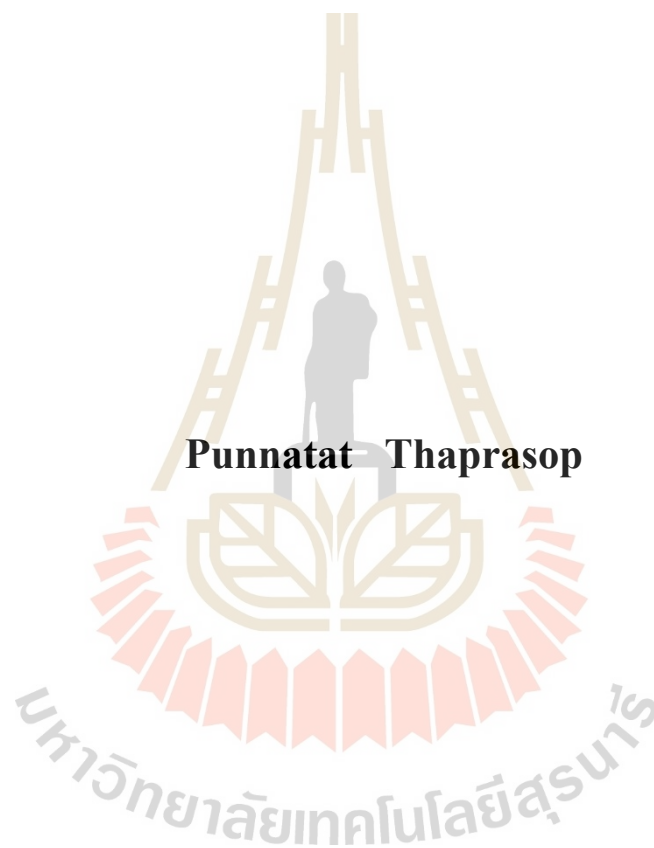


**OUTLIER DETECTION IN NUCLEAR COLLISIONS  
USING UNSUPERVISED (DEEP-) LEARNING**



**Punnatat Thaprasop**

**A Thesis Submitted in Partial Fulfillment of the Requirement for the**

**Degree of Master of Science in Physics**

**Suranaree University of Technology**

**Academic Year 2019**

การตรวจจับค่าผิดปกติในการชนกันของนิวเคลียส  
โดยใช้แอนซูปเปอร์ไวซ์ (ดีพ) เลิร์นนิ่ง



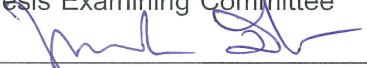
นายบัณฑิต ทะประสพ

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิทยาศาสตรมหาบัณฑิต  
สาขาวิชาฟิสิกส์  
มหาวิทยาลัยเทคโนโลยีสุรนารี  
ปีการศึกษา 2562

# OUTLIER DETECTION IN NUCLEAR COLLISIONS USING UNSUPERVISED (DEEP-) LEARNING

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master Degree

Thesis Examining Committee



---

(Assoc. Prof. Dr. Panomsak Meemon)

Chairperson



---

(Asst. Prof. Dr. Christoph Herold)

Member (Thesis Advisor)



---

(Asst. Prof. Dr. Chinorat Kobdaj)

Member



---

(Prof. Dr. Yupeng Yan)


Member



---

(Asst. Prof. Dr. Ayut Limphirat)

Member



---

(Dr. Jan Steinheimer)

Member



---

(Prof. Dr. Santi Maensiri)



---

(Assoc. Prof. Dr. Yupaporn Ruksakulpiwat)

Vice Rector for Academic Affairs  
and Quality Assurance

Dean of Institute of Science

บัณฑิต ทะประสพ : การตรวจจับค่าผิดปกติในการชนกันของนิวเคลียสโดยใช้อัน  
ซูปเปอร์ไวซ์ (ดีพ) เลิร์นนิ่ง (OUTLIER DETECTION IN NUCLEAR COLLISIONS  
USING UNSUPERVISED (DEEP-) LEARNING). อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์  
ดร.คริสตอฟ เฮโรลด์, 77 หน้า.

งานนี้เสนอวิธีการในการใช้อันซูปเปอร์ไวซ์เลิร์นนิ่งในการตรวจจับค่าผิดปกติในการ  
ชนกันของฟิสิกส์อนุภาคพลังงานสูง โดยการใช้แบบจำลอง UrQMD ในการสร้างข้อมูลซึ่ง  
ประกอบไปด้วยข้อมูลซึ่งเป็นเหตุการณ์ส่วนใหญ่ (background events) และข้อมูลที่เป็นเหตุการณ์  
ค่าผิดปกติ (outlier events) ซึ่งอาจเป็นผลจากความผิดพลาดการระบุปริมาณความใกล้เคียงกลาง  
หรือความผิดพลาดของหัววัด วิธีการที่ใช้ยังสามารถประยุกต์ใช้ในการตรวจหาเหตุการณ์การชน  
กันของอนุภาค เพื่อค้นหาฟิสิกส์ที่น่าสนใจได้อีกด้วย เพื่อที่จะตรวจจับค่าผิดปกติ อัลกอริทึมที่ลด  
ขนาดมิติของข้อมูลได้ถูกประยุกต์ใช้ ได้แก่ Principle Component Analysis และ Autoencoder โดย  
ใช้ในการวัดค่าผิดพลาดในการสร้างใหม่ (reconstruction error) เป็นตัวแปรในการแยกแยะ  
เหตุการณ์ค่าผิดปกติออกจากเหตุการณ์ส่วนใหญ่ อัลกอริทึมแต่ละตัวจะถูกเปรียบเทียบ  
ประสิทธิภาพโดยใช้กราฟ Receiver Operating Curve ซึ่งแสดงให้เห็นว่าจำนวนของมิติที่ใช้แสดง  
ข้อมูล มีนัยสำคัญโดยตรงต่อประสิทธิภาพการทำงานของอัลกอริทึม ผลสรุปพบว่า โมเดลที่สามารถ  
แยกค่าความต่างของค่าผิดพลาดในการสร้างใหม่ของเหตุการณ์ส่วนใหญ่และเหตุการณ์ค่าผิดปกติ  
ได้ดีนั้น ใช้ปริมาณของพารามิเตอร์น้อยกว่า

สาขาวิชาฟิสิกส์  
ปีการศึกษา 2562

ลายมือชื่อนักศึกษา ปิ่นทอง ทัก  
ลายมือชื่ออาจารย์ที่ปรึกษา A. H. H.

PUNNATAT THAPRASOP : COLLISIONS USING UNSUPERVISED  
(DEEP-) LEARNING. THESIS ADVISOR : ASST. PROF.  
CHRISTOPH HEROLD, Ph.D. 77 PP.

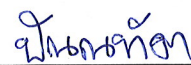
CUMULANTS/ MACHINE LEARNING/ QCD CRITICAL POINT

Different methods of supervised learning have been presented in this study which can be used for outlier detection in high energy nuclear collisions. The UrQMD model is used to generate the bulk background of events as well as different variants of outlier events that may result from misidentified centrality or detector malfunctions. The methods presented here can be generalized to different and novel physics effects. To detect the outliers, dimensional reduction algorithms are implemented, specifically Principle Component Analysis (PCA) and Autoencoders (AEN). The results show that the reconstruction error is a good measure to distinguish outliers from background. The performance of the algorithms is compared using a ROC curve. It is shown that the number of reduced (encoded) dimensions to describe a single event contributes significantly to the performance of the outlier detection task. Moreover, the model that is best suited to separate outlier events requires good performance in reconstructing events and at the same time a small number of parameters.

School of Physics

Academic Year 2019

Student's Signature



Advisor's Signature



## ACKNOWLEDGEMENTS

I would like to express my gratitude to following people for their encouragement, assistance, and support which have enabled me to complete my thesis. My deepest appreciation goes first and foremost to my thesis advisor Asst. Prof. Dr. Christoph Herold and my thesis co-advisors Dr. Jan Steinheimer and Dr. Kai Zhou, from Frankfurt Institute for Advanced Studies (FIAS) for their great support in patient guidance, consideration, and assistance throughout the study. Without their help, this work would not be possible for success.

I'm profound gratitude to Assoc. Prof. Dr. Panonsuk Meemon, Asst. Prof. Dr. Chinorat Kobdaj, Prof. Dr. Yupeng Yan, and Asst. Prof. Dr. Ayut Limphirat for being part of committee members and giving me insightful suggestions.

I am extremely grateful to FIAS, Frankfurt, Germany, for supporting computational resources, data from the UrQMD program, and also accommodation throughout the long term researching.

I am extending my thanks to my parents for their love and sacrifices for educating and preparing me for my future. I would like to say thanks to my friends and research colleagues in the EPHIC group at SUT physics.

Finally, my thanks go to all the people who have supported me to complete the research work directly or indirectly.

Punnatath Thaprasop

# CONTENTS

	<b>Page</b>
ABSTRACT IN THAI . . . . .	I
ABSTRACT IN ENGLISH . . . . .	II
ACKNOWLEDGEMENTS . . . . .	III
CONTENTS . . . . .	IV
LIST OF FIGURES . . . . .	VI
LIST OF TABLES . . . . .	VIII
<b>CHAPTER</b>	
<b>I INTRODUCTION . . . . .</b>	<b>1</b>
<b>II PHYSICS OF QUARK-GLUON PLASMA . . . . .</b>	<b>5</b>
2.1 Properties of quark and gluon . . . . .	5
2.2 Quark Gluon Plasma (QGP) . . . . .	6
2.3 The QCD phase diagram . . . . .	10
2.4 Experimental Probs, Net-Proton Number . . . . .	11
<b>III MACHINE LEARNING . . . . .</b>	<b>13</b>
3.1 Principle Component Analysis (PCA) . . . . .	14
3.2 Autoencoder (AEN) . . . . .	18
3.2.1 Artificial Neural Networks (ANNs) . . . . .	20
3.2.2 Fully Connected Neural Networks . . . . .	21
3.2.3 Convolutional Neural Networks . . . . .	22
3.3 Receiver Operating Characteristic (ROC) curve . . . . .	23
<b>IV METHODOLOGY AND RESULTS . . . . .</b>	<b>27</b>
4.1 Theoretical background and model setup . . . . .	27

## CONTENTS (Continued)

	<b>Page</b>
4.2 Result and Performance . . . . .	33
4.2.1 Radius comparison of PCA . . . . .	34
4.2.2 Reconstruction error of PCA . . . . .	39
4.2.3 PCA's radius comparison VS PCA's reconstruction error . . . . .	41
4.2.4 Reconstruction error of AEN . . . . .	42
4.2.5 Comparison of PCA and AEN . . . . .	45
4.2.6 Testing with other types of outlier . . . . .	49
<b>V SUMMARY AND CONCLUSIONS . . . . .</b>	<b>54</b>
REFERENCES . . . . .	56
APPENDIX . . . . .	62
CURRICULUM VITAE . . . . .	66



## LIST OF FIGURES

<b>Figure</b>		<b>Page</b>
1.1	Theoretical picture for visualizing QCD-phase diagram . . . . .	2
2.1	Quark-Gluon-Plasma creation by colliding two nuclei of heavy-ion . . .	9
2.2	The evolution of quark-gluon-plasma plotted on space-time diagram . .	10
2.3	QCD phase diagram shows the phase transition between quark-gluon- plasma and hadrons . . . . .	11
2.4	Energy dependence of correlated cumulant ratios . . . . .	12
3.1	Transformed axes of PCA . . . . .	15
3.2	The structure of autoencoder . . . . .	19
3.3	The example of fully-connected structure . . . . .	22
3.4	The example of convolution neural networks structure . . . . .	23
4.1	Energy dependence of cumulants and correlation functions . . . . .	28
4.2	An example of UrQMD input file . . . . .	30
4.3	Distributions of the multiplicity of charged particles . . . . .	31
4.4	Examples of charged particle multiplicity in the transverse momentum plane . . . . .	33
4.5	Distribution of events plotting in principle coordinate after operate 2 PC transformation . . . . .	34
4.6	Cumulative explained variance of applied principle components . . . . .	35
4.7	Histogram of radius after applying 2 PC transformation . . . . .	36
4.8	Histogram of radius after applying 80/320 PC transformation . . . . .	37
4.9	2 dimensional normalized histogram of the charged particle number for a single event . . . . .	38

## LIST OF FIGURES (Continued)

<b>Figure</b>		<b>Page</b>
4.10	Histogram of reconstruction error after applying 2 PC transformation . . . . .	39
4.11	Histogram of reconstruction error after applying 80/320 PC transformation . . . . .	40
4.12	ROC charts comparing the performance of PCA with different number of principle components . . . . .	42
4.13	ROC charts comparing the performance between two different methods of PCA . . . . .	42
4.14	Comparing two kinds of AEN structures . . . . .	44
4.15	Comparing input and its reconstruction applying 1 hidden layer . . . . .	45
4.16	Comparing input and its reconstruction applying 5 hidden layer . . . . .	46
4.17	Distribution of events plotting in 3 dimensional representation for PC and AEN transformation . . . . .	47
4.18	ROC charts comparing the performance of outlier detection for different algorithms . . . . .	48
4.19	Example of outliers when we randomly truncated events with 4 styles . . . . .	49
4.20	Example of information losing when we randomly null transverse momentum feature for all events . . . . .	50
4.21	Example of information losing when we randomly cut high range of transverse momentum feature for all events . . . . .	50
4.22	Distribution of events (unusual case) plotting in principle coordinate after operate 2 PC transformation . . . . .	51

## LIST OF TABLES

<b>Table</b>		<b>Page</b>
2.1	Properties of quarks . . . . .	6
2.2	Comparisons between “Big Bang” and “Micro Bang” . . . . .	7
3.1	Confusion matrix . . . . .	23
3.2	Confusion matrix . . . . .	24
4.1	Descriptions of sample input file . . . . .	30
4.2	Table summarizes the fraction of background (FP) events that are falsely identified as outliers for general case . . . . .	48
4.3	Table summarizes the fraction of background (FP) events that are falsely identified as outliers for unusual case . . . . .	53
1	Structure of the largest fully connected network for 100-bins input features	62
2	Structure of the smaller fully connected network for 100-bins input fea- tures . . . . .	62
3	Structure of the smaller fully connected network for 400-bins input fea- tures . . . . .	63
4	Structure of the convolutional AEN for $10 \times 10$ dimensional input features	63
5	Structure of the convolutional AEN for $20 \times 20$ dimensional input features	64

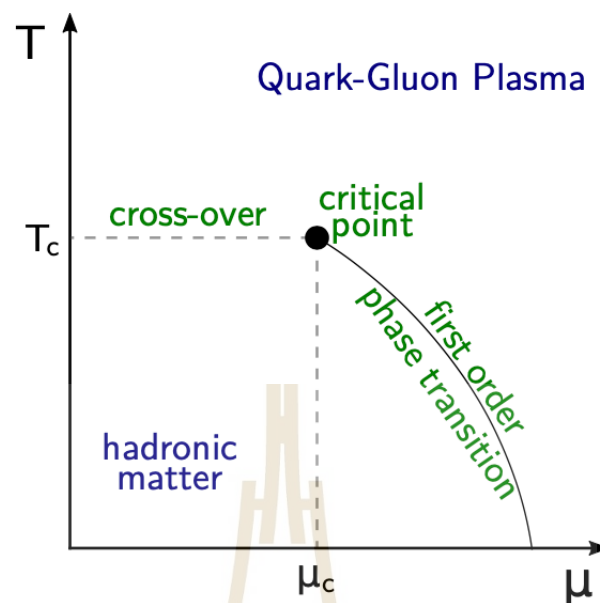
# CHAPTER I

## INTRODUCTION

The mystery of the universe is always a fascinating topic for humankind. The most popular theory that describes the starting point of the universe is known as “the Big Bang Theory”. Only a few second, after the big bang, matter was in a state of the extremely hot and dense fluid, designated “Quark-Gluon Plasma (QGP)”. Due to high temperature and density, quarks are not allowed to bind together to form hadronic matter. We call this stage “quark epoch”. The quark epoch lasts only a few milliseconds until the average energy of particle interaction is below the binding energy of hadrons. In particular following periods are known as “hadronic epoch”, When quarks are confined together within hadrons. The change of state of a strongly-interacting matter can be described by a QCD phase diagram. Theoretically, we use the QCD phase diagram to make sense of how QGP transit to the hadronic state, as shown in figure 1.1.

Heavy-ion collision experiments have been constructed to investigate the QCD phase diagram. The confirmation of QGP’s existence was formally announced in 2010 by CERN’s Super Proton Synchrotron (SPS) as “indirect evidence for a new state of matter” (Abbott, 2000). For the investigation of QGP properties, nuclei colliding experiments have been conducted in a various range of energies such as STAR (M. Aggarwal et al., 2010), PHENIX (Adcox et al., 2004), and NA61/SHINE (Abgrall et al., 2014).

The nuclear collisions at relativistic beam energies are an abundant source of particles created by the strong interaction. These particles and their correlations, in principle, carry information on the properties of matter in which they were created. The goal of all the existing and planned heavy ion experiments is to understand these properties and untangle the phases of matter from particle information that is measured in



**Figure 1.1** Theoretical picture for visualizing QCD-phase diagram. The vertical axis represents temperature. The horizontal axis represents the baryon chemical potential, which indicates the imbalance between matter and antimatter. At zero of baryon chemical potential, the crossover region is expected to be found. The first-order phase transition is expected at a higher temperature and baryon chemical potential. The critical point is expected to be found at the end of first-order phase transition Gronefeld, 2018.

the detectors. In many cases, new and interesting physics is hidden in rare events and/or rare particles as well as the correlations between these particles. Such is the case in the detection of new heavy particles, properties of charmed hadrons and higher order cumulants of particle multiplicity distributions.

