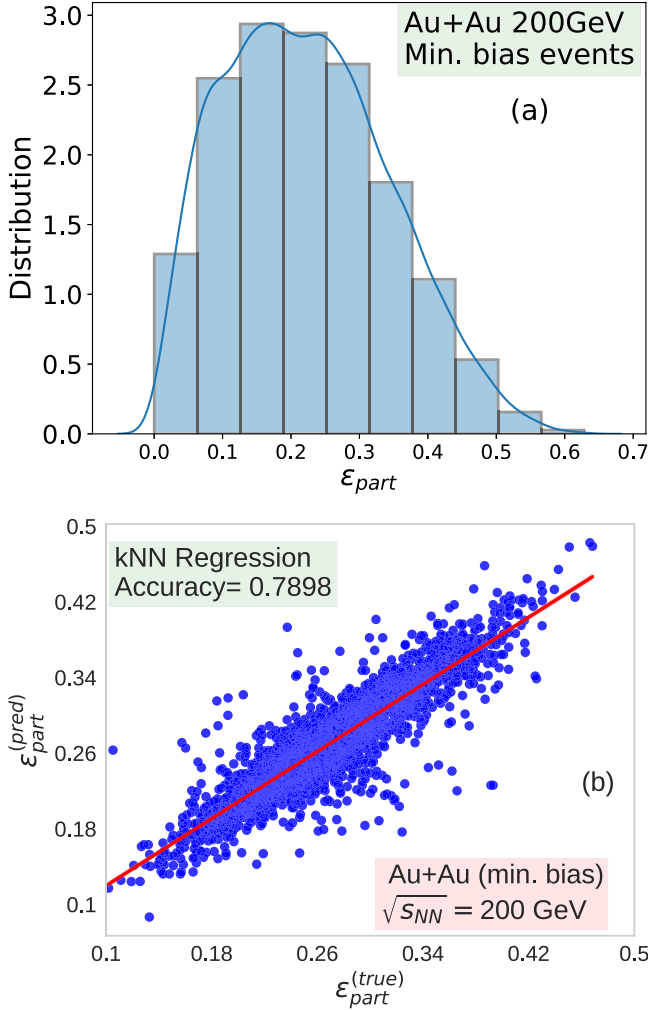


FIG. 8. (a) Histogram plot of participant eccentricity distribution and (b) prediction plot of  $\epsilon_{\text{part}}$  for higher  $\epsilon_{\text{part}}$  range using the kNN model of minimum-bias Au-Au collision events at  $\sqrt{s} = 200$  GeV given by the AMPT model.

horizontal axis gives the eccentricity range. The peak in the distribution is observed for eccentricities between 0.15 and 0.25. The distribution is thus skewed, which means that we have an imbalanced dataset. So, the eccentricity of maximum events that are occurring fall in a particular range. Hence the model fits well in this range of eccentricity because of a larger number of fitting points. From the graph, we see that the range of eccentricity can be increased further from 0.1 to 0.5. In Fig. 8(b), a prediction plot of participant eccentricity using the kNN model is given for a larger range. Here the events are considered which have eccentricities in the range from 0.1 to 0.5. So, the range has now become three times wider than the previous cases. We observe that the points are wider from the center and away from the  $45^\circ$  red line compared with the points in Fig. 7(a). We also see some points which are away and isolated from the bulk distribution. The accuracy is lowered to 78.98% from its previous value of 98.16%. The tenfold cross-validation score is 76% in this case, which is a fair amount of accuracy although it is much lower compared

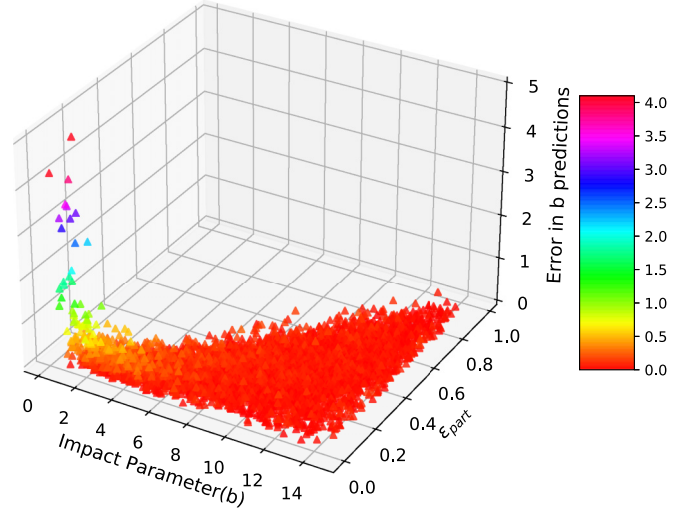


FIG. 9. Error in the prediction of impact parameter as a function of impact parameter and eccentricity distribution. This is for 200 GeV Au-Au collisions and the prediction is obtained by using a kNN model.

