ABSTRACT

Enhancing Hadron Jet Reconstruction
in the CMS Level–1 Trigger Using Machine Learning

Syed Mahedi Hasan, M.S.

Advisor: Andrew Brinkerhoff, Ph.D.

Level–1 Trigger (L1T) algorithms used in the Compact Muon Solenoid (CMS) experiment for detecting different physics objects must be optimized to ensure that CMS continues to collect the most interesting proton–proton collision events for analysis. In this thesis, a new machine learning based approach using boosted decision trees (BDTs) is presented, which improves the jet detection performance in the L1T. In the first step, a BDT is trained using 12 features of L1T jets to generate an importance ranking of the features. The results indicate that a new algorithm for mitigating the effect of simultaneous collisions ('pileup') called the 'phi–ring' algorithm could be better at detecting L1T jets than the current 'chunky donut' algorithm. New BDTs are then trained separately using phi–ring and chunky donut energies as input, to confirm the previous finding. Outputs of the BDTs that use phi–ring energies as input are found to be more stable in energy scale under varying pilepup conditions, with resolution similar to the current jet detection algorithm. Hence, we propose to use the phi–ring algorithm to calibrate jet energies and improve jet detection in the CMS L1T in Run 3 (2022–2025).

Enhancing Hadron Jet Reconstruction in the CMS Level-1 Trigger Using Machine Learning

by

Syed Mahedi Hasan, B.Sc., M.S.

A Thesis

Approved by the Department of Physics

_____

Lorin Matthews, Ph.D., Chairperson

Submitted to the Graduate Faculty of
Baylor University in Partial Fulfillment of the
Requirements for the Degree
of

Master of Science

Approved by the Thesis Committee

_____

Andrew Brinkerhoff, Ph.D., Chairperson

_____

Jay Dittmann, Ph.D.

_____

Erich Baker, Ph.D. on behalf of Pablo Rivas Ph.D

Accepted by the Graduate School

May 2022

_____

J. Larry Lyon, Ph.D., Dean

Page bearing signatures is kept on file in the Graduate School.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

All praises go to the unbeknownst almighty, the most beneficent and most merciful.

First and foremost, I offer my profound gratitude to my advisor, Dr. Andrew Brinkerhoff, who put his faith and courage in me that I can deliver this work. It is unimaginable for me to think of the current shape of my thesis without his constant assistance, knowledge and helpful discussions. In my every mistake I always found him patient and kind towards me. One simply could not wish for a better mentor than him.

I am very much grateful to Dr. Jay Dittmann and Dr. Pablo Rivas for agreeing to be the members of my thesis committee. With their constant support and comforting disposition, they eased the process of my thesis defense. They also helped me to make my thesis more well–rounded by providing invaluable suggestions.

I would like to thank Dr. Siddhesh Sawant, a postdoc of the experimental HEP group at Baylor, for his contribution in this work. Not only was the dataset I used prepared by him, but from time and again he provided me with his indispensable counsel whenever I had any queries regarding the work. I also got a lot of help in programming from Brooks McMaster and Si Sutantawibul, fellow grad students of my group. They helped me set up my skeleton code which I later used throughout this work. Their contributions to my work will never be forgotten.

I am thankful to Dr. Kenichi Hatakeyama, Dr. Bryan Caraway and Ankush Reddy Kanuganti for the support they provided me at different times which helped me integrate into the experimental HEP group at Baylor.

I want to show my heartfelt gratitude to all the teachers and the staff of the department for their teaching, guidance, and inspiration during the entire period of my study in the department.

Special thanks to my roommates, Rafi and Shourov, both are from computer science background who significantly aided me whenever I faced difficulty regarding programming in python. Their help ultimately made me better at programming and also helped in understanding various key concepts of machine learning that proved to be very crucial in later part of my work.

Last but not least, I would like to express my appreciation for the encouragement that I have received from my family members and friends, without their inspiration and support this work would not have been completed.

DEDICATION

To my family

CHAPTER ONE

Introduction

As far as our written history goes, in every civilization our curious ancestors pondered some common questions about the reality of our existence irrespective of their time and geographic location. Hence it is safe to assume that even before our recorded history, these questions stirred our psyche and many of them are still unanswered even though our understanding has increased immensely (hopefully) since ancient times. "What may be the fundamental or elementary substance with which our physical reality came into existence?" and "If we continue dividing a material, will we encounter something indivisible at the end of this process?" These are two such questions that are still difficult for us to answer properly.

Thales of Miletus (626–545 BC) was probably the first person in the western world (at least in written history) who tried to answer the first question in an 'intellectual manner', which would later come to be known as 'philosophy'. He thought water to be this mysterious fundamental substance, and argued that the proportion of water in every substance is the reason behind the variety of matter around us. Later his pupils Anaximander (610–546 BC) and Anaximenes (586–526 BC) respectively proposed 'apeiron' (an indefinite unknowable substance) and 'air' to be the fundamental substance[1]. Almost at the same time in India, a philosopher named Kanad (also spelled Kanada) came up with an idea to synthesize those two questions into one, and put forward the idea of an indivisible substance that he called 'Kana' as the elementary substance[2]. In Greece, similar ideas later came independently from Democritus (460–370 BC), who proposed the idea of an unchangeable indivisible fundamental substance called 'atomos' (Greek for 'indivisible'). His work was based on the

works of Parmenides (who lived around 500 BC) and Zeno (495–420 BC), who hailed from the Greek town of Elea and propounded a static view of the universe in which change or motion is nothing but an illusion. Their ideas were opposed by Heraclitus of Ephesus (535–475 BC), who believed in a dynamic view of the universe where everything is changing in a constant manner, which means nothing is absolute and no fundamental substance exists. About a century later Plato (428–347 BC), made an alternative between these two contrasting views in his 'theory of ideas', where he proposed that, in the 'world of ideas' there exist fundamental substances that are absolute in nature, but in our physical world everything is constantly changing. His most famous pupil, Aristotle (384–322 BC), argued that every object in the physical world thus must be infinitely divisible. As arguably the most influential thinker of the classical world, he effectively cemented these arguments on divisibility for about 2000 years[1]. In 1803, a British chemist named John Dalton, while experimenting with gases, realized there are some unit substances that do not divide during a chemical reaction. He named them 'atoms', and put forward the modern atomic theory of elements, which created a paradigm shift and made atomism mainstream once again[3]. By the end of the 19th century the scientific community was pretty convinced about the indivisibility of the atoms since that idea could explain many experimental observations with great accuracy. But with the famous cathode ray tube experiment (1897) by J. J. Thomson and alpha particle scattering experiments (1908–1913) by Hans Geiger and Ernest Marsden (under the direction of Ernest Rutherford), it was discovered that atoms are not fundamental, but they are made up of 'nuclei' and 'electrons'[4]. Further discoveries indicated that the nucleus is made up of 'protons' and 'neutrons'. As a result, starting with the atom, we ended up with three elementary particles: protons (electrically positive), electrons (electrically negative) and neutrons (electrically neutral). In later years, it was discovered that

the protons and neutrons are divisible as well, being made up of smaller fundamental particles called 'quarks'. Additionally, heavier 'electron–like' particles were also discovered, and together with electrons they are termed 'leptons'[5]. By the late 1960s particle physicists came up with a list of almost all possible elementary particles that can exist from our current understanding of the physical universe, with some of them yet to be discovered experimentally. This list of the fundamental particles would come to be known as 'the Standard Model of particle physics' or simply 'the Standard Model' (SM) in later years.

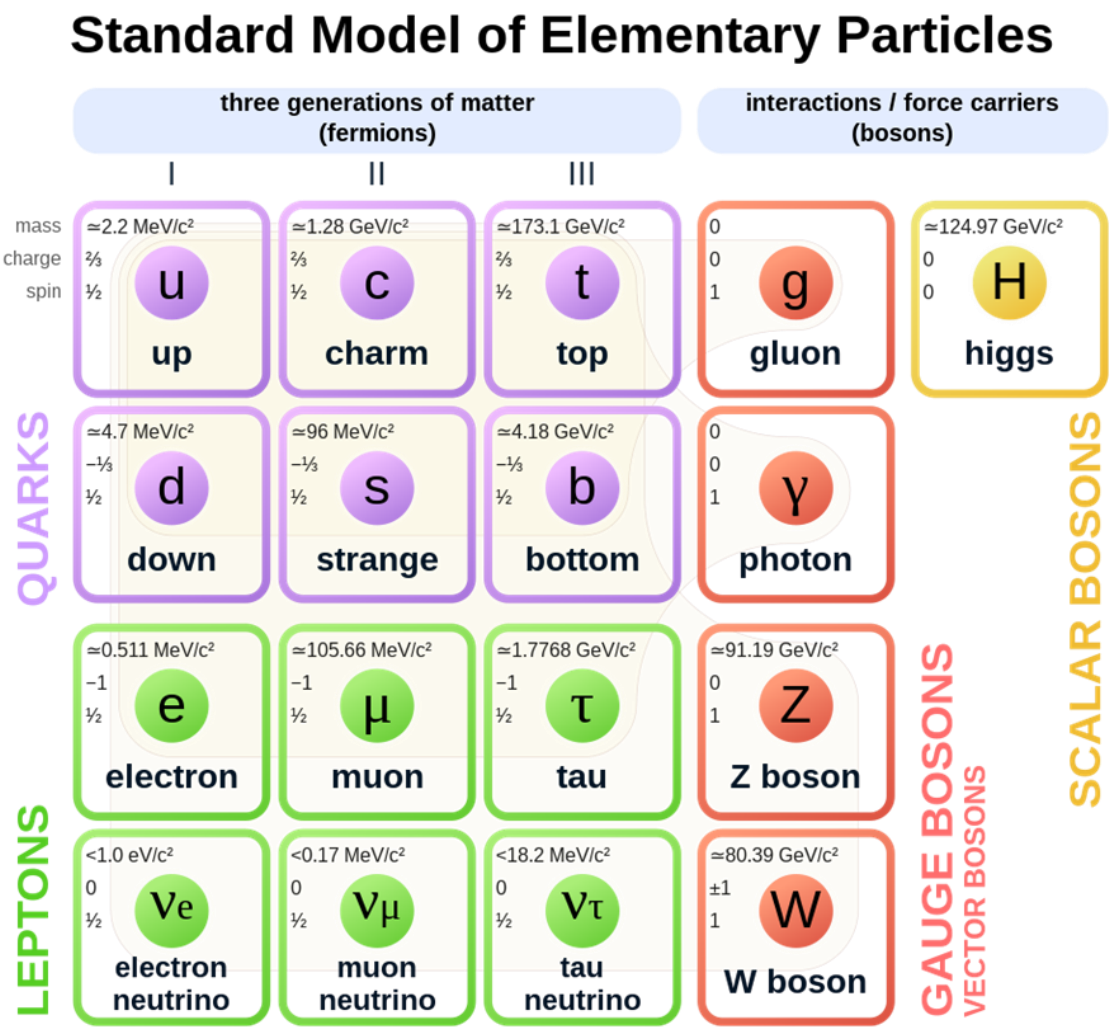

Figure 1.1: Elementary particles in the Standard Model of particle physics[7]

1.1 The Standard Model

According to our current understanding, the physical world is made up of two different groups of elementary particles[5]. Particles of the first group are called fermions and are usually associated with matter. They have half–integer spin and follow a statistical distribution called 'Fermi–Dirac statistics'. Particles of the other group have integer spin and follow a statistical distribution called 'Bose–Einstein statistics.' These particles are usually associated with force fields and are called bosons. Thus the Standard Model (Figure 1.1) is essentially a theory that incorporates all elementary matter particle fermions with all elementary field carrier bosons except gravity. Gravity is not a part of this framework for two reasons. First, the SM is obtained by extending the quantum field theory (the quantum mechanical fusion of classical field theory with the special theory of relativity) for subatomic particles, but mathematical synthesis of the general theory of relativity (the theory that describes macroscopic gravity) and quantum field theory (QFT) has not been achieved. Second, the elusive fundamental quantum excitation of gravity called the 'graviton' has not yet been discovered[6].

Fermions are responsible for making up the matter around us. They have two distinct classes ('quarks' and 'leptons') and three 'generations' in each class. The particles in the same generation in either class have no relation between them. The differences between generations in a single class are primarily attributed to the difference between their masses. There are a total of six quarks in the SM. Among them, the up, charm and top quarks are called up–type (uct) quarks, and down, strange and bottom quarks are called down–type (dsb) quarks due to their identical charges[5]. Since the names 'top' and 'bottom' are similar to 'up' and 'down', sometimes the top and bottom quarks are termed as 'truth' and 'beauty' quarks respectively. The quarks have fractional charges. Up–type quarks have charge +⅔,

and down–type quarks have charge –⅓ (as a fraction of the fundamental charge, i.e. the amount charge of an electron or a proton)[5,6]. The quark masses are as shown in Figure 1.1. Quarks can interact via all three fundamental forces described in the SM, i.e. the electromagnetic (EM) force, the weak force and the strong force. Leptons also have six members, where electron, muon and tau particles are categorized as electron–like (charged) leptons, and one neutrino is associated with each of them. In the SM, leptons only interact with neutrinos of the same type of 'flavor'. Charged leptons and quarks of the higher generations usually decay into other quarks or leptons with lower masses. The charged leptons are massive, but the neutrinos are almost massless and the reason behind neutrino's very small mass is yet to be discovered. All charged leptons have –1 charge, but neutrinos are neutral[6]. Unlike the quarks, leptons can interact only via the EM and the weak forces (Figure 1.2).

As the SM is a quantum field theory, these fundamental forces that are mediated through fields can be regarded as independent particles (discrete excitations) as well. In the SM we have four different bosons for three fundamental interactions[7]. For the EM force we have massless and chargeless photons (γ), and the interaction between photons and other SM particles depends on the electrical charge of the SM particles. Then we have $W^{\pm}$ and Z bosons which mediate the weak force. They have masses around 80 GeV/c^2 (W boson) and 91 GeV/c^2 (Z boson). Z bosons are neutral but $W^{\pm}$ bosons have a charge ±1. To carry the strong force, we have the gluons, which are also massless and neutral and only mediate quark interactions. Gluons are responsible for quarks binding together to form protons and neutrons, which eventually assemble to form nuclei. These four bosons are called vector bosons since they have a spin of 1. They are also called gauge or force carrier bosons. The graviton, if it is discovered, will be classified as a gauge boson as well, with a spin of 2.

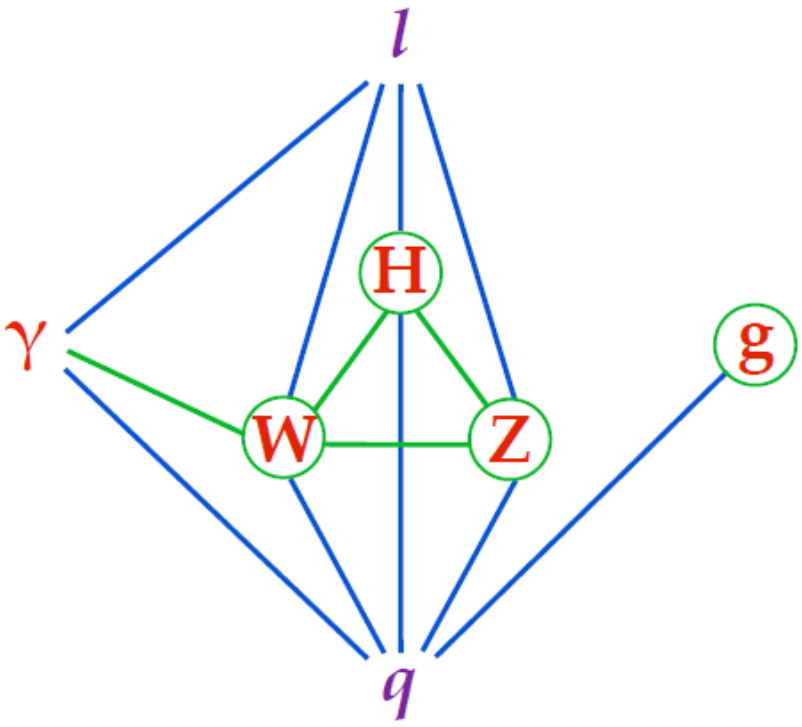

Figure 1.2: Standard Model couplings between leptons (l) and quarks (q) with fermion–boson couplings in blue and boson–boson interactions in green.[7]

In 1928, after Dirac applied the special theory of relativity to quantum mechanics, he predicted the existence of the 'positron', the positive counterpart (antiparticle) of the electron[5]. Soon it was realized that all particles have antiparticles. Antiparticles have the same mass as their SM counterparts, but all other properties are exactly opposite. When a particle and its antiparticle run into each other, they annihilate by producing gamma rays. For charged leptons they are denoted by the opposite charge sign, i.e. e^- for the electron and e^+ for the positron, while for all other antifermions they are denoted by a bar over the symbol for their SM counterparts, e.g. q for quarks and $\overline{q}$ for the antiquarks. Neutral bosons are treated as their own antiparticles, and thus are denoted by the same symbol.

Before their discovery, it was already proposed that a quark can group with two other quarks to form a baryon, or with an antiquark to form a meson. This occurs with the help of gluons to create these composite particles through a process called hadronization. Only the top quark cannot group with other quarks, since it is very heavy in comparison to other quarks and decays very rapidly before any binding with other quarks is possible[6,8]. It was also known that the nuclei–forming protons and neutrons are generated by a combination of three quarks, up–up–down (uud) and up–down–down (udd) respectively. After these discoveries, physicists realized that there must be another degree of freedom beyond charge, mass and spin that is missing from the model, since the fermions follow the 'Pauli exclusion principle', which states that two fermions in a bound system can never be in the same quantum state. Thus, the quarks inside the baryons like Δ^{++} (uuu) or Δ^{-} (ddd) can never be in the same quantum state. To resolve this issue, the idea of 'color charge' was introduced in the model. It is described as a property that determines how quarks interact with the gluons. There are three 'fundamental' colors, red (r), green (g) and blue (b), and their anticolors cyan ($\overline{\mathbf{r}}$), magenta ($\overline{\mathbf{g}}$) and yellow ($\overline{\mathbf{b}}$). They are not real colors like in our macroscopic world but an analogy, in that their combination in hadrons (two or more quarks and/or antiquarks including mesons and baryons) must yield a 'white' (colorless) state, since we don't observe them like the electric charges. With these six colors–anticolors, a total of 9 colorless combinations are possible as given in Table 1.1.

Table 1.1: Colorless combinations of color charge[8]

Octet	Singlet
rb,rg, bg,br, gr,gb,12(rr–bb),16(rr+bb–2gg)	13(rr+bb+gg)

Among these nine states, the singlet state is not allowed since it would give rise to free massless gluons with infinite range, which is not possible within the framework of the SM. Thus we have a total of 8 gluons that interact with the quarks to form different hadrons. These gluons can interact with each other through their color charges as well.

