Sofia University
Faculty of Physics

A Thesis submitted for the degree of
**Master in 'Nuclear and Particle Physics'**

# Machine Learning approach to CMS RPC HV scan data analysis

*Mihaela Pehlivanova*

*Supervisor: Assoc. prof. Peicho Petkov*

March, 2025
Sofia

# Contents

# Introduction

To explore the fundamental components of matter and the laws governing their interactions, large research infrastructures have been established.

The European Organization for Nuclear Research, known as CERN, is a world-leading research center where scientists investigate the fundamental laws of matter. The Large Hadron Collider (LHC) is the most powerful particle accelerator ever built, with a center-of-mass energy of nearly 14 TeV. Inside the LHC, two beams of protons collide at four interaction points, where large detectors are placed to study the results. One of these detectors is the Compact Muon Solenoid (CMS), a general-purpose experiment designed to explore the Standard Model and search for new physics.

At the CMS interaction point, proton bunches collide every 25 ns, with each bunch containing approximately $1.5 \times 10^{11}$ protons. On average, about 20 effective collisions occur per bunch crossing, leading to around 600,000 collisions per second. These collisions create new particles, which decay into 'stable' particles such as photons, electrons, hadrons, neutrinos, and muons, which can then be detected.

Muons, being minimal ionizing particles, pass through most of the detector and reach its outermost layers. To detect them, CMS has a dedicated muon system consisting of four types of gaseous detectors: Resistive Plate Chambers (RPCs), Cathode Strip Chambers (CSCs), Drift Tubes (DTs), and Gas Electron Multipliers (GEMs). These detectors are located in different regions of the muon system, either in the barrel or in the endcap.

RPCs, which are present in both the barrel and endcap, play a crucial role in the trigger system due to their high time resolution. This allows them to accurately assign muons to the correct bunch crossing. Their performance depends on the applied high voltage between their resistive electrodes. To ensure optimal operation, a procedure called a high-voltage (HV) scan is performed at the beginning of each year of data collection. The HV scan measures the efficiency and cluster size of the detectors over a range of voltages, and after the data is analyzed, the

correct working points are set in the system.

For years, parts of this analysis have been carried out manually, making the process time-consuming. The goal of this Master's thesis is to automate the HV scan calibration procedure for CMS RPCs using machine learning.

The Master thesis is structured into five chapters. It begins with an introduction to the LHC and CMS. Next, it covers RPCs and their role in muon detection. Key theoretical concepts, including autoencoders and the Discrete Cosine Transform, are then introduced. Finally, the thesis presents the automation of CMS RPC high-voltage scan analysis using machine learning, detailing the implemented solution and developed software.

# 1 The Large Hadron Collider and the Compact Muon Solenoid

## 1.1 LHC

In 1932, Cockcroft and Walton built the first particle accelerator and achieved the first splitting of a lithium nucleus using a 400 keV proton beam [1]. Their device operated on the principle that the energy gained by a particle in an electrostatic electric field is determined by the applied voltage and the charge of the particle, expressed as:

$$\Delta E = q\Delta U \tag{1}$$

It was a simple voltage multiplier, consisting of diodes and capacitors, that converted an AC voltage into a DC voltage. The terminal voltage achieved was a multiple of the applied voltage. This type of device was used for many years at CERN and is now displayed outside Building 143 (Fig. 1).

Figure 1: Cockcroft-Walton generator (600 kV) built by Philips, used in 1964 as a high-voltage supply for the 3 MeV Linac pre-injector to the proton synclotron Image by CERN.

This groundbreaking achievement marked the beginning of the development of increasingly powerful particle accelerators. In 2000, the construction of the LHC began, which remains the most powerful accelerator on Earth, with a center-of-mass energy of 14 TeV.

The LHC has a circumference of 27 km and surrounds an area of approximately 60 km$^2$ (Fig. 2 shows how LHC would fit within Sofia).

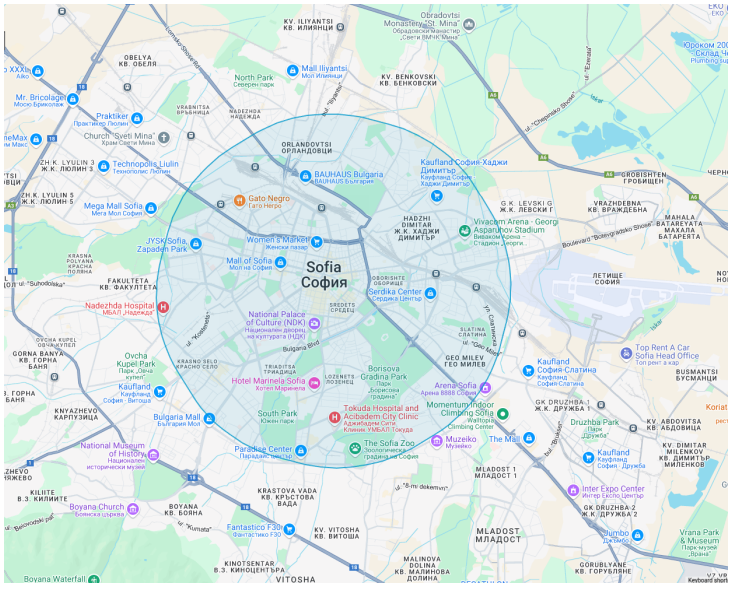



Figure 2: A visualization showing how the LHC would span across Sofia, based on the interactive map tool [2].

It is designed to collide head-on two beams of particles, either proton-proton or lead-lead. The dipole magnets that guide the protons in their orbit and the radiofrequency cavities that accelerate them are among the most powerful on Earth, allowing each proton to reach an energy of 7 TeV. Compared to fixed-target experiments, head-on collisions provide much higher energies in the center-of-mass frame, given by the formula $\sqrt{s} = 2\sqrt{E_1 E_2}$. This allows access to a wider range of masses for newly produced particles. In fixed-target collisions, the formula becomes $\sqrt{s} = 2\sqrt{E_1 m_2}$, where $E_1$ is the total energy of the moving particle and $m_2$ is the rest mass of the target particle. Achieving such high energies would require an extremely high $E_1$, which is technically very challenging.

Except for the type of particles being accelerated and the energy in the center-of-mass frame, another key characteristic of an accelerator is the instantaneous luminosity or just luminosity, which corresponds to the number of interactions between particles in crossing bunches. The average number of interactions per bunch crossing detected by an experiment is referred to as a 'pileup'. Luminosity is measured in $\mathrm{cm}^{-2}\mathrm{s}^{-1}$ and indicates the rate at which interactions are produced:

$$\frac{dN}{dt} = \mathcal{L}_{inst}\sigma \tag{2}$$

where $\frac{dN}{dt}$ represents the number of interactions per second, and $\sigma$ is the cross section for a specific process occurring during the interaction of two colliding particles. For example, knowing the cross section for Standard Model Higgs boson production at $\sqrt{s} = 13.6\,\mathrm{TeV}$, Eq. 2 can be used to calculate the expected number of Higgs bosons produced per second. For the LHC, the instantaneous luminosity for proton-proton colliding bunches is $10^{34}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$, while for lead-lead interactions, it is $10^{27}\,\mathrm{cm}^{-2}\,\mathrm{s}^{-1}$. Integrating instantaneous luminosity over time (Eq. 3) yields the integrated luminosity which corresponds to the amount of data delivered by the accelerator or the total number of interactions for some period of time. Integrated luminosity is measured in $\mathrm{fb}^{-1}$, where $1\,\mathrm{b} = 10^{-24}\,\mathrm{cm}^2$. The term 'barn' originates from the Manhattan Project during World War II, where the exact value of 1 barn was classified information, as it corresponded to the effective cross-section of a uranium nucleus for neutron-induced fission.

$$L = \int \mathcal{L}_{inst} dt \tag{3}$$

Since protons and heavy ions are significantly accelerated before reaching the LHC, it is part of the CERN's accelerator complex, as shown on Fig. 3. Both proton and heavy-ion beams are accelerated by the Proton Synchrotron (PS) and then by the Super Proton Synchrotron (SPS) before being fed into the LHC. The difference lies in the initial stages: protons are first accelerated by LINAC4 and then by the Proton Synchrotron Booster (PSB), which injects protons into the PS at an energy of about 1.4 GeV. In contrast, lead ions are initially accelerated in LINAC3 and then in the Low Energy Ion Ring (LEIR) before reaching the PS with an energy

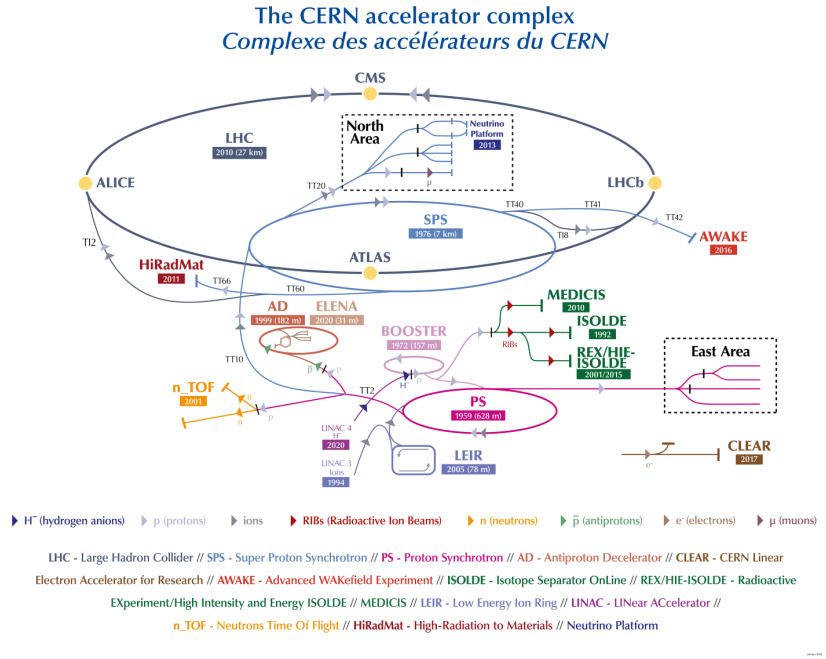



Figure 3

of 72 MeV per nucleon.

For accelerating the particles, the LHC uses a set of 8 radiofrequency (RF) cavities per beam (Fig. 4), each delivering 2 MV across a 0.4 m gap, so in total, a proton gains 16 MeV per revolution. Given that a proton makes 11,245 revolutions per second and needs to acquire 6.55 TeV of energy, in theory, it should take about 36.4 seconds for the LHC to accelerate the protons. In practice, however, it takes about 20 minutes because the protons are not fully affected by the total accelerating voltage of the cavities.

For deflecting the beams, the LHC uses dipole magnets, each 15 meters long, which generate a magnetic field of approximately 8.33 T, driven by a current of nearly 12,000 A. Using Eqs. 4 and 5, the deflection angle $\phi$ of the particle generated by each dipole magnet can be calculated, as shown in Eq. 6:

$$r = \frac{p}{qB} \qquad (4) \qquad\qquad \phi = \frac{L}{r} \qquad (5)$$

$$\phi = \frac{qLB}{p} \qquad (6)$$

where $r$ is the bending radius, $B$ is the magnitude of the magnetic field, $p$ is the momentum of the particle, $L$ is the length of the magnet, and $q$ is the elementary charge. By substituting $B = 8.33\,\mathrm{T}$, $L = 15\,\mathrm{m}$, $p = 7\,\mathrm{TeV}$, and $q = 1.6 \times 10^{-19}\,\mathrm{C}$ in Eq. 6, we get for $\phi$ an angle of $0.29°$ or $0.0051\,\mathrm{rad}$. To achieve a full $2\pi$ deflection of the particles, approximately $\frac{2\pi}{0.0051} \approx 1232$ dipole magnets are required, which is the actual number of dipole magnets used in the LHC. The trajectories of the particles passing through all deflectors are closed, i.e circles, if their velocity vector lies entirely in the horizontal plane. If their velocity has a $y$ component, their trajectories will not be stable, i.e. they will have a spiral-like motion that will not close. This means that the beam will become more and more spread out every time it passes through a deflector.

