International Journal of Modern Physics A
Vol. 37, No. 2 (2022) 2250002 (15 pages)

DOI: 10.1142/S0217751X22500026

World Scientific
www.worldscientific.com

Particles multiplicity based on rapidity in Landau and artificial neural network (ANN) models

D. M. Habashy and Mahmoud Y. El-Bakry

Physics Department, Faculty of Education,
Ain Shams University, 11771, Roxy, Cairo, Egypt

Abdel Nasser Tawfik

Egyptian Center for Theoretical Physics (ECTP), Nile University,
Juhayna Square Off 26th-July-Corridor, 12588 Giza, Egypt

R. M. Abdel Rahman

Basic Sciences Department,
Modern Academy for Engineering and Technology,
11571, Mokattam, Cairo, Egypt

Mahmoud Hanafy*

Physics Department, Faculty of Science,
Benha University, 13518, Benha, Egypt
mahmoud.nasar@fsc.bu.edu.eg

Received 4 November 2021
Revised 4 December 2021
Accepted 5 December 2021
Published 15 January 2022

In this paper, artificial neural network (ANN) model is used to estimate the multiplicity per rapidity for charged pions and kaons observed in various high-energy experiments from central Au+Au heavy-ion collisions with energies ranging from 2–200 GeV, and then compared to available experimental data, including RHIC-BRAHMS experiment, and covering the energy range of the future accelerator facilities at NICA and FAIR. We also used Landau hydrodynamical approach, which has a better description for the evolution of hot and dense matter produced in ultra-relativistic heavy-ion collisions. The approach is fitted to both results estimated from experiment and ANN simulation. We noted that the Landau model accurately reproduces the entire range of multiplicity per rapidity for all created particles at all energies. Also ANN model can reproduce the multiplicity per rapidity very well for all the considered particles. This encourages us to

*Corresponding author.

use ANN model to predict the multiplicity per rapidity for k^- particles at energies 12.2 and 17.3 GeV.

Keywords: Hydrodynamical models; RHIC; ANN method.

PACS numbers: 24.10.Nz, 25.75.−q, 05.70.−a

1. Introduction

The evolution of dense and hot matter produced in ultra-relativistic heavy-ion collisions is an interesting subject.[1,2] It is important to have a better overview of the dynamics of the matter created in such collision for solving various problems in heavy-ion collisions such as the research of a heavy quarkonium in quark–gluon plasma (QGP) and the interaction of the produced jet with the formed medium.[3–5] For these challenges, Landau hydrodynamics model provides a successful picture for the evolution of a dense matter assembly at high temperature and pressure.[1,6] Its dynamics can be solved precisely[7–9] during the longitudinal expansion in one-dimension in the initial stage. The longitudinal expansion problem in one-dimension has an approximate solution that can be applied to the fluid in its entirety. Following that, the three-dimensional (3D) behavior can be approximated to get predictions comparable to the experimental results.[7,10,11] The multiplicity per rapidity is one of the most essential observables in heavy-ion collisions. It can be considered as an important feature for particle production.[12]

Many models[8,11,13–24] have been studied and rapidity distribution has been discussed such as Landau hydrodynamical approach, which will be utilized in this research. Hydrodynamics is employed to describe the evolution of a colliding system, from the initial conditions upto freeze-out conditions.[25] In 1950, Fermi proposed that when high-energy particles collide, numerous new elementary particles are formed, each with a mean free path that is smaller than the total size of the created cloud of the hadronic matter.[26]

Artificial neural networks (ANNs) are the core components of artificial intelligence, and are frequently involved in machine learning (ML). First, the ML model is built by training the model with a training dataset. The performance is assessed using a new set of data, and if necessary, the model parameters are fine-tuned.[27–29] The model is saved and ready to be applied to actual data to solve the problem once the estimations are satisfactory. Machine learning is mostly used to solve classification, regression, and clustering problems.[27,30] The challenge we are working on is trained modeling, which means that each pair of input variables has a fixed numerical value as the goal variable.[31–34] In recent years, several successful applications of the ANNs have emerged in nuclear physics and high-energy physics,[27–30,35–39] as well as in biology, chemistry, meteorology, and other fields of science. A major goal of nuclear theory is to predict nuclear structure and nuclear reactions using the underlying theory of the strong interactions, Quantum Chromodynamics (QCD), QGP and other theories. One of the essential observables that has a substantial impact on the end state particle production is the rapidity distribution of a collision.

Each set of data corresponds to a single heavy-ion collisions event.[29,36,38] Given that the work is comparable to signal fitting, determining the degree of conformity between the original and the obtained signals is crucial. As a result, the performance of the ANNs considered in this study was assessed using a Chi-squared test deformation.[31,40–44]

The goal of this study is to compute the multiplicity per rapidity for charged pions and kaons formed from Au+Au heavy-ion collisions using the Landau hydrodynamical model and the ANN simulation model starting from low-energy region, covered by RHIC-BRAHMS experiment and shall be covered by the future accelerator facilities NICA and FAIR, upto high-energy region, that are covered by RHIC-BRAHMS experiment. The future accelerator facilities NICA at JINR, Russia and FAIR at GSI, Germany are promising heavy-ion collisions experiments and will work at intermediate center of mass energies $\sqrt{s} = 3{-}13$ GeV for Au+Au heavy-ion collisions system.[45,46] We compare our results to the experimental data produced from RHIC-BRAHMS experiment.[11,20,46–55]

This paper is organized as follows. In Sec. 2.1, we briefly introduce Landau hydrodynamical model and the ANN model in Sec. 2.2. The results are shown in in Sec. 3. The conclusion is drawn in Sec. 4.