with the maximum accuracy. This means that the range of accuracy can be fixed according to the requirement of the problem. To accommodate a wider eccentricity range, we have to compensate with accuracy. We have also applied different ML algorithms to obtain the accuracy at different collision energies from 20 to 200 GeV for the impact parameter, eccentricity, and the participant eccentricity predictions. For lower collision energies, the number of events required to train a ML model is higher compared with the number of events required for higher collision energies. This is because high multiplicity events are generated at higher collision energies. Thus, the event-by-event averages become stable.

In Fig. 9, we plot the error in the impact-parameter prediction as a function of the impact parameter and the eccentricity distribution. The error here is the relative error (RE) which is given by  $RE = |(b_{\text{pred}} - b_{\text{org}})/b_{\text{org}}|$ , where  $b_{\text{pred}}$  and  $b_{\text{org}}$  are the predicted and original value of impact parameter, respectively. We observe that, for all eccentricity and impact-parameter ranges, the error is low except for the region where the impact parameter is less than 2 fm. In the majority of the distribution, the difference in the prediction and the original impact parameter is less than 0.5 (shown by the red points) and in some cases, it is less than 1. But for the lower range of impact parameters ( $b < 2$  fm) and eccentricities, we find the difference becomes significantly larger. This is also because of imbalance in the data as discussed. There are comparably large errors for MC-Glauber model predictions of low impact parameters. Large discrepancies are also obtained for events of UrQMD and AMPT with higher charge particle when fit with MC-Glauber model data which are shown in Ref. [63]. In Ref. [64], large errors are observed for the fitting of Glauber model data to the ALICE data. Our results are similar to the recent results obtained from ML using other models like UrQMD, where it was shown that the impact parameters are determined efficiently in all the regions except the very central and the very peripheral regions [65]. This is adequately reflected in

Fig. 9, where a large amount of error is found in the very central region.

### B. Results from the different heavy-ion collision models

To check the model dependency, we have used the data from other HIC models and obtained the prediction of the impact parameter. The training of a ML model has been done using the AMPT model data, but the predictions are made for other heavy-ion collision models. The other HIC models used in this study are the VISH2 + 1 model [26] which is a hydrodynamic evolution code and a hybrid model made of the VISH2 + 1 and the UrQMD models [27]. In the hybrid model, the UrQMD code is used for later-stage hadronic rescatterings. The reason for using multiple models is because we want the test set and the training set to come from different models giving the same  $p_T$  spectra. This would mean that as long as the  $p_T$  spectra are the same, the ML algorithms will not know which model simulated the test data. As done in the previous cases, here also we have used the transverse momentum spectra of the AMPT model as the features and impact parameters of the corresponding AMPT events as targets for the ML model training. The AMPT events considered here are the minimum-bias Au-Au events of 200 GeV collision energy. The  $p_T$  spectra of VISH2 + 1 and hybrid model are obtained at the same collision energy and at specific centralities with impact parameters ranging from 0.1 to 14 fm. The parameter settings of the VISH2 + 1 model in this study is similar to the parameters considered in Ref. [37], with the Glauber initial condition, shear viscosity to entropy density ratio  $\eta/s = 0.16$ , and decoupling temperature  $T_{\text{dec}} = 160$  MeV. The box plot is obtained for the s95p-PCE equation of state. For the hybrid model, we set the  $\eta/s$  to 0.08, the equation of state used is s95p-PCE, and  $T_{\text{dec}} = 165$  MeV. We got 5000 events each from both of these models at all the impact-parameter ranges separately and fit the average  $p_T$  spectra to the experimentally obtained  $p_T$  spectra [66,67]. We considered the 0.15 to 1.4 GeV/c  $p_T$  range to fit with the experimental spectra and also for ML model training. In this range, VISH2 + 1 data fit well with the experimental data. By doing this we are ensuring that the data are similar to the experiments. In an approximate manner, we are also examining the performance of ML models in case of the use of experimental data as test data for prediction. As we do not have event-by-event experimental data at specific centralities, we have used different HIC models to generate the  $p_T$  spectra. In this way, we are able to obtain the error distribution of the predictions given by the ML model for a large number of events at specific centralities.