To find and learn more about such rare probes, new experiments, like the CBM experiments at the upcoming FAIR facility, the NICA facility and the ALICE experiment at CERN, will be able to produce a huge amount of events every second. Since the amount of data generated in such events is very large, one has to find new methods to be able to find and classify new events very rapidly online in order to save the events for later in-depth analysis. For such scenarios, it is very useful to have a model at hand which is able to quickly and reliably determine whether an event contains any interesting information or is simply worthwhile to be saved on disk for later investigation.

Another challenge related to huge amounts of experimental data is related to possible interesting physics from a statistical analysis. A prominent example hereby is the analysis of the proton number multiplicity distribution as a function of beam energy by STAR at RHIC. In Au+Au collisions at  $\sqrt{s} = 7.7$  GeV, a significant deviation from a simple binomial distribution has been reported, which some interpret as a signal for a critical endpoint in the QCD phase diagram (Luo, 2016). As discussed in a previous work, this observation can be explained by a two peak anomaly in the proton number distribution (Bzdak et al., 2018). At the moment, however, the cause for such a two-bump distribution is unknown, possibilities include an experimental artifact or the effect of the QCD phase transition.

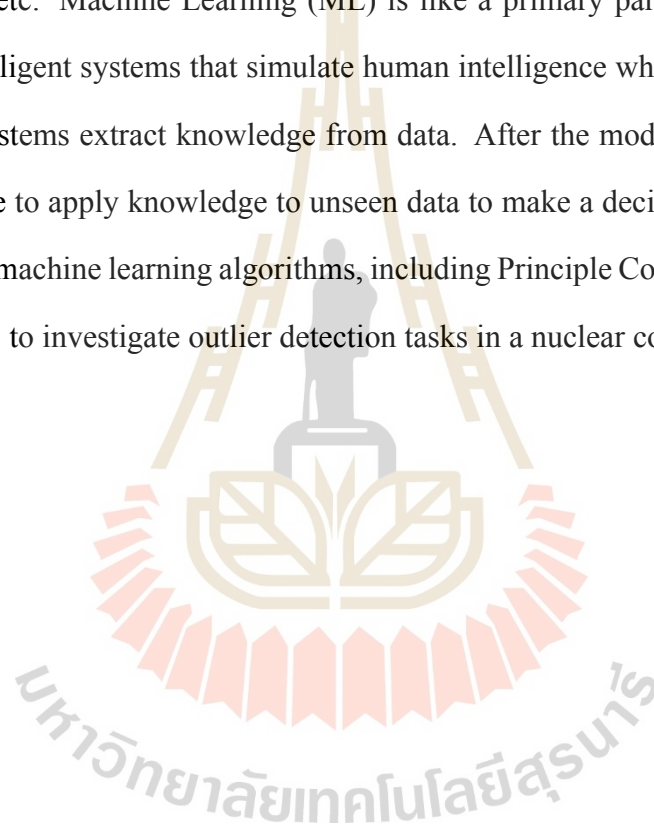
To find the actual source, a careful analysis of the events responsible for that distribution may be useful, which requires the identification of the corresponding events. If, for example, an imperfect centrality determination, a completely different event type, or even a detector malfunction would be responsible, characteristics of those events should be different from those of the bulk. Such events are called outliers.

In recent years, the detection of outliers has been an essential goal in the machine-learning community, see e.g. (Hawkins, 1980). In the present work, we will show how modern machine-learning (ML) methods can be applied to the detection of outliers in the context of high energy nuclear collisions. In particular, we will focus on collisions of gold nuclei at a center of mass energy of 7.7 GeV per nucleon due to the previously mentioned observation of interesting fluctuations and correlations. Our work here can be understood as a suggestion for an extended experimental analysis of that particular beam energy.

In general, however, the presented methods are applicable to outlier detection in various nuclear collision experiments and are not at all restricted to the specific example discussed in the following. Machine learning tools have nowadays become essential to

face experimental challenges in high-energy physics, with applications ranging from track finding at PANDA (Esmail et al., 2019) to b-jet tagging and the measurement of low-mass dielectrons at ALICE (Haake, 2017).

Nowadays, there are breakthrough developments in artificial intelligence (AI) both in the scientific field and industry. There are many real-world applications that guarantee the success of the AI field, such as a self-driving car, fraud detection, speech recognition, etc. Machine Learning (ML) is like a primary part of AI. We can think of AI as intelligent systems that simulate human intelligence while ML is the way that intelligent systems extract knowledge from data. After the model has learned enough it will be able to apply knowledge to unseen data to make a decision. In this work, we will develop machine learning algorithms, including Principle Component Analysis and Autoencoder, to investigate outlier detection tasks in a nuclear collision.



## CHAPTER II

### PHYSICS OF QUARK-GLUON PLASMA

The fundamental particles, known as hadronic matter, such as proton, neutron, and pion, are states of confined quarks and gluons. These can be studied under the concept of symmetry of the strong interaction through quantum chromodynamics (QCD), the theory of strong interaction between quarks and gluons. In the limit of vanishing quark masses, the state at high temperature/density, the chiral symmetry of QCD is restored. This process is known as “a restoration of chiral symmetry” where quarks and gluons are confined together by the strong force corresponding to Pauli exclusion principle. When the chiral symmetry is broken, the current quark mass becomes finite. However, If we consider the sum of the quark masses in a hadron, it is significantly smaller than the mass of hadron itself. The major goal of the ultra-relativistic heavy-ion experiments is to create and study a system of restored chiral symmetry, or the plasma state of quarks and gluons.

In this chapter, we will discuss idea and concept of QGP and the phase transition between QGP and hadronic matter

#### **2.1 Properties of quark and gluon**

Quark (Q) and gluon (G) are subatomic particle which are confined under the strong force to form particles we know as hadrons. The subatomic particles have properties:

There are six different flavors of quarks as shown in table 2.1 including up (u), down (d), strange (s), charm (c), bottom (b) and top (t). u-quark and d-quark are the most



stable version of all flavors. In addition to the stable version, there are also unstable versions, including c-quark and s-quark and t-quark and b-quark. With the short life span, they will decay to more stable versions.  $Q_q$  indicates the electric charge carried by each quark flavor.

**Table 2.1** Properties of quarks with different flavors are shown here.  $Q$  indicates the charge of each flavor,  $m_f$  denotes the mass of each flavor.

flavor	Quark	$Q$	$m_f(2\text{GeV})$
$u$	Up	$+\frac{2}{3}$	$3.5 \pm 2 \text{ MeV}$
$d$	Down	$-\frac{1}{3}$	$6 \pm 3 \text{ MeV}$
$s$	Strange	$-\frac{1}{3}$	$115 \pm 55 \text{ MeV}$
$c$	Charm	$+\frac{2}{3}$	$1.25 \pm 0.15 \text{ GeV}$
$b$	Bottom	$-\frac{1}{3}$	$4.25 \pm 0.15 \text{ GeV}$
$t$	Top	$+\frac{2}{3}$	$174.3 \pm 5.1 \text{ GeV}$

Quarks must have an additional quantum number aside from spin called *color* because they are fermions which should be subject to the Pauli principle\*. The particles such as  $\Delta^{++} = (uuu)$ ,  $\Delta^- = (ddd)$ , and  $\Omega^- = (sss)$  can be described according to the Pauli principle with 3 additional degeneracy factor† including red, green, and blue.

Gluons are different types of particles known as “Gauge Bosons”. They mediate the strong interaction between two or more quarks. It is usually been described analogously with photon that are considered as intermediate between charge particles.

## 2.2 Quark Gluon Plasma (QGP)

Go back to the universe’s beginnings; 10  $\mu\text{s}$  after the Big Bang, it was believed that the universe was filled with a thermalized plasma of deconfined quarks, antiquarks,

\*Pauli principle states that there is no two or more identical fermions (particles with half-integer spin) occupied the same quantum state within a quantum system simultaneously.

†If we account for anti-particle, then we have three more colors, including anti-red, anti-green, and anti-blue.

gluon and leptons (and other, heavier particles) - a Quark-Gluon Plasma (QGP). After hadronization of this QGP, it took another 3 minutes until the first small nuclei were formed from protons and neutrons called “primordial nucleosynthesis and chemical freeze-out”, another 400,000 years until atomic nuclei and electrons could combine to form electrically neutral atoms, thereby making the Universe transparent and liberating the Cosmic Microwave Background. To study the physics of QGP could imply the history of the universe and get the clues of how the matters were formed.

Atomic nuclei of lead are accelerated and collided to create an environment which existed only fractions of a second similar to what claim as the “Big Bang”. Some physicists called these experiments “Micro Bang” (or “Little Bang”). The table 2.2 shows the differences between the two.

**Table 2.2** The table makes a comparison between the Big Bang and micro bang. The time scale of the plasma expansion of a micro bang is less than of the Big Bang due to the gravitational force effect. In the laboratory, a considerable baryon number  $B$  is produced compared to total particle number  $N_b$  unlike in the early Universe. A significant of matter-antimatter asymmetry is expected to found in the micro bang. Where  $\tau$  time interval during QGP evolution. And  $N_b$  and  $N$  are the production number of baryon and total particles respectively.

Big-bang	Micro-bang
$\tau \simeq 10\mu s$	$\tau \simeq 4 \times 10^{-17}\mu s$
$N_b/N \simeq 10^{-10}$	$N_b/N \simeq 0.1$
gravitational force effect	no gravitational force effect

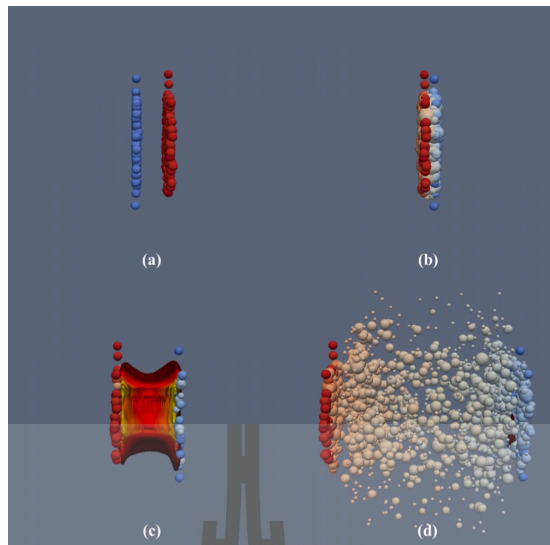
In 1987, Leon van Hove, proposed the term “quark gluon plasma” (Van Hove, 1987). If the temperature is above Hagedorn temperature, the scale of u/d quark mass, and the pressure exhibits the relativistic Stefan-Boltzmann, quark and gluon are freed from mass constituents. And indeed, in year 2000, quarks and gluons were first detected in the laboratory at CERN (Heinz et al., 2000)(Glanz, 2000).

The experiment of colliding particle creates hot and dense fireballs (Quark Gluon Plasma or QGP) that evolve through the transforming process. They found that gluons bind quarks together, acting as an exchange particle for the strong force to form hadrons

such as protons and neutrons. The process was called “hadronization”.

We use the simple transparent model of a heavy-ion collision to describe the mechanism of creating QGP and its evolution as in figure 2.1. First, the two nuclei of heavy-ion are accelerating to near the speed of light along the beam direction. They both got affected by Lorentz contraction into a disk-like shape. The factor  $\gamma = E_{beam}/M$  indicates the magnitude of contraction, where  $E_{beam}$  is the beam energy per nucleon and  $M = 0.94$  GeV is nucleon mass.  $\gamma$  is approximately 110 and 3000 at the respective top energies of RHIC and LHC. When both nuclei collide, neutrons and protons dissolve into the state of deconfined quarks and gluons. To guess what happens to them inside, one needs to draw a conclusion that QGP was finally transit to the hadronic matters at low temperature and baryon chemical density. The QCD phase diagram is used to describe the possibility of the phase transition from QGP to hadron gas. Many important probes are investigated (Luo, 2016), such as freeze out condition, nuclear modification factor  $R_{cp}$ , and net-proton number fluctuations, which we will review some of them in the future chapter. The crash produces so many thousand particles spread out with information about what happened inside. Physicists call the spread-out particles as asymptotic particles and study them actively. To scope down a little bit, we mainly focus on the strongly-interacting matter. The nature of quarks and gluons is themselves never been observed as free objects. In a normal state, they are confined. Only in the situation of a high-energy interaction can we see them separately for the short instant of time. Moreover, quarks and gluons interact only with a strong force, which is a vast experimental challenge regarding study their properties.

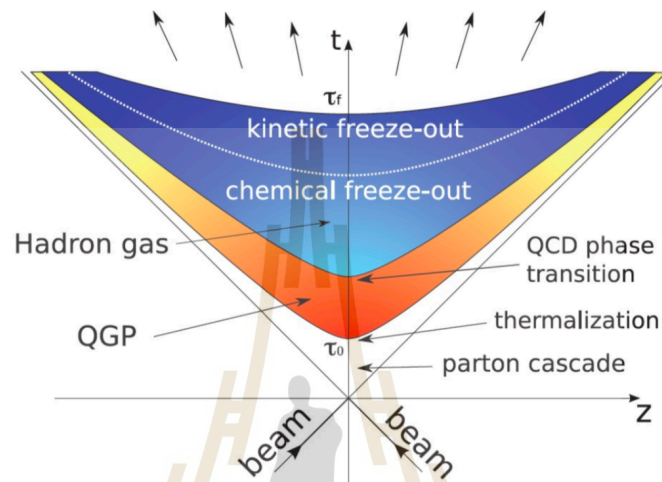
During the early collision, the system produces a large amount of heat in a tiny region, causing a huge energy density rapidly. The QCD process and interaction in high energy regions produce cascades of radiation called *a parton cascade* or *a parton shower*. The temperature keeps increasing up to the limit that the nucleus is melted at



**Figure 2.1** (a) The two nuclei of heavy-ion are accelerating to near the speed of light along the beam direction. They both got affected by Lorentz contraction into a disk-like shape. (b) Protons and neutrons dissolve for a brief instant, liberating their constituents (quarks and gluons) to form a Quark-Gluon-Plasma (very hot and dense fireball). (c) They both collide, the nuclei of neutrons and protons dissolve into the state of deconfined quark and gluon. The intermediate plasma with extremely low viscosity, behaves like a perfect fluid. (d) The crash produces thousand of particles spread out to the detectors. [ref: MADAI collaboration, Hannah Petersen and Jonah Bernhard]

$\tau_0$ . It behaves like a perfect fluid with extremely low viscosity. At this point, we can call the unconfined color charged with gluon as quark-gluon plasma (QGP). The QGP around the center of mass of the collision keep expanding cause the temperature to cool down near the equilibrium this process known as “Thermalization”. With an increase in strength of the strong force, the QGP plasma of weakly interactive quarks gets replaced with the more familiar hadronic phase in which quark appears bound in baryons and mesons. This phenomenon is called confinement. At this stage, the phase of matter changes from the QGP phase to the hadronic phase. At some time,  $\tau_f$ , the hadrons will cease to interact and stream freely into the detector where they are measured. The process known as “freeze-out”. More precisely, there are two types of freeze-out: first comes the chemical freeze-out, after which no inelastic collisions occur anymore, resulting in the fixed chemical composition of the system. Then, at the moment when

elastic collisions cease as well, one speaks of a thermal or kinetic freeze-out. From the definition, it is clear that the chemical freeze-out will happen before the kinetic freeze-out (Herold, 2017). The evolution of QGP are shown in the figure 2.3. The figure 2.2 shows the evolution of QGP.

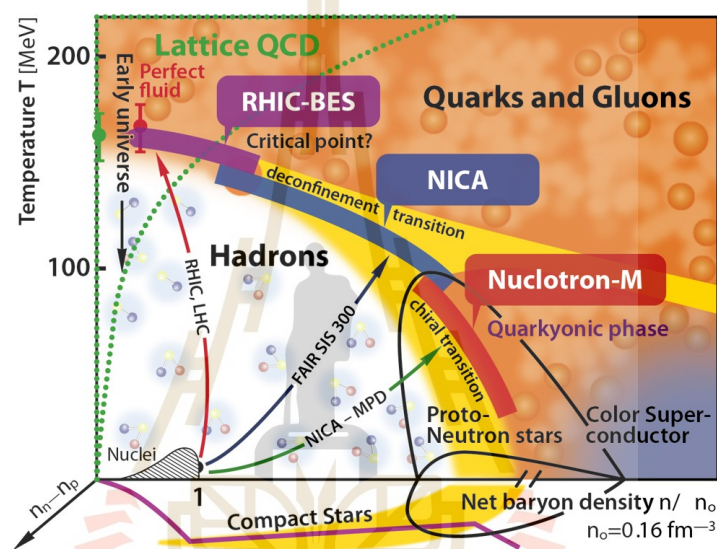


**Figure 2.2** The figure shows the evolution of QGP starting from the colliding nuclei, parton cascade, thermalization, QCD phase transition, and freeze-out. The evolution is plotted corresponding to space-time diagram (Shi, 2010).

### 2.3 The QCD phase diagram

During the phase of QGP evolution, the freeze-out dynamics are important bulk properties of the hot and dense nuclear matter. It maps out the regions where we can access in the QCD phase diagram, which provides a baseline for finding the QCD phase transition and critical point. A careful comparison between experimental data and the results from the lattice gauge theory calculations, the scientists concluded that the transition temperature (expressed in units of energy) begins at 175 MeV (175 million electron volts) (Bicudo, 2010). The exploration of QCD can be drawn as in figure 2.3. For a small net baryon density region, it is expected to found a crossover of the chiral transition while evidence for a critical point (CP) and first-order phase transition are expected

to locate at the regions with large values of net baryon density. If indeed a first order phase transition occurs at large net baryon densities and intermediate temperatures, also an associated end point must exist at which the transition becomes second order. Theoretical assumption claim that first and second order phase transitions give rise to many interesting phenomena, especially related to fluctuations which we will discuss in the next section (Koch et al., 2013).