2. Approaches

In this section, the particle multiplicity per rapidity is described briefly using Landau approach, which is dependent on the dynamical evolution of dense and hot matter, and the ANN simulation model. According to this research, these gradients appear to be crucial in particle production.

2.1. *Landau hydrodynamical approach*

A recent analysis of experimental data shows that the Landau hydrodynamical technique produces conclusions that are consistent with experiment.[1,10,56,57] A quantitative analysis uses an approximate form of the Landau rapidity distribution.

In Landau model, the equation that describes the relationship between the multiplicity per rapidity and the rapidity distribution is deduced first from the experimental data and has a Gaussian distribution shape. Later, when it became necessary to use and discuss quantitative analysis of hydrodynamic evolution, a series of calculated treatments were carried out, and the equation could be expressed as[58,59]

$$\frac{1}{\sigma^2}\frac{d\sigma}{dy} = \frac{dN}{dy}. \tag{1}$$

The number distribution of particles is calculated by integrating the single-particle inclusive distribution with respect to the transverse momentum $p_\perp$ and then dividing the inelastic cross-section[58,59]

$$E\frac{d^3N}{d^3p} = \int_{-\infty}^{\infty} \rho(y_0) E\frac{d^3N_1}{d^3p}(y - y_0)dy_{FB}, \tag{2}$$

where

$$\rho(y_0) = \frac{1}{\sqrt{2}\pi\sigma} \exp\left(-\frac{y_0^2}{2\sigma^2}\right), \tag{3}$$

and

$$\frac{d^3p}{E} = m_\perp dm_\perp dy d\phi = p_\perp dp_\perp dy d\phi, \tag{4}$$

$\frac{d^3p}{E}$ is the Lorentz invariant momentum space volume element.

The rapidity variable has the advantage of transforming linearly when subjected to a Lorentz transformation as[58,59]

$$E\frac{d^3N}{d^3p} = E\frac{d^3\sigma}{p_\perp dp_\perp dp_z d\phi} = \frac{d^3\sigma}{p_\perp dp_\perp d(y-y_0)d\phi} = \frac{d^3\sigma}{m_\perp dm_\perp d(y-y_0)d\phi}, \tag{5}$$

$$p_\perp dp_\perp = m_\perp dm_\perp, \tag{6}$$

and

$$d(y-y_0) = \frac{dp_\perp}{E}, \tag{7}$$

Eq. (5) can be rewritten as a function of rapidity distribution as follows:

$$E\frac{d^3\sigma}{d^3p} = \frac{d^3N}{m_\perp dm_\perp d(y-y_0)d\phi} = \frac{d^2N}{2\pi m_\perp dm_\perp d(y-y_0)} = \frac{d^2N}{2\pi p_\perp dp_\perp d(y-y_0)}, \tag{8}$$

The phase-space distribution function is given by

$$E\frac{d^3\sigma}{d^3p} \equiv f(E, p_z). \tag{9}$$

A link between beam energy and the number of created charged particles per pair of participants is discovered by Landau.[58] The rapidity distribution of charged particles created is written as[58,59]

$$\frac{dN}{dy} \simeq \exp\left(\sqrt{L^2 - y^2}\right), \tag{10}$$

where L is the logarithm of Lorentz contract factor γ,[58,59] and it is considered as the measure of the thickness of the Lorentz contracted disks of the colliding hadronic matter[58,59]

$$L = \ln(\gamma) = \ln\left(\frac{\sqrt{s_{\rm NN}}}{2m_p}\right) = \sigma_y^2, \tag{11}$$

where $\sqrt{s_{\rm NN}}$ is the center of mass energy, σ_y is the width of the distribution, and m_p is the mass of proton.[60]

Tha Landau Gaussian distribution of the rapidity distribution can be expressed as follows[1,43,57,58,61]:

$$\frac{dN}{dy} = \frac{N}{\sqrt{2\pi L}} \exp\left(-\frac{y^2}{2L}\right), \tag{12}$$

where N is the normalization constant.

However, this equation is an approximate representation of Eq. (1) in the region of $|y| \ll L$. In other rapidity regions, the distributions are totally different.[57,61]

Table 2 includes the values of the fitting parameters N and L according to the different used center of mass collision energies for all particles.

2.2. *Artificial neural network method*

ANNs are one of the most successful programming paradigms in the last two decades. It is widely used in a large variety of applications in various areas because of its great capability and excellent learning function.[28,36,62,63] An artificial neural network may theoretically approximate arbitrary continuous mapping with arbitrary precision. By learning,[32,41] an ANN can learn the desired job.[64,65]

It is a computer model based on the structure and behavior of biological neural networks in the brain.[66] While mathematical algorithms are well suited for linear programming, arithmetic and logic calculations, ANNs are more effective to solve problems related to pattern recognition, matching, clustering and classification.[40,67] ANN consists of a very large number of nerve cells (in humans about ten billions nerve), called artificial neurons, as a nonlinear processing units, which are linked to one another in a complex network via synaptic weights.[31] The intelligent behavior is an outcome of a large amount of interaction among interconnected parts. The input of a neuron is composed of the output signals of the neurons connected to it.[33] When the contribution of these inputs exceeds a certain threshold, the neuron generates a bioelectric signal through a suitable transfer function, which propagates through the synaptic weights to other neurons,[37] as shown in Fig. 1.