All the ML models considered in this study, e.g., kNN, RF, ET, and LR, perform reasonably well for impact-parameter prediction for unknown HIC model test data. In Figs. 10(a) and 10(b), we show the error plots of impact parameter predictions by the kNN model for VISH2 + 1 and hybrid UrQMD models, respectively. These are relative errors, and the box represents the distribution of errors. The middle line inside the box represents the median error, which is in the middle of the box. The top and bottom lines represent the 25th and 75th percentile of the error distribution. The green point is the mean error. In all the boxes, i.e., at all the centralities,

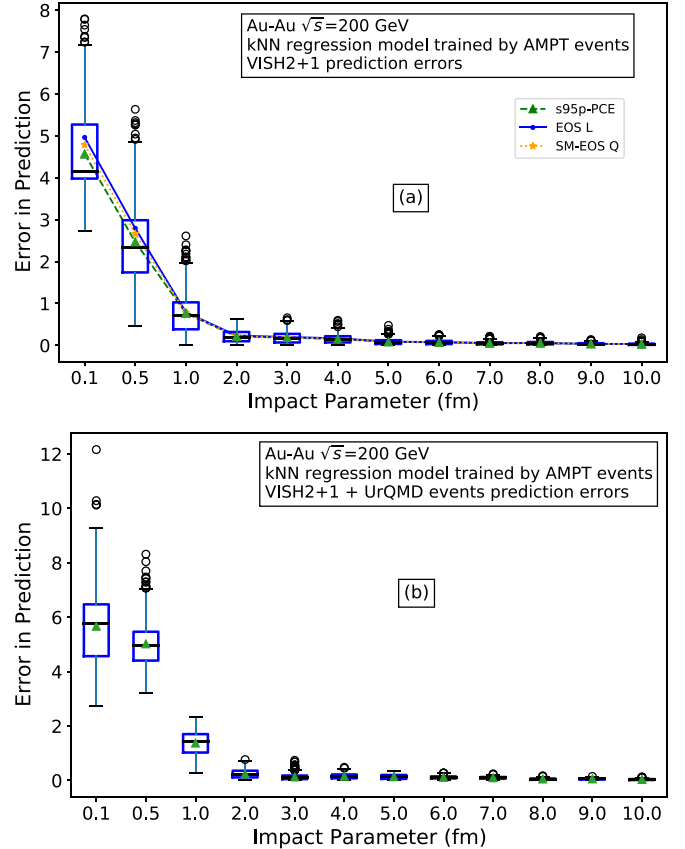


FIG. 10. Error plot of impact parameter predictions by kNN model of different centrality events of (a) VISH2 + 1 and (b) UrQMD simulations. The three lines in panel (a) show the mean errors in impact-parameter prediction for different EoS.

the errors are in a normal distribution. This shows a good prediction by the ML model. The end circles are the outliers which are less in number. In both the figures, we find that the error goes down for the higher-impact-parameter events. Above the impact parameter of 2 fm, the prediction errors are very close to zero. Above  $b = 10$  fm, the errors stay low, continuing the previous trend. The three lines in Fig. 10(a) show the mean errors in impact-parameter prediction for different equations of state (EoS). The green (dashed), blue (solid), and yellow (dotted) lines represent the mean errors for s95p-PCE, EOS-L, and SM-EoS Q equations of state, respectively. For all the EoS, the trend of error distribution is similar. In Fig. 10(a), we see that, for 0.1 and 0.5 fm impact-parameter events, the relative prediction errors are more than three times and in Fig. 10(b), it is more than five times compared with the real impact parameters. We have considered Au-Au collision events where the most-central collisions of 0%–5% centrality are comprised of events of impact parameter range 0 to 3.31 fm [16]. As has been discussed, in the case of experiments, the centrality is found by using the Glauber model. Thus, it is difficult to assign a specific impact parameter, especially in the case of the most-central events. We have seen the same nature in error prediction in Fig. 9 while working with only the AMPT events. In that case, training of the ML