Figure 4: Four cavities make up one cryomodule. In total there are four cryomodules in the LHC, two per beam.

To prevent the beam from diverging, quadrupole magnets have been installed

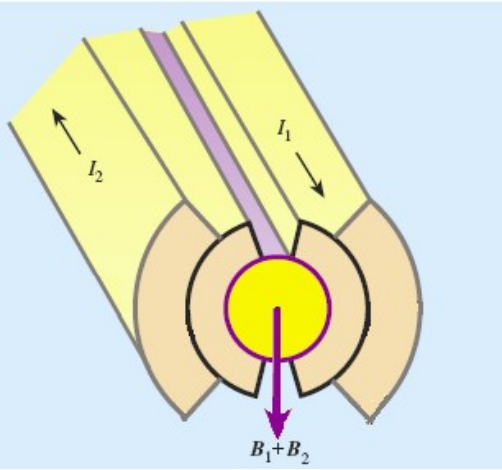



Figure 5: Scheme of a dipole magnet in the LHC. The current flows in opposite directions, creating a magnetic field vector that is perpendicular to the beam pipe. Depending on their velocity vector, the particles are deflected either to the left or to the right.

in the LHC. As can be seen in Fig. 6, the right one focuses the beam horizontally and the left one vertically.

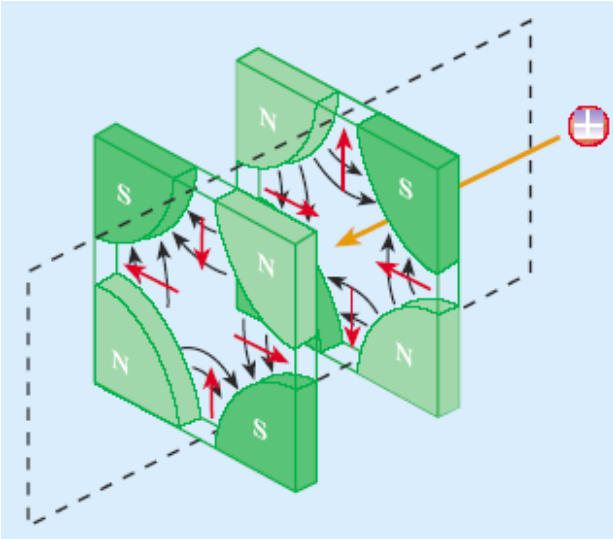



Figure 6: A scheme of a quadrupole magnets.

Quadrupoles are always in pairs because focusing the beam in one plane causes

it to diverge in the other. They are often arranged in a FODO patterns, where "F" stands for focusing vertically and defocusing horizontally, "D" is for focusing horizontally and defocusing vertically, and "O" refers to a non-focusing section, such as a deflecting magnet or drift space. An example of a FODO cell is shown in Fig. 7.

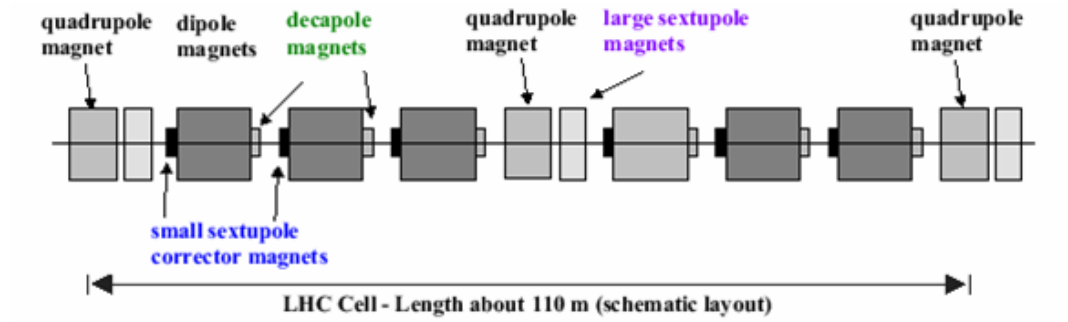



Figure 7: An example of a FODO cell. There are also multipole magnets which further focus the beams and also counteract effects such as electromagnetic interactions between bunches.

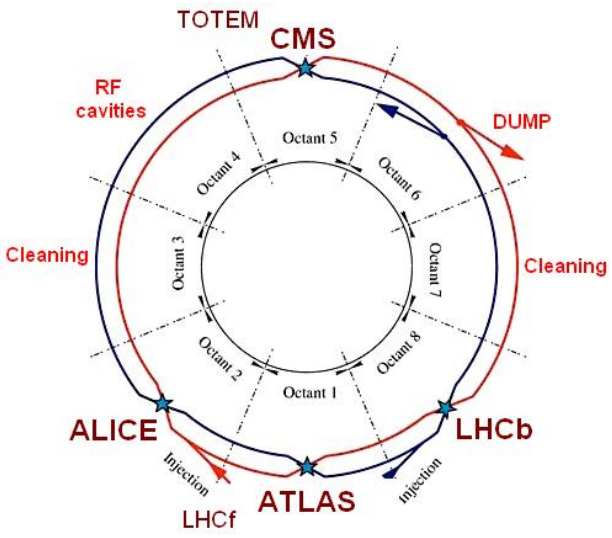



Figure 8

Since the LHC is not a perfect circle, it is made up of 8 arcs, each $2.45\,km$ long, and 8 straight sectors, each $545\,m$ in length. One arc consists of 23 FODO cells, and between each arc and the straight sector are the transition regions, also known as dispersion suppressors. The LHC is further divided into 8 octants (Fig: 8), where an octant begins at the middle of one arc and ends in the middle of the next. Each octant includes one straight sector along with its two transition regions. Those straight sectors serve specific purposes. Four of them are dedicated to key functions such

as acceleration, beam dumping, or beam cleaning, while the remaining four are located at the so-called interaction points, where the LHC's four main experiments take place.

The four experiments at the Large Hadron Collider (LHC) are ALICE, LHCb, ATLAS, and CMS. The purpose of ALICE is to investigate the state of the early universe, which is believed to be quark-gluon plasma, and to gain insights into quark confinement. LHCb is designed to investigate CP violation or explain why there is more matter than antimatter in the universe. ATLAS and CMS are general-purpose detectors that explore a wide range of topics, from the Standard Model (SM) of particle physics to Beyond the Standard Model (BSM) phenomena, including dark matter and supersymmetric theories.

## 1.2 Compact Muon Solenoid (CMS)

As mentioned earlier CMS is one of the general purpose detectors at the LHC. To increase the probability of detecting all long living particles and measure their properties with good precision, CMS has a $4\pi$ geometry and is structured in layers. The system nearest to the interaction point, and therefore the one exposed to the highest levels of radiation, is the silicon tracker. This tracker is divided into two subsystems: the pixel tracker (PIXEL) and the strip tracker (SiStrip). Its primary function is to reconstruct the tracks of charged particles and accurately measure their curvature in the magnetic field, allowing estimation of their charge and momentum (Eq. 4).

The next system is the Electromagnetic Calorimeter (ECAL), which detects and measures the energy deposited by electrons and photons. The cylindrical barrel section contains 60,000 lead tungstate scintillating crystals, while the closing parts, known as the endcap, include an additional 15,000 crystals. The third layer is the Hadronic Calorimeter (HCAL), which measures the energy of hadrons. Like the ECAL, the HCAL is also divided into barrel and endcap sections, consisting of layers of brass or steel interleaved with plastic scintillators.

The trackers and calorimeters are housed within a solenoid magnet that serves as the backbone of the experiment. This magnet is the central component around which the entire experiment is built, and it also provides most of the structural support. It generates a magnetic field of nearly $4\,T$ that causes charged particles to curve their trajectories. This, combined with high-precision measurements of the particles' positions, allows for a precise measurement of their momentum. The magnet comprises cylindrical coils made of niobium-titanium (NbTi) that are covered with copper, through which a current of $20\,kA$ flows. To produce such a strong magnetic field, not only a significant current is required, along with extremely low temperatures to minimize resistivity. The magnet is cooled to $-268.5°$, only one degree warmer than the temperature of outer space.

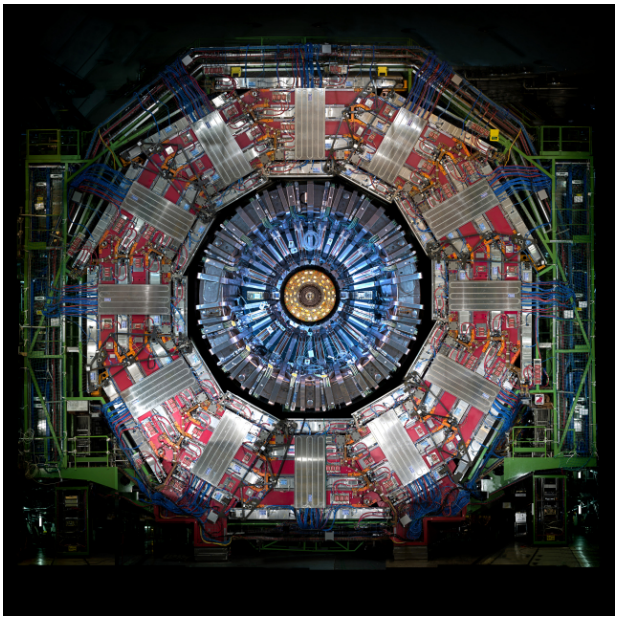


Figure 9: CMS barrel. The outermost red layer is the muon system interleaved with the iron return yoke. Calorimeters are the blue cylinders hosted by the solenoid, and in yellow is the tracking system around the beam pipe. Image by CERN.

The magnet coils are surrounded by an iron structure that not only contains and guides the magnetic field but also interleaved with the detectors of the muon system. To track muons, which along with neutrinos are the only particles that reach beyond the magnet, four types of gaseous detectors have been installed in the outermost layers of the experiment. Drift Tubes (DTs) and Resistive Plate Chambers (RPCs) are found in the barrel, covering the low pseudorapidity region. In contrast, Cathode Strip Chambers paired with RPCs cover the high pseudorapidity region of the barrel. During Long Shutdown 2 (LS2), which took place from 2018 to 2022, additional types of detectors were installed to cover the pseudorapidity range of

$1.55 < \eta < 2.18$. A total of 144 Gas Electron Multipliers (GEMs) were installed at the GE1/1 detector station. Two more GEM stations, GE2/1 and ME0, are expected to be installed during LS3 with ME0 extending the pseudorapity coverage up to $\eta = 2.8$ [3].

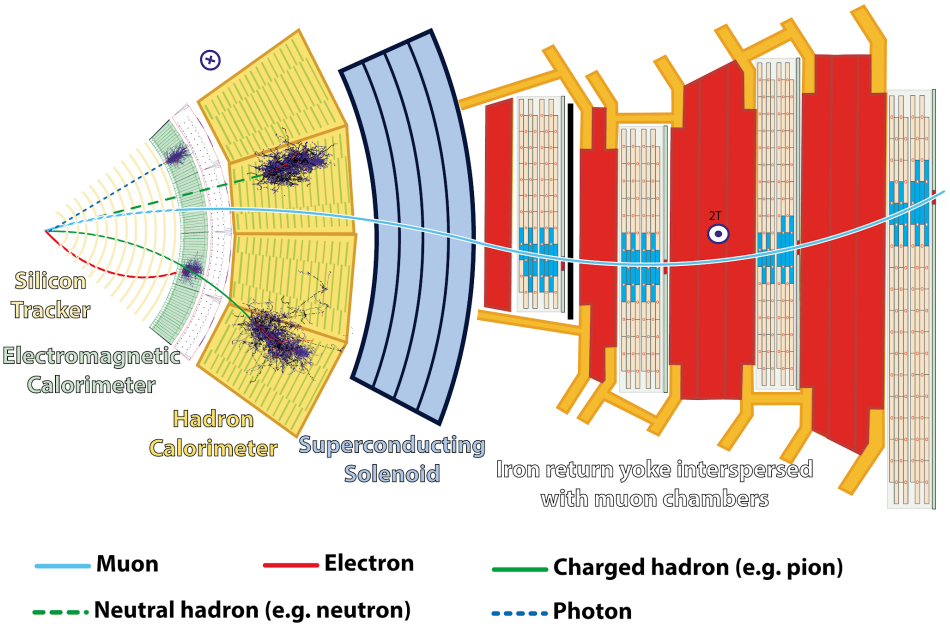



Figure 10: Slice showing CMS sub-detectors and how particles interact with them. Image by CERN.