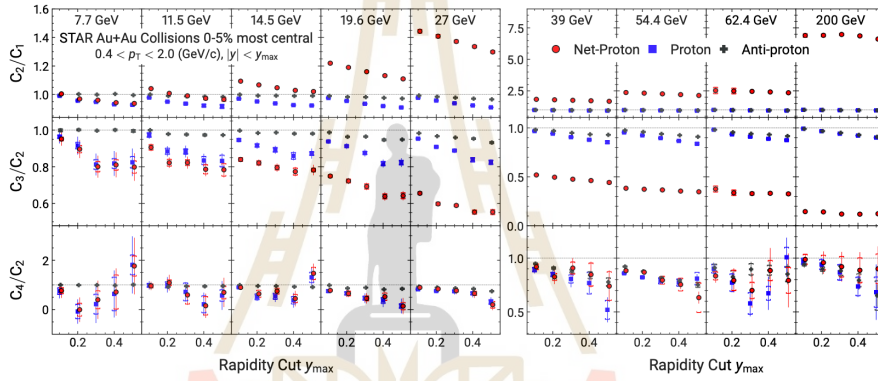


**Figure 2.3** The figure shows the various region of the QCD phase diagram indicated by the temperature  $T$  [MeV] and the net baryon density  $n/n_0$ . Many of them had explored by experiments in the nuclear collision and had investigated theoretically, including RHIC-BES, NICA, and Nuclotron-M as labels in the figure. The phase transition lies in between two-phase of matter, hadrons and quark-gluon plasma. (Bicudo et al., 2011)

## 2.4 Experimental Probs, Net-Proton Number

Fluctuations of conserved quantities, such as net baryon number ( $B$ ), had been proposed as a sensitive probe to search for the signature of a QCD critical point in heavy-ion collisions (Asakawa et al., 2009). In the final state of fluctuation, the relations between the number of proton and baryon has been explored (Kitazawa et al., 2012). It is shown that the correlations between the isospins of nucleons in the final state are almost

negligible over a wide range of collision energy. Net-proton number can be used as the approximation of net-baryon. By varying the beam energy of the collision, one is able to change the density and temperature of the created system, which allows to “scan” the phase diagram of QCD and hopefully locate the onset of phase transition signals. The STAR experiment has measured the energy dependence of  $C_n$  up to fourth order of the net-proton multiplicity distribution from Au+Au collisions with a larger acceptance of  $p_T$ ,  $0.4 < p_T < 2.0$  GeV/c (Adam et al., 2021). The results have been presented at 9 different collision energies  $\sqrt{s_{NN}} = 7.7, 11.5, 19.6, 27, 39, 54.4, 62.4$  and 200 GeV.



**Figure 2.4** Energy dependence of correlated cumulant ratios  $C_4/C_2$ ,  $C_3/C_2$  and  $C_2/C_1$  of Net-proton, Proton and Anti-proton distribution in Au+Au collisions at 0-5% most central. The cumulant ratios are plotted against Rapidity Cut  $y_{max}$  (Abdallah et al., 2021) for various collision energy,  $\sqrt{s_{NN}} = 7.7, 11.5, 19.6, 27, 39, 54.4, 62.4$  and 200 GeV.

Figure 2.4 shows the corrected cumulant of  $C_4/C_2$ ,  $C_3/C_2$ , and  $C_2/C_1$  of Net-proton, Proton and Anti-proton distribution in Au+Au collisions measured by STAR. The measurements are measured with transverse momentum  $0.4 < P_T < 2$  GeV/c and at mid-rapidity  $|y| < 0.5$ . The cumulant ratios,  $C_4/C_2$  and  $C_3/C_2$ , show a clear non-monotonic variation with  $\sqrt{s_{NN}}$  for 0-5% centrality with collision energy  $\sqrt{s_{NN}} = 7.7$  GeV. Above 39 GeV, only the  $C_4/C_2$  deviates from monotonic trends. One may note that this is only statistical errors shown in the figure, which are still large due to limited statistics. The systematical errors, which are dominated by the efficiency correction and the particle identification, are being studied.

## **CHAPTER III**

### **MACHINE LEARNING**

These days, the tremendous increase of data exceeds the human effort to make use of it. Due to the potential of learning a large amount of data, Artificial Intelligence (AI) has been used in many areas. AI can be understood as a branch of computer science concerned with building smart machines capable of performing tasks that typically require human intelligence. The core part of AI is known as machine learning. It uses the statistical method to learn from data and improve with experience. Roughly, machine learning methods can be classified into 2 groups:

- Supervised learning is the machine learning that learns from examples of input and human-labeled output pairs. We call this training data or training example. In supervised learning, each example is a pair consisting of an input object (typically a vector) and a desired output value (also called the supervisory signal). A supervised learning algorithm analyzes the training data and produces an inferred function, which can be used for mapping new examples. An optimal scenario will allow for the algorithm to correctly determine the class labels for unseen instances. This requires the learning algorithm to generalize from the training data to unseen situations in a “reasonable” way.
  
- Unsupervised learning is a type of machine learning that looks for previously undetected patterns in a data set with no pre-existing labels and with a minimum of human supervision. In contrast to supervised learning, unsupervised learning, also known as self-organization allows for modeling of probability densities over inputs.



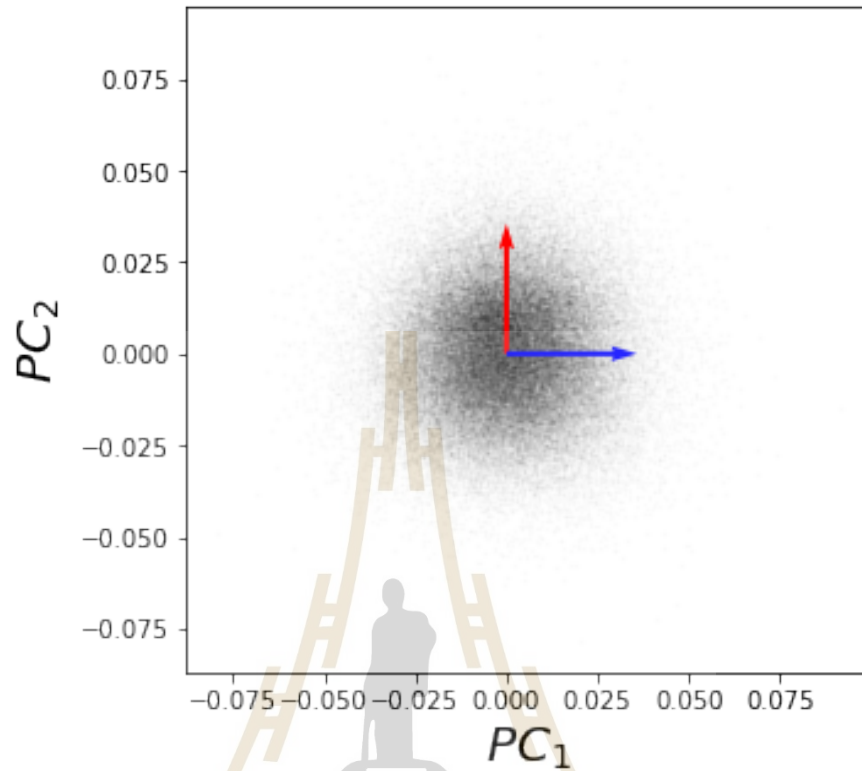
Outlier or anomaly detection is the task of detecting instances that deviate from a typical scenario or are an unlikely result of a random deviation from the expectation value. Anomaly detection has a wide range of applications e.g., email spam detection, fraud detection for credit cards, etc. In scientific fields and especially experimental data analysis, outlier detection has been used successfully (Esmail et al., 2019).

The scope of this work is restricted to unsupervised learning. We will investigate outlier detection tasks in nuclear collision physics in an unsupervised manner. The coding algorithms applied include Principle Components Analysis (PCA) and Autoencoder (AEN). Since these algorithms have different learning processes, one is a linear approach, and another one is a non-linear approach. We use python to develop an unsupervised (deep-) learning algorithm along with scikit-learn and keras packages together as a framework in this study.

### 3.1 Principle Component Analysis (PCA)

The Principle Component Analysis (PCA) is a statistical procedure that generates a low-dimensional representation of a dataset by an orthogonal linear transformation of the original data. The PCA transforms the original  $n$  coordinates of one input dataset into a new set of  $m$  coordinates called principal components (PC). These components are chosen to represent the data by maximizing the variance of the data in the new set of  $m < n$  dimensions. The first axis of PC is selected from the set of lines that pass through the origin of the original  $n$  coordinates, and it must be the axis that preserves the most variances of the dataset. So it has to maximize the distance of the projection point (from the origin) of the dataset. For the second axis of the PC, it must be orthogonal with the first axis and also maximizes the distance of the projection point. The other principle components are created in the same pattern. They must be orthogonal and be the line that preserves the most variance of dataset, figure 3.1 shows the example of it.

We proceed with these following steps to find the principle components:



**Figure 3.1** The picture shows the example of transformed axis of PCA. The transformed vectors, red and blue, are orthogonal to each other. Both preserve the most variance of datasets

1. Standardization: Calculate the mean ( $\mu$ ) of all numeric features ( $x$ ) of the target datasets. Translate  $x$  by  $\mu$ . Scale the data by a factor of the standard deviation of target datasets,  $\sigma$ . Each variable will contribute equally to analysis.

$$z = \frac{x - \mu}{\sigma} \quad (3.1)$$

We can represent a single sample with the vector  $\vec{Z}$  that contain  $z_1, z_2, \dots, z_m$ .

Where  $m$  is the number of features.

2. Covariance matrix computation: We can compute the covariance of any two ran-

dom variables  $z^j$  and  $z^k$  using the following formula:

$$A \equiv cov(Z^j, Z^k) = \frac{1}{n-1} \sum_{i=1}^n (Z_i^j - \bar{Z}^j)(Z_i^k - \bar{Z}^k)^T \quad (3.2)$$

Using equation (3.2), we will get the covariance matrix,  $A$ . The matrix  $A$  will be used in the next step. The result of the covariance matrix would be a square matrix of  $Z^j \times Z^j$  dimensions (or  $Z^k \times Z^k$  dimensions).  $Z^j$  and  $Z^k$  are the matrix of two random variables,  $z_i^j$  and  $z_i^k$ . Where  $z_i$  represents the  $i$ th sample of  $z$ .  $j$  and  $k$  are labels of correlated features, analogously to height and width for example.

3. Compute eigenvectors and corresponding eigenvalues: Using the relation of Eigenvectors and corresponding eigenvalues in the equation (3.3) to find the set of pairs of an Eigenvector and its Eigenvalue.

$$A\vec{Z} = \lambda\vec{Z} \quad (3.3)$$

4. Choose the  $k$  eigenvectors with the largest eigenvalues: Sort the eigenvectors with respect to their decreasing order of eigenvalues, choosing  $k$  out of them, where  $k$  is the number of dimensions you wish to have in the new dataset.

In this work, we implement the PCA with a varying number of PCs to investigate the performance of outlier detection as a function of the number of PCs. We are not going to fine-tune the best number of components because, for all practical purposes, the algorithms need to be varied individually to find the proper set of parameters for the specific task at hand.

Having generated the original event-by-event momentum features and then used the PCA to reduce their dimensionality to  $m$  dimensions, there are two methods available to find outliers based on the output of the PCA.

The first one is to make a so-called *radius comparison (RC)*. The radius (distance from the origin) of each data point, in the new principal components, is calculated as:

$$r(i) = \sqrt{X_{PC_1}^2(i) + X_{PC_2}^2(i) + X_{PC_3}^2(i) + \dots + X_{PC_m}^2(i)}, \quad (3.4)$$

where  $X_{PC}$  are the coordinates of the new  $m$ -dimensional principal components and  $(i)$  refers to a single event.

Using the PCA radius will help later to visualize how outliers and background are distributed in configuration space (space where they are represented by principle coordinates in a lower dimension).

The second method used to identify outliers is to calculate the *reconstruction error (RE)* of the reduced representation. The reconstruction error quantifies the error that occurs when an input feature vector is projected on the new reduced set of  $m$  dimensions and then is reconstructed from the reduced dimensions to the original dimensionality. One could also say it quantifies the information loss that occurs for a single instance by performing the PCA.

We can calculate the reconstruction error of any  $i^{th}$  event as:

$$RE(i) = \frac{1}{N} \left( \sum_{j=1}^N [X_{rec_j}(i) - X_j(i)]^2 \right)^{\frac{1}{2}} \quad (3.5)$$

Where

- $N$  is the number of dimensions of the input data.
- $X_j(i)$  is the  $j^{th}$  component of the  $i^{th}$  input event. Usually one sums over all  $N$  components/dimensions of the input event.

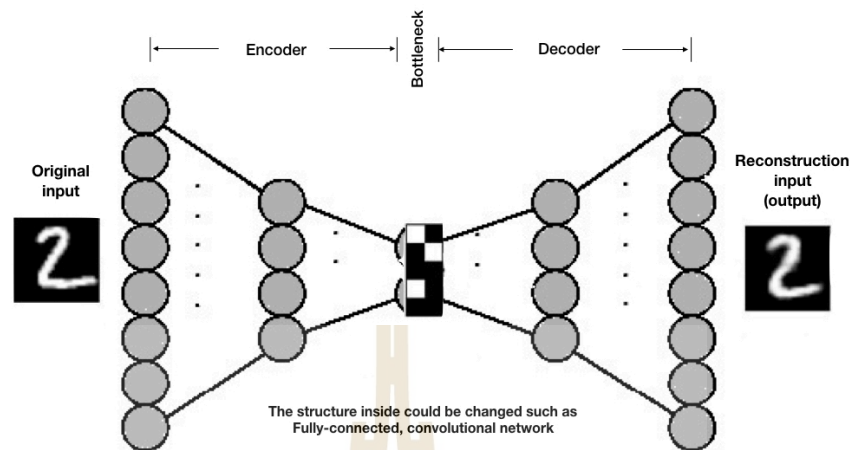
- $X_{rec_j}$  is the  $j^{th}$  component of the  $i^{th}$  reconstructed event.

When we reduce the dimensionality of data while keeping a significant portion of the information in the principal components (as defined as the components with the largest variance), the reconstruction loss is expected to differ between two types of events (outliers and background). Therefore, the properties of the reconstruction error can be used as an indicator to detect an anomaly.

### 3.2 Autoencoder (AEN)

Autoencoders (AEN) are deep artificial neural networks that learn to reduce the dimensionality of input data. They follow a similar encoding-decoding strategy as PCAs but are not limited to linear projections of the input data; thus, they can deal with more complex input. An AEN has been shown to also successfully denoise input data, which makes them very useful for the study of nuclear collision data. In an AEN, there are usually 3 components and 1 evaluator working together:

1. The encoder is the network part that learns how to reduce the dimension of the input data and compresses it into an encoded representation.
2. The Bottleneck (Hidden features, encoded representation) is the part that contains the dimensionally reduced representation of the input data. This part tries to preserve as much information as possible from the original input data.
3. The Decoder is the part that decodes the bottleneck/hidden representation back to the output data, preserving as much significant information from the input data as possible. The output of the Decoder has the same dimensionality as the input data; in an ideal case, it is an almost perfect copy of the input.
4. The Reconstruction Loss (reconstruction error) is an indicator that measures how



**Figure 3.2** The picture shows the main components of AEN, including encoder, bottleneck, and decoder. The AEN learn to encode number 2 as the input data and keep it in encoded representation or the bottleneck. Finally, the decoder reconstructs data back base on information kept in the bottleneck. Reconstruction loss has been calculated to measure efficiency.

well the output resembles the input and thus has been reconstructed from the hidden representation.

One can apply many types of architectures such as convolutional neural networks and fully-connected neural networks for the encoder and decoder parts of the network depending on which structure most efficiently can encode essential features of the input data. These different architectures are explained in more detail in the subsection 3.2.2 and 3.2.3. For the parameters used in this work, see appendix.

The data has been reduced into a lower dimension but with different mathematical algorithms compared with PCA (AENs learn from non-linear relations). However, the idea of finding the reconstruction error of AEN is the same as for PCA. Therefore, it can be used to detect an anomaly since two classes have a different range of value of reconstruction loss.