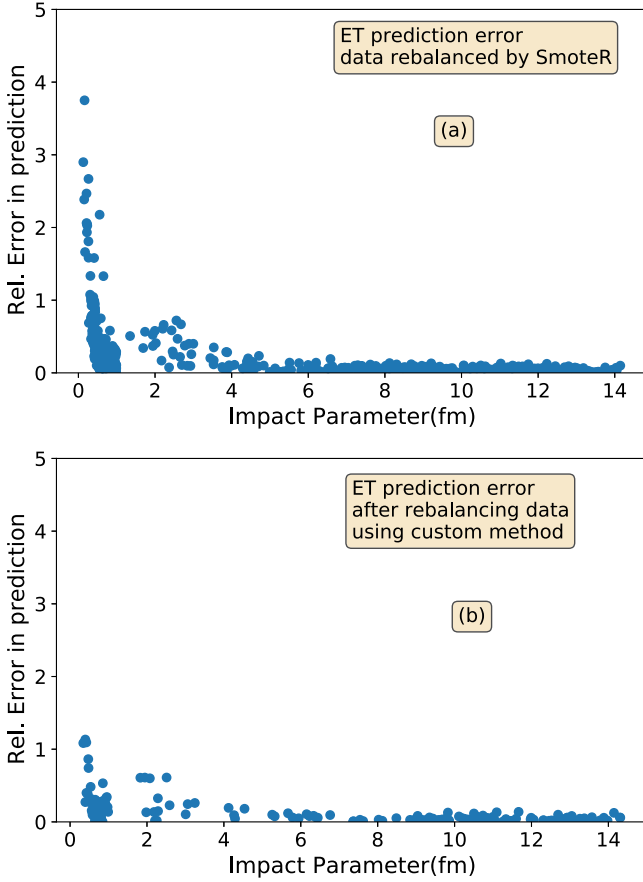


FIG. 11. Error distribution of ET model of impact-parameter predictions of Au-Au collisions at  $\sqrt{s} = 200$  GeV. The training set is rebalanced using (a) the SmoteR method, and (b) a custom method of giving weights to the input data.

model and the testing are performed with the AMPT events. This is due to the imbalance in the dataset that we are using for the ML model training. Although we get a similar nature of error distribution in all the cases, we used the Glauber initial conditions for the hydro model input. The initial condition from the color glass condensate model can give different  $p_T$  spectra. To check whether the ML models are effective in

this scenario, the parameters of the hydro model should be adjusted such that the  $p_T$  spectra obtained match well with the experimental  $p_T$  spectra.

### C. Results from rebalancing the data set

A large error [Figs. 9 and 10] is observed in the prediction in the lower impact-parameter range due to the imbalance in the impact-parameter distribution in the training set. We overcome this through a custom sample weighing method, as mentioned in Sec. III B. As mentioned previously in Sec. III B, initially, we used standard packages to rebalance the data. The results are shown in Fig. 11(a) for one of the methods. The others give similar results. Although the error comes down in the lower impact-parameter region compared with the errors obtained in Fig. 9, still we get enough errors that would give a wrong estimate for the low-impact-parameter events. Finally, we give the results of our custom rebalancing method which has been described previously in detail in Sec. III B in Fig. 11(b). With our custom method we were able to minimize the error to less than 1, as shown in Fig. 11(b). This error is acceptable in this impact-parameter range as the prediction made in this range will always fall in the most-central collision category (0%–5%) for the Au-Au collisions.

It is also interesting to see how the AMPT trained models predict eccentricities when they are introduced to other HIC model data. In Fig. 12(a), we show the distribution of eccentricity with the centrality of 200 GeV AMPT collision events. However, the color plot suggests that there is a linear relationship between the average eccentricity and impact parameter of collision events. We see that the range of eccentricity is lower for lower-impact-parameter values. As we go for higher-impact-parameter events, the range of eccentricity becomes larger. A similar observation has been shown previously in Ref. [68]. In Fig. 12(b), we show the distribution of eccentricity predictions of VISH2 + 1 events for two centrality range. Here also the ML model is trained using minimum-bias AMPT events. The orange dots are the prediction events of 40%–80% centrality and the blue dots are the prediction events of 0%–10% centrality. In the case of 0%–10% centrality range, we get eccentricity distribution in 0 – 0.15 range, which is also in the range of original