Weak isospin is another degree of freedom in the SM, which is a property of both quarks and leptons that comes from interacting with W and Z bosons[7]. It was needed in the SM to explain radioactive emission events where leptons are emitted from a decaying nucleus. While the $W^{\pm}$ bosons have a weak isospin of ±1, all left-handed neutrinos and up–type quarks (uct) have a weak isospin of +½ and the charged leptons and down–type quarks (dsb) have a weak isospin of –½. Similar to gluons, W and Z bosons can interact with each other through weak isospin. In the 1960s, it was discovered that Z bosons and photons can be used interchangeably in describing the same kind of decay. Hence, a new idea was proposed where the EM interactions and weak interactions can be regarded as the part of a force called 'electroweak force' which is mediated by photons, W and Z bosons. Since photons are massless and W and Z bosons are heavy, there is an imbalance in their interactions, e.g. decay via photons is more probable. This is known as the breaking of 'electroweak symmetry', which means there must be another degree of freedom associated with W and Z bosons that provides them their mass. To settle this, the existence of a new scalar (spin 0) boson called the 'Higgs boson' is predicted in the SM which would interact with the W and Z bosons, as well as with charged fermions, to give them their mass. The Higgs boson was discovered about fifty years after this prediction[9].

Except the electrons, neutrinos, gluons and up and down quarks, all other SM particles are very short–lived and can only be found in a measurable amount in cosmic rays and high–energy particle collisions. The primary triumph of the SM is its accuracy in

predicting the existence of charm, bottom and top quarks along with gluons, W and Z bosons long before their discovery in high energy particle colliders around the globe. The final missing piece in the current framework, the Higgs boson, was discovered to have a mass of 125 GeV/c^2 in 2012 at the 'Large Hadron Collider' (LHC) in the European Center for Nuclear Research (CERN) by the Compact Muon Solenoid (CMS) and A Toroidal LHC Apparatus (ATLAS) experiments[9].

1.2 The Large Hadron Collider

In terms of energy and size, the Large Hadron Collider is the biggest particle collider on earth. It can produce a 13 TeV collision energy with luminosity 2.1×10^{34} $cm^{-2}s^{-1}$. This collision energy is about 8 times the previous world record of collision energy made by the Tevatron collider in 1985 [10,11]. It lies about 100 meters underground in a tunnel of 27 km circumference near Geneva at the French and Swiss border. It started its first collisions in 2010, and since then, apart from the discovery of the Higgs boson, it has helped us to understand the SM in a better way and also to investigate physics beyond the standard model (BSM)[10].

The LHC houses a total of nine experiments (8 ongoing and 1 in construction). These experiments consists of collaborations with more than 10000 scientists from hundreds of institutions around the world. Among these experiments, CMS and ATLAS are dubbed as 'general purpose' detectors since they aim to investigate physical phenomena that involve all SM particles, while the other seven detectors have specific purposes and investigate particular phenomena in detail (Table 1.2)[13].

Data-taking periods of the LHC are known as 'Runs'. Run 1 was between the years of 2009 and 2013, during which the LHC operated at 7 and 8 TeV collision energy. After that, the LHC had its first long shutdown (LS1) period from 2013–2015. During this period

many collider and detector components were repaired and upgraded, including an increase in collision energy to 13 TeV. Run 2 was performed from 2015–2018[10]. The second long shutdown (LS2) period started in 2018 and is scheduled to end in April 2022. As before, LS2 was also utilized to repair and upgrade different components of the colliders and detectors. Run 3 will start in April 2022 with a collision energy of 13.6 TeV and is planned to end in 2025.

Table 1.2: LHC experiments[12]

Name	Abbreviation	Purpose
ATLAS	A Toroidal LHC Apparatus	General purpose
CMS	Compact Muon Solenoid	General purpose
ALICE	A Large Ion Collider Experiment	Investigate quark–gluon plasma
LHCb	Large Hadron Collider Beauty	Investigate phenomena related to beauty quarks
TOTEM	Total Elastic and Diffractive Crosssection Measurement	Investigate elastic and diffractive crosssections
LHCf	Large Hadron Collider Forward	Investigate neutral pions generated in the forward region
MoEDAL	Monopole and Exotics Detector at LHC	Investigate magnetic monopoles and other stable massive particles
FASER	Forward Search Experiment	Investigate high–energy neutrinos
SND (In Construction)	Scattering and Neutrino Detector	Search for collider neutrinos and feebly interacting particles

1.3 The CMS Detector

The Compact Muon Solenoid (CMS) detector is a 21 m long, 15 m wide and 15 m high gigantic detector (Figure 1.3) that weighs about 14,000 tonnes[10]. At its heart is a cylindrical superconducting solenoid that has an internal diameter of 6 m and produces a

magnetic field of 3.8 T. Within the solenoid there are silicon pixels and strip trackers. There are also two other calorimeters, an electromagnetic calorimeter (ECAL) made up of lead–tungsten crystals and a hadron calorimeter (HCAL), made up of brass and scintillator materials. These two calorimeters take data from three different regions within the detector. These regions are the barrel region (within the solenoid volume) and two endcap regions (both ends of the solenoid). HCAL also takes data from the forward region (within the detector but outside of the solenoid). Outside of the solenoid there are four gas–ionization muon collection chambers where muons are detected. These chambers are called drift tubes (DT), cathode strip chambers (CSC), resistive plate chambers (RPC) and gas electron multiplier (GEM)[14]. The absolute pseudorapidity ($|\eta|$) of DT is less than 1.2 and it covers the barrel region. The η coordinate is related to the polar angle (θ) of the emitted particle's direction with respect to the beam axis in such a way that for $\theta = 90^0$, $|\eta|$ is zero and for $\theta = 0^0$, $|\eta|$ goes to infinity[10]. Mathematically it is expressed as: $\eta = -\ln[\tan(\theta/2)]$[15]. Thus, θ corresponding to the DT ranges from 33.5^0 to 90^0. For CSC, $|\eta|$ ranges from 0.9 to 2.4 (θ from 10.4^0 to 44.3^0) which covers the endcap regions primarily. For the RPC, $|\eta|$ is less than 1.7 (θ from 20.7^0 to 90^0), and it covers both the barrel and endcap regions[10]. GEM is a new addition to the CMS detector with $|\eta|$ greater than 2.0 ($\theta < 15.4^0$). It covers primarily the forward region[10]. An r–z slice of a quadrant of the CMS detector is shown in Figure 1.4, and a cross–sectional image of the CMS detector is shown in Figure 1.5. This whole system runs with a collaboration of over 5000 scientists from more than 200 institutions in about 50 countries of the world.

Even though the LHC produces about 30 million proton–proton collision events per second with an average of around 35 simultaneous collisions per event during its typical data taking period, it would be impractical to record the information of all these events. First, we

do not have enough data storage space to do so, and second, most of these events are not 'interesting', which means that they are not able to provide any insight about novel phenomena. Thus, only about 1000 interesting events are selected per second from those 30 million for further investigation. This feat is achieved with the help of the 'Trigger System' of the CMS detector[17].

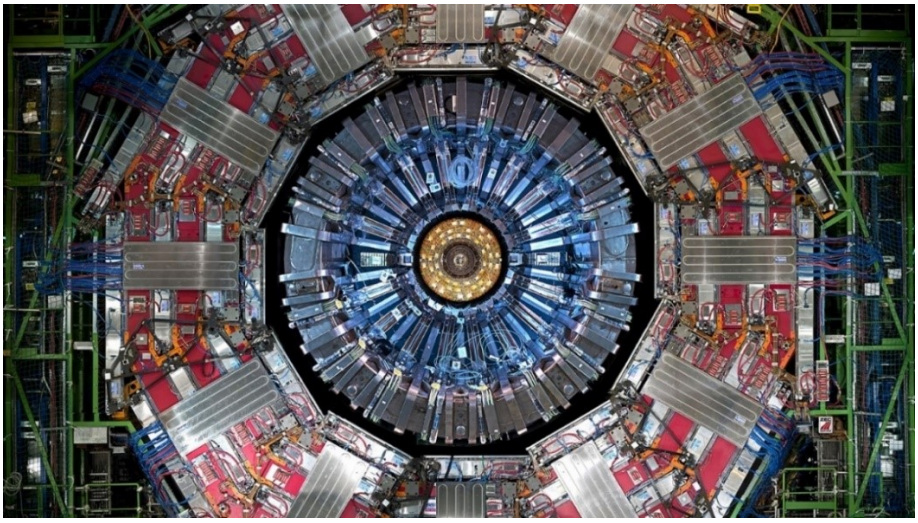

Figure 1.3: The CMS detector[16]

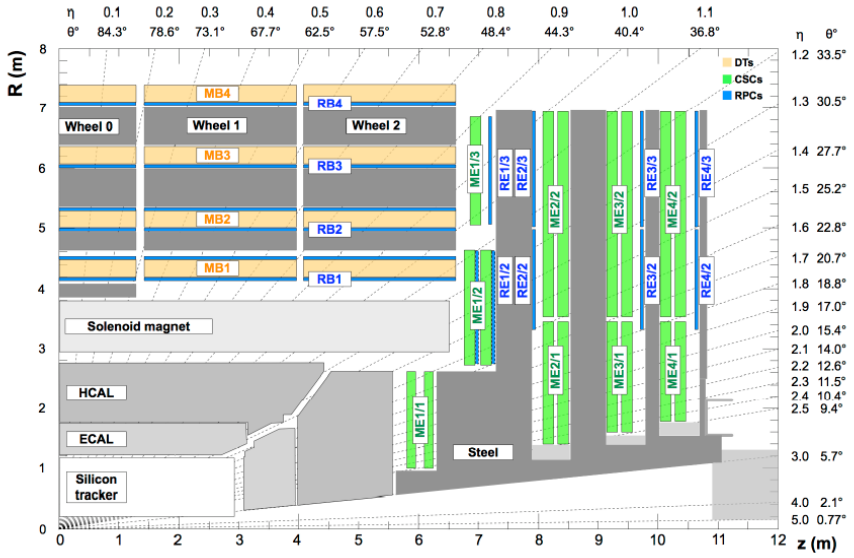

Figure 1.4: An r–z slice diagram of a quadrant of the CMS detector[18]

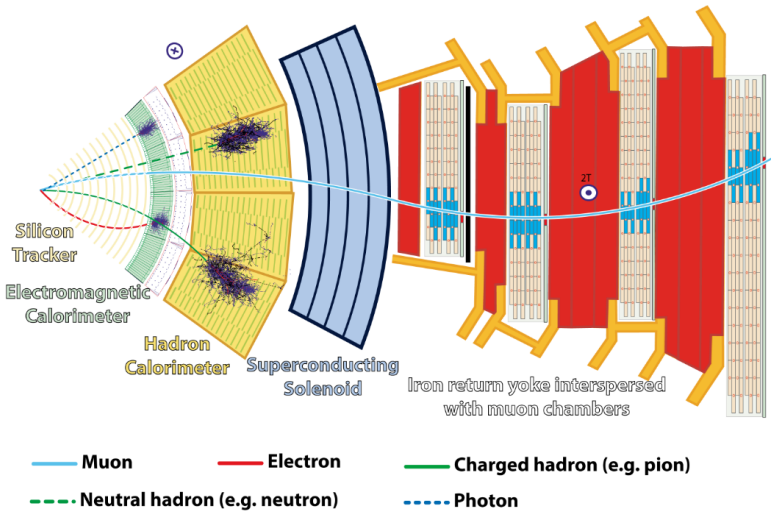

Figure 1.5: Cross–section of the CMS detector[18]

The trigger system contains two different stages for filtering out the uninteresting events from the collisions. The first stage is called the 'Level–1 Trigger' (L1T). It reconstructs and measures the energy of various entities that come out of the collisions, including muons, taus, electrons, photons and jets (the collection of hadrons). After this preliminary reconstruction L1T selects only 100 thousand events from the total number of collision events and sends them to the next stage known as the 'High Level Trigger' (HLT). To do so, the L1T uses a list of predefined algorithms known as 'seeds' that reconstruct the collision events based on some predefined criteria. The full collection of seeds is called the trigger 'menu'. If any event satisfies the conditions of at least one seed in the menu, higher–granularity data from all subdetectors (including the tracker) are sent to the HLT for further investigation. The HLT then performs a detailed reconstruction using high–performance processor farm and selects only about 1000 'most interesting' events to be stored for physics analysis. Hence, the L1T and HLT algorithms must be optimized properly on a regular basis to make sure that the most promising collision events are selected from the trillions of events[10].

With that in mind, this thesis describes a new technique to enhance the hadron jet reconstruction of the L1T using a machine learning tool called 'boosted decision trees' (BDTs). It also reports a noticeable improvement by providing a comparison between the current jet detection algorithm of the L1T and the new BDT approach.

CHAPTER TWO

Jet Reconstruction in the Level–1 Trigger

Since CMS is a 'general purpose' detector, it is designed to investigate a wide variety of physical phenomena including (but not limited to) searches for exotic particles, supersymmetry, dark matter candidates, understanding the dynamics of quark–gluon plasma generated through heavy ion collisions, the interactions of SM particles with electroweak and QCD forces, the decay of top and bottom quarks, etc[19]. After the discovery of the Higgs boson, measuring its various properties and decay modes has also become very important to the CMS experiment[20]. The experiment performs all these tasks with the help of high performance triggering systems (L1T and HLT), which run various algorithms (seeds) which together form the 'menu'. The L1T menu runs a total of 350 to 400 such seeds to select the most interesting events.

The L1T seeds can be divided into two primary categories based on their application[10]. The first category consists of seeds that apply the selection criteria only to the same type of objects. These objects include electrons, photons, muons, taus, hadronic jets, transverse energy and energy corresponding to the missing momentum ($E_{T(miss)}$)[10]. The selection criteria are applied to the object based on the threshold of their transverse momentum (p_T) or transverse energy (E_T) and the value of their absolute pseudorapidity $|\eta|$. It should be mentioned that the term p_T and E_T are used in this thesis interchangeably with the term 'energy'. About 75% of the total selection rate (100 kHz) of the L1T is covered by these types of seeds (Figure 2.1). The second category is termed 'cross' seeds, which include multiple objects of different types. Simple cross seeds involve combining multiple (usually

two) physics objects such as a jet and an electron. There are other cross seeds that work on calibrating the detectors, measuring the trigger efficiency and reconstructing more complex processes like Higgs boson production by vector boson fusion, jets decaying into leptons, etc. Trigger rate allocations for different kinds of seeds are shown in Figure 2.1. After receiving data from several components, the L1T uses these trigger algorithms to perform a reconstruction of all collision events, which are then selected or rejected by the menu[10].

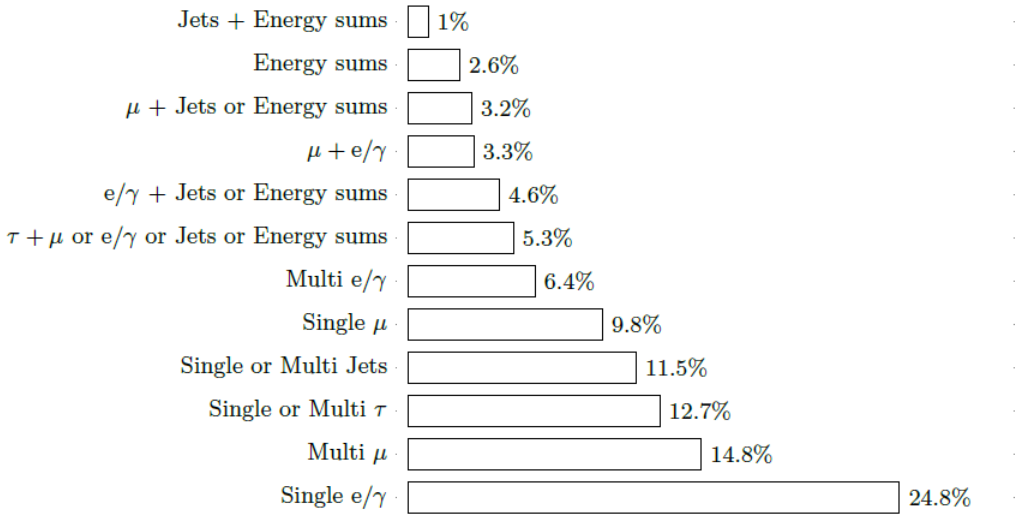

Figure 2.1: Level–1 Trigger rate allocation for simple and cross seeds[10]

2.1 Components of the Level–1 Trigger

The Level–1 Trigger works by a combination of two trigger systems called the muon trigger system and the calorimeter trigger system (Figure 2.2)[17]. The muon trigger system gets data from short tracks known as trigger primitives (TPs) from the CSC and DT along with hits from RPC. These data are then passed to the concentrator and preprocessor fanout (CPPF) and a twin multiplexer (TwinMUX). These modules pass only some selected data to three muon track finders, which reconstruct muon tracks in different regions. These track finders are called the barrel muon track finder (BMTF), overlap muon track finder (OMTF)

and endcap muon track finder (EMTF). The reconstructed muons from the track finders are then sent to the global muon trigger (μGMT, pronounced as micro–GMT) for the penultimate selection. After that, the selected data are passed down to the global trigger (μGT) for final selection[10].

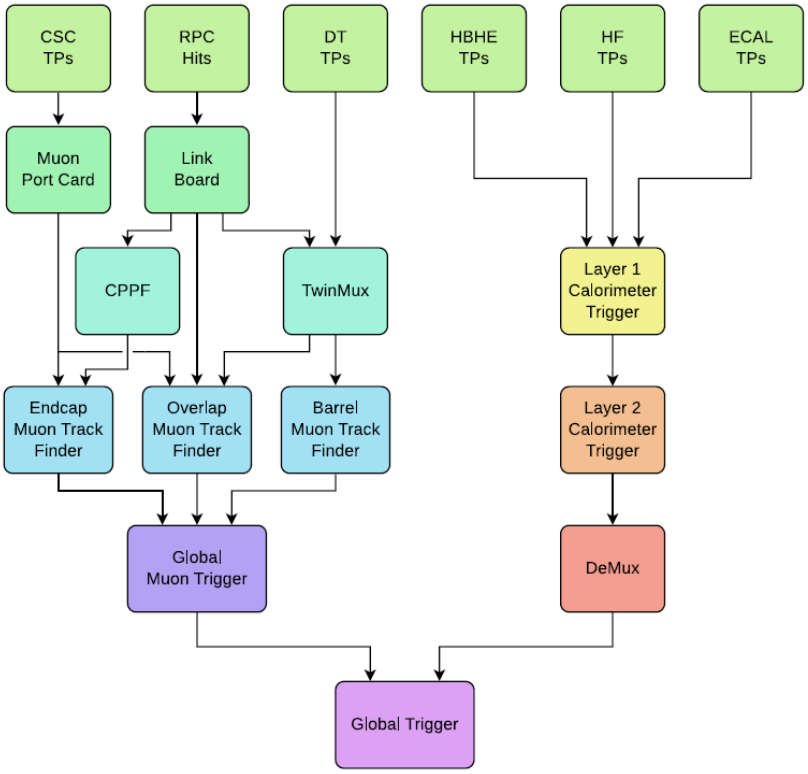

Figure 2.2: Components of the Level–1 Trigger[10]

The calorimeter trigger has two stages called 'Layers'. Layer–1 receives local energy deposit TPs from the HCAL and ECAL. These energies in calorimeters are deposited into ECAL crystals or HCAL modules of various shapes known as trigger towers (TTs). Layer–1 calibrates and serializes the data and passes them on to Layer–2. It also combines ECAL and HCAL TPs into a single TP, which includes bits for the ECAL/HCAL energy ratio. Layer–2 then uses these calibrated TPs to reconstruct the tracks of physics objects based on their

origin. TPs from ECAL are used to reconstruct electrons, and TPs from ECAL and HCAL are used to reconstruct jets, hadronic taus, etc. The selected objects are then sent to a demultiplexer (DeMUX) for reserialization and formatting before they are sent to the μGT for final selection. Each preprocessing node of the calorimeter trigger system obtains the information of an entire collision event with a granularity as small as $\Delta\eta \times \Delta\varphi = 0.087 \times 0.087$ (where η is unitless and φ is the azimuthal angle in radians). Together with the data from μGMT and DeMUX, the μGT makes the final selection of the data to be passed over to the HLT according to the menu. The HLT then performs a detailed reconstruction including information from the inner tracker on those events to further select the events to be stored[10,17].