For each artificial neuron, an input X_n is weighted by a factor W_n and added to bias b as

$$Y_n = \sum_{i}^{n} [X_n W_n + b]. \tag{13}$$

To get the output Y_n, this problem is pretty hard to solve using the conventional programming approach.[31] There are two types of models: supervised and unsupervised. A supervised model requires a teacher or desired output to learn a task, while an unsupervised model does not require a teacher, it does learn a task based on the task's objective functions. The development of the first ANN was based on

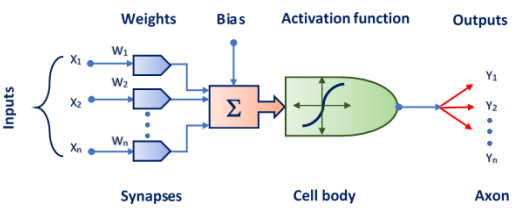

Fig. 1. The ANN data processing for a neuron input signal to be processed and converted to the desired output signal.

a very simple model of neural connections. The machine $MarkIPerceptron$ was introduced by the neurobiologist Rosenblatt. He assumes that the artificial connections between neurons could change through a supervised learning process[66] that reduces the misfit between the actual and the expected output. The expected output comes from a set of training data. The misfit between the actual and the expected responses of the network represents the necessary information for improving the learning performance.[32]

ANN can learn a task by adjusting weights. It is automatically learning from data,[32] neural network is used to execute different stages of processing systems based on learning algorithms by controlling their weights and biases.[33] The models that are executed by neural networks cannot be explained in a human symbolic language. The result must be accepted as a black box. Expressly, a neural network is able to generate a valid result or being acceptable, but it is not possible to explain how and why this result has been proceeded.

Furthermore, ANNs can achieve solutions that are either algorithmic, computationally costly, or nonexistence. As a result, ANNs are seen as a more powerful modeling technique for advanced complicated nonlinear input–output situations. Mean Square Error (MSE) is the most popular performance procedure for training ANNs, and it has been widely employed in a range of issues such as pattern recognition and machine learning. MSE is described as follows:

$$\text{MSE} = \sum_{i=1}^{n} \left[\frac{(y_s - y_o)^2}{n} \right], \tag{14}$$

where n is the number of datasets used for training the network, y_s is the mean of the simulated value, and y_o is the corresponding experimental value.

The datasets used for the training, testing and validation purpose of the model are randomized.[32,40,41] Hyperbolic tangent sigmoid function (tansig) and logarithmic sigmoid function (logsig) are the most commonly utilized activation functions in hidden layers, their mathematical formula appears in Eqs. (15) and (16), whereas a pure linear function (purelin) is utilized in the output layer.[64]

$A = \text{purelin}(N, \text{FP})$ takes N and optional parameters for the function

$$(\text{logsig}) f(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}. \tag{15}$$

$$(\text{tansig}) f(x) = \frac{1}{1 + e^{-x}}. \tag{16}$$

The Pureline function is a neural transfer function that calculates a layer's output from its net input and its formula is expressed as

$$A = \text{purelin}(N, \text{FP}), \tag{17}$$

where A is the output, N is S-by-Q matrix of net input vectors, and FP is struct of function parameters. When training the ANN model, different sets of internal network parameters were utilized to define the number of hidden layers, the number

of neurons in the hidden layer, and the transfer function. The number of neurons in the hidden layer is an important consideration when choosing a neural network architecture, and their number varies from case to case.

In the hidden layers, using sigmoid transfer functions, sometimes known as "squashing" functions, is particularly efficient. They reduce a large input range to a small output range. Sigmoid functions are distinguished by their slope, which must approach zero as the input size increases. When steepest descent is used to train a multilayer network with sigmoid functions, this results in tiny changes in the weights and biases, even though they are far from their optimal values.

This work uses a Back Propagation ANN method and Rprop training data to estimate the rapidity distribution dN/dy for particles π^-, π^+, K^-, and K^+ produced from (Au+Au) nucleus collision at energies ranging from $2-200$ GeV from various experimental high-energy physics data of different experiments, including RHIC-BRAHMS, and the future facilities NICA and FAIR. A schematic flowchart to describe the methodology of ANN method is shown in Fig. 2.

The used algorithm is $Resilient back-propagation algorithm$ (Rprop). According to this algorithm, we use the trainrp function for training process. Trainrp is a network training function that depends upon making updates for weights and biases. The Rprop algorithm is introduced by both of Riedmiller and Braun as first-order learning methods for neural networks. It is considered as one of the best performing ANN learning methods.

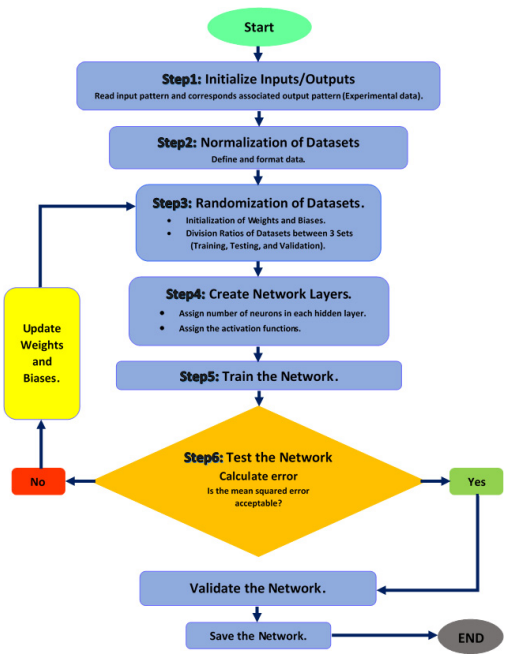

Fig. 2. A typical flowchart for ANN method work mapping.