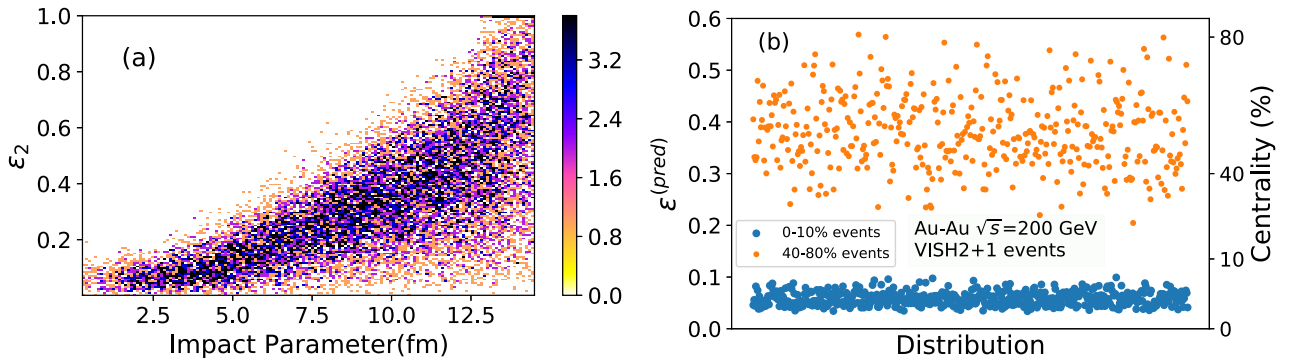


FIG. 12. (a) Distribution of eccentricity with impact parameter of minimum-bias Au-Au collision events at  $\sqrt{s} = 200$  GeV. (b) Distribution of eccentricity predictions by kNN model of 0%–10% and 40%–80% centrality events of Au-Au collisions at  $\sqrt{s} = 200$  GeV from the VISH2 + 1 model.

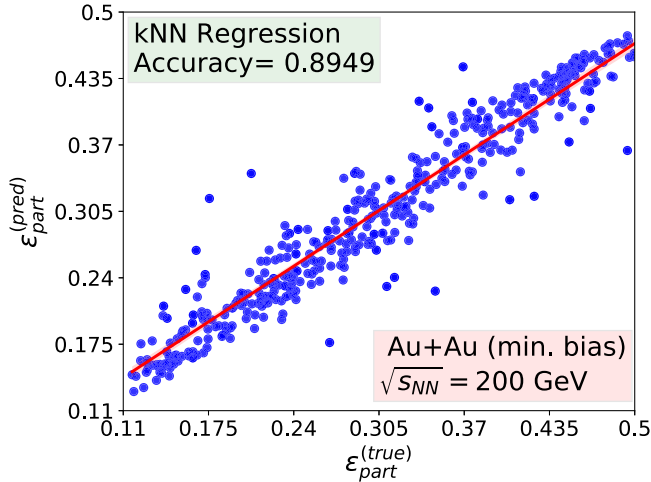


FIG. 13. Participant eccentricity predictions of Au-Au collision events at  $\sqrt{s} = 200$  GeV after rebalancing the data using the custom method.

distribution shown in Fig. 12(a). We get a larger range in eccentricity values in the higher-impact-parameter range, the prediction also gives us the same, as represented by the orange dots. Although this shows the model independence characteristics of the ML models, it is only examined for the Glauber initial conditions of VISH2 + 1 model. We have not used the color glass condensate initial conditions as it is known that it gives a larger anisotropy but it would be interesting to see how the model would perform in that case. We plan to look at these in a later work.

We have shown an imbalance in the eccentricity distribution in Fig. 8(a). Due to this imbalance in the distribution, we get an accuracy of more than 95% in eccentricity prediction only when we consider a small range. For bigger range, the CV accuracy dropped down to 76%. A good amount of accuracy can also be achieved for a higher range of eccentricity distribution if the data are rebalanced in a suitable format. We have tried a similar rebalancing technique as is done for the impact-parameter prediction. We took the same number of events from each of the distribution bins and trained the model. The prediction plot is shown in Fig. 13. We observe that the event points are much closer to the optimum accuracy line (red line) compared with Fig. 8(b), which also has the same range of eccentricity. The accuracy obtained in this case is 89.49% with a cross-validation score of 91%. So, using these data-rebalancing techniques, one can improve the performance of these ML models for the prediction of the impact parameter as well as the eccentricities.