Trigger seeds are used to select the most interesting collision events. The efficiency of these trigger algorithms depends on their ability to differentiate between multiple objects that are generated due to the collision events. Since the number of simultaneous collisions per LHC bunch crossing (pileup) is large, this process becomes increasingly difficult because of the background noise[10].

2.2 Jet Reconstruction Algorithms

Quarks cannot exist freely in nature since only the colorless states are allowed according to QCD confinement, so they interact with gluons and other quarks to form hadrons. The collections of such hadrons coming from collision events are known as jets[21]. The jet algorithm used in Run 2 is based on a square jet approach known as the 'chunky donut algorithm' (Figure 2.3)[10]. In this algorithm, a 9×9 square region is selected in TTs of ECAL and HCAL around a single jet 'seed' TT with transverse energy greater than 4 GeV. In the barrel region, this 9×9 area corresponds to a 0.783×0.783 square in $\eta \times \varphi$, approximating the 0.8 diameter circle in η vs. φ used for offline (known as particle flow or

PF) jet reconstruction, to be considered as a jet candidate, the energy of the central TT in the candidate must be equal or greater than the energy of every TT in the triangle below the diagonal of the 9×9 region, and greater than the energy of the TTs in the triangle above the diagonal of the 9×9 region. This serves two purposes. First it avoids double counting of the same jet seed, and second, it prevents vetoing the TTs with the same energies during the consideration of jet seeds. After identifying a jet candidate, the energy deposits of four neighboring 3×9 regions are also estimated. Among the four, the three regions with lowest energies are summed as the estimated pileup energy and subtracted from the jet candidate energy to obtain the final energy of the L1 jet. So, mathematically it can be written as:

E_T (PUS) = [Raw E_T – Σ three lowest E_T of 3×9 neighboring regions] × Layer–2 calib. (2.1)

Here, E_T (PUS) is the pileup-subtracted (PUS) transverse energy of the L1 Jet. Raw E_T is the raw transverse energy of the jet candidate in the 9×9 square region without Layer–2 calibration[22].

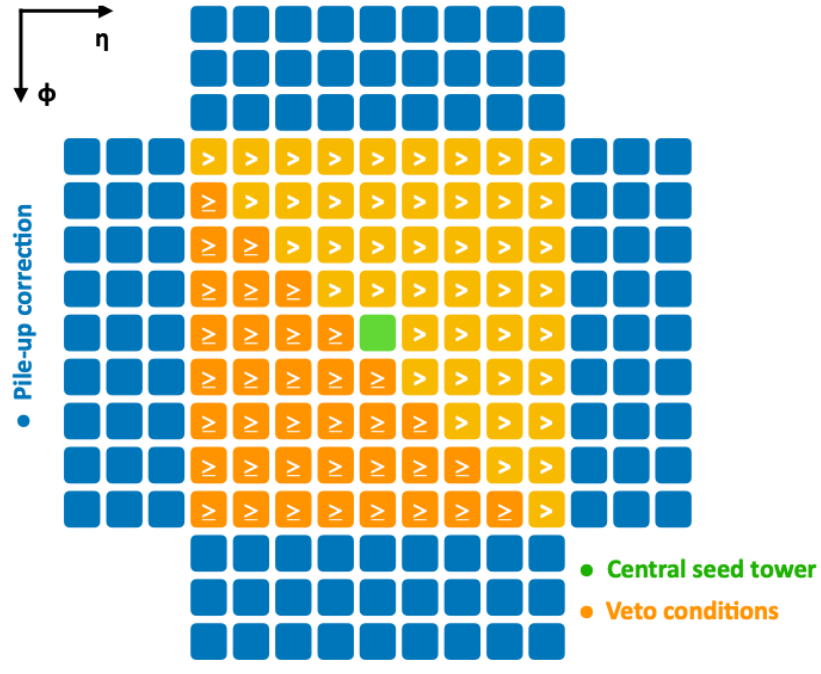

Figure 2.3: Chunky donut algorithm with a 9×9 square region in the center, and four neighboring 3×9 regions, with veto conditions[22]

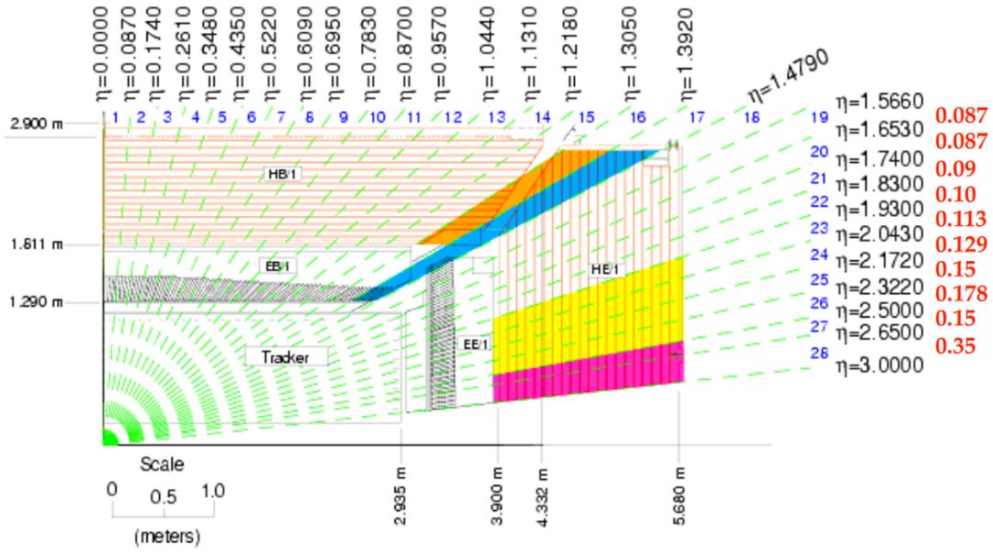

Figure 2.4: Trigger tower rings from 1 to 28 that cover $|\eta| < 3.0$[18]

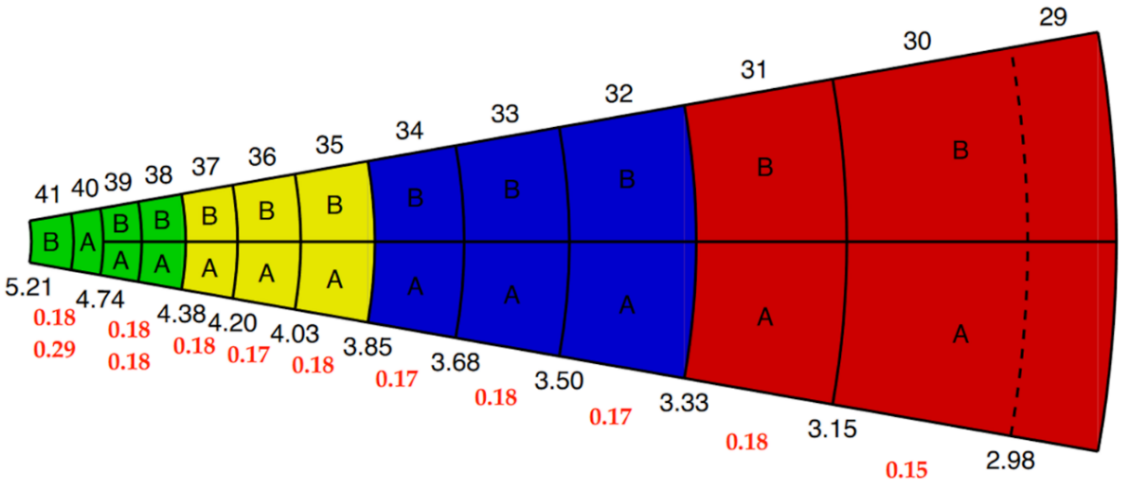

Figure 2.5: Trigger tower rings from 29 to 41 that cover $3.0 < |\eta| < 5.2$[18]

There are a total of 40 TT rings in HCAL: 16 in the barrel region, 12 in the endcap region and 12 in the forward region. Each ring covering $|\eta| < 2.65$ has 72 TTs, while rings with $|\eta|>2.65$ have 36. They are shown in Figure 2.5 and Figure 2.6. TTs in the barrel and endcap regions span an angle of $\varphi=5^0$ (except in ring 28) and cover the η values from –3.0 to 3.0. TTs in the forward region cover the $|\eta|$ values from 3.0 to 5.2 and span an angle of

φ=10^0. The TT ring numbers are often labeled with iη or iEta, i.e. iEta=1–16 is in the barrel region, iEta=17–28 is in the endcap region and iEta=30–41 is in the forward region. Even though HCAL doesn't have a TT ring at iEta=29, that number is not omitted from the iEta numbers. The width of the trigger towers in η as a multiple of Δη=0.087, are shown in Figure 2.7[18]. Here we see that towers in the forward region, which receive the most energy from pileup, are a factor of 2 to 4 larger than in the barrel region.

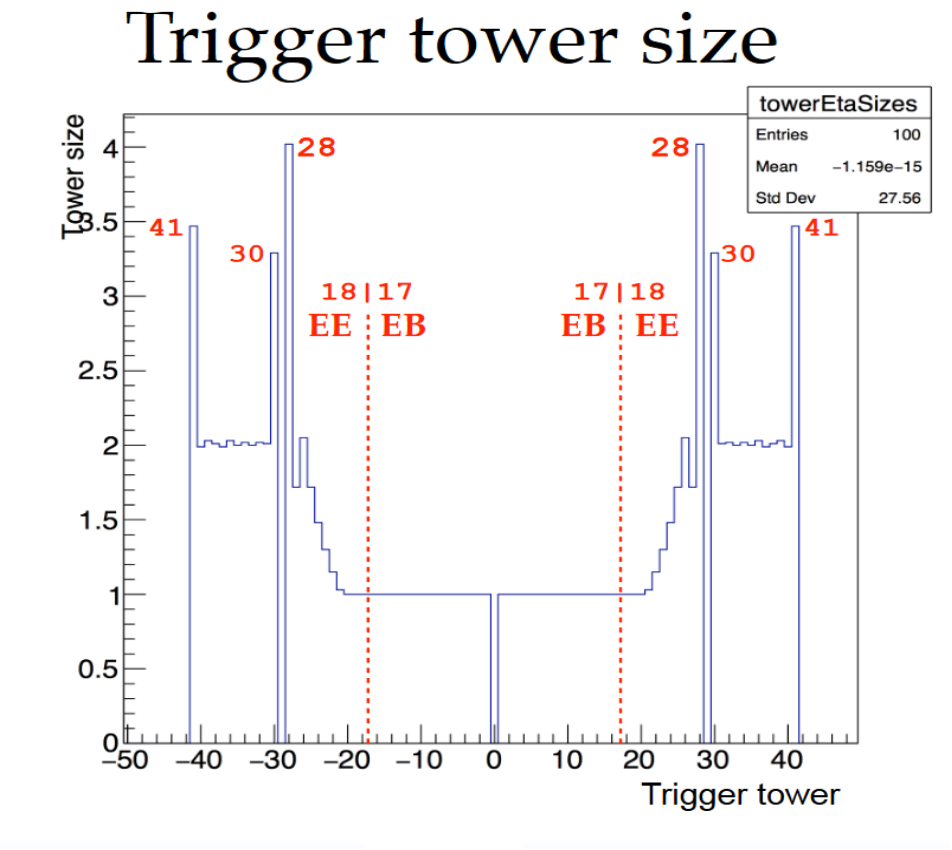

Figure 2.6: The width of the trigger towers in η as a multiple of Δη=0.087[18]

However successful, the chunky donut algorithm has a major drawback. The reconstructed energy scale of the L1T jets relative to their true energies fluctuates heavily in the forward region, which may hinder proper triggering using jets in the forward region. This

is because the L1T jet area in $\eta \times \varphi$ corresponding to the 9×9 TTs gets larger in more forward regions, while the true area of the physical jets being measured remains the same. Also the geometrical area of the three 3×9 regions used to estimate the pileup does not match the area of the 9×9 L1T jet. The first issue could be mitigated by changing the size of L1T jets in the eta direction, depending on their location in η. To address the second issue, a new algorithm called the phi–ring algorithm can be used. The mathematical formulation of phi–ring energy is:

$$\text{Phi–Ring } E_T\text{(PUS)} = [\text{Raw } E_T - (1/7)\ \text{Phi–Ring } E_T] \times \text{Layer–2 Calibration} \qquad (2.2)$$

Here, Phi–Ring E_T(PUS) is the PUS transverse energy of the jet candidate. Raw E_T is the energy deposited in the 9×9 square region before PU subtraction and Layer–2 calibration, and Phi–Ring E_T is the sum of transverse energy in the 9 full phi–rings of the 9×9 jet. The estimated PU energy in this case is 1/7 of the Phi–Ring E_T. A phi–ring consists of a total of eight 9×9 regions with the same central η values as the jet, including the jet. To estimate the pileup energy, the Phi–Ring E_T sums up the next seven 9×9 regions with the same η as the jet candidate, so the average is found by taking a mean of all these regions. The reduced energy scale fluctuation in the phi–ring algorithm, as demonstrated in Chapter 4, is attributed to its consideration of a greater surface area for calculating the pileup energy, whereas in the chunky donut algorithm the estimate is more local. Also, the phi–ring algorithm uses TTs with the same size and η location as the 9×9 jet, which gives a better prediction for PU in the 9×9 area. In our work, we have trained BDTs with both the chunky donut and phi–ring algorithms to compare their performance for calibrating L1T jets.

CHAPTER THREE

Computational Details

Due to their amazing recent advancements, different machine learning (ML) methods like neural networks, decision trees, support vector machines, and others have made their presence felt in numerous fields. They have proven to be efficient in recognizing complex patterns from a massive amount of data ('big data'). On the other hand, with advances in the collision energy and luminosity of particle colliders, experimental high energy physicists (HEP) must analyze a tremendous amount of collision data, which required a wide–scale integration of ML within HEP in recent years[23].

Machine learning can be defined as algorithms or processes that improve their accuracy in finding patterns and making predictions over a number of cycles and data. It tries to imitate how humans recognize patterns in their day–to–day experience, which makes it a branch of artificial intelligence[24]. A typical ML model starts with 'training' the model. In this process the model is fed a sufficient amount of input data along with the true or desired output data, if available. The input data are usually a number of events or occurrences that can contain one or more variables which the output of the model is dependent. These variables are known as input features[24]. During training, the model learns patterns among the features and generates output based on the input data, which are then compared to the desired output data to determine the loss or error in their decisions. If the error is higher than a predefined value (tolerance), the model is trained again with different parameters till the error is under the tolerance. After training, the models are used in classification, prediction, or regression, depending on their application[24]. Artificial feed-forward neural

networks (ANN) are perhaps the most extensively used ML techniques employed in many fields of investigation. While they can provide accurate results compared to other techniques, their training process becomes slower and slower with more and more data[25]. To alleviate this, often more computationally favorable methods based on decision tree learning techniques are used to create a balance between computational speed and accuracy. Boosted decision trees (BDTs) and bootstrap aggregating (bagging) trees are the two commonly used decision tree based techniques[26].

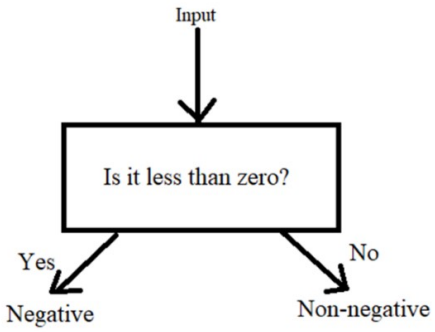

Figure 3.1: A simple decision tree

3.1 Boosted Decision Trees

A decision tree can be regarded as a tool that divides the data space into simple regions in order to classify (for discrete desired outputs) or regress (for continuous desired outputs) the provided inputs. In classification tasks the process doesn't 'count' the numbers as they are, but merely classifies them into different categories. In regression tasks the model predicts the numerical value of the target quantity. It can be used for both qualitative and quantitative features[26]. A simple decision tree is shown in Figure 3.1. A ML method based on decision trees employs a large number of such trees to classify or regress to a true output. If the outputs of the trees in the method are dependent on one another, the method is called 'boosted decision trees' (BDTs), otherwise they are called 'bootstrap aggregating trees'

(bagging trees)[26,27]. The term boosting refers to the fact that each tree is 'boosted' by (dependent on) its prior trees. Among the many types of boosting available to BDTs, the most popular ones are adaptive boosting (AdaBoost), gradient boosting and extreme gradient boosting (XGBoost)[28,29].

3.1.1 Regular and Extreme Gradient Boosting

Gradient boosting (GB) is based on a popular optimization technique called the gradient descent algorithm. In this algorithm a local minimum of a search space is determined by taking repeated steps towards the opposite direction of the gradient of the given function at the current point. The size of these steps is called the learning rate of the gradient descent. GB does something similar, but instead of taking the gradient of the output function, it uses the gradient of the error function to minimize error or loss in the decision–making process[30]. A typical error function for GB trees has the form:

$$L = [y(x) - f(x)]^2 \tag{3.1}$$

$$dL/dx = 2[y(x)-f(x)] \tag{3.2}$$

Here, L is the loss function, x is the input, y(x) is the desired (true) output and f(x) is the prediction made by the model. The term [y(x)–f(x)] represents as the error (residual) of the decision–making process[31]. GB further updates the residual by the following equation:

$$\text{New prediction (GB)} = \text{Old prediction} + \text{Learning rate} \times \text{Residual} \tag{3.3}$$

Thus, the new residual is always dependent on the previous residual and the total predictive model is a collection (ensemble) of all trees in the model. The individual trees in the ensemble are weak learners, i.e. they have a high error rate, but collectively an ensemble of weak learners can outperform models with strong learners due to their scalability (ease of modification) and computational speed[30].