### 3.2.1 Artificial Neural Networks (ANNs)

Artificial neural networks are essentially mapping functions that map an  $n$ -dimensional input on an  $m$ -dimensional output. In the specific case of an autoencoder network, the input dimension is equal to the output dimension. Besides these so-called input and output layers a neural networks consists of a varying numbers of hidden layers, with neurons who themselves perform a non-linear transformation on their input. The value  $y = ax + b$  of any given neuron (where  $a$  and  $b$  are parameters and  $x$  is the input of the neuron) serves as argument of a so-called *activation function* which can take on different forms (often sigmoid and Relu functions are used). Equation (3.6) and (3.7) show the formula of sigmoid and Relu functions respectively.

$$S(y) = \frac{1}{1 + e^{-y}} \quad (3.6)$$

$$R(y) = \begin{cases} 0 & \text{for } y \leq 0 \\ y & \text{otherwise} \end{cases} \quad (3.7)$$

Depending on the structure of the network, the neurons of one single layer can be connected to any number of neurons in the next layer. For example, in a fully connected neural network, the output of the  $j^{\text{th}}$  neuron in the  $(i + 1)^{\text{th}}$  layer is the sum of all outputs of the  $i^{\text{th}}$  layer  $y_{i+1,j} = f(\sum_k a_{k,i+1}x_k + b)$  where  $k$  is the index of a neuron in the  $i^{\text{th}}$  layer and  $f()$  is an activation function. Such a network can easily have a large number of parameters  $a_{j,i}$  and  $b$  to be determined. The determination of these parameters is done during the training phase of the network where a loss function is minimized. In the autoencoder network, the loss function is simply defined as the reconstruction error of the input vs. output comparison. In our specific case this is done by calculating the mean squared error (*mse*) of the networks output with respect to the networks input, where

the squared errors for all dimensions are summed as shown in equation (3.8). The parameter values of the network are changed using a gradient descent method in order to minimize errors by keep changing weights. The equation (3.9) shows the example of gradient descent.

$$MSE = \frac{1}{n} \sum_i^n (y'_i - y_i)^2 \quad (3.8)$$

$$a_i = a_0 - r \nabla y_i \quad (3.9)$$

Where

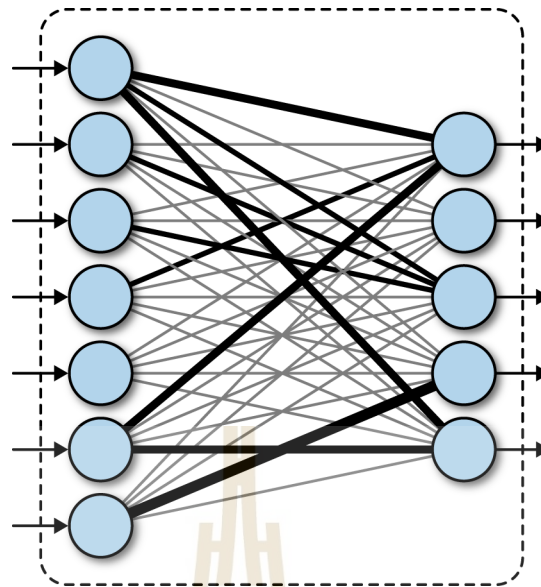
- $n$  is the number of samples.
- $y'_i$  and  $y_i$  are the label and the output of the  $i^{\text{th}}$  layer.
- $a_i$  and  $a_0$  are updated weight and old weight parameters.
- $r$  is the learning rate parameter
- $\nabla y_i$  represents the magnitude of a weight change.

The gradient descent in our calculations will be an Adam optimizer (*adam*) which is provided by the *Tensorflow* library. For a much more in-depth explanation on neural networks we refer the interested reader to (Mehta et al., 2019)

### 3.2.2 Fully Connected Neural Networks

The term fully connected comes from the fact that each node in one layer of a network is connected to all nodes of the next layer. A Fully connected layer is a function from  $\mathbb{R}_m$  to  $\mathbb{R}_n$  shown in Figure 3.3. The advantage of fully connected neural networks is they are good at pattern recognition without need for special assumptions to be made for the input data.

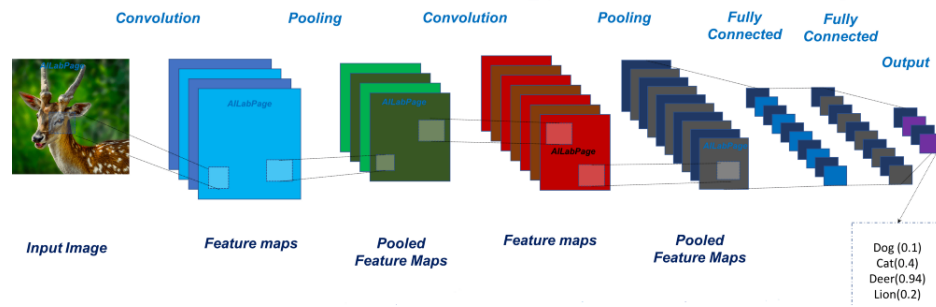




**Figure 3.3** A Fully-connected layer is a function mapping from  $\mathbb{R}_m$  to  $\mathbb{R}_n$ . Each node in one layer is connected to all nodes of the next layer

### 3.2.3 Convolutional Neural Networks

Due to information loss across spatial dimensions of fully connected network, Convolutional Neural Networks are a special kind of network structure which takes into account the two dimensional structure. This type of neural network is suitable to deal with data structure like an image. The CNN takes a two dimensional array as input as well as output. Instead of mapping every input pixel to an independent neuron the CNN uses convolutional kernels on the input image. These convolutional kernels have the dimension  $m \times m$  or written  $(m, m)$ . In each convolutional layer these kernels can take the form of  $n$  so called feature maps (FM) which constitute the trainable parameters of the model. The general training procedure is similar to that of the fully connected network in that a gradient descent algorithm is used to change the parameters such that the loss function (again the mean squared error) is minimized. In addition to the kernel size and the feature maps, the CNN also can use strides (step size with which the kernels scan the image) and padding (additional pixels which are added on the images boundary) to select the sliding patterns in particular parts of the two dimensional data, or images.



**Figure 3.4** The convolutional neural networks (CNNs) take the two dimensional input image into account as well as output. The CNNs use convolutional kernels on the input image or feature map. In addition to the kernel size and the feature maps, the CNN also can use strides and padding

### 3.3 Receiver Operating Characteristic (ROC) curve

We will introduce some concepts of a machine learning tool known as a *confusion matrix* to later on define what is the ROC curve. Let's start with an example of a confusion matrix for a binary classifier (which can be used as a reference concept in this work).

**Table 3.1** The table give an example of confusion matrix.

N = 3300	Predicted: NO	Predicted: YES
Actual: NO	1000	200
Actual: YES	100	2000

The table shows an example of a confusion matrix for a disease classifier. Each cell informs the number of patients that are classified by a doctor. For example, the doctor predicts 1000 healthy (no disease) patients as healthy (correctly-classify), 200 healthy patients as affected (wrongly-classify), 100 affected patients as healthy (wrongly-classify), and 2000 affected patients as affected (correctly-classify). We can see the performance of the doctor through the confusion matrix. He diagnoses 3000 cases correctly and 300 cases wrongly. Now we assume that one doctor is a machine learning algorithm with

particular parameters. The next step, we need to compare which, algorithm and parameters, say doctors, is the best to classify outliers (a disease in this case).

**Table 3.2** The table give an example of 4 classes of confusion matrix including True Positive (TP), True Negative (TN), False Positives (FP), False Negatives (FN).

N = 3300	Predicted: NO	Predicted: YES
Actual: NO	1000	200
Actual: YES	100	2000

- True Positive (TP): These are cases in which a doctor predicted yes (they have the disease), and they do have the disease.
- True Negative (TN): A doctor predicted no, and they truly don't have the disease.
- False Positives (FP): A doctor predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- False Negatives (FN): A doctor predicted no, but they actually do have the disease. (Also known as a "Type II error.")

To quantify the quality of the different models (doctor's ability) to find outliers, a *Receiver Operating Characteristic* (ROC) curve is used. In other words, it is used to describe the performance of a classification model. The ROC curve is a method to estimate and compare the performances of ML algorithms since it tells how much backgrounds are detected from relevant signals.

In this work, we use the logarithm of the inverse of the fraction of incorrectly classified events ( $\log \frac{1}{\epsilon_b}$ ) plotted against the fraction of correctly classified signal events ( $\epsilon_s$ ), as ROC curve. Here, the subscripts  $b$  and  $s$  stand for background and signal, respectively. For our specific examples with two centrality classes, peripheral (signal)

and central (background), we define the fraction of events that are correctly classified as signal out of all events classified as signals by:

$$\epsilon_s = \frac{TP}{TP + FN} . \quad (3.10)$$

The fraction of events incorrectly classified as signal out of all events classified as background is given by:

$$\epsilon_b = \frac{FP}{FP + TN} . \quad (3.11)$$

Hereby we use:

- $TP$  (True Positives): number of correctly classified signal events
- $FN$  (False Negatives): number of wrongly classified background events
- $FP$  (False Positives): number of wrongly classified signal events
- $TN$  (True Negatives): number of correctly classified background events

According to equations (3.10) and (3.11),  $\epsilon_s$  and  $\epsilon_b$  are proportional to the number of correctly classified events and the number of wrongly classified events, respectively. Due to a large number of background compared to outlier events, log scaling is applied to  $1/\epsilon_b$ . This makes the relation between  $\epsilon_s$  and  $\log(\frac{1}{\epsilon_b})$  more obvious to notice.

To find the relation between  $\epsilon_s$  and  $\log(\frac{1}{\epsilon_b})$ , a histogram of binary classes will be cut on one value of the horizontal axis \* separating the left-hand side from the right-hand side starting from the minimum value until the maximum value. The TN and FN will be measured only from the left-hand side while the TP and FP will be measure only from the right-hand side per each cut. Then, we can calculate the values of  $\epsilon_s$  and  $\log(\frac{1}{\epsilon_b})$  using equations (3.10) and (3.11), respectively. The first cut will always gives the value

---

\*which are one value of radius and reconstruction error in this work

of  $\epsilon_s$  and  $\log\left(\frac{1}{\epsilon_b}\right)$  as 1. When we vary parameters, for better performance,  $\epsilon_s$  trends to slowly decrease, while  $\log\left(\frac{1}{\epsilon_b}\right)$  trends to increase quickly. Therefore, the better curves will stay on the upper-right of the graph, see in section 4.2.



# CHAPTER IV

## METHODOLOGY AND RESULTS

In this part, we will present methodology and results in the following steps:

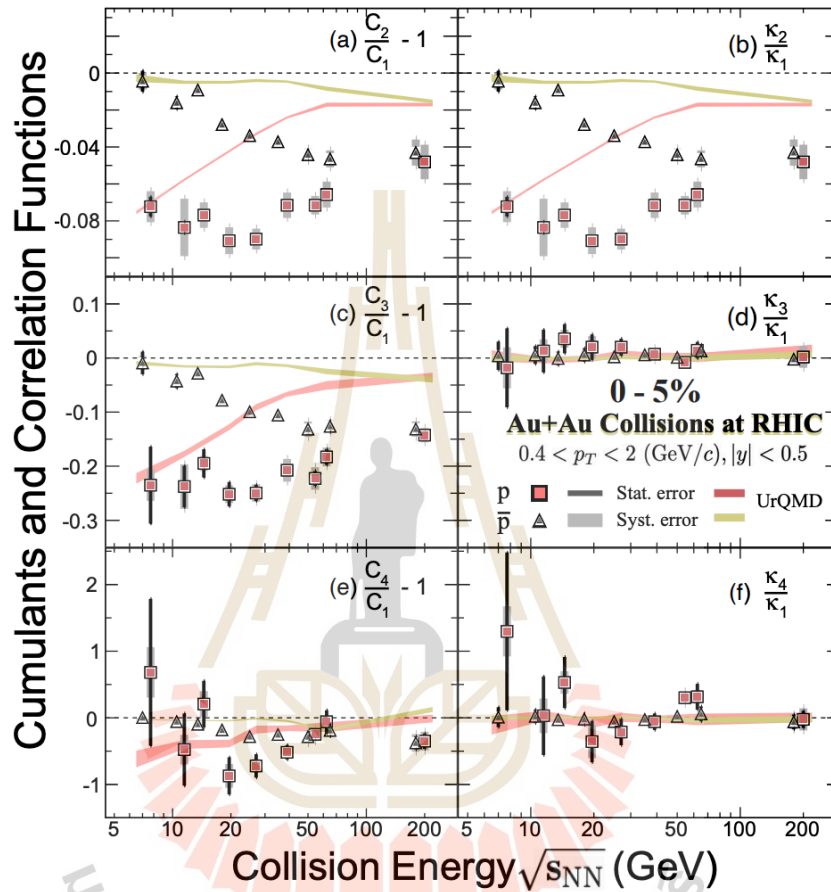
1. Introduction of the theoretical background and model setup of UrQMD including system parameters and how is the data can be generated.
2. Discussion of the results and implication for experiments. Then we show and compare the performance for the different methods for the specific task.

### 4.1 Theoretical background and model setup

The main goal of the Beam Energy Scan (BES) program at the BNL Relativistic Heavy Ion Collider (RHIC) is to study the QCD phase structure. This is expected to lead to the mapping of the phase diagram for strong interactions in the space of temperature ( $T$ ) versus baryon chemical potential ( $\mu_B$ ). Both theoretically and experimentally, several advancements have been made towards this goal. The magnitude of conserved quantity fluctuations are distinctly different in the hadronic and the QGP phases. The different areas in phase diagram indicate different values of temperature and pressure. The susceptibility, which is defined as the derivative of free energy density or pressure of a thermodynamic system at a given temperature with respect to the chemical potential, can be related to the cumulants of the event-by-event distribution of the associated conserved quantity (Stephanov, 2009). STAR detector at RHIC measures energy dependence ( $\sqrt{s_{NN}} = 7.7-62.4$  GeV) of net proton cumulant ratios.

The latest STAR experiment, (Abdallah et al., 2021) shows an interesting scaled cumulant ratios deviates from monotonic trends at  $\sqrt{s_{NN}} = 7.7$  GeV. In figure 4.1 on

the bottom left, one does observe non-monotonic energy dependence of proton  $C_4/C_1$  in the 0 – 5% central collision. This is because the forth-order cumulants ( $C_4$ ) contains contributions from second, third, and forth order factorial cumulants ( $\kappa_n$ ), see appendix.



**Figure 4.1** Energy dependence of the scaled proton and anti-proton cumulants and correlation functions in 0 – 5% central Au + Au collision at  $\sqrt{s_{NN}} = 7.7, 11.5, 14.5, 19.6, 27, 39, 54.4, 62.4$  and 200 GeV. The error bars and bands represent the statistical and systematic uncertainties, respectively.

The anomaly could be due to new physics, critical phenomena, a phase transition, or be a result of detector issues or misidentified events. It was suggested earlier that such an anomalous distribution could be created by misidentification of a few peripheral events as central events. In any case, it is important to identify those events which are responsible for the anomaly and investigate them in more detail. In, (Bzdak et al., 2018), the authors investigate the proton multiplicity distribution as the superposition

of two event classes including Poisson and binomial. Poisson and binomial are the representation of the combination between small number of rare events (outliers) and a large bulk of background. This situation is a perfect analogy to the world of high energy physics collision that outlier and background events are entangle. Nowadays, machine/deep Learning (ML/DL) algorithms are well known for the feature extraction problem. To determine the significant causes of distribution with as least as human assumption, we propose to implement unsupervised learning to disentangle any two classes based on the feature of momentum space. This investigation can be also used as an alternative cross-check of the previous studies (Bzdak et al., 2018). Moreover, machine learning has no bias in analyzing data. We can train the machine in various situations to handle e.g. broken events etc.

In this work, we use UrQMD (Ultrarelativistic Quantum Molecular Dynamics) model (S. Bass et al., 1999) as our source of data. UrQMD is a transport model for simulating heavy-ion collisions in the energy range from SIS to RHIC. It is a simulation package based on an effective Monte Carlo solution of the Boltzmann transport equations (Bleicher et al., 1999). That means that hadrons are propagated on straight lines until they scatter according to experimentally known cross-sections. It is widely used in high energy physics studies as it gives realistic results for the yields and momentum spectra of produced particles over a wide range of beam energies. It is designed as a multipurpose tool for studying a wide variety of heavy-ion related effects ranging from multifragmentation and collective flow to particle production and correlation. We can generate data from UrQMD by setting the system parameter in the input file and compiling through the command line in a UNIX based system. Figure 4.2 shows an example of an input file. All parameters are explained in detail in table 4.1.



```

# this is a sample input file for urqmd
# Ap Zp
pro 197 79 # projectile (here: gold nucleus)
# optional: special projectile: ityp, iso3 (here: pi plus meson)
# PRO 101 2
# At Zt
tar 197 79 # target (here: gold nucleus)
nev 10 # number of events
tim 200 200 # time to propagate and output time-interval (in fm/c)
elb 160.0 # incident beam energy in AGeV
imp -3.0 # weighted impact parameter distribution (from 0-3 fm)
eos 0 # Equation of State (used in transport part): CASCADE mode
# some options and parameters
cto 4 1 # output of initialization
ctp 1 1.d0 # scaling for decay width of Resonances
f15 # no output to file15
# end of file
xxx

```

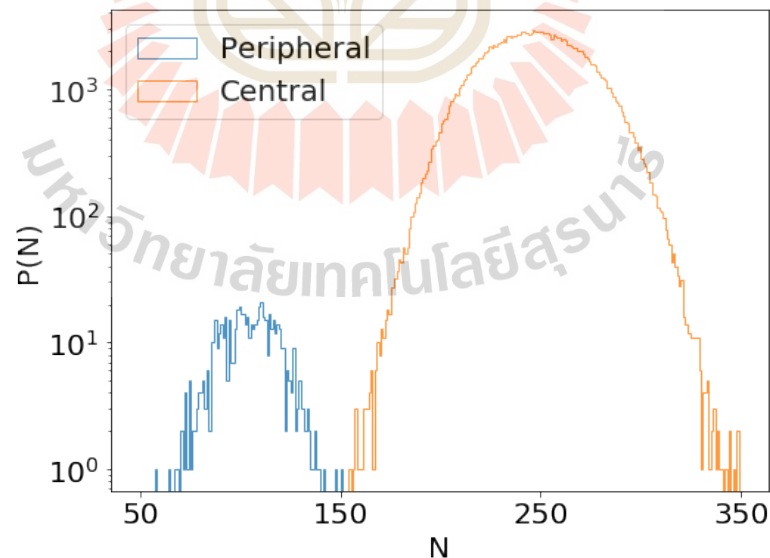
**Figure 4.2** This picture shows the example format of the UrQMD input file.