## V. CONCLUSIONS

We have trained different machine-learning models to predict various initial stage parameters of a heavy-ion collision system using the AMPT model. We have used the  $p_T$  spectra for training and testing of the ML models. We have chosen these spectra because they are one of the direct observables in heavy-ion collision experiments. We have observed their learning processes and made changes to the hyperparameters

to get an optimum accuracy in the predictions. Out of the various models tested, we have chosen four models, kNN, RF, ET, and LR, for the prediction of the impact parameter. All the models performed well in the impact-parameter prediction. Most of the algorithms have shown an accuracy of more than 90% in the prediction of the impact parameter. In the case of the eccentricity, and the participant eccentricity prediction, three models, i.e., the kNN, ET, and RF, have performed exceptionally well and have given an accuracy of more than 90% after a tenfold cross-validation. These three models along with the decision tree and the light gradient boosting machine have a tenfold cross-validation score of more than 75% in almost all cases. There is a range of eccentricity (0.2–0.32) where the optimum accuracy is obtained for the eccentricity predictions. A greater range of eccentricity (0.1–0.5) has also been taken into consideration. We find that the choice of the range in eccentricity affects the prediction accuracy of the eccentricity due to the imbalance in the training data distribution.

We have also performed an analysis of how the model would possibly perform in predicting the centrality class using experimental data as test data. We have considered two heavy-ion collision models, a viscous hydrodynamic model (VISH2 + 1) and a hybrid model (HYDRO + UrQMD), which are different from the AMPT model that is used for training the machine-learning models. The ML model predictions of impact parameters are obtained for the events of the VISH2 + 1 model and the hybrid model. The hydro and hybrid model events considered for testing are taken at specific impact-parameter ranges from 0.5 fm to 14 fm. The ML models (kNN results shown specifically) predicted the centrality classes of these events meticulously well. Although in both cases, we have obtained higher errors for the 0.5 fm events, the errors are very small at other impact parameters. The reason behind this is the lack of balance in the data set. When the data set is normalized, it is found that the peak of the distribution is not at the center. This indicates that the distribution of impact parameter and eccentricity over events are not isotropic.

To minimize these errors, we have used various sampling methods. Although there are several standard packages that help to rebalance the data, we finally see that the accuracy is improved in the lower-impact-parameter region if we assign different weights to the data at different impact parameters. For the extratrees model, rarer events are given four times the weight-age as the weight-age given to impact parameters with a large number of events. This has helped improving the accuracy in the lower-impact-parameter range. Our rebalancing technique resulted in a cross-validation accuracy of more than 90% for a higher range of eccentricity distribution. This meant an overall improvement from 75% accuracy before the rebalancing to an accuracy of 90% after the rebalancing. Our study therefore shows a rebalanced data set will be useful in making accurate prediction close to the head-on collisions.

Finally in conclusion, we have shown that it is possible to use the  $p_T$  spectra only to make accurate predictions of the initial parameters such as the impact parameter, the eccentricity, and the participant eccentricity using the ML algorithms. Even though the algorithm is trained by a single model, it can make accurate predictions from the data generated by

other models as long as all the models are able to generate the experimental data accurately. This means that any of the models may be used to train the data set. We have also found that the inaccuracies in the prediction are due to the imbalance in the data set. Proper rebalancing techniques can be used to rebalance the data set and this can be used to predict more accurate results in the low-impact-parameter regime.

## ACKNOWLEDGMENTS

For computational infrastructure, we acknowledge the Center for Modeling, Simulation and Design (CMSD) at the University of Hyderabad, where part of the simulations was carried out. A.S. is supported by INSPIRE Fellowship of the Department of Science and Technology (DST) Govt. of India, through Grant No. IF170627.