Extreme gradient boosting (XGBoost) is an advanced version of the GB technique[29]. In XGBoost, to avoid overfitting (when a model performs well on training data but works poorly on test data), regularization or penalty terms are added to the residual of GB. The search space is divided into further regions based on their distance from the mean or median of the total error. The regularization terms depend on this new error metric. Since XGBoost focuses more on smaller search spaces it reaches the optimized solution faster than GB[29]. It takes the form:

$$\text{New prediction (XGB)} = \text{New prediction (GB)} + \lambda \times \text{Error}_{\text{Branch}} \tag{3.4}$$

Here, λ is a regularization parameter and $\text{Error}_{\text{Branch}}$ is either the mean or the median of the error of the current branch. It is possible to add another regularization term to XGBoost to increase the optimization speed further. The second regularization parameter is often denoted by α. The values of λ and α are usually small and depend on the residual and distribution of values in the search space[29].

3.1.2 Algorithms

Let us consider an example of a dataset containing 4 observations where the desired outputs are 10, 12, 14 and 16. Let the initial predictions for both GB and XGB be 13 (mean of the distribution) for all cases. Further steps in GB are performed as follows:

Step 1: The residual of each prediction is determined by subtracting the prediction from the true or desired outputs.

Step 2: New predictions are made using Equation (3.3). Thus, for a learning rate of 0.1, for iteration 1 the predictions are:

$13+(0.1\times-3)=12.7$, $13+(0.1\times-1)=12.9$, $13+(0.1\times1)=13.1$, $13+(0.1\times3)=13.3$

Step 3: Step 1 and Step 2 are repeated till the error is under a predefined value (tolerance). Table 3.1 shows another iteration of the prediction for GB.

Table 3.1: Regular gradient boosting

True	Prediction	Residual	Prediction (Iteration 1)	Residual (Iteration 1)	Prediction (Iteration 2)	Residual (Iteration 2)
10	13	–3	12.7	–2.7	12.43	–2.43
12	13	–1	12.9	–0.81	12.81	–0.81
14	13	1	13.1	0.9	13.19	0.81
16	13	3	13.3	2.7	13.57	2.43

Table 3.2: Extreme gradient boosting

True	Prediction	Residual	Prediction (Iteration 1)	Residual (Iteration 1)	Prediction (Iteration 2)	Residual (Iteration 2)
10	13	–3	12.5	–2.5	12.09	–2.09
12	13	–1	12.7	–0.7	12.47	–0.47
14	13	1	13.3	0.7	13.53	0.47
16	13	3	13.5	2.5	13.91	2.09

In XGBoost a cutoff point is defined at first, and the data are divided into multiple regions based on the cutoff. In the example above, the mean residual after the initial prediction is (–3–1+1+3)/4=0. Let us set the cutoff at zero. In real applications of XGBoost, where the number of observations are far higher than 4, such cutoffs are created periodically after a few iterations to reach the optimized solution faster. The steps of the algorithms are as follows:

Step 1: Residual of each prediction is determined by subtracting the prediction from the true or desired outputs.

Step 2: New predictions are made using Equation (3.4). As the cutoff is at zero, 4 data points can be divided into two branches: on the first branch the mean error is (–3–1)/2=–2 and on

the second branch the mean error is (3+1)/2=2. Thus, for a learning rate of 0.1 and λ=0.1, for iteration 1 the predictions are:

13+(0.1×−3)+(0.1×−2)=12.5, 13+(0.1×−1)+(0.1×−2)=12.7,

13+(0.1×1)+(0.1×2)=13.3, 13+(0.1×3)+(0.1×2)=13.5

Step 3: Step 1 and Step 2 are repeated till the error is under a predefined value (tolerance).

Table 3.2 shows another iteration of the prediction for GB. From Table 3.1 and 3.2, we can see that after 2 iterations the residual is less in XGBoost, which means it converges faster than GB.

3.2 BDT Training

In our work, we implemented two different BDTs using the 2018 single muon data. The dataset for the first BDT (BDT1) contained 15 jet variables (features) and had about 4.5 million jets of all energy ranges. The jet variables are listed in Table 3.3.

These jet variables target different jet sizes and regions along the path of the jets. Since we wanted our BDT to differentiate between the high (more than 50 pileup vertices, nVtx > 50) and low pileup events (less than 25 pileup vertices, nVtx < 25), exactly 50% of the jets were in high pileup events and the remaining 50% were in low pileup events. We used 50% (randomly chosen) of our data as the training data and 50% were used as the test data. BDT1 was trained with a total of 12 variables (all except PFJetEtCorr, L1JetType and PileupEnvironment) and the output of the BDT was regressed to PFJetEtCorr as the desired output.

Since 'PFJetEtCorr' is the target variable (desired output), it was omitted from the list of input features. The other two variables, L1JetType and PileupEnvironment, are omitted because, firstly, all jets of the dataset were of emulated type, so there is no point of including that in our training and secondly, as described above, we wanted our BDT to

differentiate between high and low pileup events on its own. We trained BDT1 with all possible variables to perform a feature ranking on them so that the features which are more important in reconstructing jets can be used in future modifications to the L1T jet algorithm.

Table 3.3: List of jet variables

Variable	Definition
PFJetEtCorr	Transverse energy (E_T) of particle flow offline (PF) jets
L1JetType	Type of L1T jet. Either emulated or unpacked jets. Current dataset only has emulated jets
L1JetToweriEtaAbs	Absolute value of iEta of L1T jet seed
L1JetDefault_EtPUS	E_T of 9×9 ($\eta \times \varphi$) L1T jets with PU subtraction and calibrations from Run 2
L1JetDefault_RawEtPUS	PU subtracted E_T of 9×9 jets with chunky donut PU subtraction without Layer–2 calibration
L1JetDefault_PU	PU for 9×9 jets estimated with chunky donut algorithm
L1Jet9x9_RawEt	E_T of 9×9 jets before PU subtraction and without Layer–2 calibration
L1Jet9x9_EtSum7PUTowers	Sum of E_T in full phi–ring of the 9×9 jet, excluding the 9×9 jet area
L1Jet7x9_RawEt	E_T of 7×9 ($\eta \times \varphi$) jets before PU subtraction and without Layer–2 calibration
L1Jet7x9_EtSum7PUTowers	Sum of E_T in full phi–ring of the 7×9 jet
L1Jet5x9_RawEt	E_T of 5×9 ($\eta \times \varphi$) jets before PU subtraction and without Layer–2 calibration
L1Jet5x9_EtSum7PUTowers	Sum of E_T in full phi–ring of the 5×9 jet
L1Jet3x9_RawEt	E_T of 3×9 ($\eta \times \varphi$) jets before PU subtraction and without Layer–2 calibration
L1Jet3x9_EtSum7PUTowers	Sum of E_T in full phi–ring of the 3×9 jet
PileupEnvironment	High (nVtx > 50) or low (nVtx<25) pileup

After analyzing the performance in high and low PU events, and the feature ranking plots of the BDT1, we realized that the phi–ring algorithm is better at detecting jets than the current chunky donut algorithm. Hence, we moved on to the training of energy calibration BDTs. At first, we calculated the phi–ring energies for all events in the dataset using equation (2.2), then we filtered out the jets with phi–ring energy values less than 10 GeV/c^2 since the vast majority of low E_T jets come from pileup. With all the jets having phi–ring energy equal to or more than 10 GeV/c^2, the size of the dataset was reduced to 2.9 million jets with about 1.6 million high PU and 1.3 million low PU jets. Then we set up three new sets of energy calibration BDTs with two input variables and four BDTs in each set. They are tabulated in Table 3.4.

Table 3.4: BDT training with two input variables

BDT	Input Variable	Target Variable
Phi–Ring BDT	L1JetToweriEtaAbs Phi–RingEnergy	PFJetEtCorr Phi–RingEnergy/PFJetEtCorr log(PFJetEtCorr) log(Phi–RingEnergy/PFJetEtCorr)
L1Jet BDT	L1JetToweriEtaAbs L1JetDefault_EtPUS	PFJetEtCorr L1JetDefault_EtPUS /PFJetEtCorr log(PFJetEtCorr) log(L1JetDefault_EtPUS /PFJetEtCorr)
L1JetRaw BDT	L1JetToweriEtaAbs L1JetDefault_RawEtPUS	PFJetEtCorr L1JetDefault_RawEtPUS /PFJetEtCorr log(PFJetEtCorr) log(L1JetDefault_RawEtPUS /PFJetEtCorr)

We trained separately for chunky donut and phi–ring algorithms to check their performance against one another. Again, we used 50% of the data as the training data, and the rest were used to validate the performance of the training. In the training process, we trained each BDT 10 times with different random subsets of data. After each training we used a pseudo test dataset containing iEta values from 1 to 40 (except 29) with energy (chunky donut or phi–ring) values from 1 to 200 GeV/c^2 for each iEta to obtain the scale factors of the trained BDTs associated with each iEta for a particular energy. These events mimic all possible iEta and E_T values seen by the L1T system. After each set is trained, we took the values of mean scale factors from the 10 BDTs and plotted them against the input E_T values for each iEta. For each iEta and E_T pair as input, the BDT would produce the calibrated energy as the output. The energy calibration scale factors are then computed as the ratio of the BDT output energy to the BDT input energy.

All BDTs were set up using xgboost library with python. The learning rate was set at 0.01. The maximum depth in all BDTs was set at 5, and the number of trees in each depth was 1000.

CHAPTER FOUR

Results and Discussion

This section is divided into two parts. In the first part, we discuss the results from the 12–variable BDTs where we compare the energy scale and resolution (the ratio of the square root of the variance and the energy scale) of the BDT output with the energy scale and resolution of the chunky donut algorithm, both with and without Layer–2 calibration. Importance rankings of input jet variables in different regions of the CMS detector are also provided. In the second part, we go into the performance of the 2–variable Phi–Ring, L1Jet and L1JetRaw BDTs by comparing their energy scales and resolutions.

4.1 12–Variable BDT

4.1.1 Energy Scale Plots

In each of the energy scale plots of the 12–variable BDT, we have compared three energy distributions for training and testing datasets. These distributions are the output of the BDT, L1Jet energy (with Layer–2 calibration) and L1JetRaw energy (without Layer–2 calibration). While there are no training and testing datasets associated with L1Jet p_T and L1JetRaw p_T, we provided their corresponding values for training and testing datasets to compare with the BDT output. Figure 4.1 shows the energy scale comparison of high and low PU for PF jets with all ranges of pT (mostly below 30 GeV), and PF jets with p_T between 60 and 90 GeV, a common range for L1T multijet triggers. Here, we can see that the energy scale of the L1Jet (orange) and L1JetRaw (green) are different in the barrel and endcap regions, but merge together in the forward region. This happens because in the L1Jet distribution, Layer–2 scale factors were applied but were turned off in the forward region.

Thus, 2018 L1Jet and L1JetRaw both have the same p_T in the forward region. The output of the BDT (blue) does not fluctuate much in comparison to the orange and green lines, indicating stable performance.

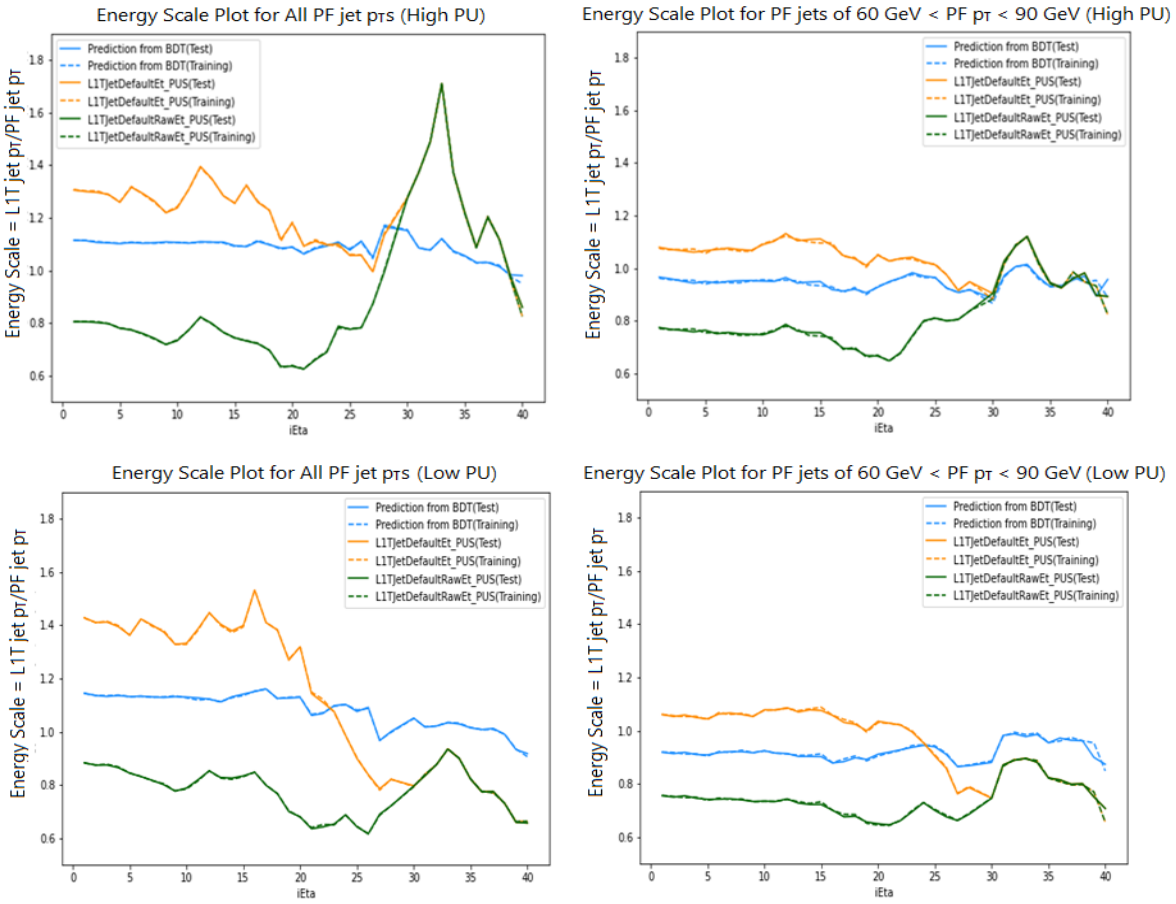

Figure 4.1: Energy scale for high and low PU (top to bottom) for all PF jet p_Ts (left) and PF jets with p_T between 60 and 90 GeV (right)

Another superiority of the BDT output compared to the other two distributions is in high and low PU plots, the shapes of the orange and green lines are different in the forward region. In high PU plots, they go up sharply between iEta 30 and 35 then come down, for both 'all p_T' and '60 GeV to 90 GeV' ranges. But in low PU plots, the orange line comes down sharply in the forward region and merges onto the green line, whereas in

all plots irrespective of the PU type, the BDT output remains comparatively consistent. This is clearly visualized in Figure 4.2 where the ratio of high and low PU energy scale is nearly equal to 1 in all regions but for orange and green lines its is far higher in the forward region. This indicates that the BDT output is capable of differentiating between high and low PU events on its own, even if the input data didn't explicitly include this information.

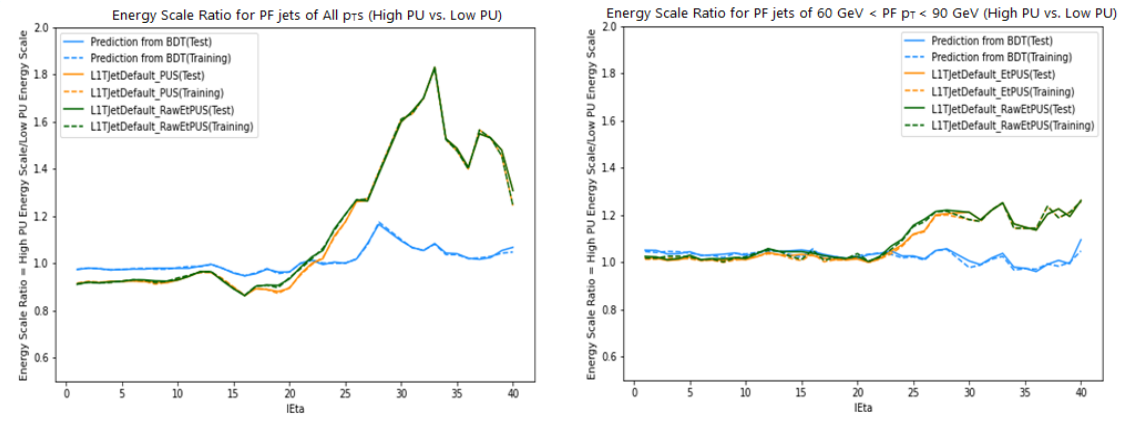

Figure 4.2: Energy scale ratio (high vs low PU) for all PF jet pTs (left) and PF jets of p_T between 60 GeV and 90 GeV (right)

4.1.2 Resolution Plots

Similar to the energy scale, we have compared the resolutions of three aforementioned energy distributions as well. The resolution is the ratio of the square root of the variance over the mean energy scale, thus a smaller variation among the data indicates better resolution and vice versa. From the resolution plots of Figure 4.3, it can be seen that for the full PF p_T range the resolution of the BDT output is slightly better than the resolutions L1Jet and L1JetRaw. But in the 60–90 GeV range, the resolution of L1Jet is slightly better than the BDT output in the barrel and endcap regions, but similar in the forward region. The resolution ratio of the high and low PU energy distributions of the BDT

output is found to be very similar to the other distributions, thus we don't observe any noticeable improvement in the resolution for BDT training as was observed for the energy scale. The difference between L1Jet and the 12–variable BDT is much larger for the energy scale (up to 20% of the jet energy) than it is for the resolution (less than 5%).