The purpose of the *Resilientbackpropagation* (Rprop) training algorithm is to eliminate these harmful effects of the magnitudes of the partial derivatives of arbitrary error measure that is differentiable with respect to the weights. The direction of the weight update is determined solely by the sign of the derivative; the size of

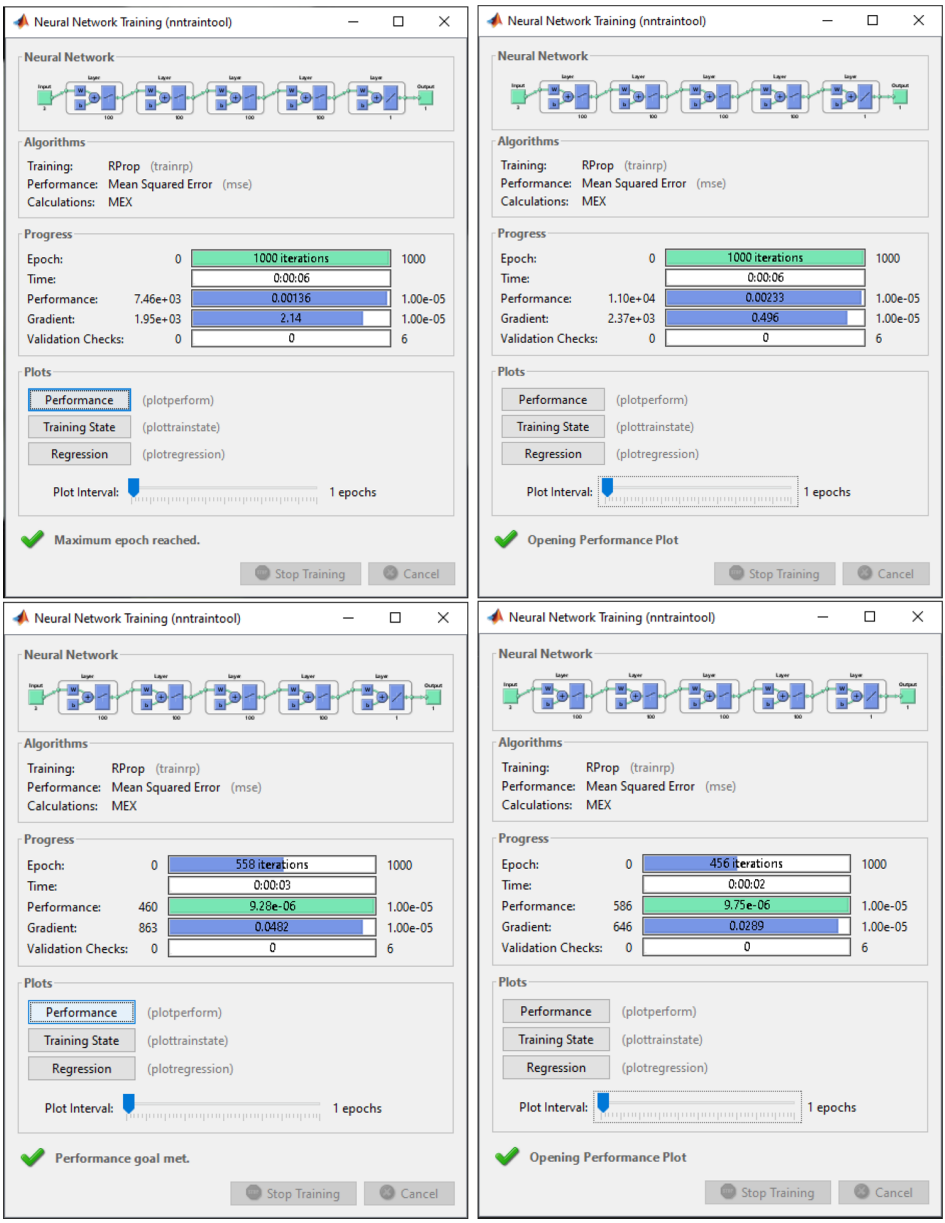

Fig. 3. Matlab neural network training tool used for estimating the rapidity distribution, dN/dy, for particles π^-, π^+, K^-, and K^+.

the derivative has no bearing on the weight update. A separate update value determines the size of the weight change.

Simulation is done by using Matlab programming language interface as shown in Fig. 3 for all considered particles. The input parameters are rapidity sets, the center of mass energy and the mass number of the gold nucleus (Au), (Z). For a better rapidity distribution, dN/dy, the used hidden layers are four, each one includes 100 neurons and the first started applying epochs (refers to one cycle through the full training dataset) is 1000. Usually, training a neural network takes more than a few epoch. The suitable activation function used for hidden layers is logsig while purelin function is used for the output layer.

3. Results and Discussion

In this research, we calculate the rapidity distribution, dN/dy, produced from various high-energy experimental results in central (Au + Au) heavy-ion collisions at energies spanning from 2–200 GeV, including RHIC-BRAHMS, for particles π^-, π^+, K^-, and K^+.[11,20,46–55] The obtained results are confronted to those estimated from ANN simulation model.[31,67]

The ANN model is perfectly trained, based on the available experimental data, using the input parameters, which are shown in Table 1.