- 
- [1] J. Adams *et al.* (STAR Collaboration), *Nucl. Phys. A* **757**, 102 (2005).
- [2] B. Back *et al.* (PHOBOS Collaboration), *Nucl. Phys. A* **757**, 28 (2005).
- [3] A. Arsene *et al.* (BRAHMS Collaboration), *Nucl. Phys. A* **757**, 1 (2005).
- [4] K. Adcox *et al.* (PHENIX Collaboration), *Nucl. Phys. A* **757**, 184 (2005).
- [5] L. Adamczyk *et al.* (STAR Collaboration), *Phys. Rev. C* **96**, 044904 (2017).
- [6] D. Prorok, *Eur. Phys. J. A* **26**, 277 (2005).
- [7] S. Jowzaee, *Nucl. Phys. A* **967**, 792 (2017).
- [8] G. Odyniec, *J. Phys.: Conf. Ser.* **455**, 012037 (2013).
- [9] K. Aamodt *et al.* (ALICE Collaboration), *Phys. Rev. Lett.* **106**, 032301 (2011).
- [10] J. Adam *et al.* (ALICE Collaboration), *Phys. Rev. Lett.* **116**, 222302 (2016).
- [11] L. X. Han, G. L. Ma, Y. G. Ma, X. Z. Cai, J. H. Chen, S. Zhang, and C. Zhong, *Phys. Rev. C* **84**, 064907 (2011).
- [12] K. Adcox *et al.* (PHENIX Collaboration), *Phys. Rev. Lett.* **86**, 3500 (2001).
- [13] S. Acharya *et al.* (ALICE Collaboration), *Phys. Lett. B* **790**, 35 (2019); **788**, 166 (2019); R. Rath, S. Tripathy, R. Sahoo, S. De, and Md. Younus, *Phys. Rev. C* **99**, 064903 (2019).
- [14] M. L. Miller, K. Reyggers, S. J. Sanders, and P. Steinberg, *Annu. Rev. Nucl. Part. Sci.* **57**, 205 (2007).
- [15] C. David, M. Freslier, and J. Aichelin, *Phys. Rev. C* **51**, 1453 (1995).
- [16] C. Loizides, J. Kamin, and D. d'Enterria, *Phys. Rev. C* **97**, 054910 (2018).
- [17] F. Li, Y. Wang, Z. Gao, P. Li, H. Lü, Q. Li, C. Y. Tsang, and M. B. Tsang, *Phys. Rev. C* **104**, 034608 (2021).
- [18] J. De Sanctis, M. Masotti, M. Bruno *et al.*, *J. Phys. G* **36**, 015101 (2009).
- [19] M. Omana Kuttan, J. Steinheimer, K. Zhou, A. Redelbach, and H. Stoecker, *Phys. Lett. B* **811**, 135872 (2020).
- [20] Y. Huang, L.-G. Pang, X. Luo, X.-N. Wang, *Phys. Lett. B* **827**, 137001 (2022).
- [21] S. A. Bass, A. Bishoff, C. Hartnack, J. A. Maruhn, J. Reinhardt, H. Stoecker, and W. Greiner, *J. Phys. G* **20**, L21 (1994).
- [22] Z.-W. Lin, C. M. Ko, B.-A. Li, Bin Zhang, and S. Pal, *Phys. Rev. C* **72**, 064901 (2005).
- [23] Z. W. Lin, S. Pal, C. M. Ko, B. A. Li, and B. Zhang, *Phys. Rev. C* **64**, 011902(R) (2001).
- [24] T. Hirano and Y. Nara, *Phys. Rev. C* **79**, 064904 (2009).
- [25] N. Magdy, *J. Phys. G: Nucl. Part. Phys.* **49**, 015105 (2022).
- [26] H. Song and U. Heinz, *Phys. Rev. C* **77**, 064901 (2008).
- [27] C. Shen *et al.*, *Comput. Phys. Commun.* **199**, 61 (2016).
- [28] X. N. Wang and M. Gyulassy, *Phys. Rev. D: Part. Fields* **44**, 3501 (1991).
- [29] W.-T. Deng, X.-N. Wang, and R. Xu, *Phys. Rev. C* **83**, 014915 (2011).
- [30] S. Pal and M. Bleicher, *Phys. Lett. B* **709**, 82 (2012).
- [31] T. Sjöstrand, *Comput. Phys. Commun.* **82**, 74 (1994).
- [32] B. Zhang, *Comput. Phys. Commun.* **109**, 193 (1998).
- [33] B. A. Li and C. M. Ko, *Phys. Rev. C* **52**, 2037 (1995).
- [34] J. Xu and C. M. Ko, *Phys. Rev. C* **83**, 034904 (2011).
- [35] U. Heinz, H. Song, and A. K. Chaudhuri, *Phys. Rev. C* **73**, 034904 (2006).
- [36] H. Song and U. Heinz, *Phys. Rev. C* **78**, 024902 (2008).
- [37] C. Shen, U. Heinz, P. Huovinen, and H. Song, *Phys. Rev. C* **82**, 054904 (2010).
- [38] P. F. Kolb, J. Sollfrank, and U. Heinz, *Phys. Lett. B* **459**, 667 (1999); *Phys. Rev. C* **62**, 054909 (2000).
- [39] P. Huovinen and P. Petreczky, *Nucl. Phys. A* **837**, 26 (2010).
- [40] W. Israel, *Ann. Phys. (NY)* **100**, 310 (1976).
- [41] W. Israel and J. M. Stewart, *Ann. Phys. (NY)* **118**, 341 (1979).
- [42] J. P. Boris and D. L. Boon, *J. Comput. Phys.* **11**, 38 (1973).
- [43] M. He, R. J. Fries, and R. Rapp, *Phys. Rev. C* **85**, 044911 (2012).
- [44] F. Cooper and G. Frye, *Phys. Rev. D: Part. Fields* **10**, 186 (1974).
- [45] G. S. Denicol, C. Gale, S. Jeon, A. Monnai, B. Schenke, and C. Shen, *Phys. Rev. C* **98**, 034916 (2018).
- [46] <https://madai.phy.duke.edu/index1ccc.html?pageid=204>
- [47] S. A. Bass *et al.*, *Prog. Part. Nucl. Phys.* **41**, 255 (1998).
- [48] M. Bleicher *et al.*, *J. Phys. G* **25**, 1859 (1999).
- [49] I. H. Sarker, *SN Comput. Sci.* **2**, 160 (2021).
- [50] K. Taunk, S. De, S. Verma, and A. Swetapadma, in *2019 International Conference on Intelligent Computing and Control Systems (ICCS)* (IEEE, Piscataway, NJ, 2019), pp. 1255–1260.
- [51] L. Breiman, *Mach. Learn.* **45**, 5 (2001).
- [52] P. Geurts, D. Ernst, and L. Wehenkel, *Mach. Learn.* **63**, 3 (2006).
- [53] M. Ojala and G. C. Garriga, *J. Mach. Learn. Research* **11**, 1833 (2010).
- [54] S. G. Patro and K. Kumar Sahu, *IARJSET*. Vol. 2, Issue 3, March 2015.
- [55] Pedregosa *et al.*, *Scikit-Learn: Machine learning in python*, *J. Mach. Learn. Res.* **12**, 2825 (2011).
- [56] T. Doan and J. Kalita, in *2015 IEEE International Conference on Data Mining Workshop* (IEEE, Atlantic City, NJ, USA, 2015), pp. 1498–1505.
- [57] I. T. Jolliffe and J. Cadima, *Philos. Trans. R. Soc., A* **374**, 20150202 (2016).