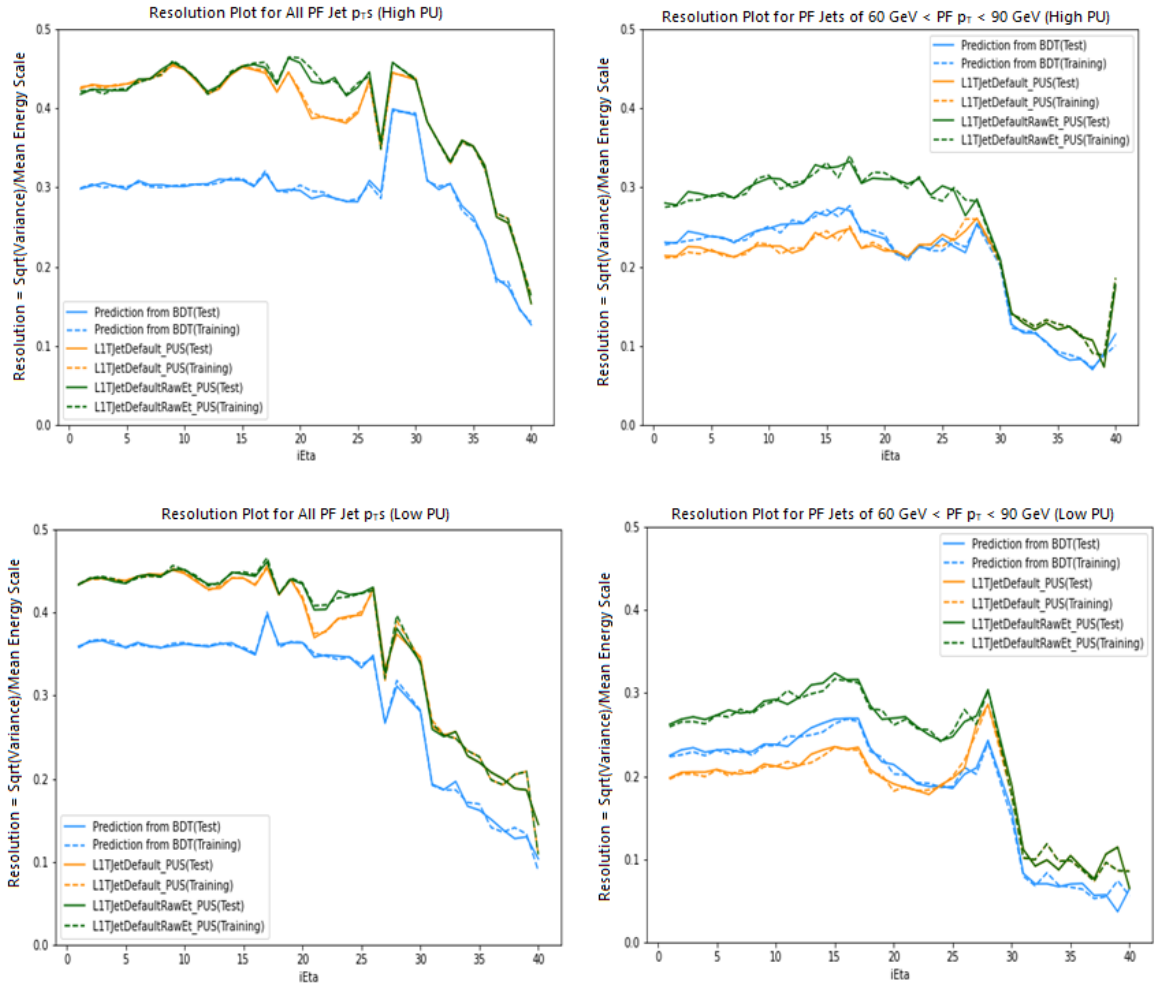

Figure 4.3: Resolution for high and low PU (top to bottom) for all PF jet p_Ts (left) and PF jets of p_T between 60 and 90 GeV (right)

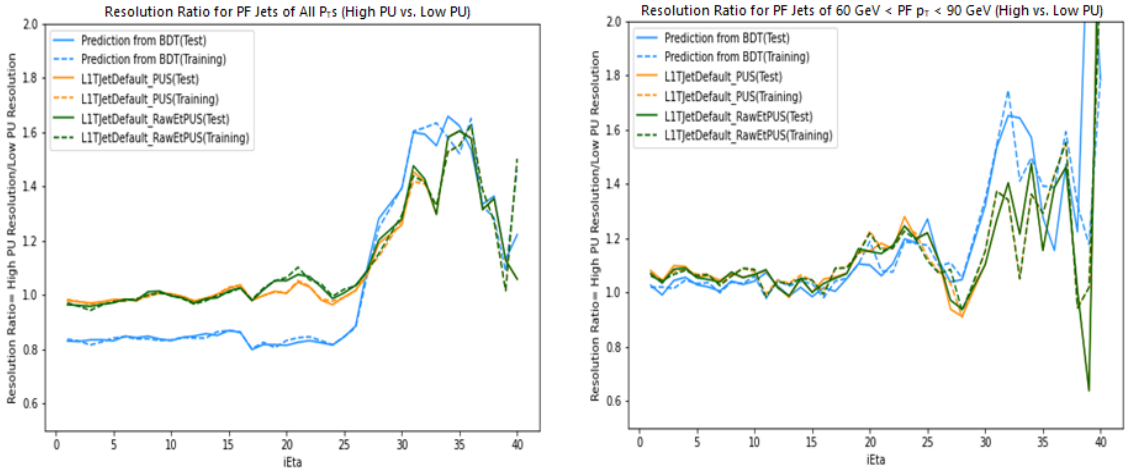

Figure 4.4: Resolution ratio (high vs low PU) for all PF jet p_Ts (left) and PF jets of p_T between 60 and 90 GeV (right)

4.1.3 Ranking Plots of the BDT Variables

Variable ranking plots for all regions (Figure 4.5) indicate that the variable L1JetDefault_EtPUS (L1 jet pT with PU subtraction) is the jet variable that is preferred most by the BDT in determining the PF jet energies. But when we probed deeper and trained five new BDTs in five separate regions of the HCAL, we obtained a different insight. These regions are: barrel (iEta less than or equal 16), endcap 1 (iEta 17–20), endcap 2a (iEta 21–25), endcap 2b (iEta 26–28) and forward (iEta 30–41). In the barrel region (Figure 4.5), L1Jet9x9_RawEt (a variable from equation 2.2 which helps in determining the value of the Phi–Ring p_T) is found to be the most important variable. In the endcap 1 region, the same variable is found to be the most significant with some strong contribution from L1Jet7x9_RawEt.

This indicates that the BDT starts preferring smaller jet sizes (i.e. fewer trigger tower rings in iEta) as it progresses through different regions. We claim this to be the case since we see a strong contribution from L1Jet5x9_RawEt in the endcap 2a region (even though L1Jet

pT has the highest contribution in this region) and from L1Jet3x9_RawEt in endcap 2b and the forward region.

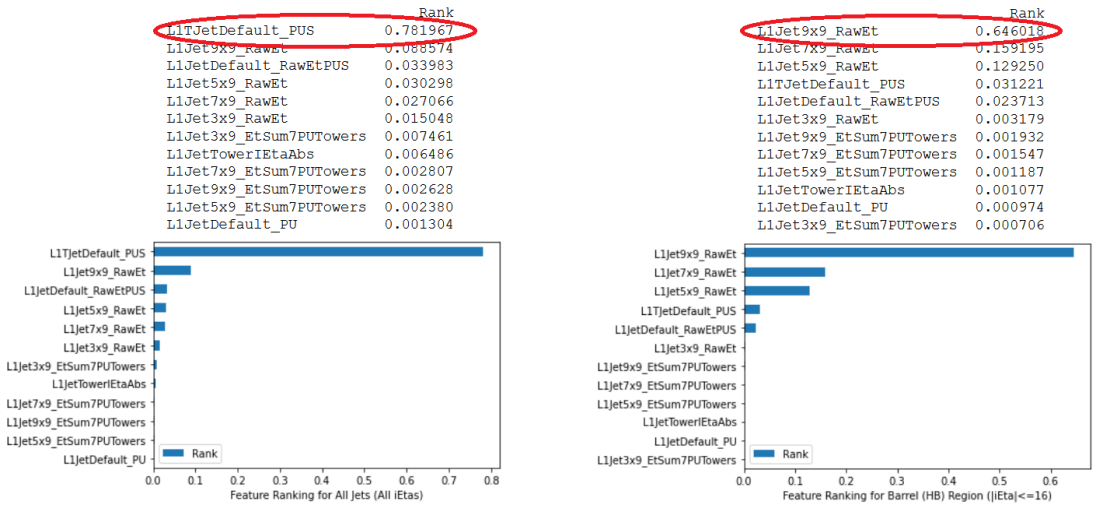

Figure 4.5: Ranking plots for all regions (left), barrel (HB) region (right)

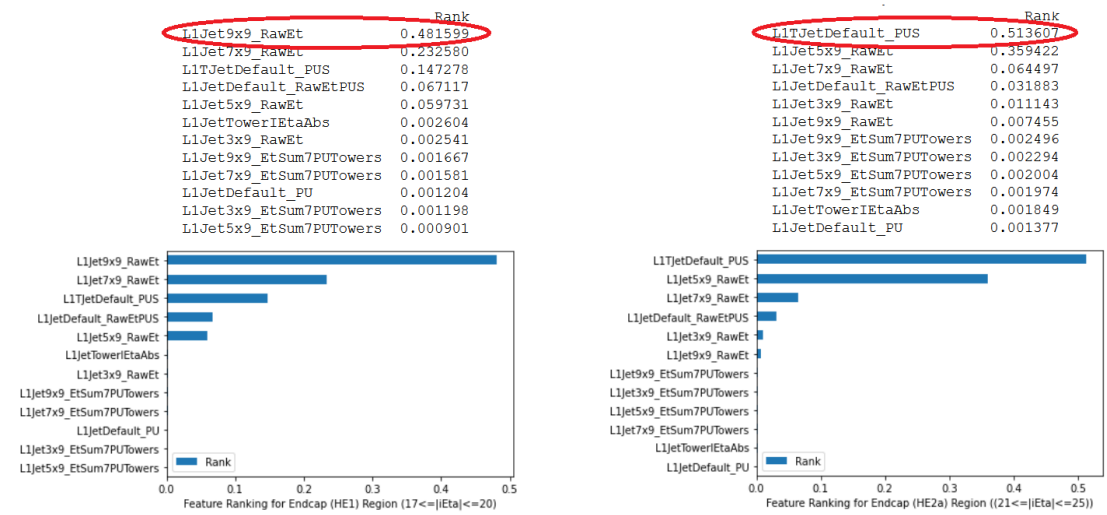

Figure 4.6: Ranking plots for endcap 1 region (left), endcap 2a (HE2a) region (right)

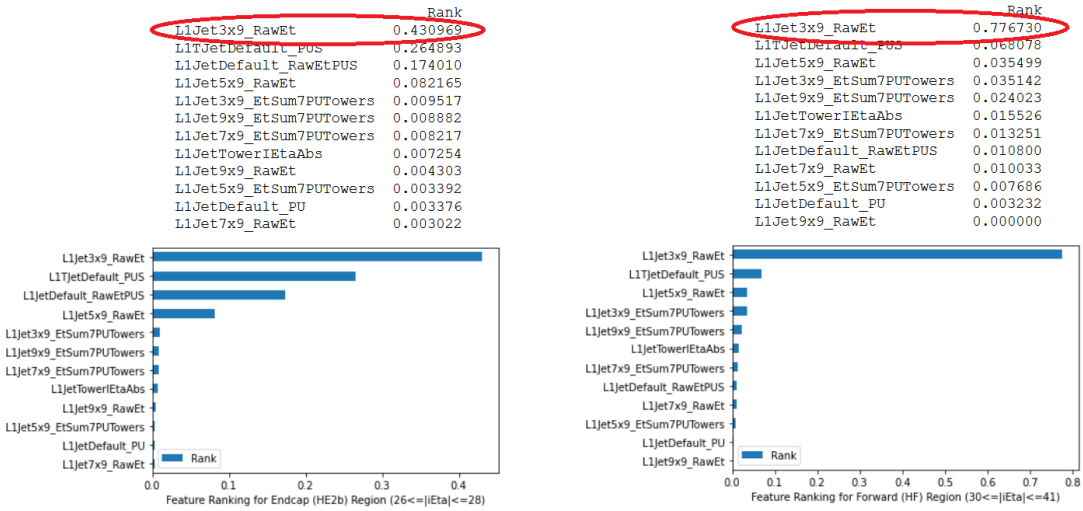

Figure 4.7: Ranking plots for endcap 2b (HE2b) region (left), forward (HF) region (right)

4.2 Energy Calibration BDT

4.2.1 Phi–Ring BDTs

As described in Table 3.4, we trained four BDTs with Phi–Ring p_T and iEta values as inputs and compared their outputs with the values of Phi–Ring p_T, L1Jet p_T and L1JetRaw p_T. From the graphs it is evident that Phi–Ring p_T (pink) is more consistent in all regions than L1Jet p_T (violet) and L1JetRaw p_T (brown). The BDTs are regressed to PF p_T, Phi–Ring p_T/PF p_T, log (PF p_T) and log(Phi–Ring p_T/PF p_T) and their outputs are plotted in blue, orange, green and red respectively. These lines are quite similar in shape and differ mainly in scale factors. That is why we will keep the focus of our discussion primarily on the output of the PF p_T BDT (blue). We generated energy scale and resolution plots for four PF p_T ranges: 25–35 GeV, 40–55 GeV, 60–90 GeV and 100–200 GeV (Figure 4.8–Figure 4.11) to compare the performance in separate regions of the p_T values for high and low PUs. Here we got results similar to the previous training with 12 variables, except now our BDT outputs have greater stability than the previous training. It can be understood from the high and low PU plot that better stability is achieved in the energy scale for BDT outputs than L1Jet and

L1JetRaw p_Ts. Since the resolution plots tell us that the resolution is similar to other distributions, we can conclude that Phi–Ring p_T will be better for detecting jets with all ranges of p_T. To confirm this, we trained two similar sets of BDTs with L1Jet and L1JetRaw p_Ts respectively as inputs. Their results are described in the following sections. We also notice that the energy distributions become flatter as the energies of the PF jets increase.

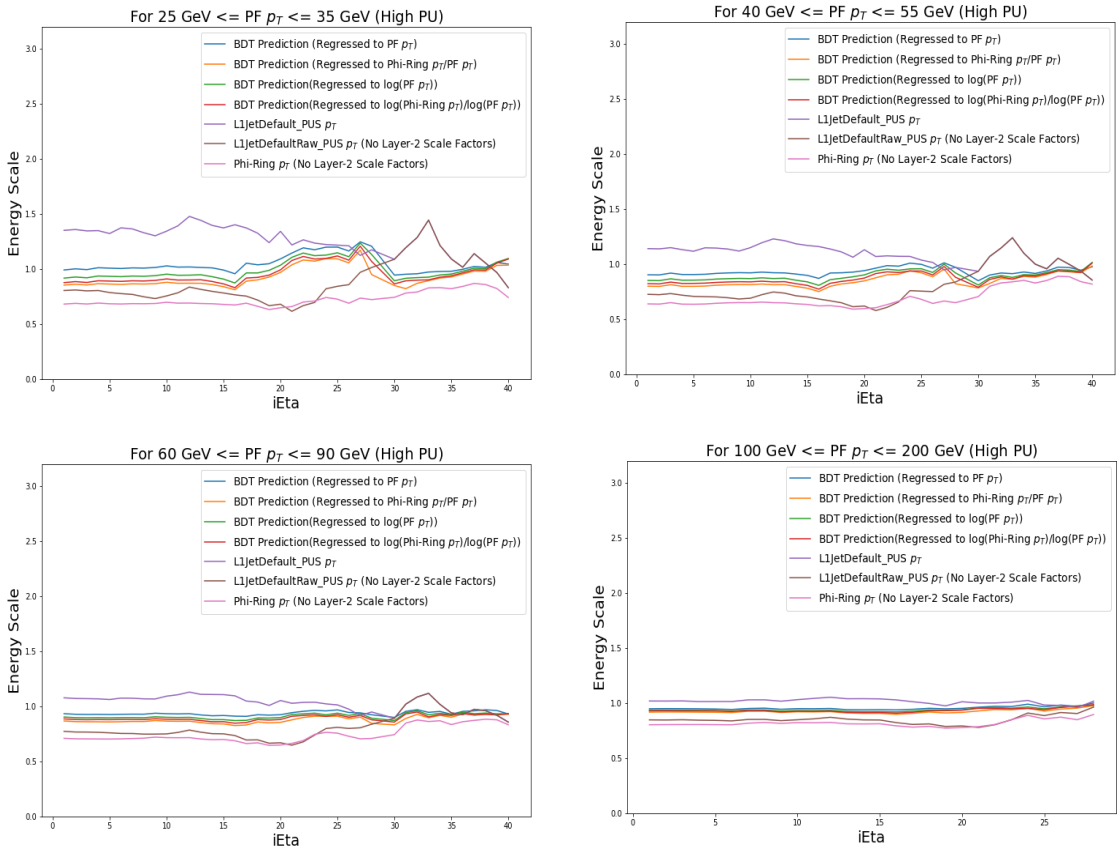

Figure 4.8: Energy scale of the Phi–Ring BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in high PU

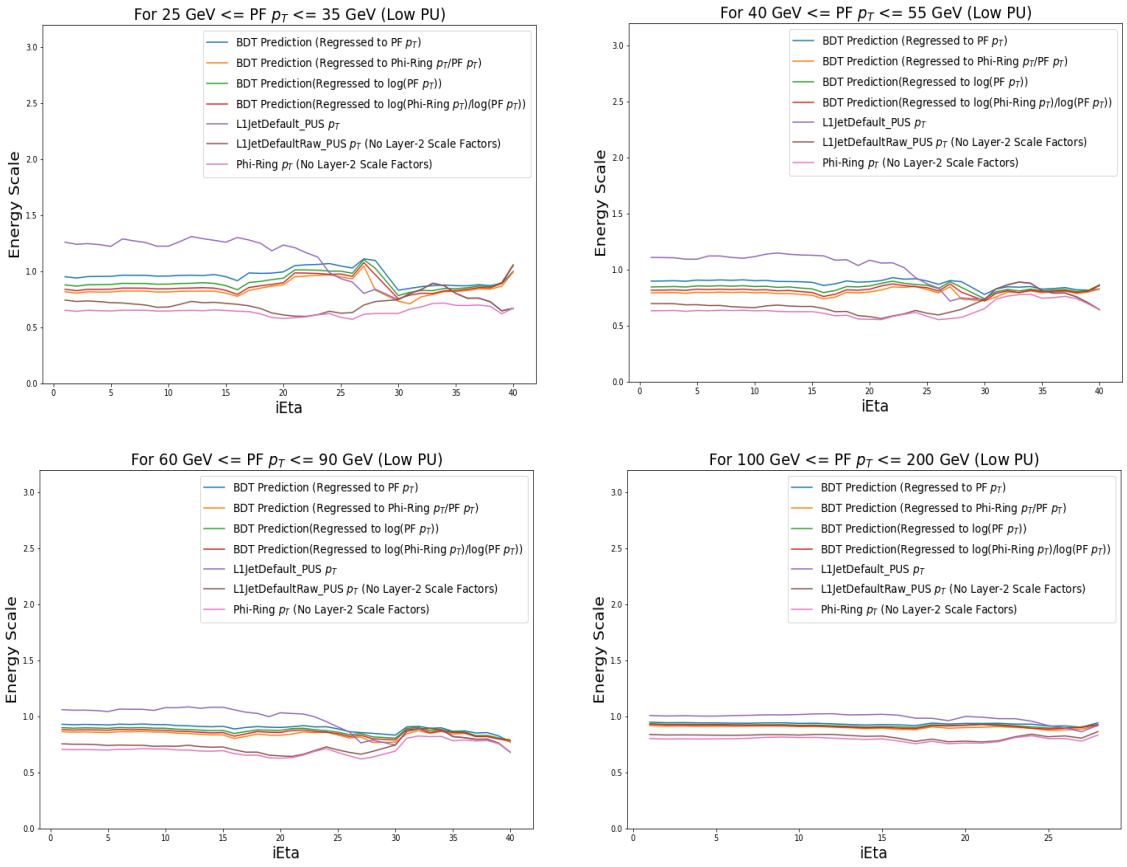

Figure 4.9: Energy scale of the Phi–Ring BDT for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in low PU