**Table 4.1** The description of sample input file of the figure 4.2.

label	arguments	description
#	(none)	comment line
pro	Ap Zp	mass and proton number of projectile ion
tar	At Zt	mass and proton number of target ion
nev	nevents	number of events to calculate
tim	totime outtime	time of calculation and output
elb	ebeam	incident kinetic beam energy (lab frame)
imp	bmax	impact parameter
eos	EoS	equation of state
cto	index value	set option in CTOption array
ctp	index value	set optional parameter in CTParam array
f15	(none)	suppress output to unit 15
xxx	(none)	last line of input-file

Since, in general, we do not know how an outlier looks like, i.e., what the characteristics of an outlier are, we have to generate “unusual” events by hand and present our method with these artificial outliers. In the following, we will deal with heavy-ion (Au+Au) collision events motivated by recent STAR data (Abdallah et al., 2021).

For the following study, we generated 184000 central ( $b = 3$  fm) events and 600 peripheral ( $b = 7$  fm) events from Au+Au collisions at  $\sqrt{s_{NN}} = 7.7$  GeV. The number of events was explicitly chosen to differentiate the number of outliers and backgrounds, resembled the status of the experimental data from STAR experiment. However, our results can be generalized to any number of events. The reason for this choice of systems is that such a combination of central (background) and peripheral (signal) events would lead to an anomalous proton number distribution, as observed by the STAR experiment. Even though, such a choice of events would be easily distinguishable by the conventional method of counting the total number of charged particles per event, as is shown in figure 4.3, it’s a good choice for investigating the ML/DL performance.

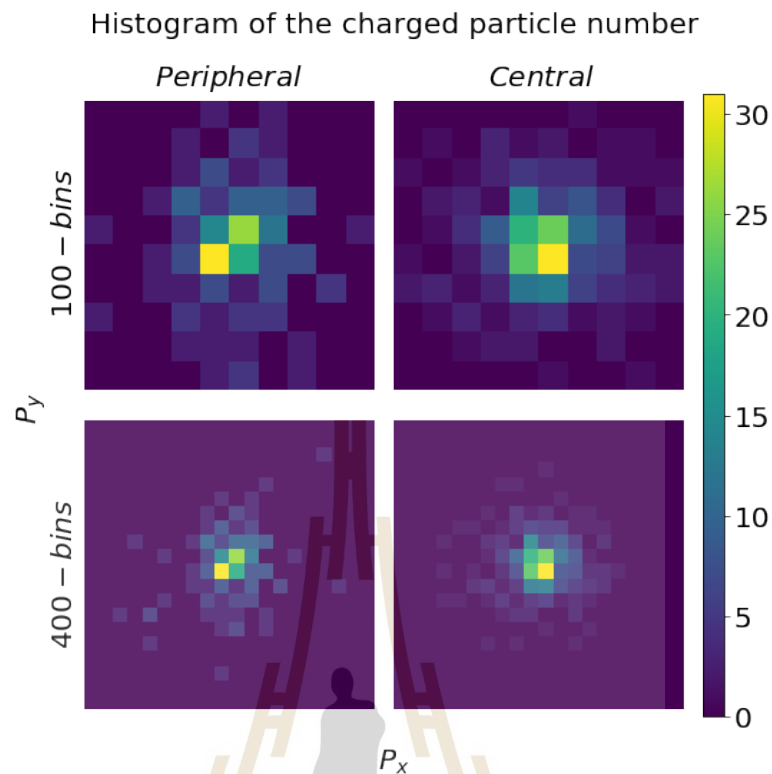


**Figure 4.3** Distributions of the multiplicity of charged particles (excluding protons)  $N$ , using  $|y| < 0.5$  and  $P_T \leq \sqrt{2}$  GeV for the peripheral and central events. The two event classes show no overlap and are therefore clearly distinguishable by their charged particle number distribution.

In order to use any outlier detection algorithms, we first have to create characteristic features (or feature-vectors) for all individual events. For this purpose, we define the *momentum feature* of every event. As a first step, we create “momentum features” of charged particles within a specific mid rapidity ( $|y| \leq 0.5$ ) window. These “momentum features” then characterize every single event and form the basis of the training and validation datasets. From the generated UrQMD data, and within the mid rapidity range, only charged particles are selected and binned in 2 Dimensional histograms of transverse X and Y momentum in two different momentum ranges to study the effect of the number of dimensions of the features:

- 10x10 momentum bins along the  $P_x$  and  $P_y$  direction, i.e. a total of 100 bins, for:  
 $-1 \text{ GeV} \leq P_{X/Y} \leq 1 \text{ GeV}$
- 20x20 momentum bins along the  $P_x$  and  $P_y$  direction, i.e. a total of 400 bins, for:  
 $-2 \text{ GeV} \leq P_{X/Y} \leq 2 \text{ GeV}$

The momentum features are shown in the figure 4.4 for both 100-bins and 400-bins cases. Each event class is presented here, including peripheral events and central events.

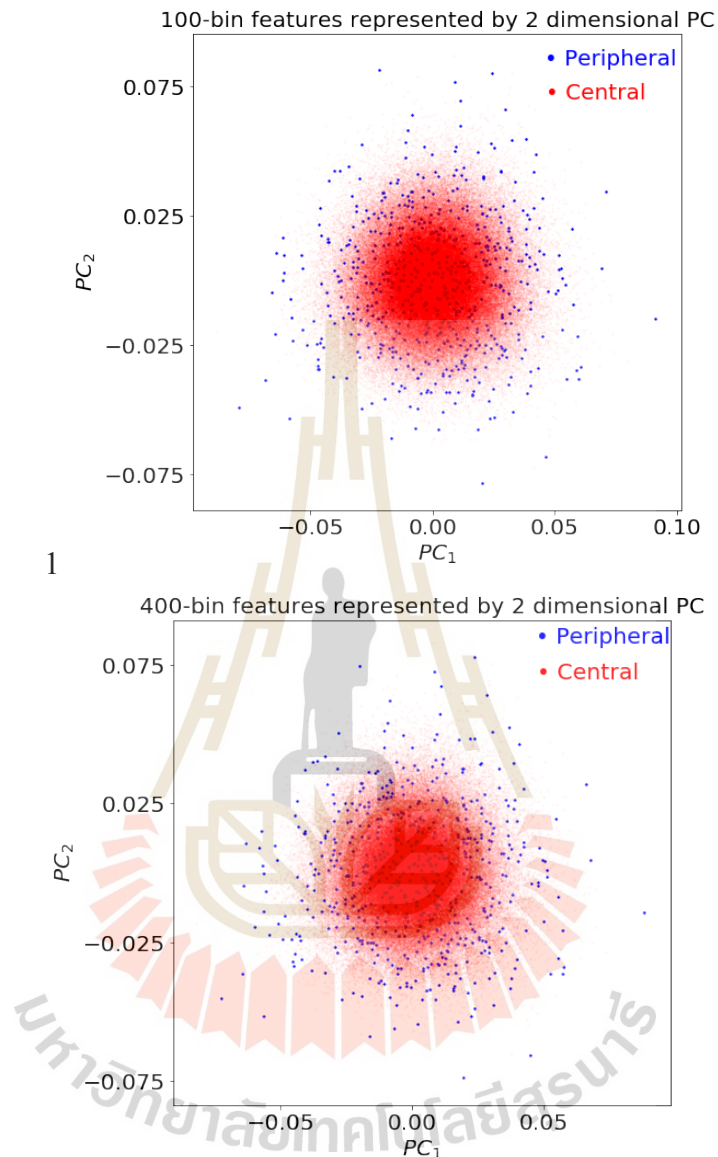


**Figure 4.4** Examples of charged particle multiplicity in the transverse momentum plane for a peripheral event (left column) and a central event (right column). The momentum ranges are  $P_T \leq \sqrt{2}$  GeV, divided into  $10 \times 10$  bins in the upper row, and  $P_T \leq \sqrt{2}$  GeV, divided into  $20 \times 20$  bins in the lower row.

## 4.2 Result and Performance

In this section, we present the result of implementing machine-learning algorithms on generated data. The algorithms that we have described in the previous chapter, PCA radius comparison, PCA reconstruction error, and AEN reconstruction error, are applied and compared. We test the learning performance of each algorithm by varying parameters and applying it to different feature dimensions. Moreover, we study the case of unusual events in section 4.2.6. Calculations and data visualizations are done using python libraries; see Appendix for details.

### 4.2.1 Radius comparison of PCA



**Figure 4.5** We show representation of the 100-bins (top) and 400-bins (bottom) feature after PCA transformation from 100/400 to 2 dimensions (or principal components). Peripheral and central events are mixed, but the former ones are distributed more sparsely and spread over a larger area.

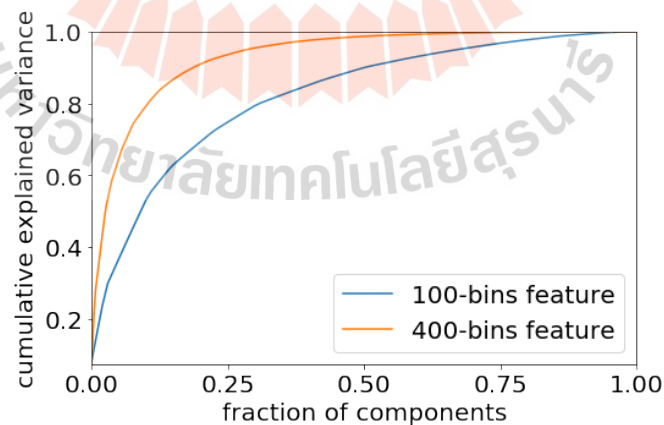
As a first step, the PCA is used on 100- and 400-bins features with two PCs. The resulting distribution of events in 2-dimensional principle coordinates  $PC_1$  and  $PC_2$  is shown in figure 4.5 for illustrative purposes. We can see that the distribution of both event classes is spherically symmetric here, with peripheral events distributed

more sparsely and over a larger area. There is, however, significant overlap, and it is clear that peripheral or signal events could only be uniquely identified if their radius, cf. equation (3.4), is sufficiently large.

In a next step, we compare the performance of 100- and 400-bins features using the cumulative explained variance  $\sigma_{cev}^2$  which is given by the cumulated variance of the  $n$  used PCs divided by the cumulated variance of the maximum number  $n_{max}$  of PCs which in our case is equal to 100 or 400, respectively. It reads

$$\sigma_{cev}^2 = \frac{\sigma_n^2}{\sigma_{n_{max}}^2} = \frac{\sum_{i=1}^n \sigma_{PC_i}^2}{\sum_{i=1}^{n_{max}} \sigma_{PC_i}^2}. \quad (4.1)$$

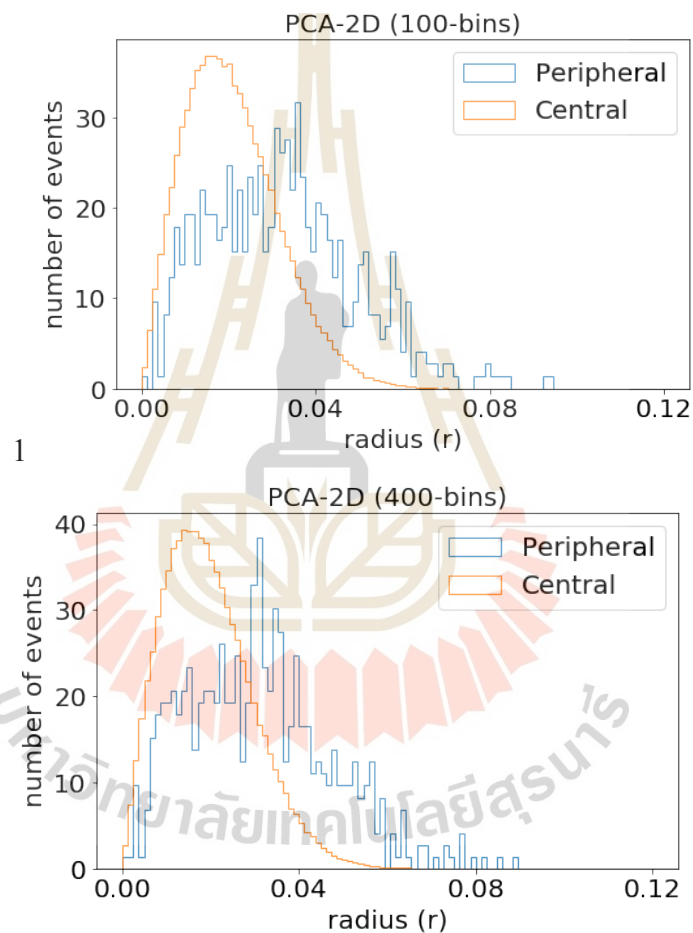
Hereby,  $\sigma_{PC_i}^2 = \sum_{j=1}^N [(x_{PC_i})_j - \bar{x}_{PC_i}]^2 / N$  is the variance of PC  $i$  over a total of  $N = 184600$  events. The cumulative explained variance is shown as function of the fraction of PCA reduced dimensions  $n/n_{max}$  in figure 4.6. We see that for small values of the fraction of components,  $\sigma_{cev}^2$  stays close to zero and approaches 1 only for fractions of components close to 1. With few components only a small amount of information is captured.



**Figure 4.6** Cumulative explained variance represents a function of the fraction of components, which is equal to the number of PC divided by the number of input bins, for 100-bins and 400-bins feature. The more components we employ, the better variance of data.

We now study two cases: First, a low number of PCs yielding a low cev, and

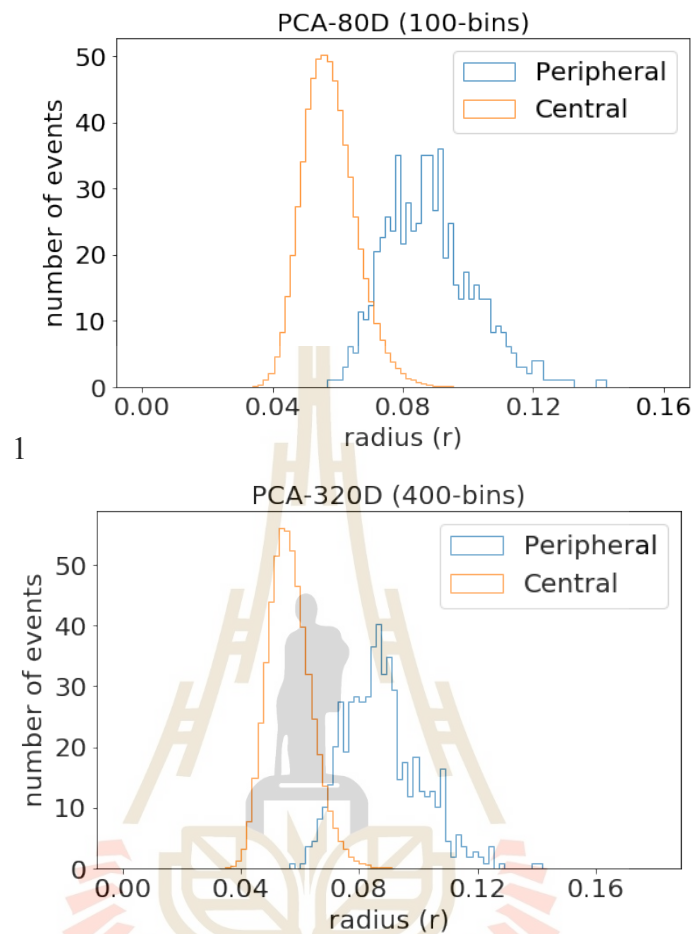
second, a large number of PCs yielding a higher cev. As seen in figure 4.6, a small number of principal components capture only a small amount of information. We are interested to see whether the datasets with a few principal components are able to capture enough information to differentiate our two event classes, signal and background compared to many principle components applied. We study this for both 100-bins feature and 400-bins feature.



**Figure 4.7** Histogram of the PCA radius for an input dimension of 100-bins (top) and 400-bins (bottom) features, reduced to 2 PCs. We see that the two event classes, signal and background, overlap. With a few numbers of PC applied, are not able to distinguish outliers from background

According to figure 4.7, we see that a low number of component yield, low variabilities from the dataset. The two event classes significantly overlapped in this case. For the second case, if we increase the number of PCs, it will improve the separation

between two classes of the histogram as shown in figure 4.8.

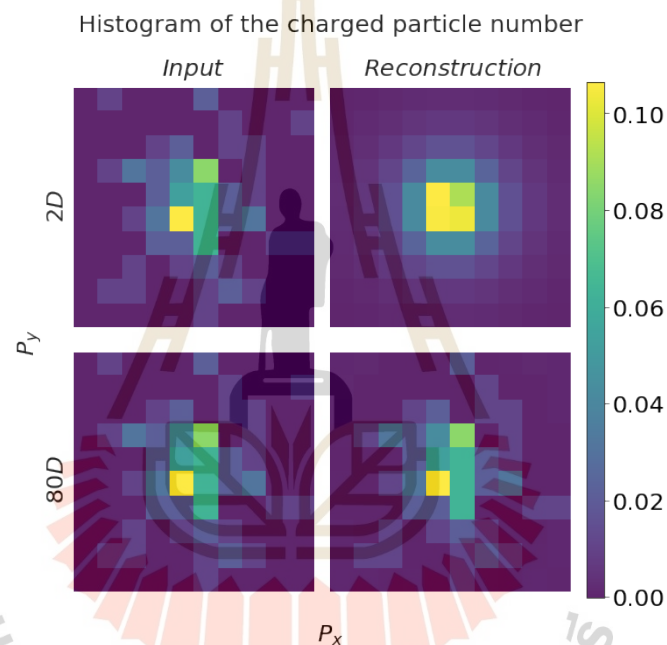


**Figure 4.8** Histogram of the PCA radius for an input dimension of 100-bins (top) and 400-bins (bottom) features, reduced to 80 and 320 PCs, respectively. We see that the two event classes, signal and background, overlap less. With a large number of PC applied, we are able to classify the outliers and background.