- [58] K. Grebieszko, [Phys. Rev. C \*\*76\*\*, 064908 \(2007\)](#).
- [59] S. A. Bass, A. Bischoff, J. A. Maruhn, H. Stocker, and W. Greiner, [Phys. Rev. C \*\*53\*\*, 2358 \(1996\)](#).
- [60] F. Hu and H. Li, [Math. Probl. Eng. \*\*2013\*\*, 694809 \(2013\)](#).
- [61] H. He, Y. Bai, E. A. Garcia, and S. Li, in *2008 IEEE International Joint Conference on Neural Networks* (IEEE, Piscataway, NJ, 2008), pp. 1322–1328.
- [62] G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T. Liu, *LightGBM: A Highly Efficient Gradient Boosting Decision Tree*, *NIPS* (Curran Associates, Inc., 2017).
- [63] P. Parfenov, D. Idrisov, V. B. Luong, and A. Taranenko, [Particles \*\*4\*\*, 275 \(2021\)](#).
- [64] B. Abelev *et al.* (ALICE Collaboration), [Phys. Rev. C \*\*88\*\*, 044909 \(2013\)](#).
- [65] C. Y. Tsang *et al.*, [arXiv:2107.13985](#).
- [66] I. Arsene *et al.* (BRAHMS Collaboration), [Phys. Rev. C \*\*72\*\*, 014908 \(2005\)](#).
- [67] B. I. Abelev *et al.* (STAR Collaboration), [Phys. Rev. Lett. \*\*97\*\*, 152301 \(2006\)](#).
- [68] A. Wolterman, *Macalester J. Phys. Astron.* **8**, 17 (2020).