# 2 Resistive Plate Chambers (RPCs)

Despite the good spatial resolution, the limited time resolution of DTs and CSCs introduces the need for another type of detector, the RPCs. These are also gaseous detectors, but instead of generating a signal through the collection of charge at an anode, as in wire chambers, the signal is induced in copper pickup strips located outside the gas volume. This results from the changing electric field caused by the movement of electrons and ions in the gas. The time resolution of an RPC is approximately $1\,ns$, while the time resolution for DTs and CSCs ranges from 2 to 3 $ns$. Although this difference may seem minor, it becomes significant

when considering the precision of the trigger system 12.5 $ns$ and the interval 25 $ns$ between the crossings of the bunches [5]. A detector with better time resolution can improve the assignment of detected muons to the correct bunch crossing.

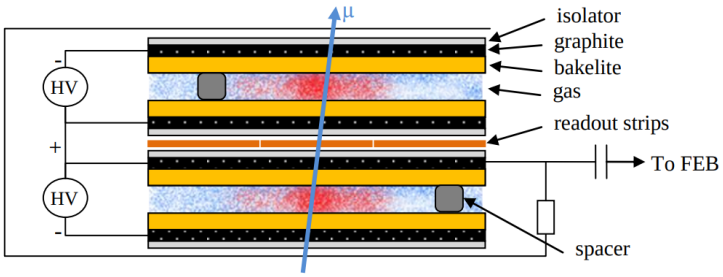



Figure 11: Cross-section of a double-gap RPC. [4]

A single gap RPC consists of two bakelite resistive plates with a gap filled with a gas mixture of 95.2% $C_2H_2F_4$ + 4.5% $iC_4H_{10}$ + 0.3% $SF_6$. However, in CMS, RPCs are designed as double-gap chambers (Fig. 11). There are 1,056 RPCs in the barrel and endcap regions of the detector, spanning a pseudorapidity range of $0 < \eta < 1.6$ [5]. In addition to the Bakelite plates and the gas gap, there are graphite layers to apply voltage and an insulating layer to prevent charge leakage.

Once a particle transverses the gas volume, it interacts with the gas atoms to produce electron-ion pairs (Fig. 12a), referred to as primary ionization. This primary ionization then initiates an avalanche of such pairs, with the positive ions moving toward the cathode and the electrons toward the anode. This avalanche starts to locally influence the electric field (Fig. 12b).

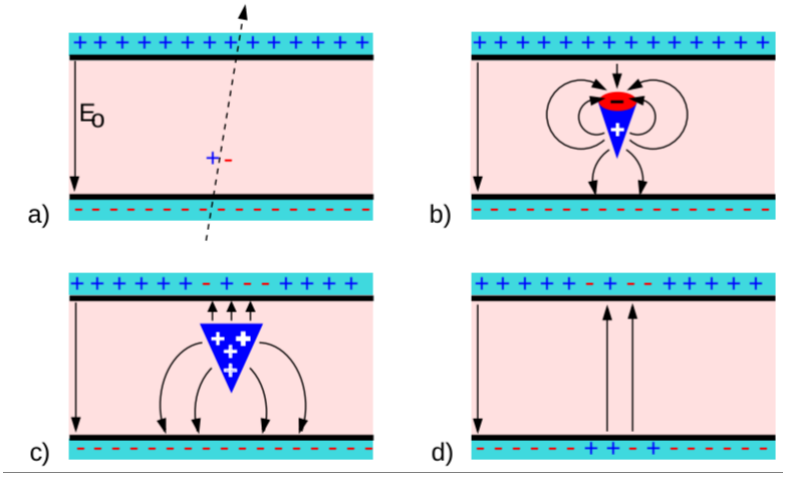



Figure 12: Different stages of avalanche development in an RPC where a constant field of magnitude $E_0$ is applied between the electrodes.

The electrons reach the anode first due to their higher drift velocity, followed by the ions reaching the cathode (Fig. 12c). While the charges do not recombine with the electrode material, the electric field remains locally affected, blinding the detector in this region (Fig. 12d). The time constant that determines how long the charge takes to recombine is a very important parameter of an RPC. By taking a quasi-static approximation of Maxwell's equations for weakly conducting media, we can compute the time needed for recombination at the interface between the gas and the electrode (Eq. 7). This can be seen more in depth in [6].

$$\tau_{RPC} = \frac{\epsilon_e + \epsilon_g}{\sigma_e + \sigma_g} \tag{7}$$

where $\epsilon_e$ and $\epsilon_g$ are the specific permittivity constants of the electrode and the gas, and $\sigma_e$ and $\sigma_g$ are their conductivity constants. Approximating $\epsilon_g = \epsilon_0$ and $\sigma_g = 0$, and using $\epsilon_r = \epsilon_e\epsilon_0$ and $\sigma_e = 1/\rho_e$, where $\rho_e$ is the resistivity of the electrode, we obtain:

$$\tau_{RPC} = (\epsilon_r + 1)\epsilon_0\rho_e \tag{8}$$

Equation 8 demonstrates that the time required for recombination is proportional to the material's resistivity. The resistivity of materials used in RPCs typically ranges from $10^9$ to $10^{12}\,\Omega cm$. While lower resistivity results in a longer recombi-

nation time, this choice helps avoid distortion of the electric field across the entire plates.

One important consideration is the width of the gas gap in the RPCs. Experiments have shown that wider gaps can improve the detection rate of RPCs, but they tend to decrease the time resolution. This led to the development of multi-gap RPCs. By incorporating multiple narrower gaps, it became possible to maintain excellent time resolution while simultaneously increasing the rate of detected events.

Another consideration is the strength of the applied electric field. Initially, RPCs operated in a streamer mode, which was convenient for readout since no additional amplification was required. However, Crotty demonstrated that a weaker electric field results in a weaker signal due to a smaller ion cloud and reduced electronic gain which helps the avalanche be more localized and hence increase the rate capability of the detector (Fig. 13).

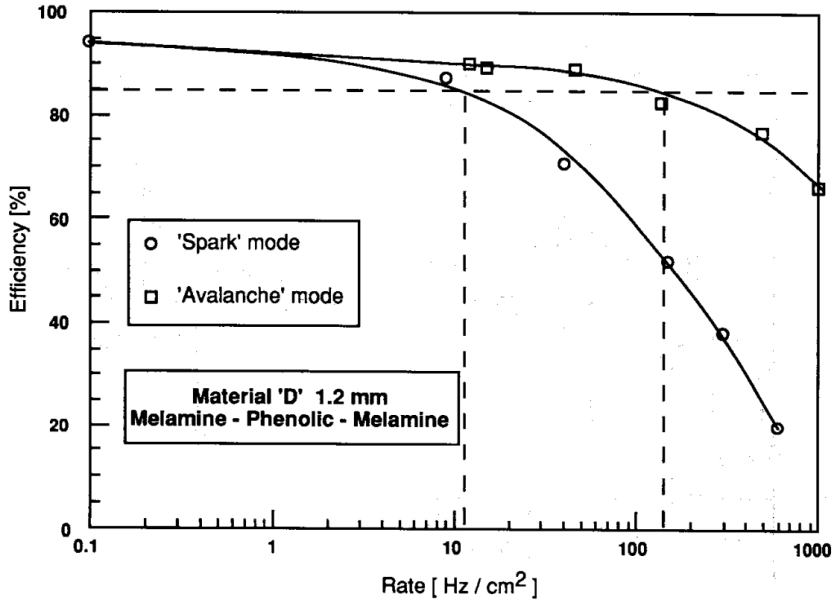



Figure 13: Comparison of rate compatibility for the two modes of work of an RPC. In avalanche mode, increasing the rate does not affect the efficiency of the chamber as much as in 'spark' mode [7].

The efficiency of the RPCs in CMS is defined as the ratio of number of tracks detected in the RPC divided by the number of extrapolated tracks identified in CSC and DT detectors, $\epsilon_{RPC} = n_{RPC}/n_{extrap}$ [8]. For defining the working point of an RPC an efficiency measurement as a function of the effictive voltage is conducted. The data can be fitted with a sigmoidal function described by Eq. 9:

$$\varepsilon = \frac{\varepsilon_{max}}{1 + e^{-slope_{50\%}(HV - HV_{50\%})}} \tag{9}$$

where $slope_{50\%}$ characterizes the steepness of the slope of the sigmoid, $\varepsilon_{max}$ represents the level of the plateau, and $HV_{50\%}$ denotes the voltage at which the efficiency reaches 50% of its maximum [8]. An example can be seen in Fig. 14.

CMS defines the working point $HV_{WP}$ as:

$$HV_{WP} = HV_{knee} + \begin{cases} 100\,\mathrm{V} & \text{barrel} \\ 120\,\mathrm{V} & \text{endcap} \end{cases} \tag{10}$$

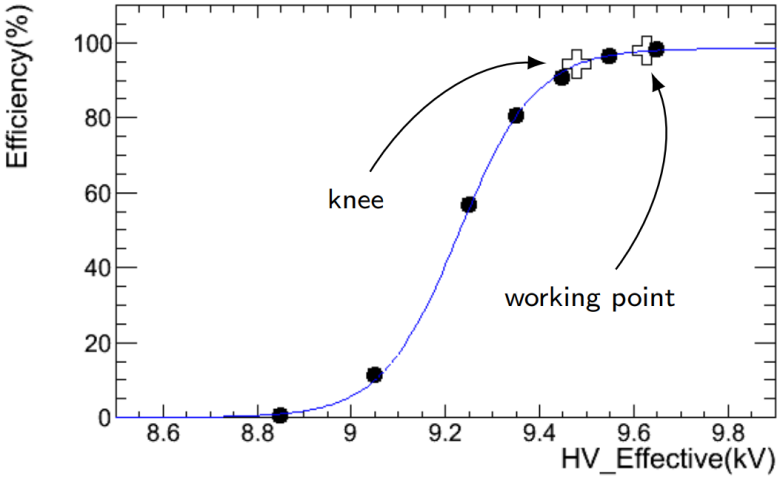



Figure 14: A typical efficiency measurement of CMS RPC. The black dots are the measured data points and the blue curve represents the fit to these data points. The cross marking the 'knee' ($HV_{knee}$) shows the voltage at which the efficiency reaches 95 % of its maximum value. This point is part of the fit parameters and is linked to the calculation of the working point [9] (Eq. 10).