To judge whether one can use the radius as defined in equation (3.4) to separate the two classes, we calculate  $r(i)$  for each event and make histograms for two cases with different numbers of principal components, i.e., cases with significantly different  $c_{ev}$ . We compare 100-bins and 400-bins features. For a few principle components applied in figure 4.7, both distributions, the signal and background, show a large overlap. For a large number of principal components applied in the figure 4.8, the peaks are well separated in the case of 80D or 320D and the distributions share a much smaller overlap. It is obvious that, as more principal components are used, better separation between the



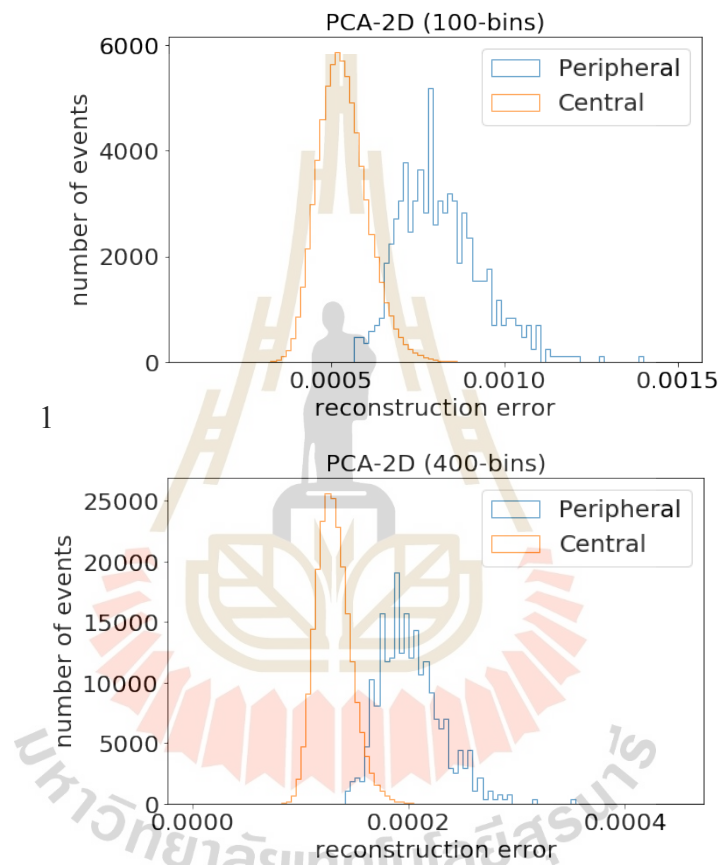
two classes can be achieved. To understand this behavior, we show in figure 4.9 a comparison of an arbitrary input event with its reconstruction by PCA using 2 and 80 PCs. the reconstruction with 80 PCs shows much more variability and closely resembles the input. This increased variability or variance is then captured in the radius, which is calculated as the sum of squares of all PCs, and therefore closely related to the variance. In other words, the PCs of the outlier events show a larger variance, and more PCs make the identification of signal events easier.



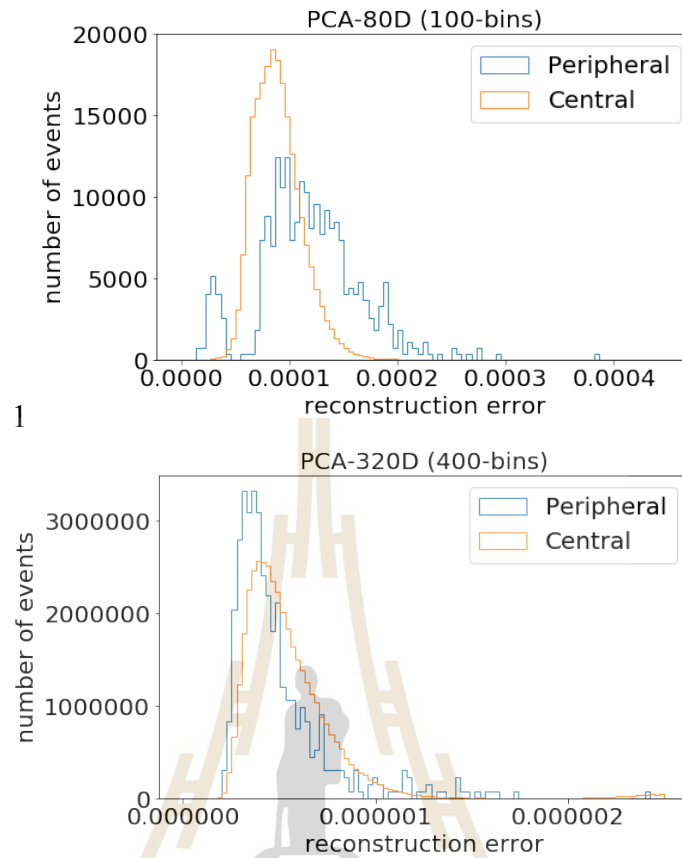
**Figure 4.9** 2 dimensional normalized histogram of the charged particle number for a single event. We compare the input feature with its reconstructed output after implementing PCA with 2 and 80 principal components. For 2 PC, the reconstruction feature looks similar to the event-averaged input feature. For 80 PC, the reconstruction feature is approaching the input feature.

## 4.2.2 Reconstruction error of PCA

A different method to quantify the appearance of outlier events is to calculate the reconstruction error according to equation (3.5). Instead of focusing on the variability of the dataset (as in the case of the radius comparison), this method focuses on the similarity of data points.



**Figure 4.10** Histogram of PCA reconstruction error for an input dimension of 100 (top) and 400 bins (bottom), reduced to 2 PC. With a low number of PC applied, the reconstruction error is quite large compared to a higher number of PC. The histogram shows that the distributions share a small overlap.



**Figure 4.11** Histogram of PCA reconstruction error for an input dimension of 100 (top) and 400 bins (bottom), reduced to 80 and 320 PCs respectively. With a high number of PC, the reconstruction error is quite small compared to a lower number of PC. The histogram shows that the distributions share a large overlap.

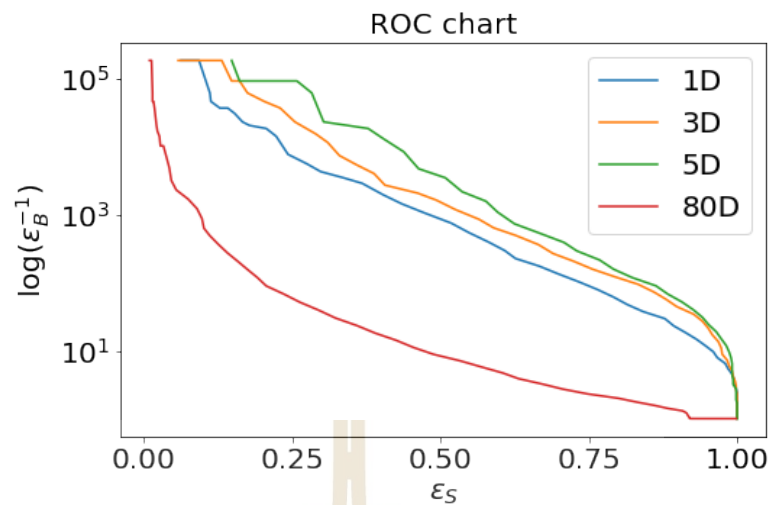
We calculate the reconstruction error for both cases,  $m = 2$  for a few numbers of PCs and  $m = 80$  and  $320$  for a large number of PCs. Figure 4.10 and 4.11 show the distributions of the reconstruction error for these cases. We first note that the reconstruction error is overall much smaller for 80(or 320) PCs than for 2 as one would expect. We also see that the distributions for 2 PCs are better separated than for 80 (or 320). In the former case, background events are, on average, better constructed than outlier events. Increasing the number of PCs, the RE approaches the same lower range of values for both background and outlier events, because the algorithm learns from both significant and insignificant components. In this case, signal and backgrounds are not separable using the RE.

### 4.2.3 PCA's radius comparison VS PCA's reconstruction error

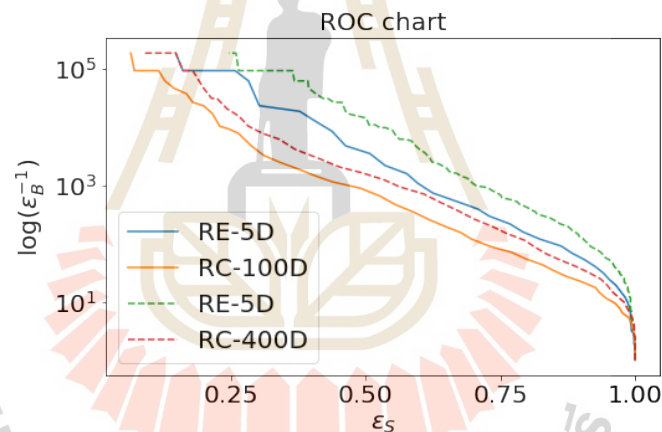
The reconstruction error method is based on the similarity of events with a reduced representation, and thus less PC work better to separate signal and background events. In contrast to that, the radius comparison method differentiates the two classes by the variance of components, thus choosing a high number of principal components leads to better discrimination. Therefore, the number of components needs to be chosen carefully and in accordance with the method to be used.

To make the comparison of the different methods more quantitative, we begin with ROC curves of 100-bins features for the reconstruction error methods with various numbers of PCs, 1, 3, 5, and 80, in figure 4.12. Amongst these, 5 PCs give the best performance, clearly better than the large number of 80 PCs but also significantly above the lower values of 3 and 1. Presumably, for 5 PCs, enough significant information is preserved, while events are not reconstructed with too high accuracy, and consequently, the outlier detection here works best. One should keep in mind that this specific choice of the number of PC may be different for a different dataset and may have to be adjusted according to the specific experimental setup used.

Now it's time to compare the radius comparison and reconstruction error method of PCA. ROC curves of 100- and 400-bins features are plotted in the figure 4.13. The curves of radius comparison method and reconstruction error methods are labeled as RC and RE, respectively. For the RC method, the number of PCs is equal to the number of input bins. While for the RE method, the number of PCs is equal to 5. We see that RE with a low dimension yields better performance for both input features. The PCA reconstruction error method works better because it tries to optimize principal component (PC) by choosing as least as PCs that are able to capture the significance of data while the radius comparison method needs viabilities as much as possible to differentiate two classes.



**Figure 4.12** ROC charts for event separation with PCA using the reconstruction error (RE), compared for different numbers of PC. We obtain the best result for 5 PC, clearly outperforming the high value of 80 PC.



**Figure 4.13** ROC charts for comparison of two PCA methods, reconstruction error (RE) with 5 PC and radius comparison (RC) with the number of PCs equal to the number of input bins. The dashed line represents 400-bins input. We see that RE with a low dimension yields better performance for both input features. The PCA reconstruction error method works better because it tries to optimize principal component (PC) by choosing as least as PC that is able to capture the significance of data while the radius comparison method needs viabilities as much as possible to differentiate 2 classes.

#### 4.2.4 Reconstruction error of AEN

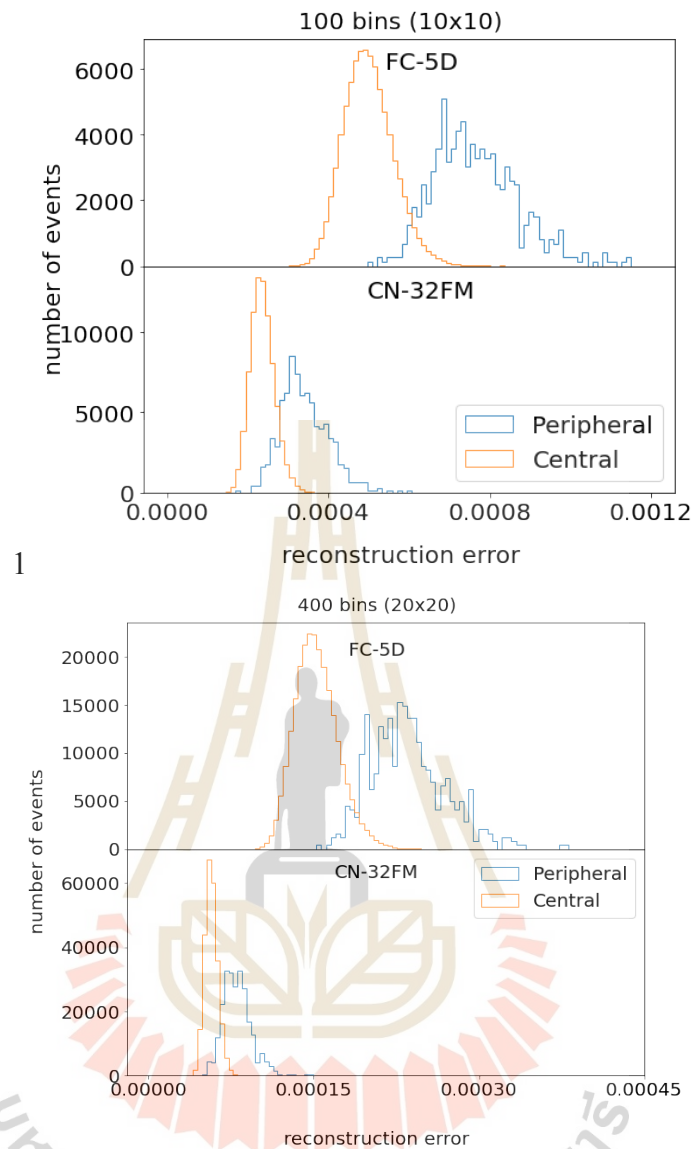
In this section we explore the possibilities of using an AEN for outlier detection in the same scenario as presented earlier.

Different kinds of neural network structures can be employed as an AEN. In

the following, two different kinds of network structures are employed. These include fully-connected neural networks, convolutional neural networks. The details of these network structures are explained in appendix.

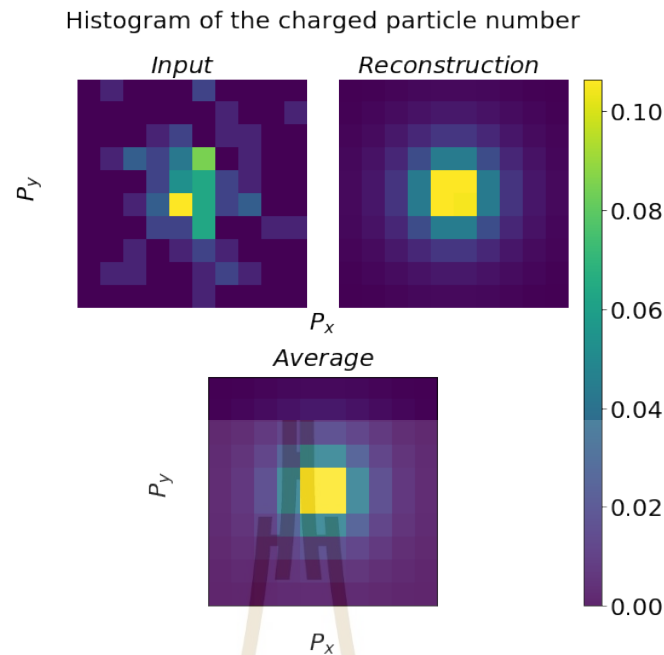
The same input features as used before, 100 and 400 bins, are encoded by the AEN to a low dimensional representation at the bottleneck and subsequently decoded back to some output which has the same dimension as the input feature. The reconstruction error is then calculated according to equation (3.5) for each event. Figure 4.14 shows the histograms of the reconstruction error for a 100-bins feature and a 400-bins feature, respectively. The network structures are fully-connected (FC) and convolutional (CN) neural networks. Fully-connected neural networks are encoded to 5D in the bottleneck layer (FC-5D) and convolutional neural network are encoded with (2,2,32) in the bottleneck layer (CN-32FM). From the figure, for both 100- and 400-bins feature, we use 5 hidden layers for both FC and 7 hidden layers for CN neural networks to be the representative of a big network structure. We see that the distributions of the errors for peripheral and central events are in the same range and have nearly the same shape for both types of structures. Consequently, the separation of the event classes is almost the same for the both types.

In addition to understand the process of encoding in AEN, we compare the input feature with its reconstruction applying fully-connected neural networks, encoded into 5D, for one randomly chosen event in figure 4.15. We show the averaged input feature, determined over the whole set of input spectra. It clearly resembles the reconstructed feature. The decoder network learns to reproduce the average of all input features as output. Thus, the information encoded in the bottleneck is not that relevant for this autoencoder. This indicates that, for a better reconstruction of the original input, more neurons in the bottleneck would be required. In the case of employing large network structure (FC with 5 hidden layers), we have also tried to vary the number of encoded



**Figure 4.14** Comparing two kinds of AEN structures applied to the 100-bins feature (top) and 400-bins feature (bottom). The network structures with 5 and 7 hidden layers are fully-connected (FC) and convolutional (CN) neural networks, respectively. The encoded representations in the bottleneck are 5D and (2,2,32), for FC and CN respectively. The results show a reasonable separation between event classes with roughly the same ranges of reconstruction error.

dimension and found that it does not contribute significantly to the performance of AEN. This simple comparison already yields a reasonable separation accuracy. On the other hand, if we use small network structure (FC with 1 hidden layer), the reconstruction now is different from the average input feature as shown in figure 4.16 and the result of separation accuracy is better.