The ANN target is to obtain the best MSE value about (10^{-5}) according to Eq. (14). The obtained training results for particles π^-, π^+, K^-, and K^+ are shown in Figs. 3 and 4. The network's MSE value reduced from a huge value to a lower value, as shown in Fig. 4. Furthermore, the network was evolving. After the network has learned the training set, training was finalized. The optimal ANN simulation model training, for particles π^-, π^+, K^-, and K^+, is chosen as a result of MSE value of 0.00136, 0.00233, 9.28×10^{-6}, and 9.75×10^{-6}, respectively. The MSE

Table 1. ANN parameters which are used for estimating the multiplicity per rapidity of all considered particles.

ANN	Particles			
parameters	π^-	π^+	K^-	K^+
Inputs	y, $\sqrt{S_{\rm NN}}$, Z	y, $\sqrt{S_{\rm NN}}$, Z	y, $\sqrt{S_{\rm NN}}$, Z	y, $\sqrt{S_{\rm NN}}$, Z
$\sqrt{S_{\rm NN}}$	2, 4, 6, 8, 8.8,	2, 4, 6, 8,	10.7, 62.4, 200	8.8, 12.2, 17.3,
(GeV)	12.2, 17.3, 62.4, 200	10.7, 62.4, 200		62.4, 200
Z	$Z_{\rm Au} = 197$	$Z_{\rm Au} = 197$	$Z_{\rm Au} = 197$	$Z_{\rm Au} = 197$
Outputs	dN/dy	dN/dy	dN/dy	dN/dy
Hidden layers	4	4	4	4
Neurons	100,100,100,100	100,100,100,100	100,100,100,100	100,100,100,100
Epochs	1000	1000	558	456
Training algorithms	Rprop	Rprop	Rprop	Rprop
Training functions	trainrp	trainrp	trainrp	trainrp
Transfer functions	logsig	logsig	logsig	logsig
Performances	0.00136	0.00233	9.28×10^{-6}	9.75×10^{-6}
Output functions	Purelin	Purelin	Purelin	Purelin

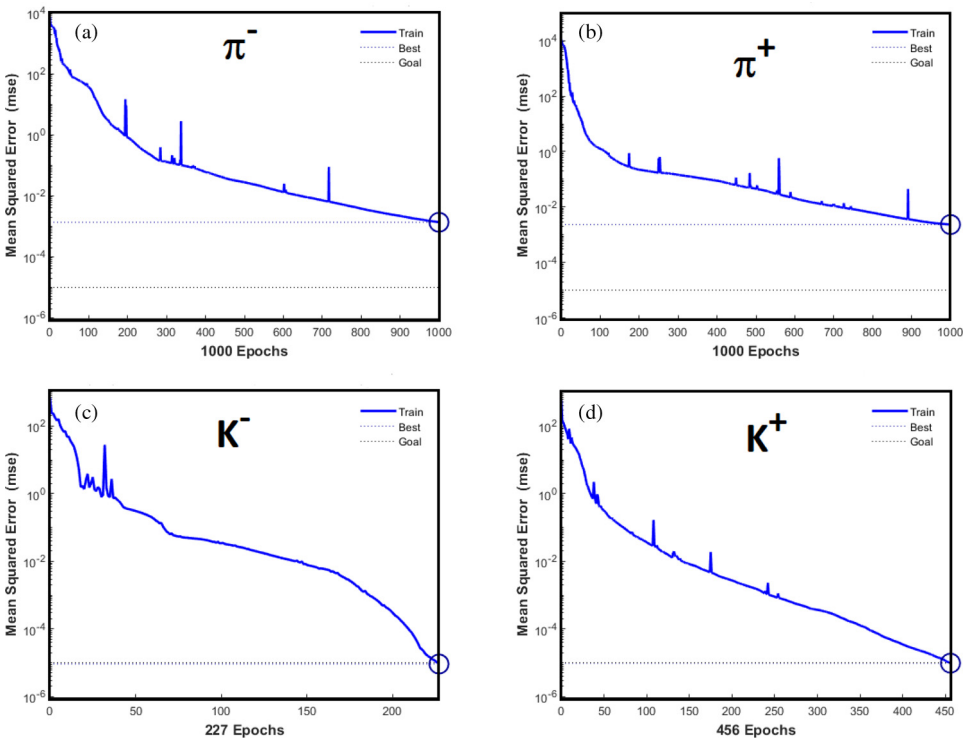

Fig. 4. The performance of the used neural network for particles (a) π^-, (b) π^+, (c) K^-, and (d) K^+.

value for pion particles is slightly high in comparison with that of kaons despite of considering the same training parameters. This encourages for further investigation.