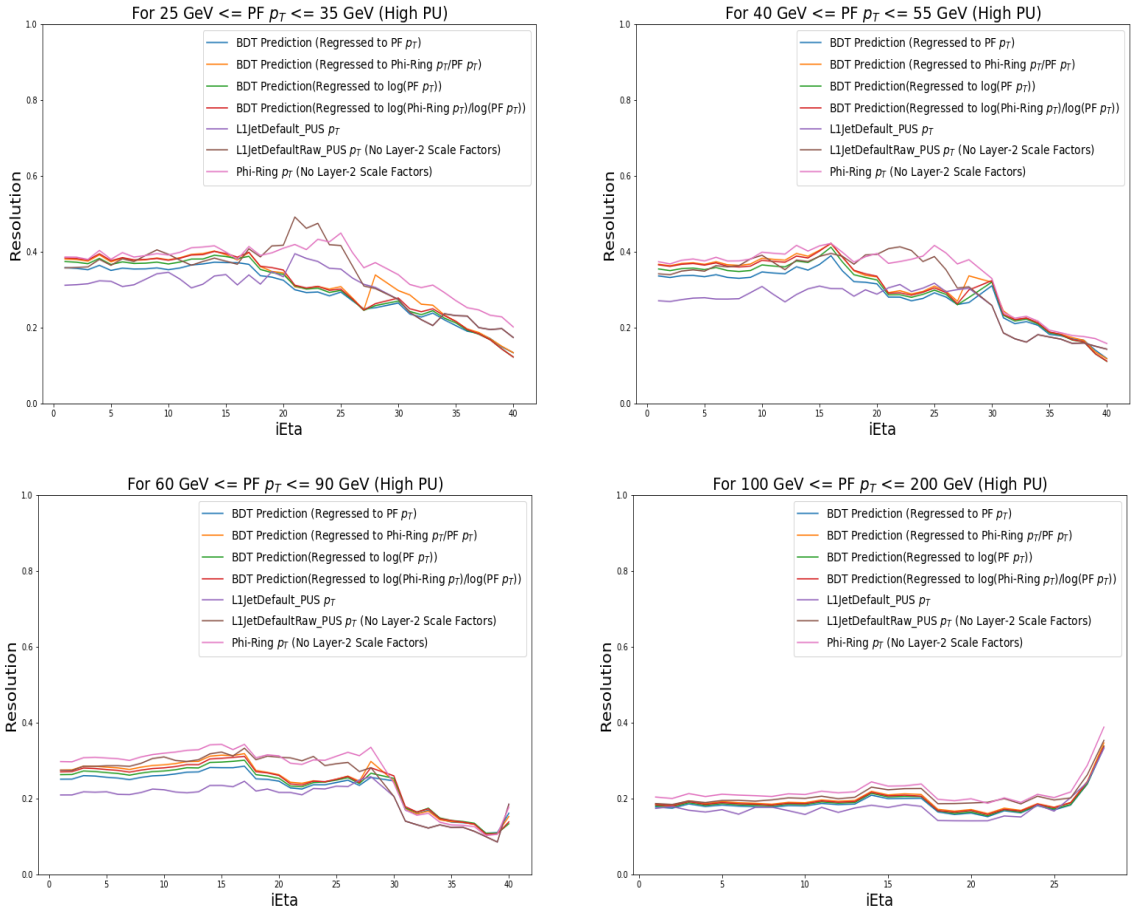

Figure 4.10: Resolution of the Phi–Ring BDT for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in high PU

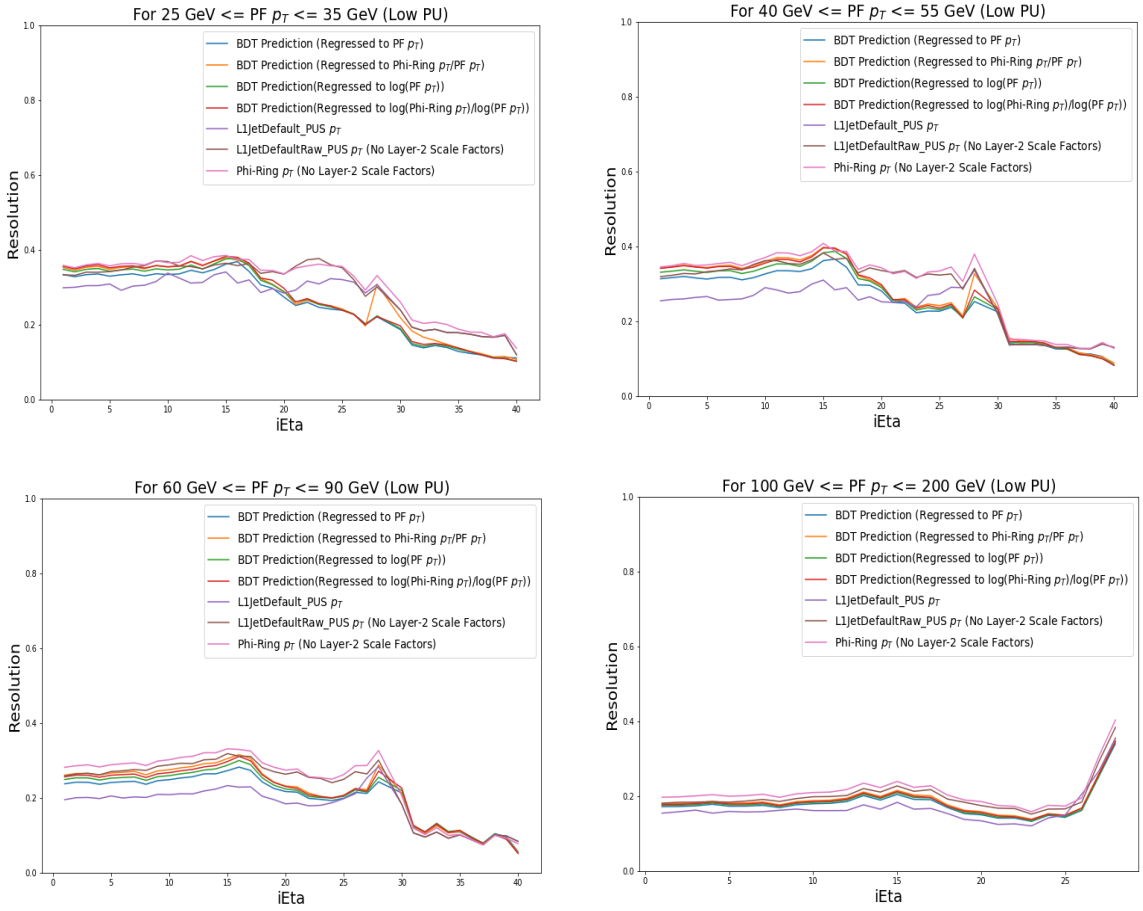

Figure 4.11: Resolution of the Phi–Ring BDT for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in low PU

4.2.2 L1Jet and L1JetRaw BDTs

We have generated similar plots described in the previous section for L1Jet BDTs (Figures 4.12–4.15). Although the individual outputs of the L1Jet BDTs are flatter when PF p_T with all ranges are considered, they suffer a serious lack of stability in the lower p_T ranges. It is also evident from the high to low PU plots that they do not achieve better stability in the energy scale compared to the Phi–Ring BDTs, i.e. the ratio is not close to 1 in the forward region (Figure 4.20). We suspect the reason for this involves the inputs of the BDT training. Since the BDT takes L1Jet p_T (which fluctuates a lot in the forward region for both high and

low PU) as an input variable, and the output of the BDT also has similar tendencies. The resolution of the L1Jet BDT from resolution plots is found to be similar to other energy distributions as well. The resolution ratio (Figure 4.21) between the Phi–Ring and L1Jet BDTs also confirms this result. This is an expected outcome as we could not improve the resolution in the previous 12–variable trainings.

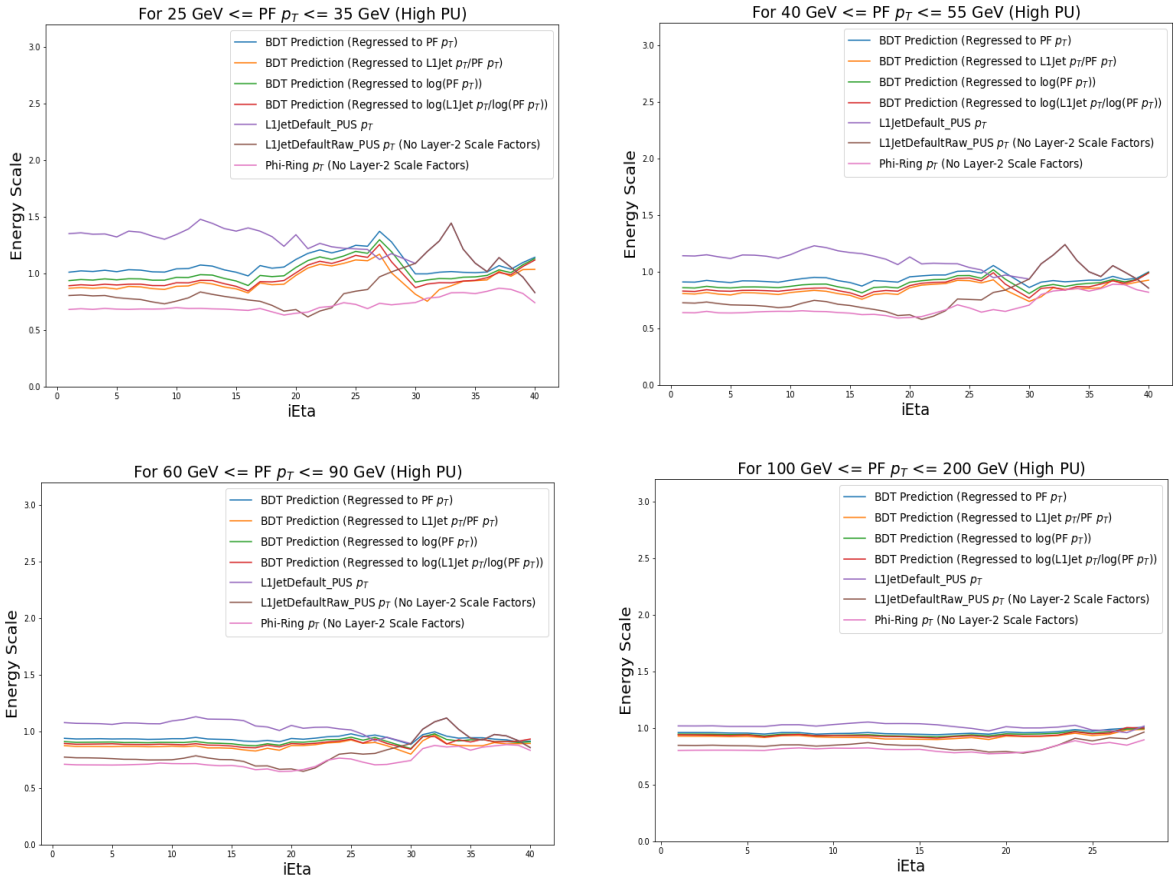

Figure 4.12: Energy scale of the L1Jet BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in high PU

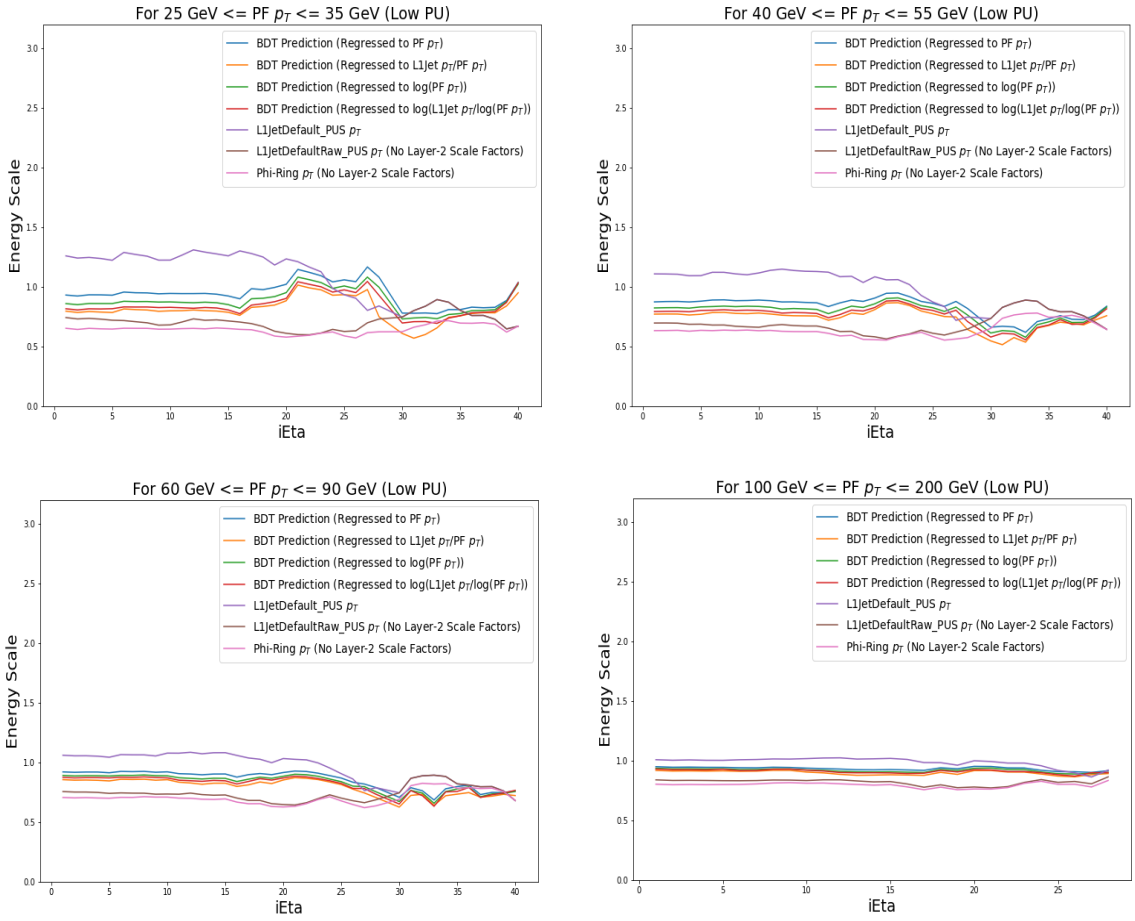

Figure 4.13: Energy scale of the L1Jet BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in low PU

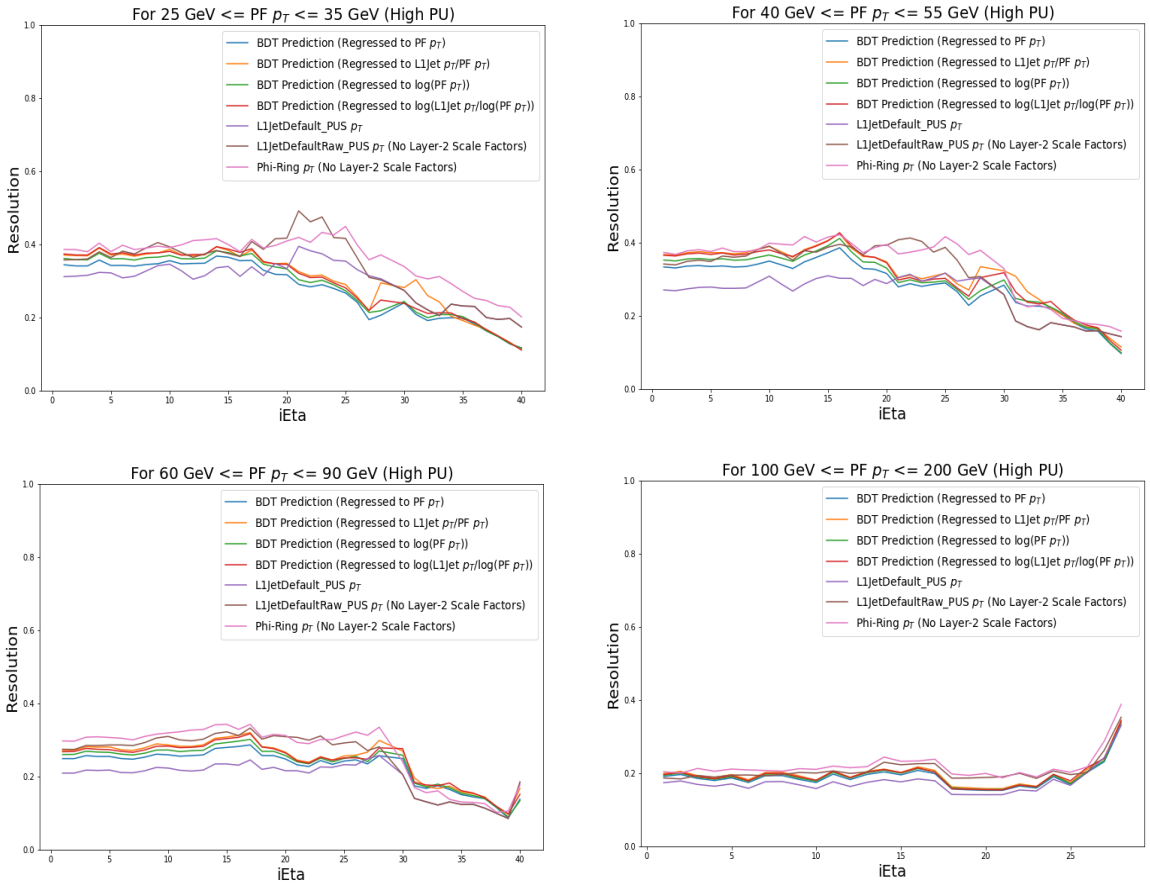

Figure 4.14: Resolution of the L1Jet BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in high PU