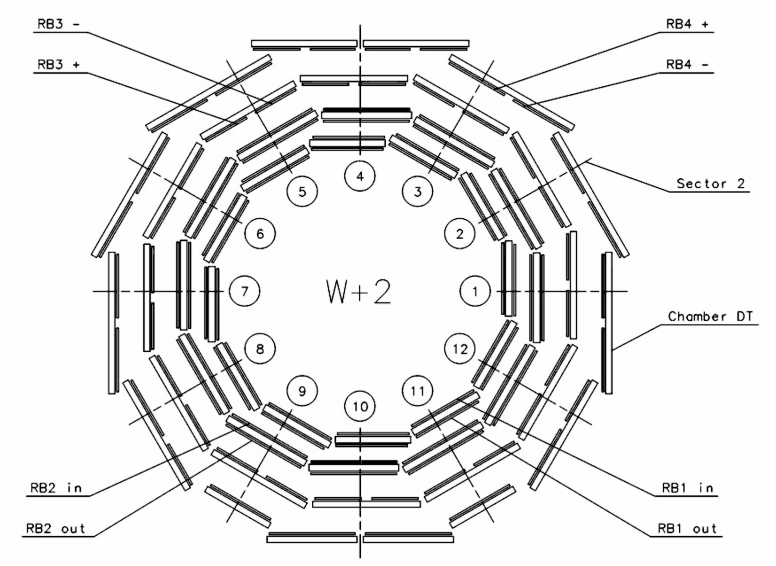



Figure 15: A schematic view of wheel $W + 2$ in the barrel of CMS. Image by CERN.

The effective voltage is defined with reference values for temperature at $T_0 = 293.15\,K$ and pressure at $P_0 = 965\,hPa$. Changes in environmental conditions can influence gas density and electrode resistivity. That is why a correction for both temperature and pressure is needed. A standard practice is to maintain the factor $\frac{HV \cdot T}{P}$ as constant by using the following formula:

$$HV_{app} = HV_{eff} \left(1 - \alpha + \alpha \frac{P_0}{P}\right) \frac{T}{T_0}, \tag{11}$$

where $\alpha = 0.8$ is experimentally obtained.

The barrel consists of 5 wheels, each containing 4 detector sections. Within each section, there are 12 sectors that provide full coverage over $2\pi$ radians. RPCs in the barrel are labeled as "RB$n \pm w$," where $n$ denotes the station number, and $\pm w$ indicates the wheel number. For stations 1 and 2, there are additional designations of "in" and "out" to differentiate the chambers within the same sector. There are two RPC chambers positioned on either side of a DT module, with "in" referring to the chamber that is closer to the beam pipe. Stations 3 and 4 use the

labels "-" and "+" to distinguish the RPCs that are installed side by side along a DT chamber (with some exceptions) where"-" indicates the chamber with the smaller $\phi$ value (Fig. 15).

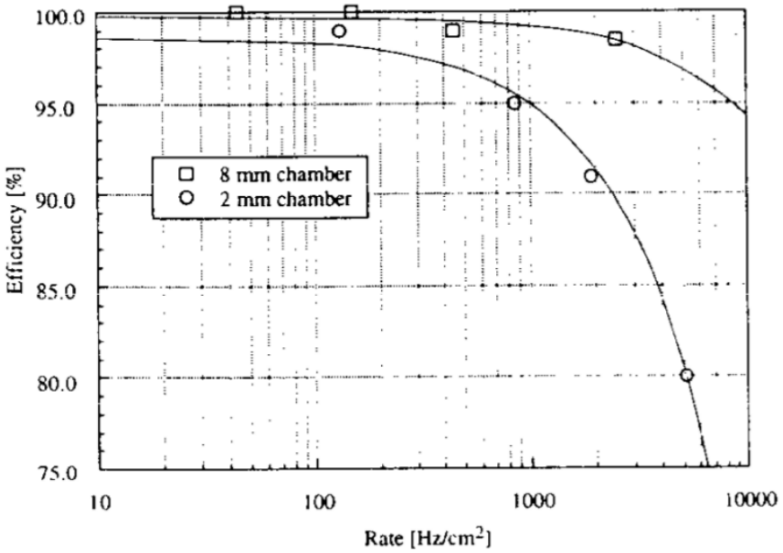



Figure 16: Comparison of the rate capabilities of an $8\,mm$ gas gap and a $2\,mm$ gas gap [10].

The endcap region consists of eight disks, four on each side of the detector. Each disk is composed of three rings, and each ring contains 36 trapezoidal RPC detectors. The detectors are labeled "RE$\pm\,n/r$", where $n$ indicates the disk number, and $r$ represents the ring number, increasing with the distance from the pipe. Note that the first ring has not been installed.

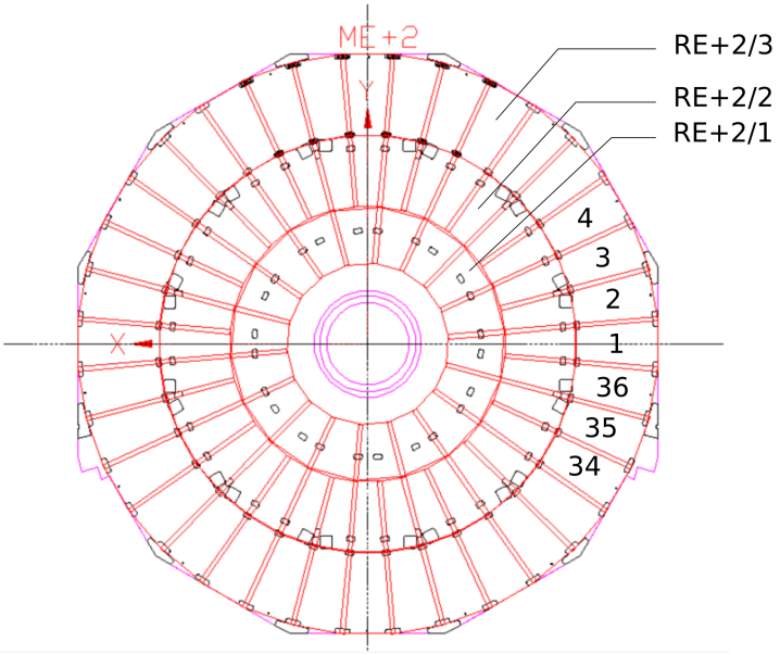



Figure 17: A schematic view of disk +2 in the endcap of CMS.

# 3 Automation of CMS RPC HV scan data analysis using ML

The aim of this master's thesis was to automate the analysis of high-voltage (HV) scan data for the RPCs in the muon system of CMS through the implementation of machine learning techniques. The HV scan is performed annually during data collection to measure the efficiency and cluster size of the chambers by varying the pressure-corrected voltage applied between the detector plates. The purpose of this procedure is to define the correct working points per roll and per channel. A roll refers to a double gap with a distinct $\eta$ partition and its own readout electronics within an RPC. In the barrel, a single chamber can contain 2 or 3 rolls, while in the endcap there are 6 different $\eta$ partitions. A typical result of such a measurement is illustrated in Fig. 14. The relationship between efficiency and HV demonstrates a distinct pattern characterized by a sigmoid function, which is defined by the formula Eq. 9. Formula 10 specifies the working point per roll, and

per channel, it is calculated as follows:

$$WP_{CH} = \begin{cases} \langle WP_{roll} \rangle & \text{if } WP_{roll}^{\text{Max}} - WP_{roll}^{\text{Min}} \leq 100 \text{ V} \\ \\ WP_{roll}^{\text{Min}} + 100 \text{ V} & \text{if } WP_{roll}^{\text{Max}} - WP_{roll}^{\text{Min}} > 100 \text{ V} \end{cases} \tag{12}$$

## 3.1 Autoencoders

An autoencoder is a type of unsupervised deep learning model that serves as a generalization of Principal Component Analysis (PCA). Its main objective is to learn a compressed representation of the original data by transforming it into a lower-dimensional space. This process helps to gain insights into the data.

A typical autoencoder consists of two functions, an encoder that maps the data to the latent (bottleneck) space by a function $g : \mathbb{R}^d \to \mathbb{R}^k$, where $k < d$, and a decoder that maps the data from the lower-dimensional space back to the original one, $h : \mathbb{R}^k \to \mathbb{R}^d$. A one-layer neural network (NN) encoder-decoder is depicted in Fig. 18. The $d$-dimensional input space is represented in a $k = 3$ latent space by the encoder function $g(x; W^1, W_0^1) = f_1({W^1}^T x + W_0^1)$. Here, $W^1 \in \mathbb{R}^{d \times k}$ represents the weights, $W_0^1 \in \mathbb{R}^k$ is the bias, and $f_1$ denotes the activation function, which applies nonlinearity to each dimension. To restore the representations from $\mathbb{R}^k$ to $\mathbb{R}^d$, a decoder defined as $h(x; W^2, W_0^2) = f_2({W^2}^T a + W_0^2)$ is used, where $f_2$ serves as the activation function. The weights and biases are adjusted through backpropagation to make the outputs $\tilde{x}_i$ similar to the inputs $x_i$ [11]. It is possible to perfectly reconstruct the input if, for example, $d = k$. In this scenario, the model does not compress the data at all. To gain valuable insights, a bottleneck should be incorporated into the model. This constraint forces the model to learn connections within the data, allowing for representation of the information in the simplest way possible.

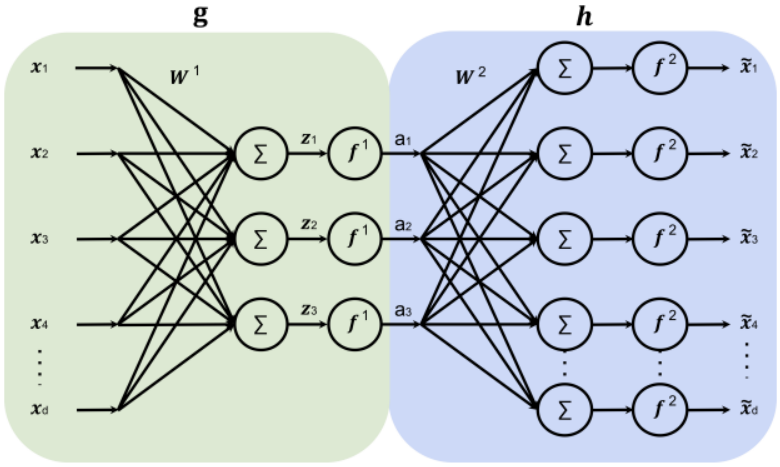



Figure 18: An autoencoder with one layer NN in the encoder $g$ (left) and the decoder $h$ (right). The input $x_i$ is represented in 3-dimensional space $(a_1, a_2, a_3)$, and is then reconstructed into the original space as $\tilde{x}_i$ [11].

## 3.2 Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is a transformation that converts real-valued data points into a sum of cosine functions. It is similar to another Fourier-related technique known as the Discrete Fourier Transform (DFT). However, instead of repeating the data periodically, the DCT mirrors it. This mirroring helps eliminate sharp discontinuities, which often lead to higher frequency oscillations. As a result, the higher frequencies in the transformed data tend to have very small amplitudes, often around zero. This makes for a perfect technique for data compression.

There are eight DCT variants, but only four are commonly used. Among these, two are particularly prevalent: DCT II (Eq. 13) and DCT III (Eq. 14) [12]. DCT II is the originally proposed variant and is often referred to simply as DCT. DCT

III, on the other hand, is the inverse DCT.

$$X_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right), \quad k = 0, 1, 2, \ldots, N-1 \tag{13}$$

$$X_k = \frac{1}{2}x_0 + \sum_{n=1}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n+\frac{1}{2}\right)k\right), \quad k = 0, 1, 2, \ldots, N-1 \tag{14}$$

## 3.3 ML solution

Until now, parts of the HV scan were done manually, which made the procedure time-consuming and caused delays in updating the working points of the chambers. One full analysis took about 3 months. One possible approach to make the analysis more efficient is to use machine learning. Various models were tested, but the best results came from a specialized autoencoder that operates in Fourier space. A brief description of some of the models that were tested can be found in Appendices A and B.

The chosen architecture of the model can be seen in Fig. 19. The model begins with an input layer that receives the measured efficiencies. The second layer performs Discrete Cosine Transform (DCT) on the data, and then feeds the output of this layer to the autoencoder. After processing the data through the decoder function, an inverse DCT is applied to revert the data back to its original space. In the end, 26 data points are produced. These 26 data points are fitted to a sigmoidal function (Eq. 9) to obtain the parameters of the fit needed for determining the new working points. Points of the cluster size data are excluded by comparing the predicted and measured efficiencies with three times the measured efficiency's error. The reason is that if an efficiency measurement is untrustworthy, a cluster size measurement for the same HV cannot be trusted either.