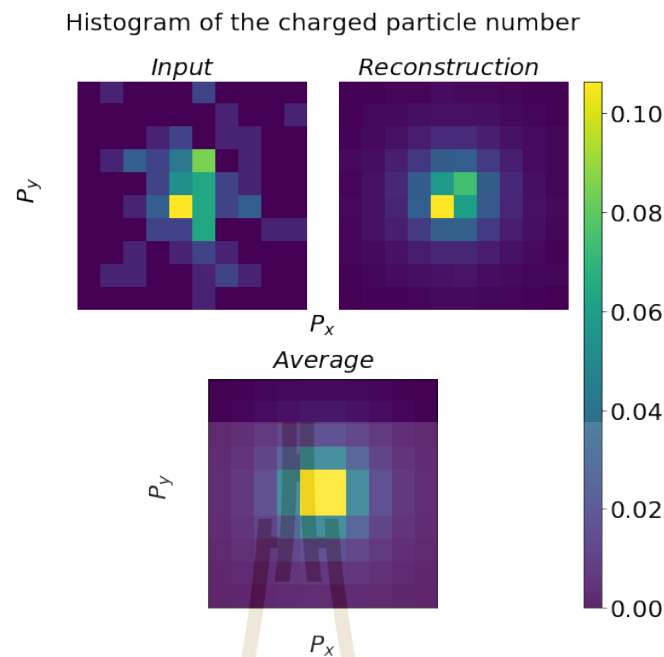
**Figure 4.15** Normalized momentum features. The first row shows the input and output of AEN for a 10x10 momentum feature. The bottom figure shows the average of input data over 184600 events. Note that the output feature of AEN-FC with 5 hidden layers applied and encoded into 5D is similar to the averaged input feature.

#### 4.2.5 Comparison of PCA and AEN

In this subsection, we will present the differences between PCA and AEN. As explained earlier, both algorithms have different learning processes. PCA is a linear approach while AEN is a non-linear approach. We also investigate the features of principle components (bottleneck) for PCA (AEN) which show interesting information. We compare the performances of these two algorithms graphically via a ROC curve and numerically via a table. We will discuss in details of how the learning algorithm works.

First, we transform (encode) the 100 features into 3 dimensions for PCA (autoencoder). The encoded features of both are plotted in the figure 4.17 with a 2-dimensional representation. PCA is explained by the first three top graphs. The events are distributed in each PC axis similar to what we got on the section 4.2.1, while the feature in AEN, explained by the last three bottom graphs, is clearly different. The data are encoded to 0 at one node ( $N_1$ ) in the bottleneck. For the values of  $N_2$  and  $N_3$ , it behaves similar to



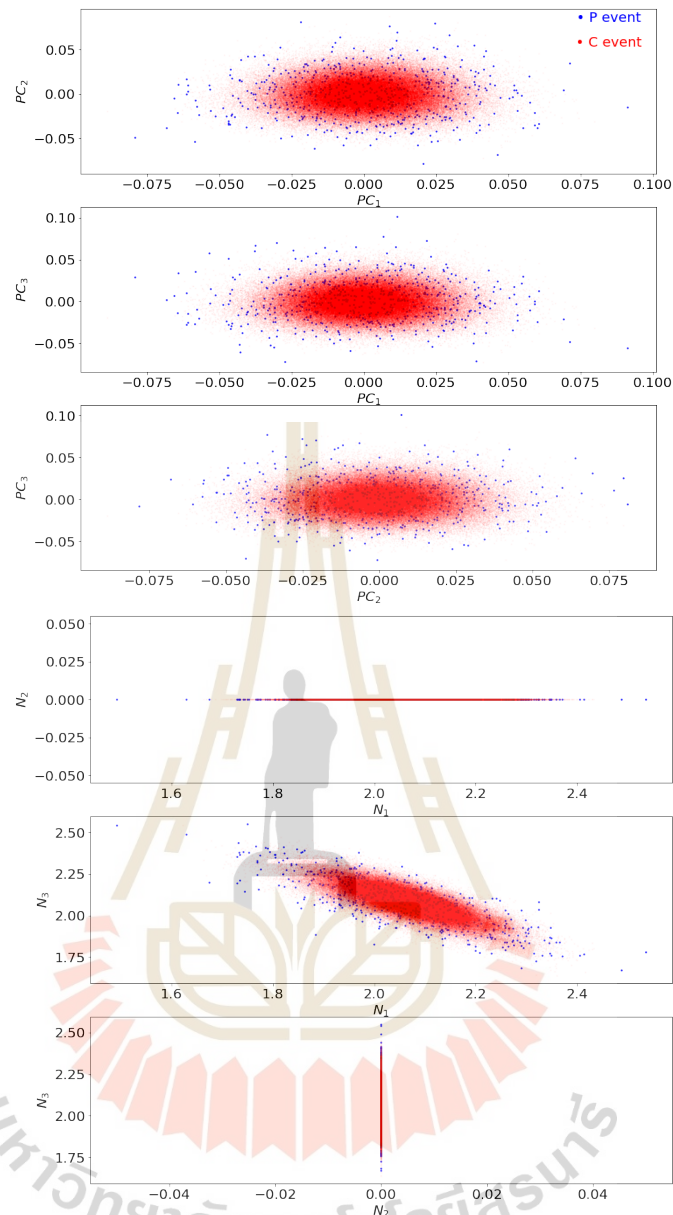


**Figure 4.16** Normalized momentum features. The first row shows the input and output of AEN for a 10x10 momentum feature. The bottom figure shows the average of input data over 184600 events. Note that the output feature of AEN-FC with 1 hidden layer applied and encoded into 5D is now different from the averaged input feature.

the case of PCA but with a narrower spreading.

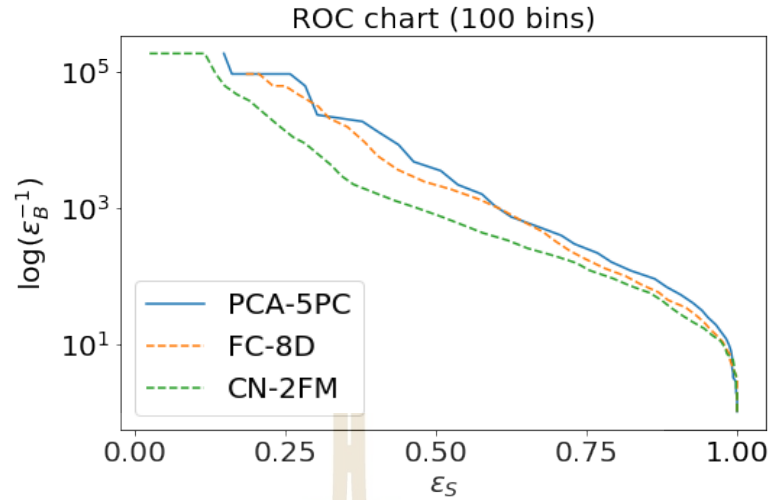
The ROC chart of 100-bins (10x10) feature, for two network structures and a PCA, is shown in figure 4.18. We see that the PCA curve mostly lies above the other two, indicating a better performance. For deeper insight, we show the performance numerically in table 4.2 by assuming if we cut the histogram to get 90% of signal how much, in percentage, backgrounds will smear in. For network structures, details are put in the appendix. Overall, we note that out of the tested structures, the best performance is given by PCA-5D, for both 100- and 400-bins input features. The performance will be discussed more in the end of this section.

We compare the output of AEN-FC, shown in Figures 4.15 and 4.16, with PCA in the previous section. We find that the output of 2D-encoded AEN is similar to the averaged input features. On the other hands, the output of 2D-encoded PCA deviates significantly from the averaged input features. the reconstruction with two PCs essen-



**Figure 4.17** We transform (encoded) the dimensions of input from 100 bins into 3 and plot pairs of encoded dimensions,  $(PC_i, PC_j)$  where  $i \neq j = 1, 2, 3$ , for PCA (top) and  $(N_i, N_j)$  where  $i \neq j = 1, 2, 3$ , for AEN (bottom), on 2-dimensional coordinates. For PCA, the events are distributed all over the 2-dimensional configuration (principle) space for all 3 pairs of components. On the other hand, the first two subplot of AEN, the value of  $N_1$  vanishes ( $N_1 = 0$ ).

tially only recovers the average distribution. Interestingly, when features are encoded into low dimensional representation, AEN encodes data by considering the relation between the 100 inputs (each pixel). This implies that neural networks are connected and investigate features more globally. The more important features are enough to encode



**Figure 4.18** Performance of PCA with 2D compared to two kinds of AEN structures. The ROC chart shows that the performance of PCA is generally better than of an AEN.

**Table 4.2** This table summarizes the fraction of background (FP) events (in percent) that are falsely identified as outliers if we cut on the reconstruction error to select 90% of true outliers (TP). Different numbers of encoded dimensions 2PC, 3PC, and 5PC for PCA and 2,3, and 8 for the fully connected AEN for both 10x10 and 20x20 momentum bins are compared to a varying number of feature maps for the CNN-AEN. All results using the input of 20x20 momentum bins separate the two classes better than the input of 10x10 momentum bins. An optimal number of parameters is found for all models.

method	10x10		20x20	
PCA	2 PC	3.5%	2 PC	1.9%
	3 PC	2.2%	3 PC	1.1%
	5 PC	1.6%	5 PC	0.8%
AEN	FC-2D	3.3%	FC-2D	1.8%
	FC-3D	2.5%	FC-3D	1.2%
	FC-8D	2.0%	FC-8D	0.9%
AEN	CN-1FM	4.7%	CN-1FM	3.4%
	CN-2FM	3.6%	CN-2FM	8.1%
	CN-4FM	4.2%	CN-4FM	1.4%

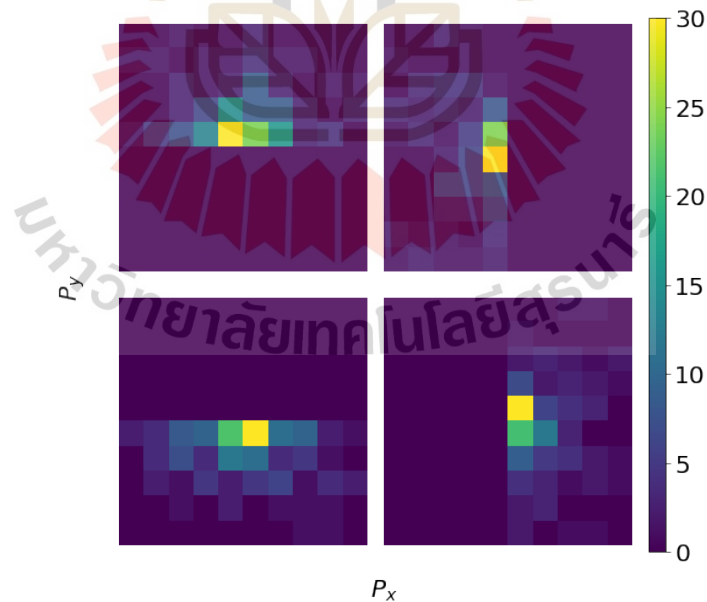
into the small number of hidden layers. This is the reason why when we increase the number of encoded dimensions in an AEN, in the case of large network structures, the performance is slightly changed. In contrast, encoded components of a PCA are selected from sub-spaces that contribute the most variance of the data. This make the weight in principle components more specific and local. However, the performance of AEN could be improved by fine-tuning hyper-parameter, the number of layers and the number of

nodes, corresponding to the size of input feature.

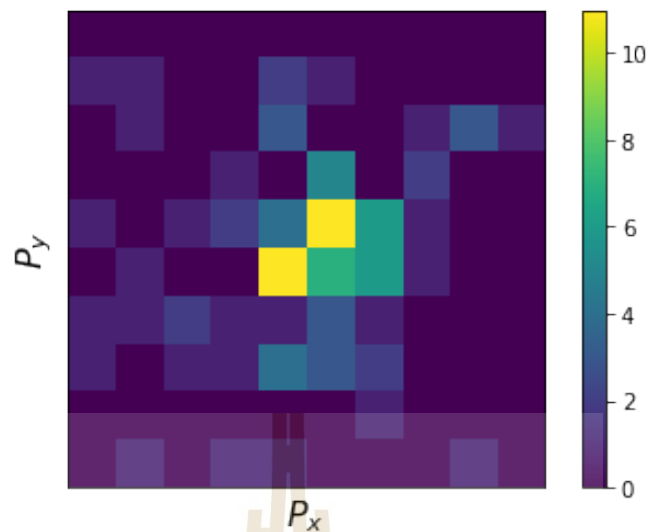
#### 4.2.6 Testing with other types of outlier

After using our neural networks for separating central and peripheral event classes, we are now ready to test them on a different type of outlier which are again constructed artificially as follows: We generate 600 central events and their corresponding momentum features as detailed in section 4.1. We then truncate one quarter of these features, i.e. we set the values in one randomly chosen quadrant of the  $P_X$ - $P_Y$  plane equal to zero as shown in figure 4.19. We vanish the number of charged particle in the high momentum range  $P_X$ - $P_Y$  and also in the random manner as shown in figure 4.21 and 4.20 respectively. These events are then normalized and used as input for our ML algorithms. We do this to both 100- and 400-bins feature.

Histogram of the charged particle number (truncated events)



**Figure 4.19** shows the example of how the outliers have been created. We randomly truncated 600 central events with 4 styles. The picture shown here is for the case of 100-bins event.

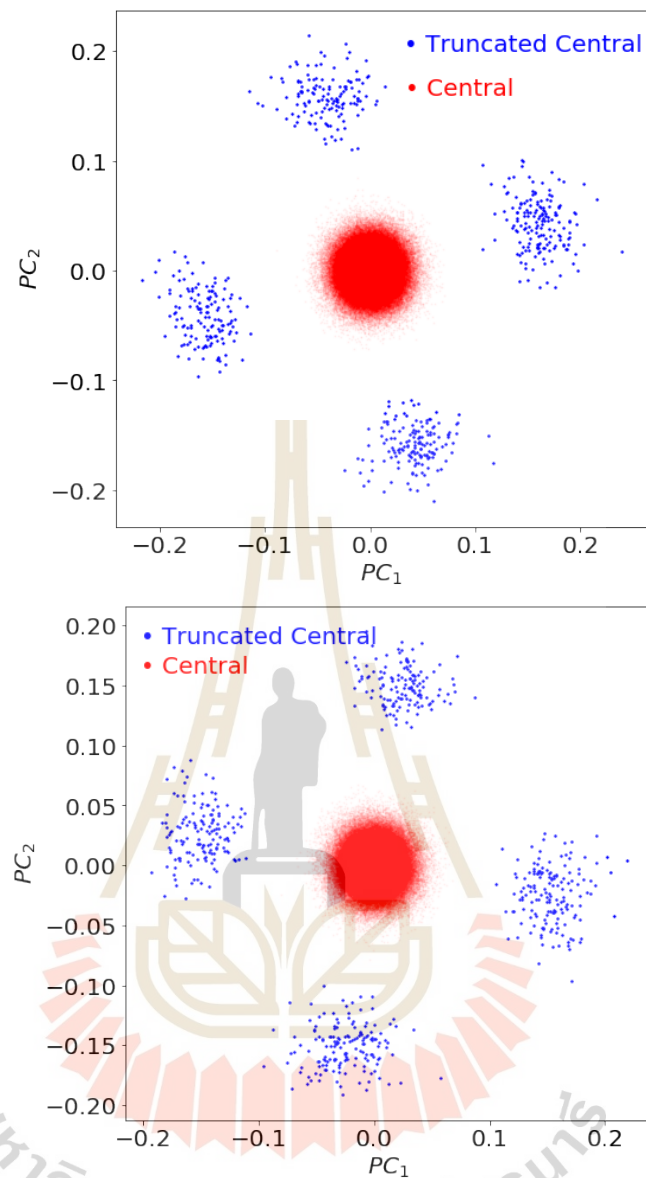


**Figure 4.20** shows the example of information losing. We randomly null all events. The picture shown here is for the case of 100-bins event.



**Figure 4.21** shows the example of information losing. We cut high range value of momentum out for all events. The picture shown here is for the case of 100-bins event.

We begin by checking the 2-dimensional representation in PCs after applying the PCA, see figure 4.22. While the central events naturally occupy the same space as shown earlier in figure 4.5, the truncated events or outliers are clearly separated from these into four distinct areas which are symmetrically positioned around the central events. Each of four these areas contains outlier events where one specific quadrant has been set to



**Figure 4.22** The 2-dimensional representation of the 100-(top) and 400-bins feature (bottom) is plotted after applying the PCA transformation from the 100 dimensions to 2 dimensions. This plot shows that the outliers are separated from backgrounds as 4 groups with the same radius' range.

zero.

PCA learn to select the best subspace, resulting in the maximum variance of the original data set, compared to others. The chosen subspace will be new principle coordinate that can group original data into 5 groups including 4 of outliers and 1 of background. Since backgrounds trivially contribute to variances, in contrast to outliers,

we see backgrounds in the middle and outliers more closely to the margin as shown in figures 4.22.

Now, imagine that if we do 50 styles of truncated events this way instead of 4. We will get 50 groups of outliers uniformly distribute around the backgrounds with the same range of radius values. These kinds of outliers are artificially generated following a certain pattern. And this pattern is then easily detected by the network. On the other hand, the previously considered task of identifying peripheral events is more challenging as these are not uniformly distributed. One could say that these types of outliers come from non-uniformly events or imply different physics.

We have quantified the smearing backgrounds from signal of our neural network structures (PCA-5D (100 bins), AEN-FC-5D (100 bins), AEN-CN-2FM (10x10), AEN-CN-4FM (20x20)) using the reconstruction criterion. As expected, in the case of 4-style truncated events, outlier and background events are separated perfectly. All outliers are detected without any background smearing in. If we cut the reconstruction error to get 90% of outliers, we will get 0% of backgrounds smeared in for all structures and both 100- and 400-bins features, see table 4.3. This demonstrates that the algorithms fully capable of detecting uniformly created outliers. We then test the separation of peripheral and central events, creating two additional obstacles that might be encountered in the context of experimental particle physics:

- Failed tracking of high-momentum particles: we truncate 20% of the  $P_X$  and  $P_Y$  range, leaving only the inner  $8 \times 8$  or  $16 \times 16$  bins of the 100- or 400-bins features unaltered,
- Detector malfunction or nonuniform noise: with a probability of 10%, we set each single momentum bin to zero.