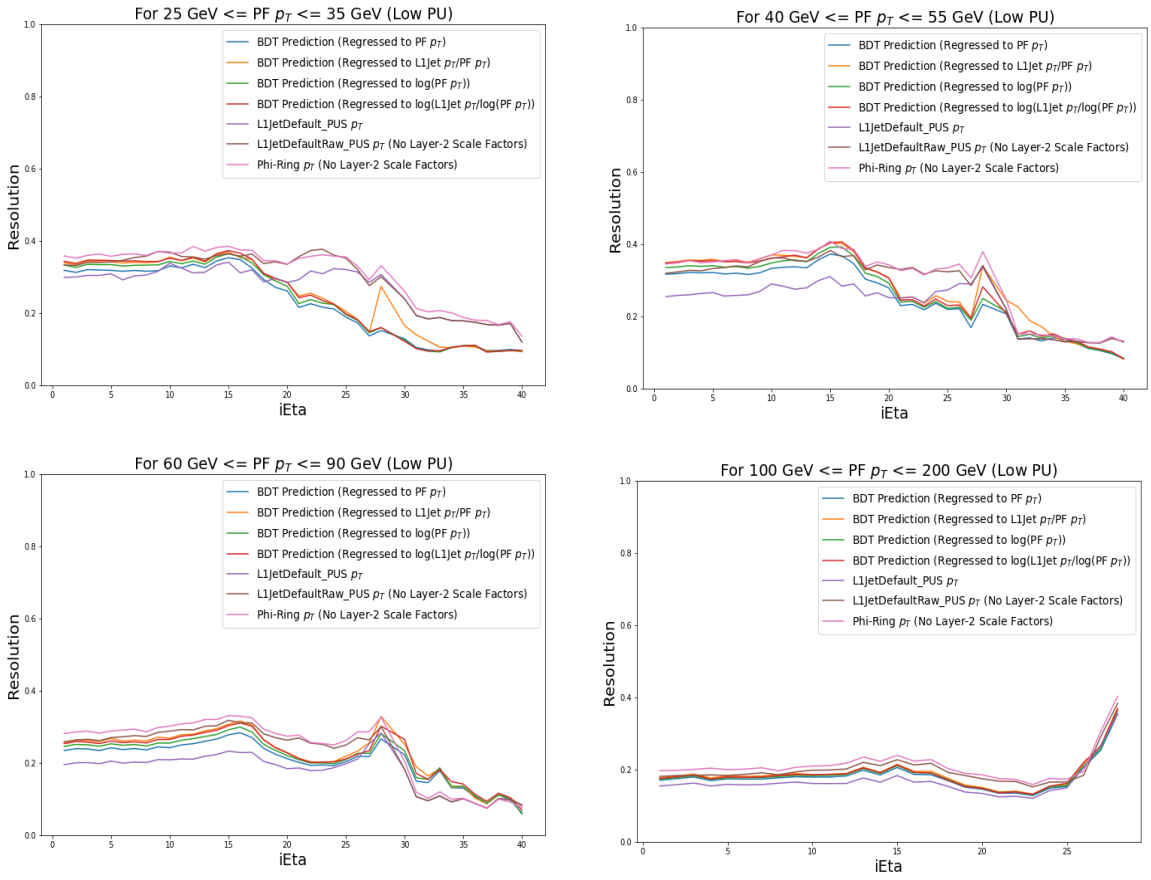

Figure 4.15: Resolution of the L1Jet BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in low PU

Training with L1JetRaw BDT generated similar results (Figures 4.16–4.19) to the L1Jet BDT (i.e. they also suffer from a lack of stability in lower energy ranges and in the forward region), except that the resolutions are slightly better for L1Jet BDTs in the barrel and endcap regions as they have Layer–2 calibration applied to them.

These results confirm that the Phi–Ring BDT is superior in performance to the L1Jet and L1JetRaw BDTs, and hence would be more suitable for reconstructing L1T jets.

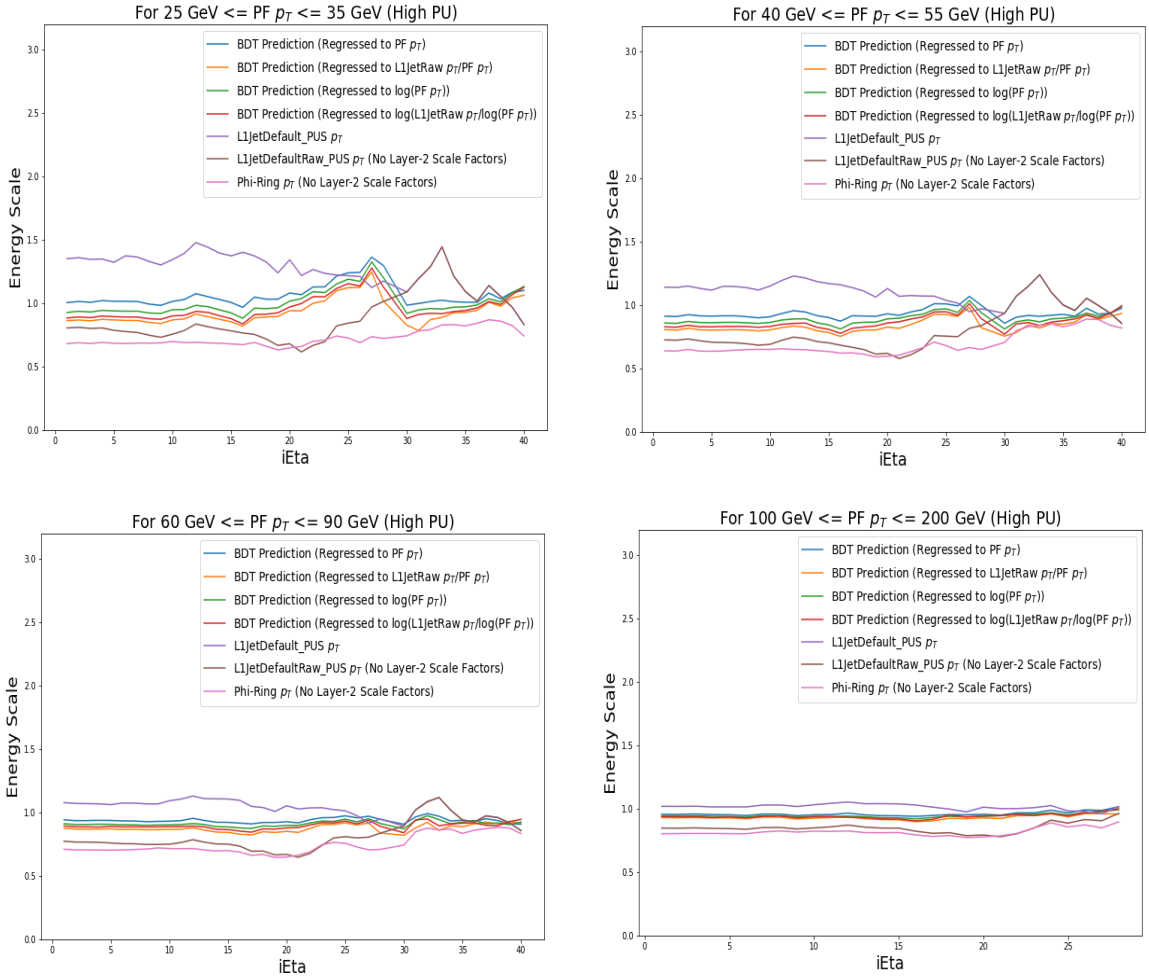

Figure 4.16: Energy scale of the L1JetRaw BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 Gev (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in high PU

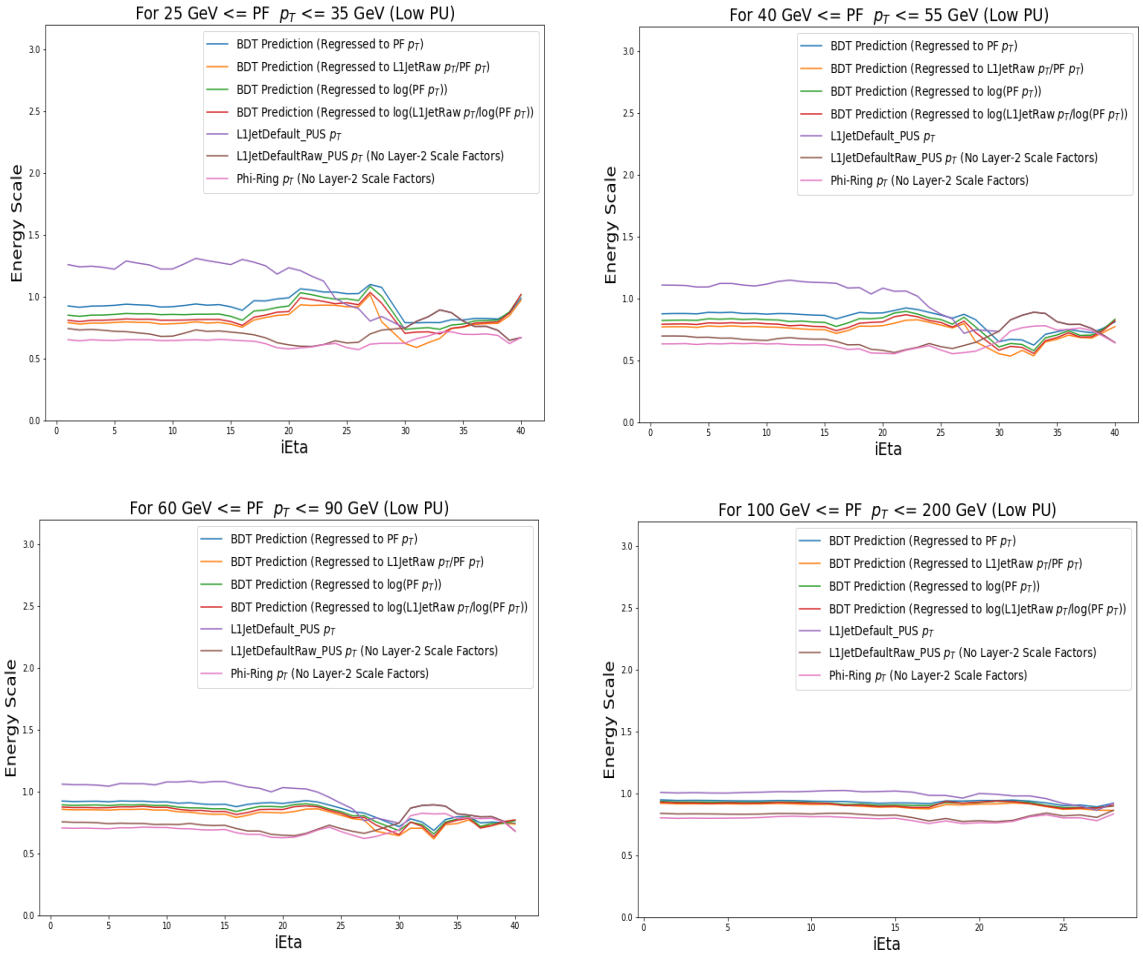

Figure 4.17: Energy scale of the L1JetRaw BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in low PU

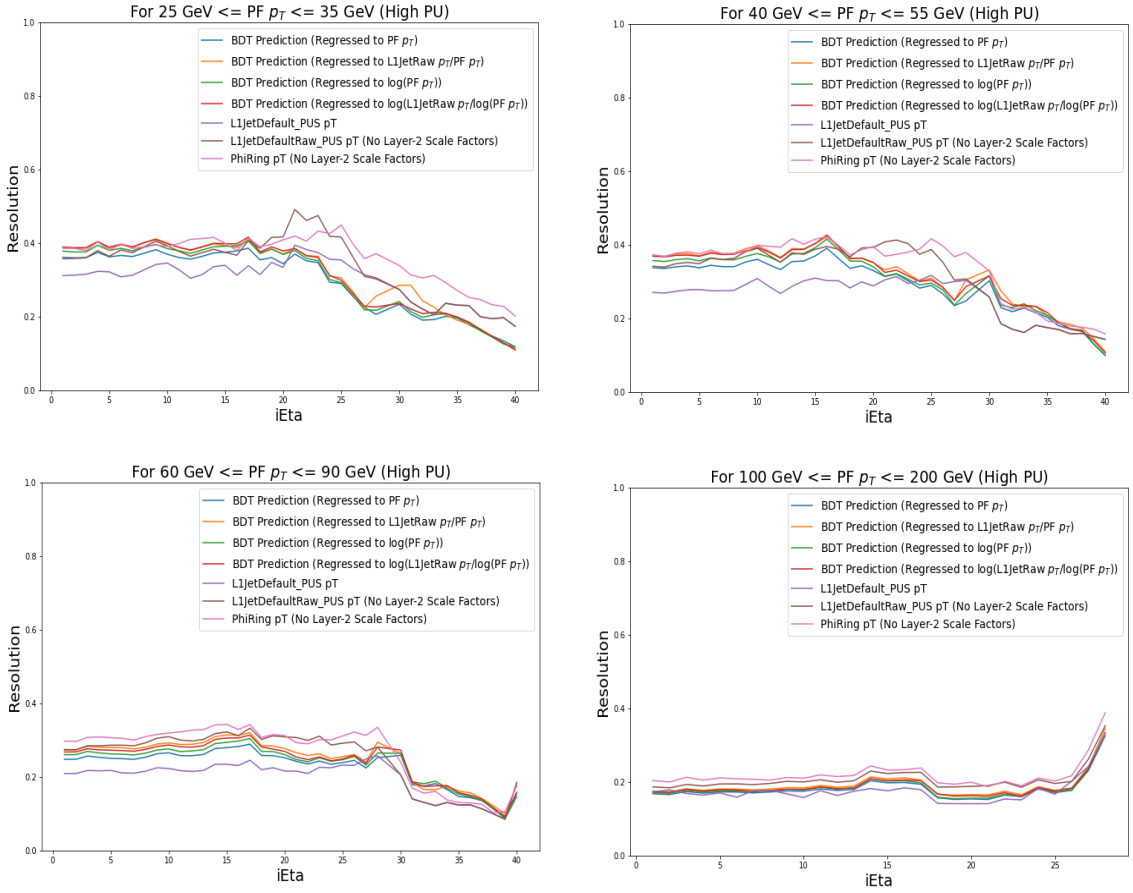

Figure 4.18: Resolution of the L1JetRaw BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in high PU

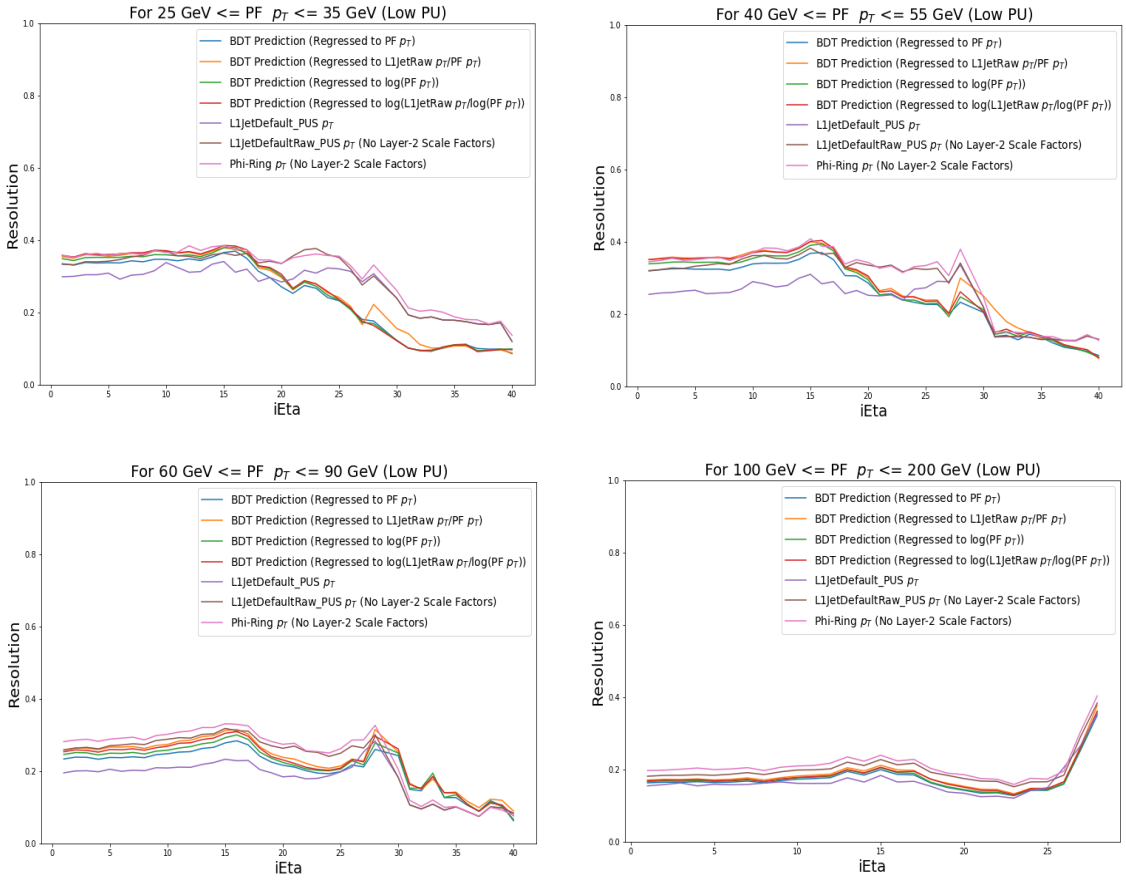

Figure 4.19: Resolution of the L1JetRaw BDTs for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right) in low PU

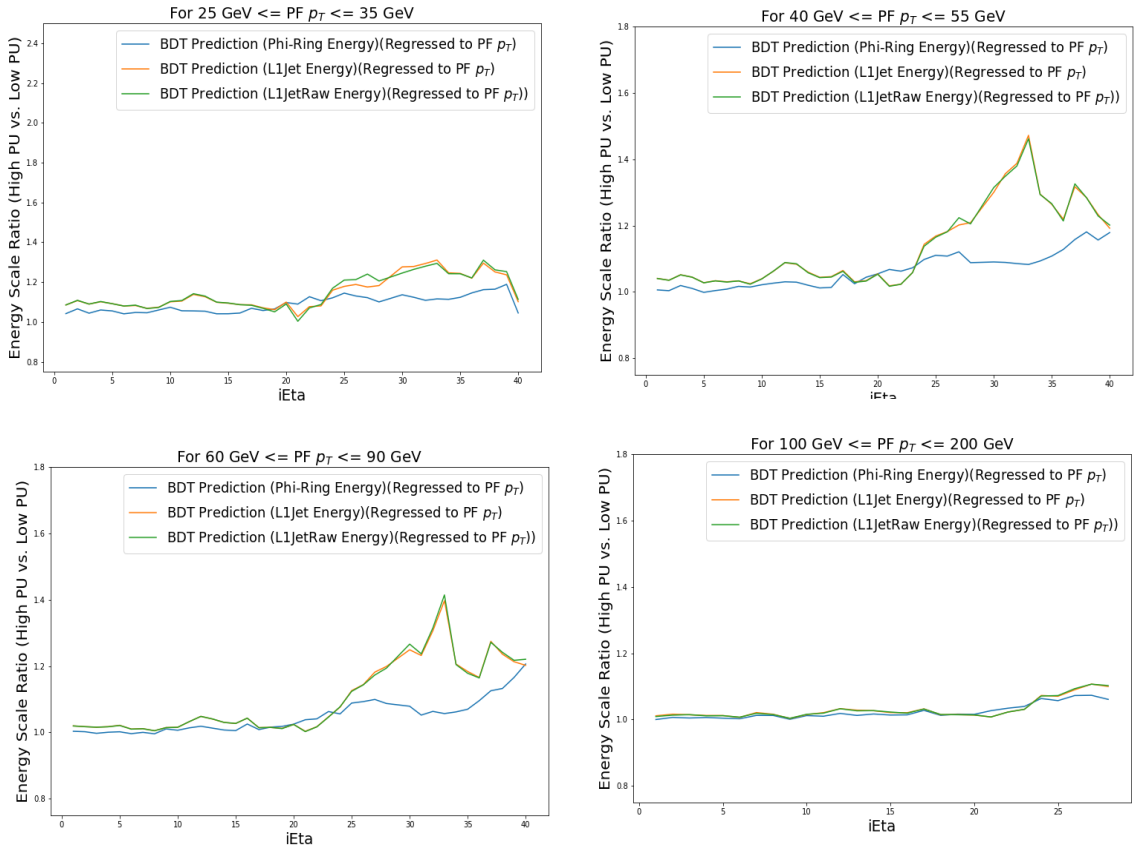

Figure 4.20: BDT energy scale ratios (high vs. low PU) for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right)