For training the model, synthetic data generated by sigmoidal functions was used, and random uniformly distributed noise was added to 4 of the 11 points. The generated data set consists of 100 000 data points. Training and testing were performed using Python's *TensorFlow* [13] and *Keras* [14] libraries. *LeakyReLU*

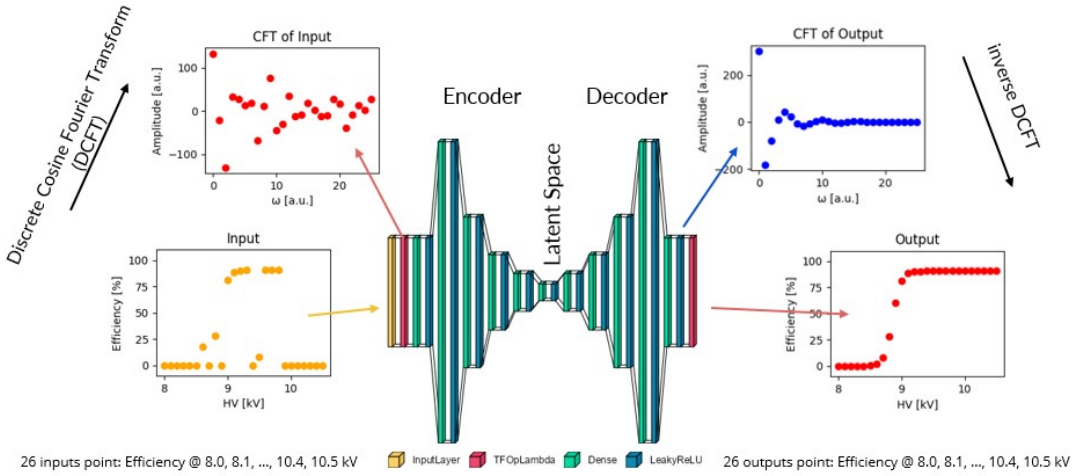



Figure 19: Architecture of the Fourier Space Autoencoder (FSAC). The inset images demonstrate how sample data transforms between the HV and reciprocal space domains.

was used to introduce non-linearity into the output of the layers, or the so-called activation function (Appendix C). The loss function was user defined and its form can be seen in Eq. 15. In the loss function, an $\alpha$ is defined as a small constant that controls the weight assigned to the mean squared error (MSE) of the derivatives between the prediction and the original data. The optimization algorithm used for minimizing the loss and for updating the weights was *Adams* (Appendix D). Results from applying the model to the 2024 RPC HV scan data are shown in the figures 20 to 24 .

$$l(w_i) = \frac{1}{n}\sum(y_{pred_{w_i}} - y_i)^2 + \alpha\frac{1}{n}\sum(dy_{pred_{w_i}} + dy_i)^2, \tag{15}$$

FSAC's strength lies in its ability to detect and manage anomalies in the data, preventing outliers and incorrect data points from skewing predictive modeling. In Fig. 20, we see an example concerning the Backward partition of the RB2in chamber (the second barrel station, with "in" indicating the chamber closer to

the detector's interaction point) in sector 1 of wheel +2. Here, the data follows a standard efficiency curve, and FSAC successfully predicts the correct efficiencies for different effective voltages. For the data point at a voltage of $8.8\,kV$, the prediction and measurement differ by more than three times the error of the measurement; therefore, this data point is excluded from the cluster size graphic.

In Fig. 21, we observe a case where the current flowing through the chamber was too high, resulting in the detector being switched off and measuring 0 efficiency at a voltage of $9.8\,kV$. If this point were not fixed or excluded, it could skew the data fit. As illustrated, FSAC is capable of predicting the correct efficiency even for data points where the measured efficiency is 0.

In Fig. 22, we see how FSAC performs in the presence of multiple anomalies in the dataset. The model effectively addresses the problem, predicting efficiencies that are comparable to those observed in a sigmoid function.

In Fig. 23, we present a case where the plateau is missing, making it difficult to calculate the correct working points without extrapolating the curve. In this instance, the model outperformed other models by predicting efficiencies in ranges even where data is missing. Finally, Fig. 24 illustrates how the model handles both the absence of a plateau and the presence of outliers by correctly predicting the efficiencies.

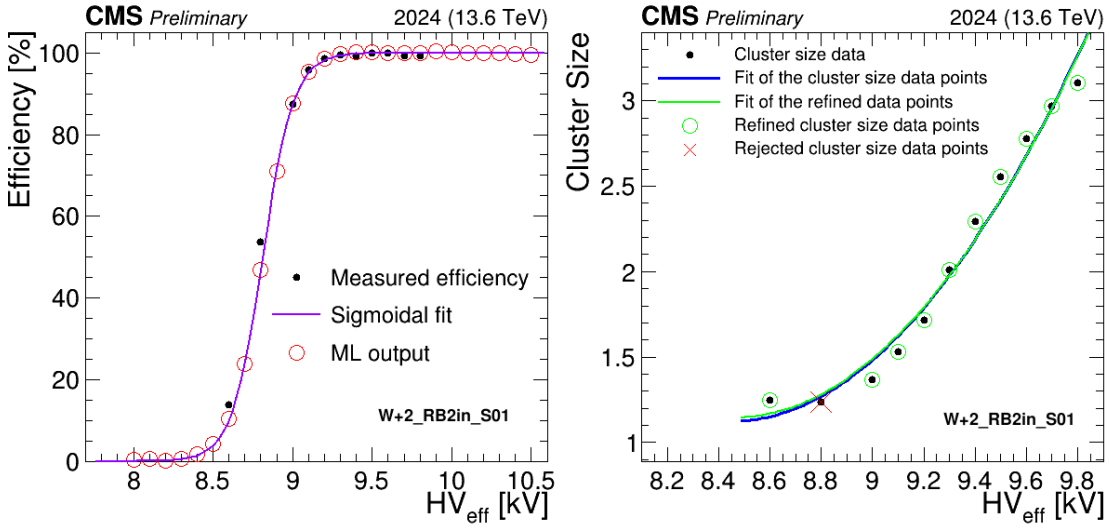



Figure 20: FSAC analysis on a typical efficiency curve: excluded 8.8 kV data point due to mismatch between predicted and measured efficiency value, which exceeds the error of the measurement [15].

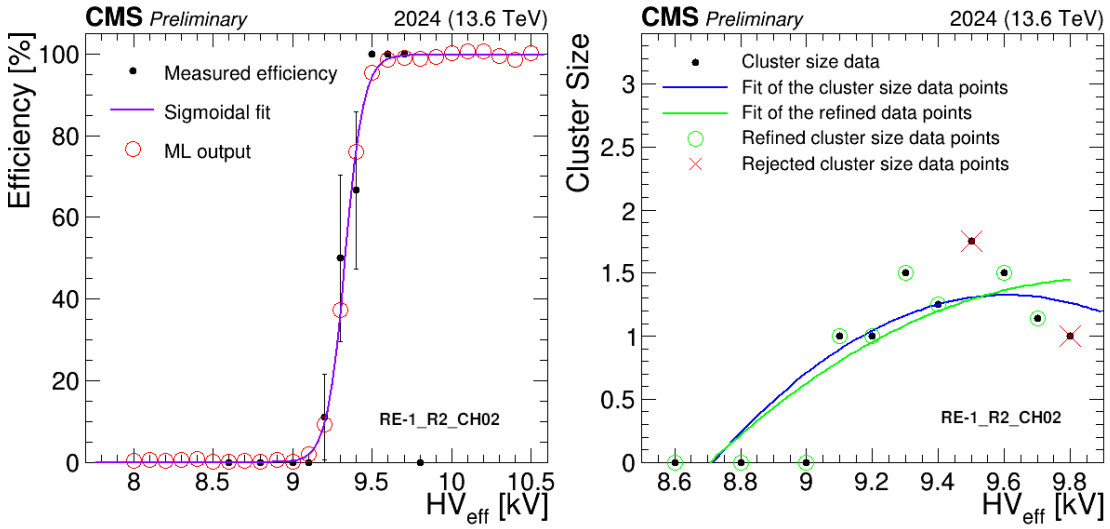



Figure 21: FSAC applied to a case with zero efficiency at last point: excluded 9.5 kV and 9.8 kV due to mismatch between predicted and measured efficiency values which exceeds the error of the measurement [15].

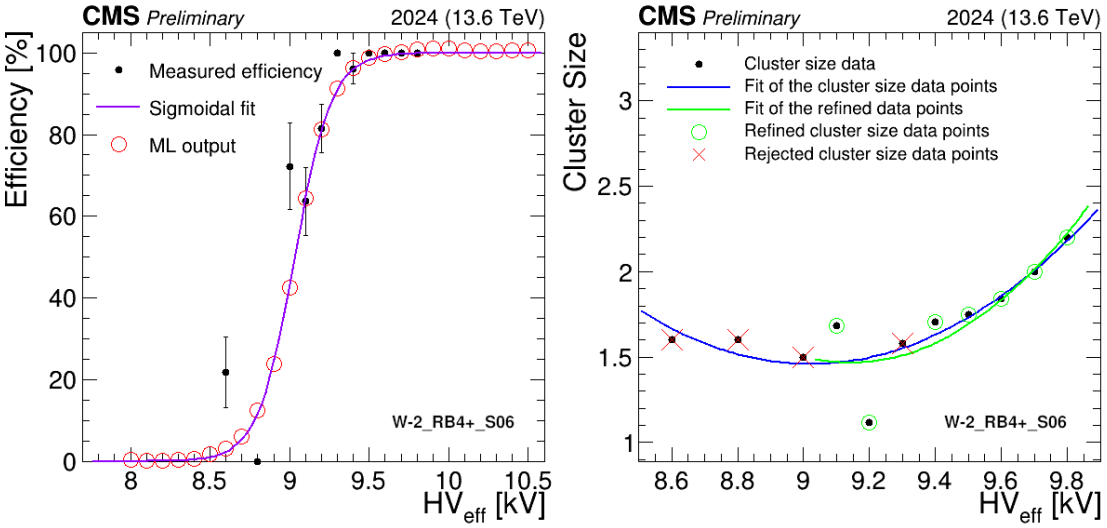



Figure 22: FSAC applied to a case where outliers are present in efficiency plot: excluded 8.6, 8.8, 9, and 9.3 kV due to mismatch between predicted and measured efficiency values which exceeds the error of the measurement [15].

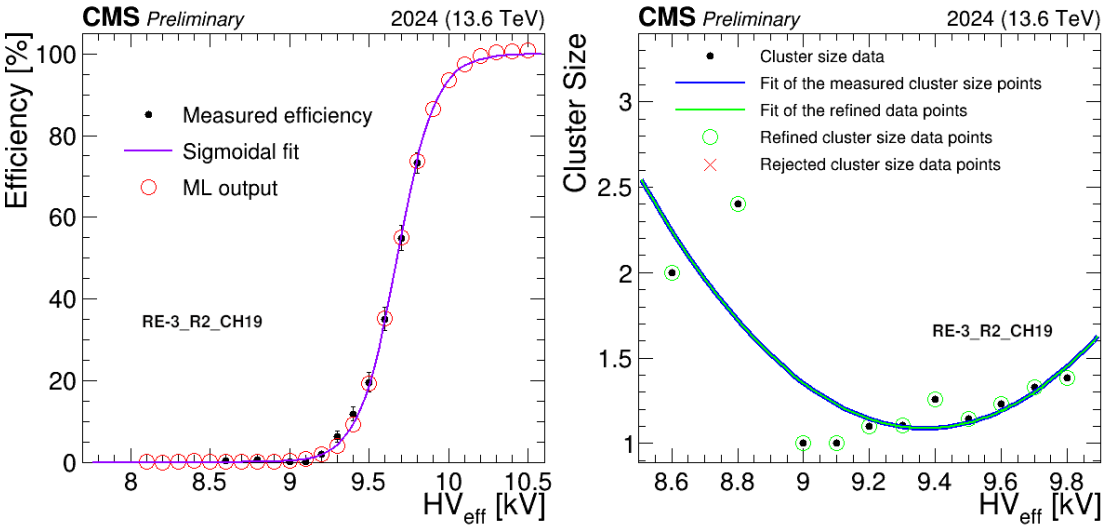



Figure 23: FSAC applied to a case where there is a missing plateau in the efficiency curve [15].