The rapidity distribution equation dN/dy can be estimated from ANN as follows:

$$\begin{aligned} dN/dy = \text{purelin}\,[\text{net. } LW\{5,4\}\ \text{logsig}\ (\text{net. } LW\{4,3\}\ \text{logsig}\ (\text{net. } LW\{3,2\} \\ \times \text{logsig}\ (\text{net. } LW\{2,1\}\ \text{logsig}\ (\text{net.IW}\ \{1,1\}R + \text{net.b}\ \{1\}) \\ + \text{net.b}\ \{2\}) + \text{net.b}\ \{3\}) + \text{net.b}\ \{4\}) + \text{net.b}\ \{5\}], \end{aligned} \tag{18}$$

where R contains the input parameters (rapidity, center of mass energy, and Au mass number), which is used to calculate dN/dy. IW and LW are the linked weights where net. $IW\{1,1\}$, net. $LW\{2,1\}$, net. $LW\{3,2\}$, net. $LW\{4,3\}$, and net. $LW\{5,4\}$ represent the linked weights between the input layer and the first hidden layer, first and the second hidden layers, the second and the third hidden layers, the third and the fourth hidden layers, and the fourth and the output layers, respectively. The parameter b stands for the bias where net.b $\{1\}$, net.b $\{2\}$, net.b $\{3\}$, net.b $\{4\}$, and net.b $\{5\}$ represent the bias of the, first, second, third, and fourth, hidden layers, and the output layer, respectively.

Table 2. The Landau hydrodynamical approach fit parameters for multiplicity per rapidity dN/dy for particles π^-, π^+, K^-, and K^+ at the considered energies.

Particle	$\sqrt{S_{\mathrm{NN}}}$ GeV	N	L
π^-	2	32.91 ± 0.75	0.34 ± 0.07
	4	71.31 ± 3.1	0.52 ± 0.08
	6	122.07 ± 3.06	0.93 ± 0.08
	8	143.19 ± 5.76	0.948 ± 0.07
	8.8	316.78 ± 11.17	1.21 ± 0.08
	10.7	146.66 ± 10.1	0.75 ± 0.05
	12.2	478.95	1.591 ± 0.06
	62.4	1025.17 ± 10.76	3.289 ± 0.05
	200	1809.23 ± 12.97	5.4785 ± 0.07
π^+	2	18.091 ± 0.43	0.328 ± 0.05
	4	44.05 ± 2.671	0.402 ± 0.06
	6	95.55	0.93 ± 0.06
	8	110.37 ± 7.01	1.11 ± 0.054
	10.7	130 ± 6.17	0.84 ± 0.03
	62.4	1044.02	3.3588 ± 0.06
	200	1694.12 ± 12.65	5.1688 ± 0.03
K^-	10.7	3.92 ± 0.581	0.59 ± 0.06
	62.4	119.223 ± 0.06	2.4779 ± 0.04
	200	226.753 ± 0.46	4.2672 ± 0.03
K^+	8.8	61.39	1.211 ± 0.05
	10.7	23.44 ± 0.08	0.85 ± 0.05
	12.2	83.241 ± 0.5	1.951 ± 0.05
	17.3	112.38 ± 0.05	2.06 ± 0.05
	62.4	163.777 ± 0.06	3.5056 ± 0.07
	200	280.372 ± 0.03	5.6778 ± 0.05

After that, the rapidity distribution, dN/dy, obtained from Landau hydrodynamical analysis, according to Eq. (12), is fitted to the experimental data and compared to those obtained from the ANN simulation model.

This work aims to study and estimate the rapidity distribution, dN/dy, in the framework of Landau hydrodynamical model based on Eq. (12) that ensures a Gaussian distribution for all used particles. The experimental data of the rapidity distribution's dependence dN/dy as function of rapidity y is fitted to Landau hydrodynamical model using Eq. (12) which has a Gaussian distribution shape. The best fitting parameters are shown in Table 2.

Figure 5 shows the rapidity distribution, dN/dy, versus rapidity, y, for particles π^-, π^+, K^-, and K^+ measured in (Au + Au) central nuclear heavy-ion collisions, including RHIC-BRAHMS experiment and covering the future NICA and FAIR experiments results, at energies 2, 4, 6, 8, 8.8, 12.2, 17.3, 62.4, and 200 GeV[10,47] at a wide rapidity range, $-8 < y < 8$. The experimental data is represented as symbols, while the Landau hydrodynamical model results are represented as solid lines, and the results from ANN simulation method, using Matlab neural network training tool (nntraintool), are depicted as dashed lines.

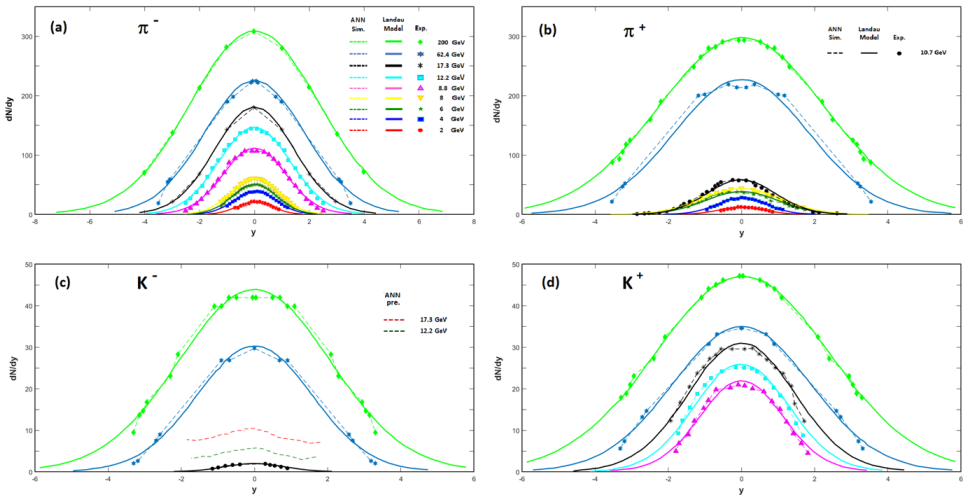

Fig. 5. The rapidity distribution dN/dy is plotted as a function of the rapidity, y, for particles (a) π^-, (b) π^+, (c) K^-, and (d) K^+. The experimental results (symbols) from Au + Au central heavy-ion collisions, at 2, 4, 6, 8, 8.8, 12.2, 17.3, 62.4, and 200 GeV,[11,20,46–55] are confronted to numerical simulation by using ANN method (dashed lines) according to Eq. (18), and also fitted to the Landau hydrodynamical approach (solid lines), Eq. (12).