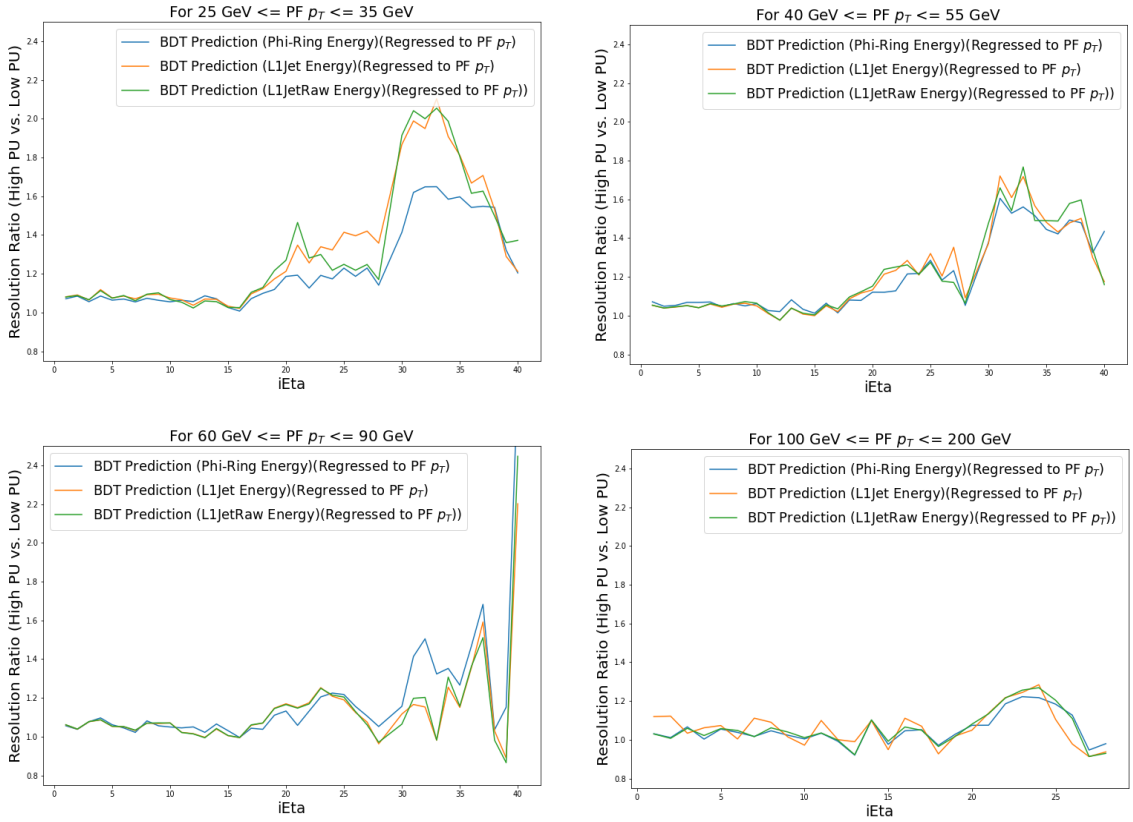

Figure 4.21: BDT resolution ratios (high vs. low PU for PF jets with p_T: 25–35 GeV (top left), 40–55 GeV (top right), 60–90 GeV (bottom left) and 100–200 GeV (bottom right)

4.2.1 Scale Factor Plots

After the training of each energy calibration BDT, we applied a pseudo test data set containing iEta values from 1 to 40 (except 29) with energy (chunky donut or phi–ring) values from 1 to 200 GeV for each iEta to obtain the scale factors of the trained BDTs associated with each iEta for a particular energy. After each set of 10 BDTs (each with a random 50% subset of events) is trained, we took the values of mean scale factors and plotted them against their input energy for each iEta. We obtained a total of 120 scale factor plots, 40 iEtas for each set of BDTs. Here we present 4 from each set for different regions of HCAL. These plots are for iEta=1 (Barrel), iEta=20 (Endcap 1), iEta=28 (Endcap 2) and iEta=33 (Forward).

These scale factors can be encoded in look–up tables (LUTs) in the Layer–2 firmware to perform online energy calibration in the Level–1 Trigger. This allows us to achieve high granularity value (separate scale factors for each iEta and p_T) with high precision, as the BDT output is smooth across similar regions of iEtas for all p_Ts. This feat cannot be obtained with analytic fits which require division or 'binning' of iEta and p_T regions by hand.

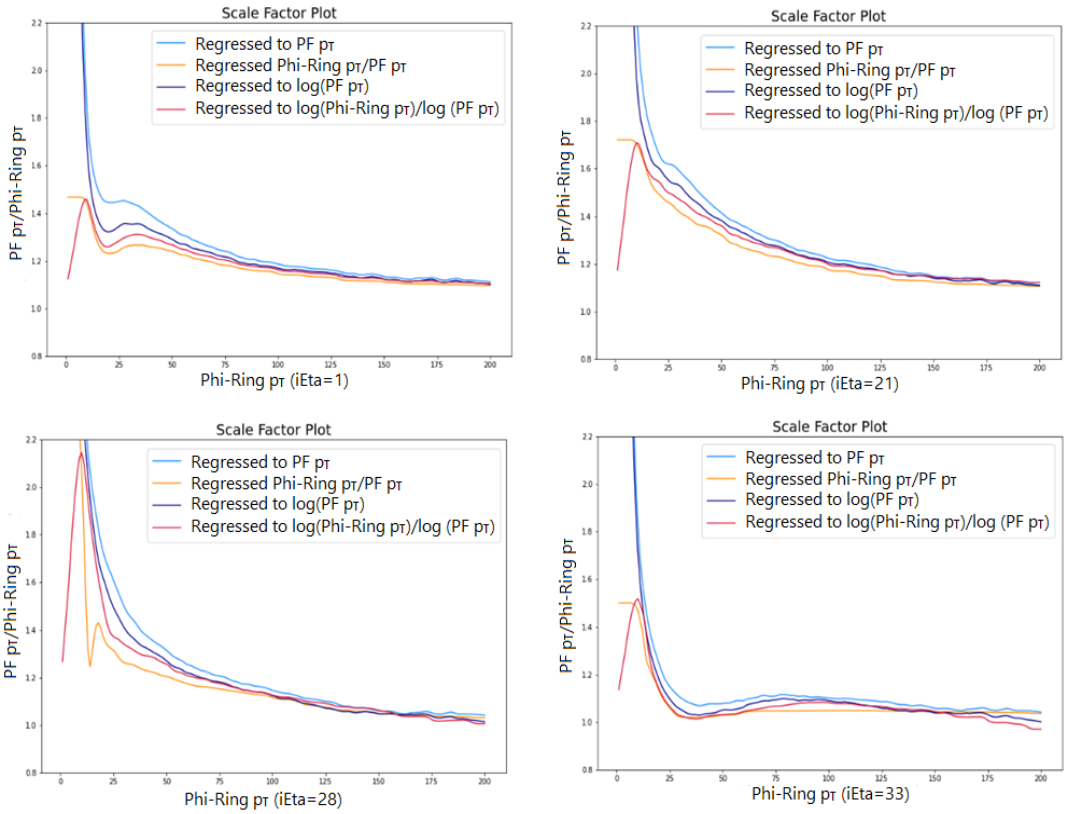

Figure 4.22: Scale factors of the Phi–Ring BDTs for iEta=1 (top left), iEta=20 (top right), iEta=28 (bottom left) and iEta=33 (bottom right)

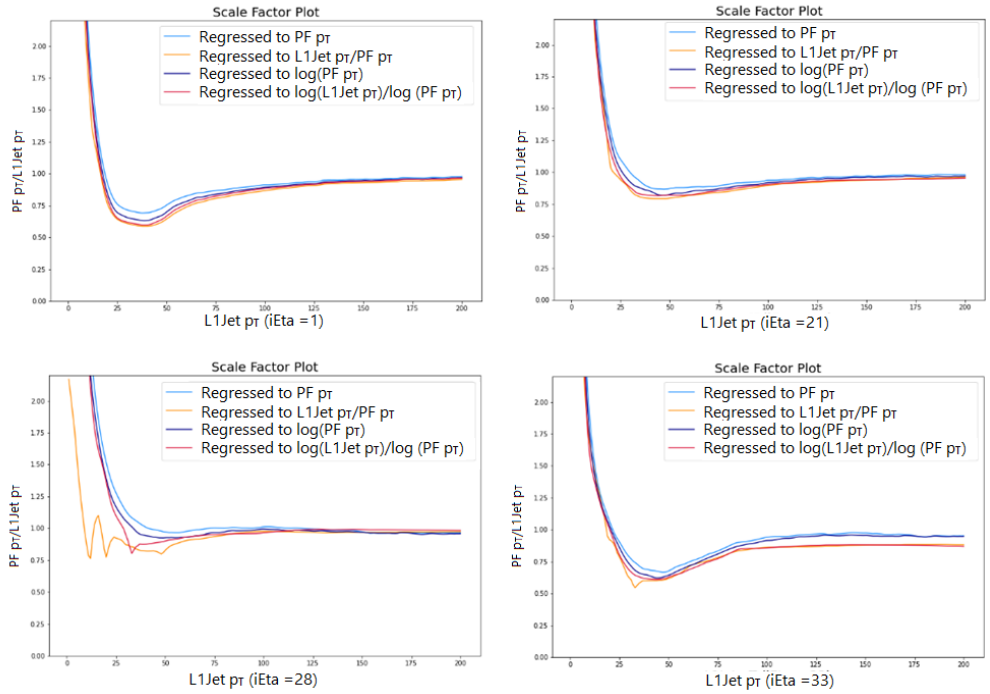

Figure 4.23: Scale factors of the L1Jet BDTs for iEta=1 (top left), iEta=20 (top right), iEta=28 (bottom left) and iEta=33 (bottom right)

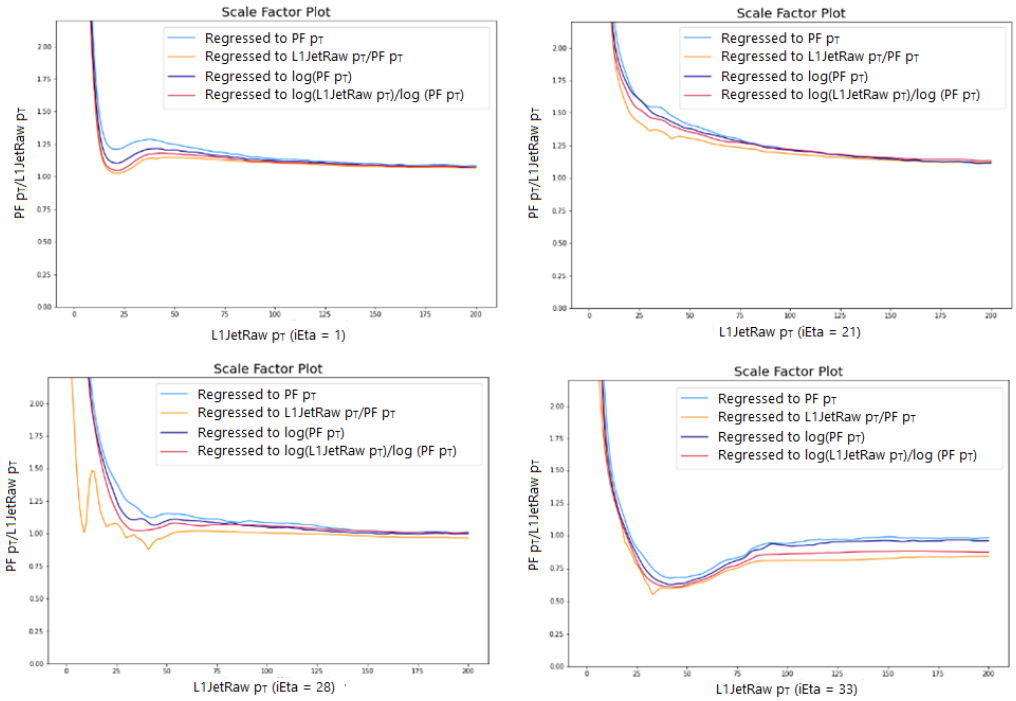

Figure 4.24: Scale factors of the L1JetRaw BDTs for iEta=1 (top left), iEta=20 (top right), iEta=28 (bottom left) and iEta=33 (bottom right)

CHAPTER FIVE

Conclusions

A systematic investigation using the machine learning technique of boosted decision trees (BDTs) on the 2018 single muon dataset from CMS to improve the reconstruction of hadron jets in the Level–1 Trigger (L1T) is presented in this work. At first we implemented a BDT with 12 jet features to check if it could differentiate between high and low PU, and also to perform a variable ranking on the inputs. Our results indicate that the BDT output achieves much better stability in energy scale compared to the energy scales obtained from the current (Run 2) jet detection algorithm of L1T (chunky donut), without any deterioration in resolution. From the variable ranking plots, we observed that the BDT prefers the energies from fewer trigger tower rings in η for jets that are further forward. The data also indicate that the phi–ring algorithm (which is based on trigger tower rings) should perform better than the Run 2 algorithm. We further trained energy calibration BDTs by providing energies of phi–ring and chunky donut algorithms (with and without Layer–2 calibration) separately to compare their performance with one another. Our results indicate that the BDT with phi–ring energy as input performs better in terms of energy scale vs. pileup and η than chunky donut, with similar resolution. With the BDT approach we can obtain smooth energy scale factors, which may be used to calibrate jets online with high granularity in Run 3. Our study can further be strengthened by performing similar investigations with Run 3 Monte–Carlo simulation events.

BIBLIOGRAPHY

[1] *The Cambridge Companion to Ancient Greek and Roman Science* (2020).

[2] D. Riepe, *The Naturalistic Tradition in Indian Thought*, Journal of the American Oriental Society **81**, (1961).

[3] L.K. Nash, *The Origin of Dalton's Chemical Atomic Theory*,Isis **47**, (1956).

[4] N. Zettili, *Quantum Mechanics: Concepts and Applications 2nd Ed.* (2009).

[5] D. Griffiths, *Introduction to Elementary Particles* (1987).

[6] A. Bettini, *Introduction to Elementary Particle Physics* (2008).

[7] A. Brinkerhoff and K. Lannon, *Observation of Top Quark Pairs Produced In Association With a W or Z Boson* (2015).

[8] T.A. Scarborough and K. Hatakeyama, *Studies on the Missing Energy Measurement and Supersymmetry Search in 7 TeV Proton–Proton Collisions* (2012).

[9] CMS Collaboration, *Precise determination of the mass of the Higgs boson and tests of compatibility of its couplings with the standard model predictions using proton collisions at 7 and 8 TeV*, European Physical Journal C **75**, (2015).

[10] CMS Collaboration, *Performance of the CMS Level–1 trigger in proton–proton collisions at* $\sqrt{s}$ *= 13 TeV*, Journal of Instrumentation **15**, (2020).

[11] S. Holmes, R.S. Moore, and V. Shiltsev, *Overview of the Tevatron collider complex: Goals, operations and performance*, Journal of Instrumentation **6**, (2011).

[12] en.wikipedia.org/wiki/List_of_Large_Hadron_Collider_experiments.

[13] D. Froidevaux and P. Sphicas, *General–purpose detectors for the Large Hadron Collider*, Annual Review of Nuclear and Particle Science **56**, (2006).

[14] CMS Collaboration, *The CMS experiment at the CERN LHC*, Journal of Instrumentation **3**, (2008).

[15] C.Y. Wong, *Introduction to High–Energy Heavy–Ion Collisions* (2011).

[16] https://cms.cern/detector.

[17] CMS Collaboration, *The CMS trigger system*, Journal of Instrumentation **12**, (2017).

[18] https://indico.cern.ch/event/1054320/contributions/4430616/attachments/2273295/3861234/ECAL_HCAL_geometry.pdf

[19] CMS Collaboration, *Search for natural and split supersymmetry in proton–proton collisions at √s=13 TeV in final states with jets and missing transverse momentum*, Journal of High Energy Physics **2018**, (2018).

[20] CMS Collaboration, *Measurements of properties of the Higgs boson decaying to a W boson pair in pp collisions at s=13TeV*, Physics Letters, Section B: Nuclear, Elementary Particle and High–Energy Physics **791**, (2019).

[21] CMS Collaboration, *Search for new physics with jets and missing transverse momentum in pp collisions at √s = 7 TeV*, Journal of High Energy Physics **2011**, (2011).

[22] V. Milesovic, *Search for Invisible Decays of the Higgs Boson at √ s = 13 TeV* (2020).

[23] M. Migliorini, R. Castellotti, L. Canali, and M. Zanetti, Computing and Software for Big Science **4**, (2020).

[24] M. Kubat, *An Introduction to Machine Learning* (2017).

[25] A. Shrestha and A. Mahmood, *Review of deep learning algorithms and architectures*, IEEE Access **7**, (2019).

[26] S.B. Kotsiantis, *Decision trees: A recent overview*, Artificial Intelligence Review **39**, (2013).

[27] Y. Himeur, A. Alsalemi, F. Bensaali, and A. Amira, *Robust event–based non–intrusive appliance recognition using multi–scale wavelet packet tree and ensemble bagging tree*, Applied Energy **267**, (2020).

[28] A. Shahraki, M. Abbasi, and Ø. Haugen, *Boosting algorithms for network intrusion detection: A comparative evaluation of Real AdaBoost, Gentle AdaBoost and Modest AdaBoost*, Engineering Applications of Artificial Intelligence **94**, (2020).

[29] T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, in Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2016).

[30] A. Kadiyala and A. Kumar, *Applications of python to evaluate the performance of decision tree–based boosting algorithms*, Environmental Progress and Sustainable Energy **37**, (2018).

[31] J. Ye, J.H. Chow, J. Chen, and Z. Zheng, *Stochastic gradient boosted distributed decision trees*, in *International Conference on Information and Knowledge Management, Proceedings* (2009).