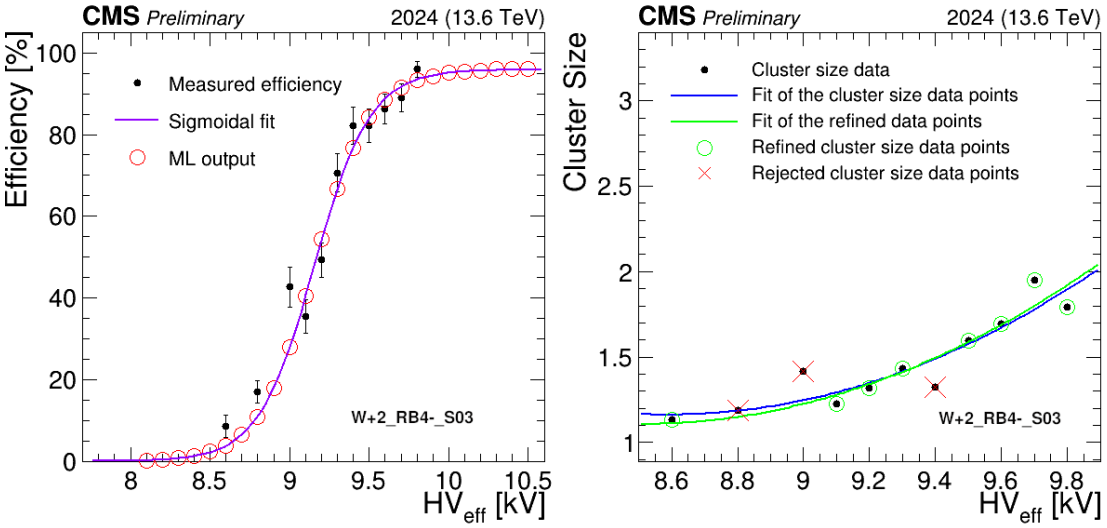



Figure 24: FSAC applied to a case where outliers and missing plateau are present: excluded 8.8, 9.0, and 9.4 kV due to mismatch between predicted and measured efficiency value which exceeds the error of the measurement [15].

## 3.4 Software

A dedicated software tool written in Python was developed to automate the analysis of CMS RPC high-voltage scan data. It processes user-input data, applies a machine learning model to generate predictions, and fits the results using a sigmoid function. The software then extracts key parameters, calculates the optimal working points for each roll and channel, and generates a CSV file. This file can be directly uploaded to the detector via the DCS (Detector Control System). The data flow can be seen in Fig. 25.

The data from the 2024 HV scan was analyzed using the software, and the resulting output was uploaded to the detector. To accomplish this, the software was downloaded from GitLab and executed in LXPLUS, where the necessary data was retrieved from a dedicated EOS directory.

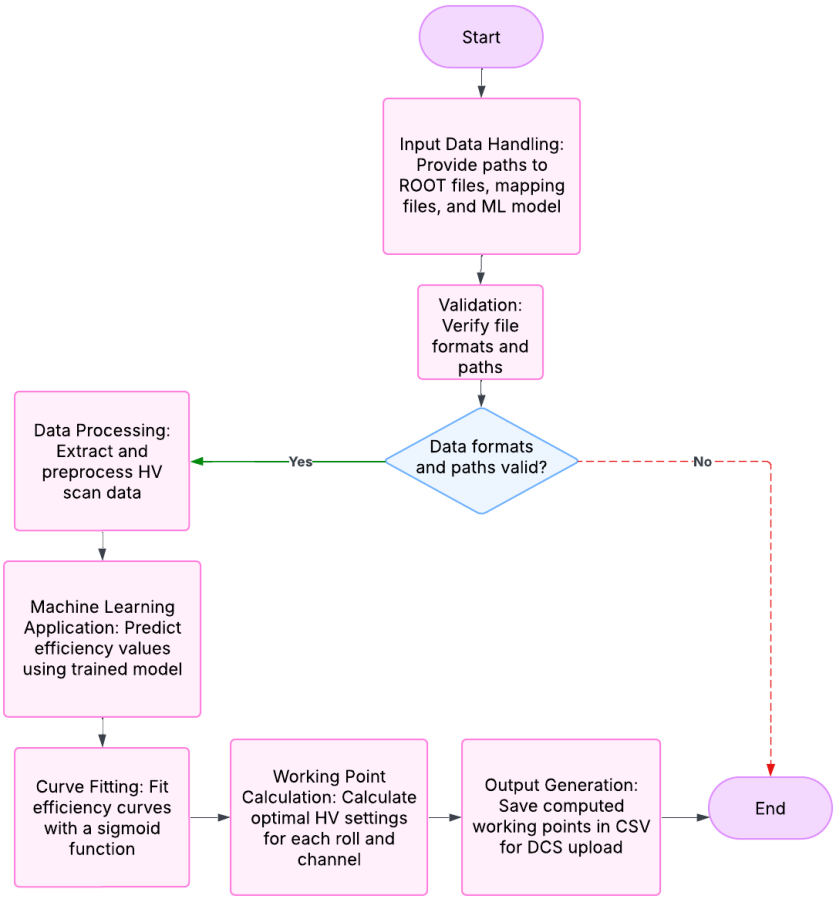
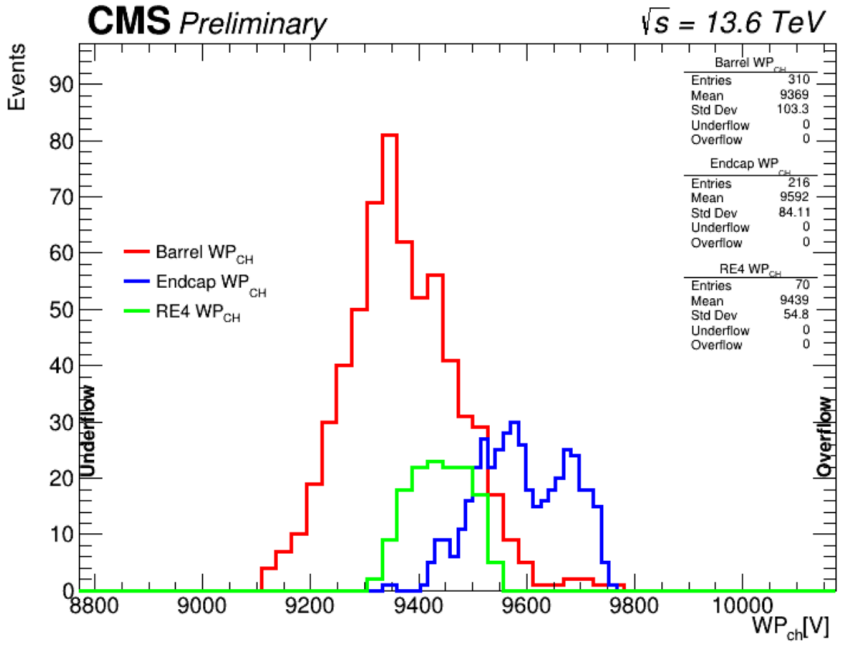
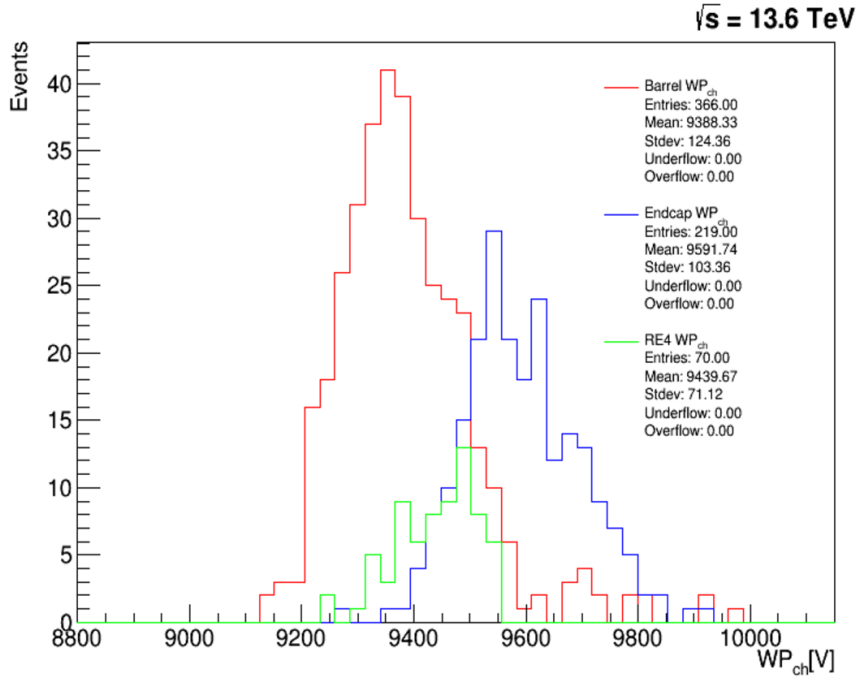



Figure 25: A chart illustrating the data flow in the developed software.

The results from the 2022 HV Scan were compared between the standard procedure and the new approach. The distributions of the WP per channel for both the standard and ML procedures can be seen in Fig. 26. In Figure 26a, it is evident that there are fewer entries for the barrel and endcap regions compared to Figure 26b. This difference occurs because the rolls that cannot be fitted with a sigmoid function have been excluded from the histogram.




(a)




(b)

Figure 26: Distributions of the working points for the two procedures: (a) shows the results with the standard procedure, and (b) shows the results using machine learning.

# Conclusion

The Large Hadron Collider is the largest and most powerful particle accelerator ever built. Operating at an energy of nearly 14 TeV in the center-of-mass frame, it smashes protons every 25 nanoseconds, resulting in thousands of newly created particles every second. The Compact Muon Solenoid is one of four detectors situated around the LHC, designed to explore questions within and beyond the Standard Model of particle physics.

Muons, which are Minimum Ionizing Particles, are detected in the outermost layers of the detector. The muon system comprises several types of gaseous detectors, including Resistive Plate Chambers. For the detectors to work properly and yield satisfactory results, they need to be calibrated by establishing the correct working points. This calibration is achieved through a procedure known as HV Scan.

The aim of this thesis is to automate the analysis of the HV Scan, parts of which were previously conducted manually. The automation uses machine learning techniques. Several ML models were tested on the data, and the FSAC model was ultimately chosen, as it successfully addressed all challenges. Additionally, a software application was developed using Python, allowing the entire analysis to be executed with a single command-line input.

The output is the correct working points for each HV channel, provided in a CSV data format that can be directly uploaded to the detector via the Detector Control System. As a result, the analysis, which originally took about three months, can now be completed in less than a week.

# Acknowledgements

I would like to express my sincere gratitude to my academic supervisor, Assoc. Prof. Peicho Petkov, for his patience, guidance, and support throughout the preparation of this thesis. Without his efforts and assistance, I would not have been able to complete this work. I am also deeply thankful to the Elementary Particle Physics group, led by Prof. Leandar Litov, for their time, attention, and valuable contributions. Furthermore, I am grateful for the opportunity to work on this project, as it allowed me to apply machine learning techniques in a real research setting. I have long been eager to explore this field, and this experience provided an engaging and insightful introduction to it.