All obtained spectra are again normalized to 1. We do this with both 100-bins features and 400-bins features. So the effect of granularity is taken into account.

**Table 4.3** Comparison of the test results for the three different models. Here the fraction of falsely as outlier identified background is shown for the three outlier options that are used in the testing. Both PCA and the AEN-FC perform equally well.

Input	PCA		AEN-FC		AEN-CN	
dim.	100	400	100	400	10x10	20x20
cut	5PC	5PC	5D	5D	2FM	4FM
half	0%	0%	0%	0%	0%	0%
rand	5PC	5PC	5D	5D	2FM	4FM
cut	2.5%	1.4%	2.5%	1.4%	3.3%	2.8%
high	5PC	5PC	5D	5D	2FM	4FM
cut	4.3%	1.0%	5.0%	1.1%	12.2%	1.5%





# CHAPTER V

## SUMMARY AND CONCLUSIONS

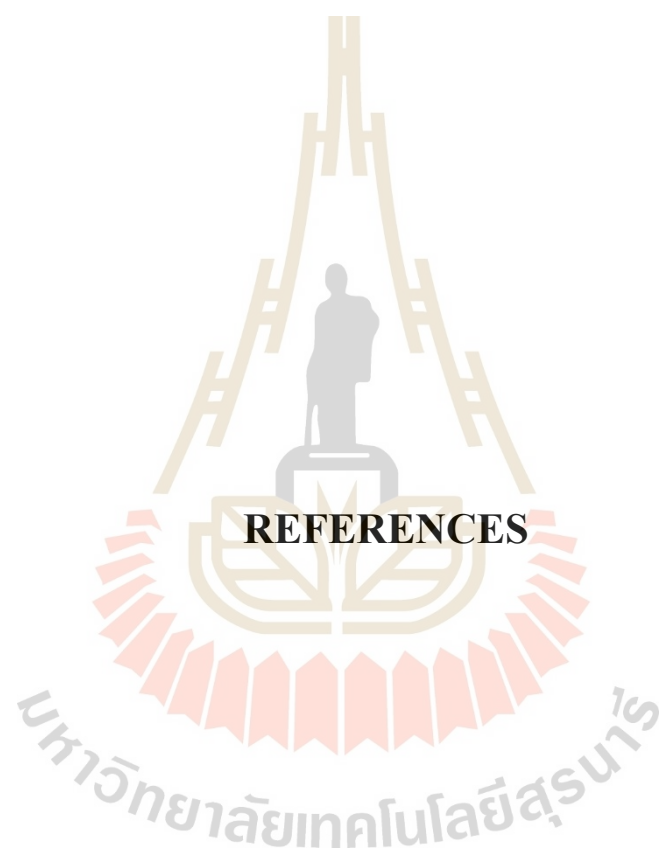
We have presented and explored methods of outlier detection using unsupervised learning which can prove useful for data analysis in high energy nuclear collision experiments. For this purpose, we have compared several unsupervised machine learning models such as the Principal Component Analysis (PCA) and Autoencoder networks (AEN). In a specific example we use the unsupervised learning to separate misidentified peripheral events from a background of central events. This example was motivated by the yet unexplained finding of large factorial cumulants at the STAR experiment. In this specific example, the transverse momentum spectra served as input features for the ML algorithms.

It was found that the reconstruction error in PCA or AEN can be a useful tool to identify outlier events. Furthermore, using the reconstruction error, it was found that a model which is too complex, i.e. gives a very small reconstruction error, gives a larger overlap of background and outlier events. Thus, a model which is less complex (has fewer parameters), but complex enough to capture the most essential features of the event is preferred. This is consistent with the result that a higher dimensional input feature also provides a better separation capability, as it is harder to reconstruct exactly but general features can be captured also by less complex models. This therefore provides an advantage for the direct application of outlier detection in an online analysis tool for heavy ion experiments. Since the model can be less complex it will be able to handle more events in a shorter time with fewer computational resources.

In a practical application the methods presented here will have to be adjusted to the actual output of the experiment. However, we believe that our work can provide a

solid guideline for the application of unsupervised outlier detection in nuclear collision experiments.





**REFERENCES**

## REFERENCES

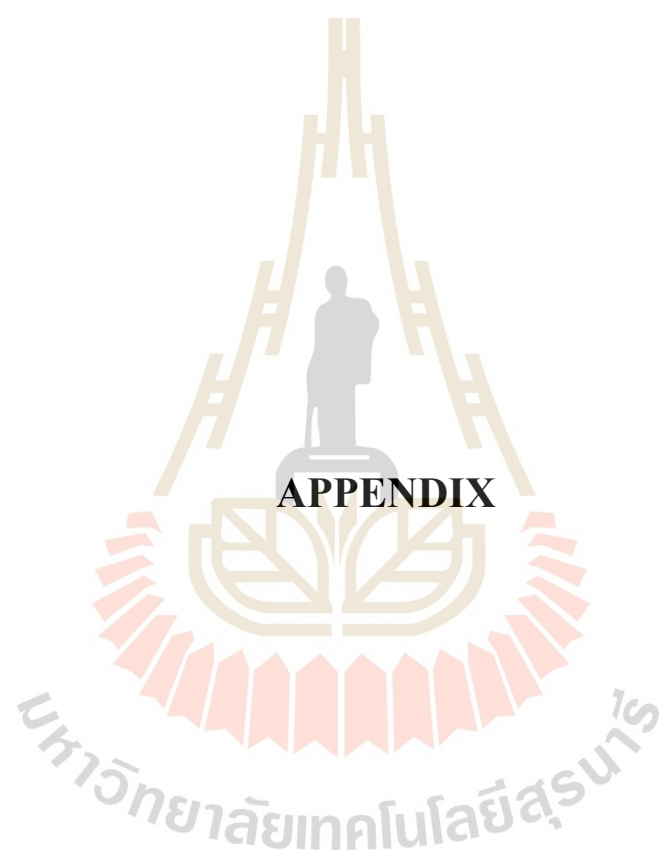
- Abbott, A. (2000). CERN claims first experimental creation of quark-gluon plasma. **Nature** 403(6770): 581–582.
- Abdallah, M., Adam, J., Adamczyk, L., Adams, J., Adkins, J., Agakishiev, G., Aggarwal, I., Aggarwal, M., Ahammed, Z., Alekseev, I., et al. (2021). Cumulants and Correlation Functions of Net-proton, Proton and Antiproton Multiplicity Distributions in Au+ Au Collisions at RHIC. **arXiv preprint arXiv:2101.12413**.
- Abgrall, N., Andreeva, O., Aduszkiewicz, A., Ali, Y., Anticic, T., Antoniou, N., Baatar, B., Bay, F., Blondel, A., Blumer, J., et al. (2014). NA61/SHINE facility at the CERN SPS: beams and detector system. **Journal of Instrumentation** 9(06): P06005.
- Adam, J., Adamczyk, L., Adams, J., Adkins, J., Agakishiev, G., Aggarwal, M., Ahammed, Z., Alekseev, I., Anderson, D., Aparin, A., et al. (2021). Nonmonotonic Energy Dependence of Net-Proton Number Fluctuations. **Physical review letters** 126(9): 092301.
- Adcox, K., Collaboration, P., et al. (2004). Formation of dense partonic matter in relativistic nucleus-nucleus collisions at RHIC: Experimental evaluation by the PHENIX collaboration. **arXiv preprint nucl-ex/0410003**.
- Aggarwal, M., Ahammed, Z., Alakhverdyants, A., Alekseev, I., Anderson, B., Arkhipkin, D., Averichev, G., Balewski, J., Barnby, L., Baumgart, S., et al. (2010). An experimental exploration of the QCD phase diagram: the search for the critical point and the onset of de-confinement. **arXiv preprint arXiv:1007.2613**.
- Asakawa, M., Ejiri, S., and Kitazawa, M. (2009). Third moments of conserved charges as probes of QCD phase structure. **Physical review letters** 103(26): 262301.

- Bass, S., Belkacem, M., Bleicher, M., Brandstetter, M., Bravina, L., Ernst, C., Gerland, L., Hofmann, M., Hofmann, S., Konopka, J., et al. (1999). The Ultra Relativistic Quantum Molecular Dynamics (UrQMD) model is a transport model for simulating heavy ion collisions in the energy range from SIS to RHIC. It runs on various UNIX-based computing platforms. Current implementations include IBM/AIX (xlf), GNU/Linux (g77, ifc), SGI/IRIX, DEC-UNIX and Sun/Solaris. UrQMD is designed as multipurpose tool for studying a wide variety of heavy ion related effects rang. **J. Phys** 25: 1859–1896.
- Bicudo, P. (2010). QCD confinement and chiral crossovers, two critical points? **PoS** FACESQCD: 015. arXiv: 1102.5531 [hep-lat].
- Bicudo, P., Cardoso, M., and Cardoso, N. (2011). QCD confinement and chiral crossovers, two critical points.
- Bleicher, M., Zabrodin, E., Spieles, C., Bass, S. A., Ernst, C., Soff, S., Bravina, L., Belkacem, M., Weber, H., Stöcker, H., et al. (1999). Relativistic hadron-hadron collisions in the ultra-relativistic quantum molecular dynamics model. **Journal of Physics G: Nuclear and Particle Physics** 25(9): 1859.
- Bzdak, A., Koch, V., Oliinychenko, D., and Steinheimer, J. (2018). Large proton cumulants from the superposition of ordinary multiplicity distributions. **Phys. Rev. C** 98(5): 054901. arXiv: 1804.04463 [nucl-th].
- Esmail, W., Stockmanns, T., and Ritman, J. (2019). Machine Learning for Track Finding at PANDA. **Connecting the Dots and Workshop on Intelligent Trackers**. arXiv: 1910.07191 [physics.ins-det].
- Glanz, J. (2000). *Particle physicists getting closer to the bang that started it all*. Tech. rep.

- Gronefeld, J. M. (2018). Transverse Momentum Distributions and Nuclear Modification Factors in Heavy-Ion Collisions with ALICE at the Large Hadron Collider. PhD thesis. Technische Universität.
- Haake, R. (2017). Machine and deep learning techniques in heavy-ion collisions with ALICE. **PoS EPS-HEP2017**. Ed. by P. Chechia et al. arXiv: 1709 . 08497 [physics.data-an].
- Hawkins, D. M. (1980). **Identification of outliers**. Vol. 11. Springer.
- Heinz, U. and Jacob, M. (2000). Evidence for a new state of matter: An assessment of the results from the CERN lead beam programme. **arXiv preprint nucl-th/0002042**.
- Herold, C. (2017). Introduction to Heavy-ion collisions, lecture notes.
- Kitazawa, M. and Asakawa, M. (2012). Relation between baryon number fluctuations and experimentally observed proton number fluctuations in relativistic heavy ion collisions. **Physical Review C** 86(2): 024904.
- Koch, V., Bzdak, A., and Skokov, V. (2013). Fluctuations and the QCD Phase Diagram. **AIP Conference Proceedings** 1520: 232–237.
- Luo, X. (2016). Exploring the QCD Phase Structure with Beam Energy Scan in Heavy-ion Collisions. **Nucl. Phys. A** 956. Ed. by Y. Akiba, S. Esumi, K. Fukushima, H. Hamagaki, T. Hatsuda, T. Hirano, and K. Shigaki: 75–82. arXiv: 1512 . 09215 [nucl-ex].
- Mehta, P., Bukov, M., Wang, C.-H., Day, A. G., Richardson, C., Fisher, C. K., and Schwab, D. J. (2019). A high-bias, low-variance introduction to machine learning for physicists. **Physics reports** 810: 1–124.
- Shi, S. (2010). Event anisotropy  $v_2$  at STAR. PhD thesis. Hua-Zhong Normal U.
- Stephanov, M. (2009). Non-Gaussian fluctuations near the QCD critical point. **Physical review letters** 102(3): 032301.

Van Hove, L. (1987). Theoretical prediction of a new state of matter, the "Quark-Gluon plasma" (also called "Quark Matter"). **Proceedings of the XVII international symposium on multiparticle dynamics.**





**APPENDIX**



## APPENDIX A

### ARTIFICIAL NEURAL NETWORK BACKBONE

**Table 1** Structure of the largest (with respect to the number of parameters) fully connected network used. Shown is the type of layer, the number of neurons in that layer (nodes), the activation function used for that layer and the output dimensionality of that layer. The total number of parameters of this model is 35,976.

FC (100 bins), Opt='adam', Loss='mse'				
layer	types	nodes	act fn	output
1	Input	100		100
2	Dense	88	relu	88
3	Dense	64	relu	64
4*	Dense	52	relu	52
5	Dense	64	relu	64
6	Dense	88	relu	88
7	Dense	100	sigmoid	100

**Table 2** Structure of the smaller (with respect to the number of parameters) fully connected networks used with the  $10 \times 10$  dimensional input features. Shown is the type of layer, the number of neurons in that layer (nodes, which is either 2, 3 or 5), the activation function used for that layer and the output dimensionality of that layer. The total number of parameters of this model are 502/703/1,105 for 2/3/5 encoded dimensions in the bottleneck respectively.

FC (100 bins), Opt='adam', Loss='mse'				
layer	types	nodes	act fn	output
1	Input	100		100
2*	Dense	2/3/5	sigmoid	2/3/5
3	Dense	100	linear	100

**Table 3** Structure of the smaller (with respect to the number of parameters) fully connected networks used with the  $20 \times 20$  dimensional input features. Shown is the type of layer, the number of neurons in that layer (nodes, which is either 2, 3 or 5), the activation function used for that layer and the output dimensionality of that layer. The total number of parameters of this model are 2,002/2,803/4,405 for 2/3/5 encoded dimensions in the bottleneck respectively.

FC (400 bins), Opt='adam', Loss='mse'				
layer	types	nodes	act fn	output
1	Input	400		400
2*	Dense	2/3/5	sigmoid	2/3/5
3	Dense	400	linear	400

**Table 4** Structure of the convolutional AEN used with the  $10 \times 10$  dimensional input features. Shown is the number of feature maps (FM) and the kernel size in that layer, the strides, the activation function used and the output dimensionality of that layer. The total number of parameters of this model are 216/771/2,901/175,521 for 1/2/4/32 FMs in the bottleneck respectively.

CN (10x10), Opt='adam', Loss='mse', 700 epochs					
layer	FM, kernel	strides	padding	act fn	output
1					(10,10,1)
2	4/8/16/128, (2,2)	(2,2)	same	relu	(5,5,4)
3	2/4/8/64, (3,3)	(2,2)	same	relu	(3,3,2)
4*	1/2/4/32, (3,3)	(2,2)	same	sigmoid	(2,2,1)
5	2/4/8/64, (2,2) <sup>T</sup>			relu	(3,3,2)
6	4/8/16/128, (3,3) <sup>T</sup>			relu	(5,5,4)
7	1, (2,2) <sup>T</sup>	(2,2)	same	relu	(10,10,1)

**Table 5** Structure of the convolutional AEN used with the  $20 \times 20$  dimensional input features. Shown is the number of feature maps (FM) and the kernel size in that layer, the strides, the activation function used and the output dimensionality of that layer. The total number of parameters of this model are 212/747/2,789 for 1/2/4 FMs in the bottleneck respectively.

CN (20x20), Opt='adam', Loss='mse', 700 epochs					
layer	FM, kernel	strides	padding	act fn	output
1					(20,20,1)
2	4/8/16, (2,2)	(2,2)	same	relu	(10,10,4)
3	2/4/8, (2,2)	(2,2)	same	relu	(5,5,2)
4	2/4/8, (3,3)	(2,2)	same	relu	(3,3,2)
5*	1/2/4, (3,3)	(2,2)	same	sigmoid	(2,2,1)
6	2/4/8, (2,2) <sup>T</sup>			relu	(3,3,2)
7	2/4/8, (3,3) <sup>T</sup>			relu	(5,5,2)
8	4/8/16, (2,2) <sup>T</sup>	(2,2)	same	relu	(10,10,4)
9	1,(2,2) <sup>T</sup>	(2,2)	same	relu	(20,20,1)

The cumulants of a given data sample could be written in terms of moments as follows:

$$C_1 = \langle N \rangle$$

$$C_2 = \langle (\delta N)^2 \rangle$$

$$C_3 = \langle (\delta N)^3 \rangle$$

$$C_4 = \langle (\delta N)^4 \rangle - 3 \langle (\delta N)^2 \rangle^2$$

Where N is the number of sample.

The factorial cumulant,  $\kappa_n$ , are related to the corresponding cumulant ( $C_n$ ) through the following relations:

$$\kappa_1 = C_1$$

$$\kappa_2 = -C_1 + C_2$$

$$\kappa_3 = 2C_1 - 3C_2 + C_3$$

$$\kappa_4 = -6C_1 + 11C_2 - 6C_3 + C_4$$

$$C_2 = \kappa_2 + \kappa_1$$

$$C_3 = \kappa_3 + 3\kappa_2 + \kappa_1$$

$$C_4 = \kappa_4 + 6\kappa_3 + 7\kappa_2 + \kappa_1$$

## CURRICULUM VITAE

**NAME:** PUNNATAT THAPRASOP    **GENDER:** Male

**NATIONALITY:** Thai

**DATE OF BIRTH:** FEBUARY 24, 1995

**E-MAIL:** a.Thaprasop.J@icloud.com

### **EDUCATIONS:**

- 2010 - 2012, High school at Kaengkhrowittaya school
- 2013 - 2016, Bachelor of Science, Physics department, Institute of science, Khon Kean University
- 2017 - 2019. Master of Science, School of Physics, Institute of Science, Suranaree University of Technology