We observed that the estimated rapidity distribution dN/dy, for particles π^-, π^+, k^-, and k^+, exhibits a Gaussian distribution shape. The estimated rapidity distribution, dN/dy, around the mid-rapidity ($y = 0$), is increased while going from low center of mass energy, $\sqrt{S} = 2$ GeV, upto high center of mass energy, $\sqrt{S} = 200$ GeV. Accordingly, as the center of mass energy increased, the curves shifted upward.[11] A general observation is Landau model, which is presented in Eq. (12), can reproduce the experimental data at the considered range of rapidity, successfully. Also, the ANN simulation method results seem to coincide with the used experimental data.[11,20,46–55] In case of K^-, there is a lack of the experimental data, thus we use ANN simulation model to predict the multiplicity per rapidity at energies $\sqrt{s} = 12.2$ and 17.3 GeV. Only results of the rapidity distribution dN/dy for, K^-, are available at energies 10.7, 62.4, and 200 GeV measured in (Au + Au) central heavy-ion collisions at various experimental high-energy physics,[10,47] including RHIC-BRAHMS and covering the future experimental facilities, NICA and FAIR. These are general observations that can be used to draw conclusions.

The multiplicity per rapidity, dN/dy, for particles π^-, π^+, K^-, and K^+, measured from (Au+Au) central nuclear heavy-ion collisions from various experimental data, including RHIC-BRAHMS and the future experimental facilities FAIR and NICA, at energies 2, 4, 6, 8, 8.8, 12.2, 17.3, 62.4, and 200,[11,20,46–55] is compared to those obtained from the ANN simulation model. The rapidity range considered lies in $(-8 < y < 8)$. Also, the outcomes of the Landau hydrodynamical approach are

fitted to the used experimental data and compared to the results of ANN simulation model. The ANN method can reproduce the rapidity distribution, dN/dy, at both low and high energies. The low-energy region shall be covered by the future NICA and FAIR facilities.

4. Conclusions

We have estimated and calculated the multiplicity per rapidity, dN/dy, for particles π^-, π^+, K^-, and K^+, measured from (Au+Au) central nuclear heavy-ion collisions from various experimental data, including RHIC-BRAHMS and the future experimental facilities FAIR and NICA using two different models, namely, the ANN simulation model; the most successful single programming paradigms ever invented in the last two decades and the Landau hydrodynamical model. The Landau hydrodynamical model is fitted to the considered experimental data and compared to the results of the ANN simulation model. The comparison between the simulation and the measurements shows an excellent agreement. Also, the rapidity distribution and the considered rapidity range are successfully recalculated using the Landau hydrodynamical model. The mathematical expression described experimental data was obtained using the ANN. The success of the ANN simulation model allows us to predict the multiplicity per rapidity for the particle k^- where there is a lack of the experimental data.

References

1. C.-Y. Wong, *EPJ Web Conf.* **7**, 01006 (2010).
2. A. N. Tawfik, *Int. J. Mod. Phys. A* **29**, 1430021 (2014).
3. T. Niida and Y. Miake, *AAPPS Bull.* **31**, 12 (2021).
4. C. Ding, W.-Y. Ke, L.-G. Pang and X.-N. Wang, *Chin. Phys. C* **45**, 074102 (2021).
5. W. Busza, K. Rajagopal and W. van der Schee, *Ann. Rev. Nucl. Part. Sci.* **68**, 339 (2018).
6. L. Du, X. An and U. Heinz, *Phys. Rev. C* **104**, 064904 (2021).
7. L. D. Landau, *Nucl. Phys.* **13**, 181 (1959).
8. T. Hirano, K. Morita, S. Muroya and C. Nonaka, *Pramana* **60**, 1103 (2002).
9. A. Monnai, *Phys. Rev. C* **100**, 014901 (2019).
10. M. J. Murray, *J. Phys. G* **30**, S667 (2004).
11. C. Blume, *J. Phys. G* **31**, S57 (2005).
12. I. G. Bearden *et al.*, *Phys. Rev. Lett.* **94**, 162301 (2005).
13. R. Hagedorn and J. Rafelski, *Phys. Lett. B* **97**, 136 (1980).
14. L. Marques, J. Cleymans and A. Deppman, *Phys. Rev. D* **91**, 054025 (2015).
15. S. K. Tiwari, P. K. Srivastava and C. P. Singh, *Phys. Rev. C* **85**, 014908 (2012).
16. J. D. Bjorken, *Phys. Rev. D* **27**, 140 (1983).
17. E. Schnedermann, J. Sollfrank and U. W. Heinz, *Phys. Rev. C* **48**, 2462 (1993).
18. P. Braun-Munzinger, J. Stachel, J. P. Wessels and N. Xu, *Phys. Lett. B* **365**, 1 (1996).
19. P. Braun-Munzinger, J. Stachel, J. P. Wessels and N. Xu, *Phys. Lett. B* **344**, 43 (1995).
20. S.-Q. Feng and Y. Zhong, *Phys. Rev. C* **83**, 034908 (2011).
21. K. Morita, S. Muroya, C. Nonaka and T. Hirano, *Phys. Rev. C* **66**, 054904 (2002).
22. U. Mayer and U. W. Heinz, *Phys. Rev. C* **56**, 439 (1997).