# Appendix A Multi-Layered Perceptron

A Multi-Layer Perceptron (MLP) is a type of feedforward deep neural network where data flows from the input layer to the output layer without recurrent connections. It consists of at least one hidden layer with fully connected neurons, each applying a nonlinear activation function. MLPs are used for classification and regression tasks. During training, they compute a weighted sum of inputs, apply activations, compute loss at the output, and update weights via backpropagation using different optimization techniques such as basic gradient descent or something more sophisticated. A simple MLP can be seen in Fig. 27

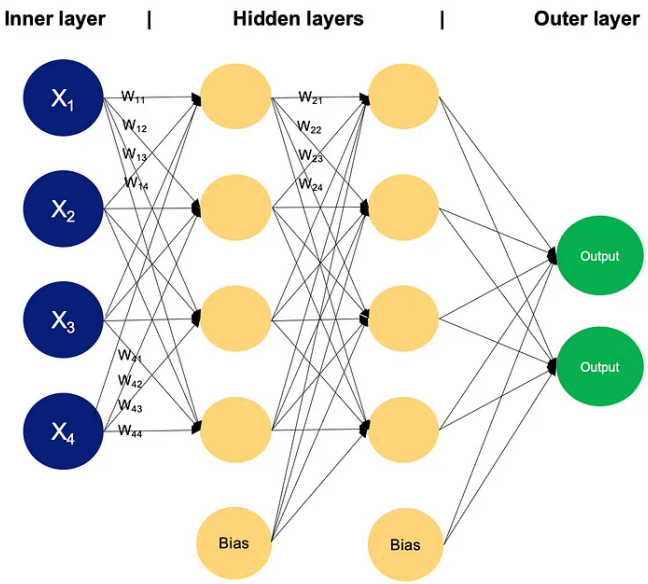



Figure 27: A simple MLP with two hidden layers.

The proposed model for automating the data analysis of RPC HV scan data is illustrated in Fig. 28. It is an MLP consisting of an input layer with 11 nodes representing the measured efficiencies. The network has three hidden layers, followed by an output layer with three nodes predicting the parameters of the sigmoidal function: $\epsilon_{max}$, $HV_{50\%}$, and $slope_{50\%}$.

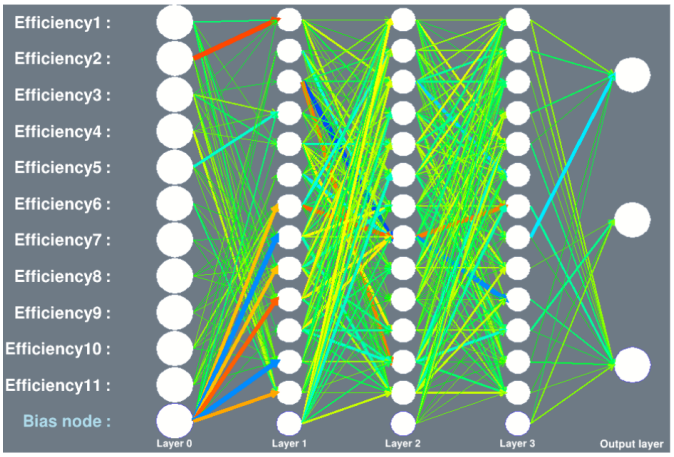



Figure 28: The tried MLP architectecture consists of one input layer, three hiddedn layers and one ouput layer.

While the model performed well in most cases, it occasionally predicted incorrect parameter values, leading to poor estimation of the working point. Another issue is that this model cannot address the problem of missing data in the plateau region, which is observed in some cases. Figures 29 and 30 illustrate examples of the good and bad performance of the model, respectively.

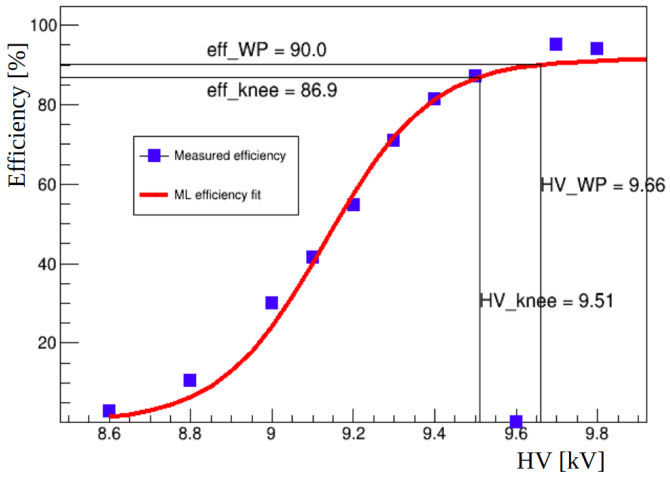



Figure 29: An example of how MLP performs well on data and accurately predicts the parameters, resulting in a good fit for the data points.

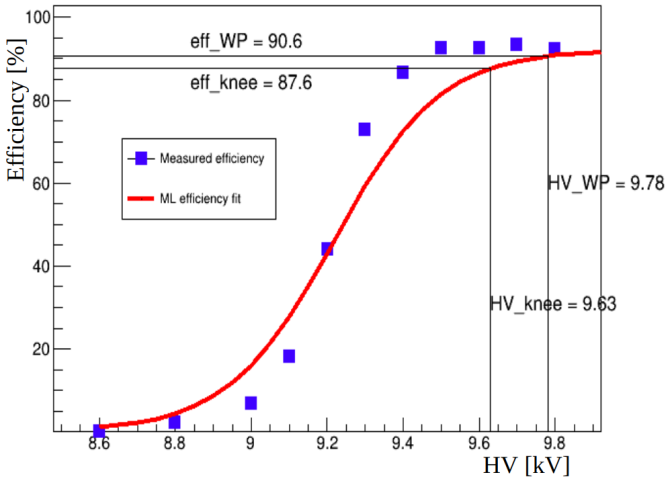



Figure 30: An example where the MLP does not perform well on data showing its inability to accurately predict the slope of the curve at $HV_{50\%}$, resulting in a poor fit of the data points.

# Appendix B 1D Convolutional Neural Network

Convolutional Neural Networks (CNNs) are a type of feedforward artificial neural network (ANN) typically used in supervised learning, where labeled data is required for training. Unlike standard neural networks with fully connected layers, CNNs use local receptive fields and shared weights, which reduces the number of parameters. This structure results in fewer weights compared to a fully connected neural network, helping reduce the time required for training and the memory usage. A Convolutional Neural Network typically consists of three main types of layers: the convolutional layer, the pooling layer, and the dense layer. The convolutional layer contains kernels (also called filters), which are matrices for 2D convolutions or vectors for 1D convolutions. These kernels perform a convolution operation with the input data, sliding over the input and performing a dot product between the kernel and local regions of the input. The values of the kernel are learned during training and are considered the model's parameters. The number of kernels in a convolutional layer determines the number of output feature maps. The kernels are used to extract specific features, such as edges or textures, from

the input data.

The pooling layer follows the convolutional layer and serves to reduce the spatial dimensions of the feature maps. It does so by applying a pooling operation, typically max pooling or average pooling. In max pooling, the maximum value within a defined region is taken, while in average pooling, the average value is calculated. Pooling helps reduce computational load and prevents overfitting by reducing the number of parameters in the network.

The dense layer is a fully connected layer that comes at the end of the network, where the features extracted by the convolutional and pooling layers are combined. In the dense layer, the features undergo a weighted sum, with each weight learned during training. The result is then passed through an activation function to produce the final output, such as for classification or regression tasks.

The architecture of the tested 1D CNN is shown in Fig. 31. It consists of an input layer that takes the 11 measured efficiencies of the RPCs, followed by 6 hidden layers, with four of them being convolutional layers. The network has 11 output nodes, which classify the data as either an outlier or not.

This model classified the data well but also excluded points that were clearly not outliers. Another issue is that it cannot address the problem of missing data in the plateau region. Good and bad examples of the behavior of the model are shown in Figures 32 and 33.

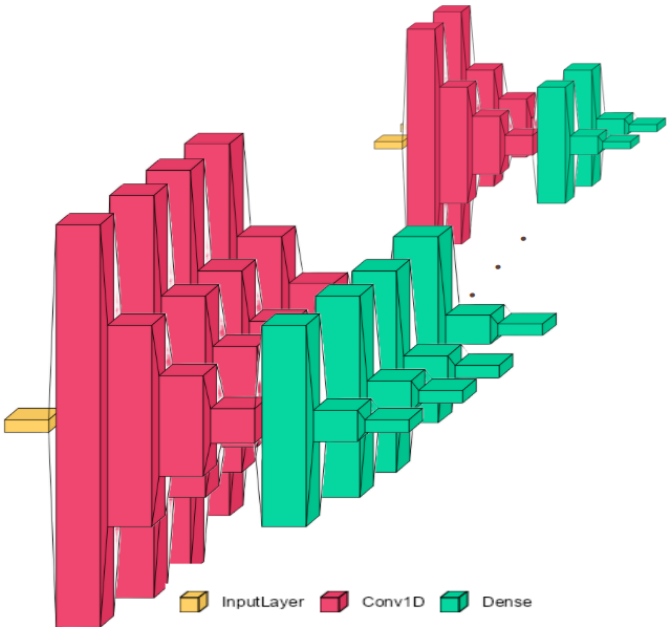



Figure 31: 1D CNN architecture. It consists of 11 input nodes, 6 hidden layers, and 11 outputs that say if the data point is and outlier or not.

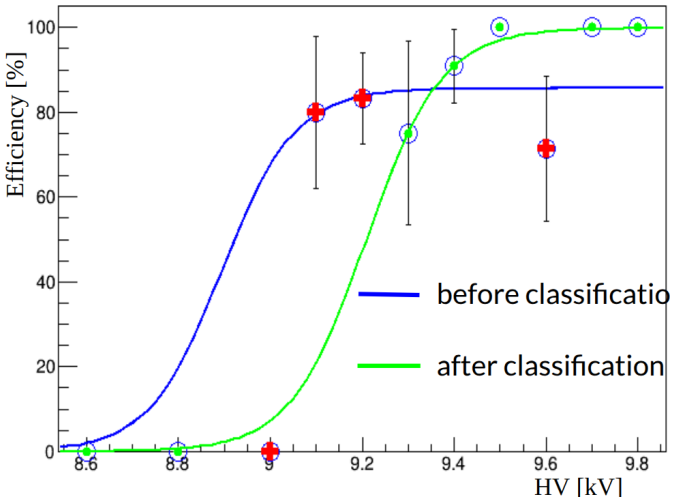



Figure 32: An example of how the CNN performs on the data, accurately classifying the outliers. The red crosses represent the data points identified as outliers, while the green dots correspond to the points that passed the test. The blue curve represents the fit of all data points, demonstrating how it is skewed due to the presence of outliers, and the green curve shows the fit of the data points that were not classified as anomalies by the model.

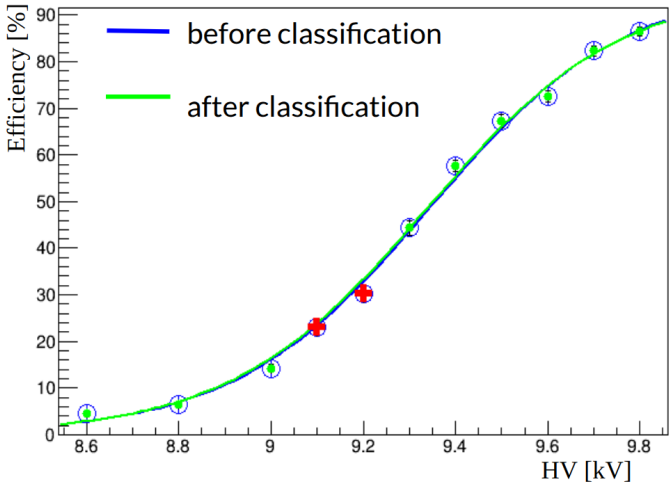



Figure 33: An example of how the CNN performs on data that does not require refinement. The model correctly classifies most of the data points, but fails to classify two points as outliers. The red crosses represent the data points identified as outliers, while the green dots correspond to the points that passed the test. The blue curve shows the fit of all data points, and the green curve represents the fit of the data points that were not classified as anomalies by the model. Although the fit remains good even after removing the two incorrectly classified data points, incorrect classification is undesirable.