23. W. Broniowski and W. Florkowski, *Phys. Rev. C* **65**, 064905 (2002).
24. T. I. Haweel, M. Y. El-Bakry and K. A. El-Metwally, *Chaos Solitons Fractals* **18**, 159 (2003).
25. S.-X. Liu, H.-J. Wang and F.-M. Liu (2012), arXiv:1210.2833.
26. E. Fermi, *Phys. Rev.* **81**, 683 (1951).
27. L.-G. Pang, *Nucl. Phys. A* **1005**, 121972 (2021).
28. J. Duarte and J.-R. Vlimant (2020), arXiv:2012.01249.
29. N. Mallick, S. Tripathy, A. N. Mishra, S. Deb and R. Sahoo, *Phys. Rev. D* **103**, 094031 (2021).
30. W. J. Brouwer, J. D. Kubicki, J. O. Sofo and C. L. Giles (2014), arXiv:1405.3564.
31. R. Rojas, *Neural Network: A Systematic Introduction* (Springer, Berlin, Heidelberg, 1996).
32. H. Ali and D. Habashy, *Commun. Theor. Phys.* **72**, 105701 (2020).
33. H. Zahran, H. Soliman, D. A. Abd El-Rehim and D. Mahmoud, *Crystals* **11**, 481 (2021).
34. A. Darwish *et al.*, *Superlattice Microstruct.* **83**, 299 (2015).
35. L.-G. Pang *et al.*, *Nucl. Phys. A* **982**, 867 (2019).
36. F. Li *et al.*, *Phys. Rev. C* **104**, 034608 (2021).
37. Y. Liu *et al.*, *Nucl. Sci. Technol.* **30**, 148 (2019).
38. L. Apolinario *et al.*, *J. High Energ. Phys.* **2021**, 219 (2021).
39. M. El-Bakry, *Chaos Solitons Fractals* **18**, 995 (2003).
40. S. Haykin, *Neural Networks and Learning Machines*, 3rd edn. (Pearson Prentice Hall, 2008).
41. D. A. Abd El-Rehim, D. Habashy, H. Zahran and H. Soliman, *Met. Mater. Int.* **27**, 4084 (2021).
42. A. F. Abd El-Rehim, H. Y. Zahran, D. M. Habashy and H. M. Al-Masoud, *Crystals* **10**, 290 (2020).
43. J. Stachel and P. Braun-Munzinger, *Phys. Lett. B* **216**, 1 (1989).
44. R. Nada *et al.*, *Mater. Sci. Eng. A* **567**, 80 (2013).
45. P. Senger, *Particles* **4**, 214 (2021).
46. H. Liu *et al.*, *Phys. Rev. Lett.* **84**, 5488 (2000).
47. I. G. Bearden *et al.*, *Phys. Rev. Lett.* **90**, 102301 (2003).
48. E. L. Bratkovskaya *et al.*, *J. Phys. Conf. Ser.* **878**, 012018 (2017).
49. I. C. Arsene *et al.*, *Phys. Lett. B* **687**, 36 (2010).
50. I. C. Arsene *et al.*, *Phys. Lett. B* **677**, 267 (2009).
51. I. Arsene *et al.*, *Nucl. Phys. A* **757**, 1 (2005).
52. I. G. Bearden *et al.*, *Phys. Rev. Lett.* **94**, 162301 (2005).
53. J. L. Klay *et al.*, *Phys. Rev. Lett.* **88**, 102301 (2002).
54. S. Feng and X. Yuan, *Sci. China G* **52**, 198 (2009).
55. J. H. Lee *et al.*, *J. Phys. G* **30**, S85 (2004).
56. P. Steinberg, *Nucl. Phys. A* **752**, 423 (2005).
57. Z. J. Jiang, Q. G. Li and H. L. Zhang, *J. Phys. G* **40**, 025101 (2013).
58. L. D. Landau and D. Ter-Haar, *Collected Papers of L.D. Landau* (Pergamon, Oxford, 1965).
59. P. Steinberg, *Acta Phys. Hung. A* **24**, 51 (2005).
60. J. Cleymans, J. Strumpfer and L. Turko, *Phys. Rev. C* **78**, 017901 (2008).
61. P. K. Netrakanti and B. Mohanty, *Phys. Rev. C* **71**, 047901 (2005).
62. S. Akkoyun, *Nucl. Instrum. Methods B* **462**, 51 (2020).
63. M. U. Anil, T. Malik and K. Banerjee (2020), arXiv:2004.14196.

64. D. S. Pandey, S. Das, I. Pan, J. J. Leahy and W. Kwapinski, *Waste Manage.* **58**, 202213 (2016).
65. B. Krse, B. Krose, P. van der Smagt and P. Smagt, *J. Comput. Sci.* **48**, 19676 (1993).
66. F. Rosenblatt, *Psychol. Rev.* **65**, 386408 (1958).
67. K. Gurney, *An Introduction to Neural Networks* (Imprint CRC Press, London, 1997).