# Appendix C LeakyReLU

Leaky Rectified Linear Unit, or Leaky ReLU, is one of the most commonly used activation functions in machine learning models.It is called "leaky" because unlike the basic ReLU function that leads to inactive neurons for negative inputs, Leaky ReLU introduces non-zero gradient for negative input values. This non-zero gradient prevents the 'dying ReLU problem' by preserving some activity in the neurons. The function is defined as:

$$\text{Leaky ReLU}(x) = \begin{cases} x & \text{if } x > 0 \\ \alpha x & \text{if } x \leq 0 \end{cases} \tag{16}$$

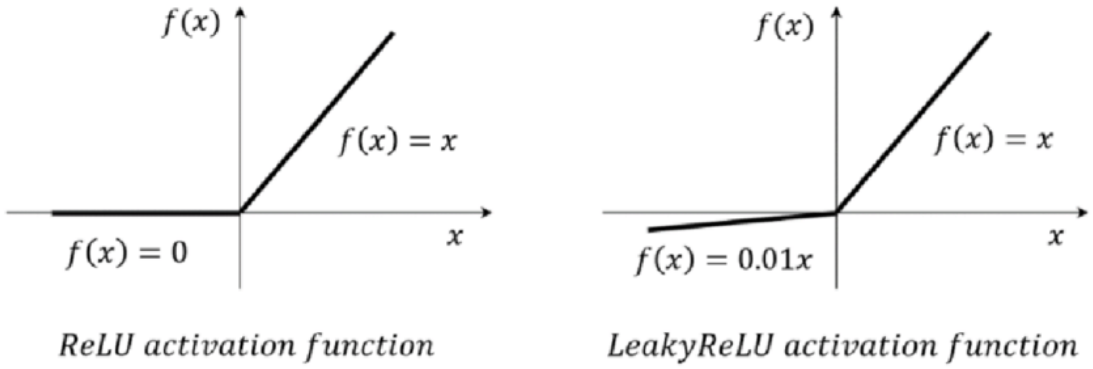



Figure 34: ReLU activation function vs Leaky ReLU activation function. The small slope introduced for negative inputs in Leaky ReLU keeps the neurons active, meaning they can still learn [16].

where $\alpha$ is a small positive constant that defines the slope of the negative part of the function. Comparison between basic ReLU and LeakyReLU can be seen in Fig. 34.

# Appendix D *Adams* optimizer

Adams is an optimization method that integrates two extensions of stochastic gradient descent (SGD): the Adaptive Gradient Algorithm (AdaGrad) and Root Mean Square Propagation (RMSProp). The key innovation of these extensions is the use of different learning rates for each weight (or parameter), in contrast to traditional SGD, which applies the same learning rate for all weights throughout the entire training process [17]. The issue with Stochastic Gradient Descent (SGD) is that, while it converges to the minimum faster than standard gradient descent, it can be highly sensitive to the initial weight values. This sensitivity may cause SGD to become trapped in a local minimum, preventing it from reaching the global minimum [17].

AdaGrad addresses this by adjusting the learning rates based on the accumulation of past squared gradients. This means that if the gradients for a particular weight are consistently large over several steps, the learning rate for that weight

will decrease rapidly. The update rule for the AdaGrad algorithm is as follows:

$$w_i(t) = w_i(t-1) - \frac{\eta}{\sqrt{G_i} + \epsilon} \cdot \nabla_{w_i} J(w)\,, \tag{17}$$

where $G_i = \sum_{t=1}^{T} (\nabla_{w_i} J(w))^2$ is the sum of the gradients of the loss function in relation to the weight $w_i$ over all previous steps. In some instances, the algorithm can reduce the learning rate excessively, causing the learning process to become very slow.

RMSProp uses a different method to adjust the learning rate by relying on the moving average of squared gradients. This approach addresses the issue of the learning rate decreasing too rapidly, as seen in AdaGrad. The algorithm is described by the following equations:

$$E_i(t) = \beta E_i(t-1) + (1-\beta)(\nabla_{w_i} J(w_i))^2 \tag{18}$$

where $E_i(t)$ is the moving average of the squared gradients for parameter $w_i$, $\beta$ is the decay factor (typically $\beta = 0.9$), $\nabla_{w_i} J(w)$ is the gradient of the loss function $J(w)$ with respect to parameter $w_i$ at iteration $t$.

The weights are updated by scaling the gradient by the inverse of the square root of the moving average $E_i(t)$ (with a small constant $\epsilon$ to prevent division by zero). The update rule is given by:

$$w_i(t) = w_i(t-1) - \frac{\eta}{\sqrt{E_i(t)} + \epsilon} \nabla_{w_i} J(w) \tag{19}$$

where $w_i(t)$ is the updated parameter at iteration $t$, $\eta$ is the learning rate, $\epsilon$ is a small constant (typically $10^{-8}$) to avoid division by zero.

Adams combines the advantages of both the AdaGrad and RMSProp. It computes two 'moments': the exponential moving average of the gradient and the squared gradient of the weights. By providing additional information about past weight updates and the direction of the gradient, the optimization process improves due to better control over the algorithm's functioning. Adams is described

by the following equations:

$$\begin{aligned} m_t &= \beta_1 m_{t-1} + (1-\beta_1) g_t, \\ v_t &= \beta_2 v_{t-1} + (1-\beta_2) g_t^2 \end{aligned} \tag{20}$$

where $m_t$ and $v_t$ are the moving average of the gradient (fist moment or also referred to as 'mean' of the gradients) and the moving average of the squared gradients (second moment or also referred also as 'variance' of the gradients). Since $m_t$ and $v_t$ are initialized as vectors of zeros, they tend to be biased towards zero during the initial steps, especially when the gradients are small (due to $\beta_1$ and $\beta_2$ being close to 1). This is why a bias correction is necessary. The bias correction is performed using the following equations:

$$\hat{m}_t = \frac{m_t}{1-\beta_1^t}, \qquad \hat{v}_t = \frac{v_t}{1-\beta_2^t} \tag{21}$$

Keeping this in mind, the rule for updating the weights is as follows:

$$w_{t+1} = w_t - \frac{n}{\sqrt{v_t} + \epsilon} m_t, \tag{22}$$

Comparision between different algorithms for minimizing the function $f(x1, x2) = 1.3x_1^2 + 2x_2^2$ can be seen in Fig.35.

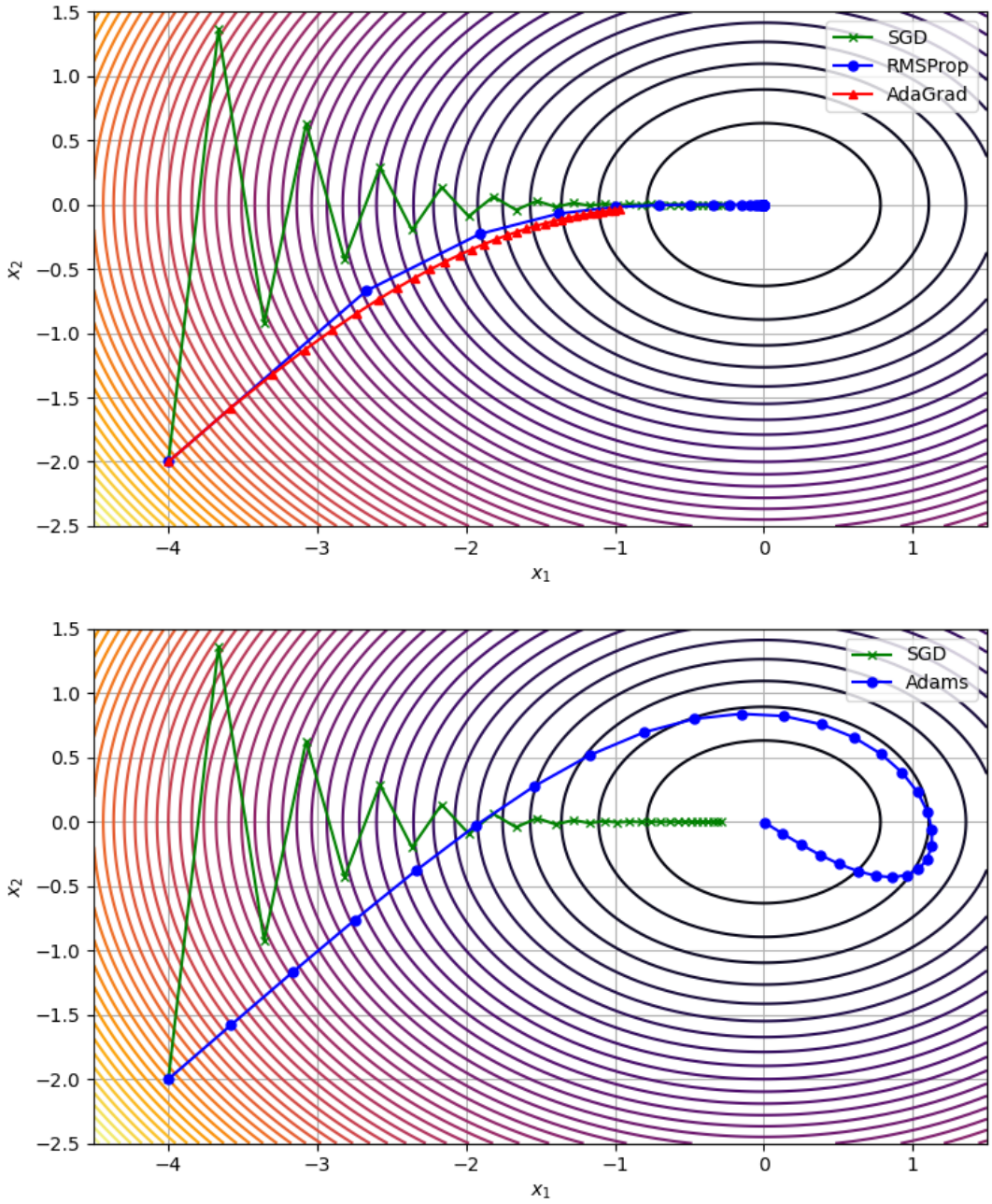



Figure 35: The top plot compares the performance of SGD, AdaGrad, and RMSProp, each using 30 iterations for updating $x_1$ and $x_2$. RMSProp converges the fastest and successfully reaches the minimum. While SGD outperforms AdaGrad in terms of speed in this instance, both AdaGrad and RMSProp exhibit smoother convergence. The bottom plot compares SGD and Adam, also with 30 iterations for updating $x_1$ and $x_2$. Adam demonstrates smoother convergence and successfully finds the minimum, whereas SGD does not reach it. All algorithms utilize a learning rate of 0.42.

# References


[1] B. J. Holzer, "Introduction to Particle Accelerators and their Limitations," in *CAS - CERN Accelerator School: Plasma Wake Acceleration*, pp. 29–50. CERN, Geneva, 2016. arXiv:1705.09601 [physics.acc-ph].

[2] CERN, "How large is the LHC?." https://howlargeisthelhc.com/.

[3] S. Calzaferri and on behalf of the CMS Muon Group, "Analysis of discharge events in the cms ge1/1 gem detectors in presence of lhc beam," *Journal of Instrumentation* **19** no. 02, (Feb, 2024) C02003. https://dx.doi.org/10.1088/1748-0221/19/02/C02003.

[4] K. Bunkowski, *Optimization, Synchronization, Calibration and Diagnostic of the RPC PAC Muon Trigger System for the CMS detector*. PhD thesis, Warsaw U., 2009. https://cds.cern.ch/record/1308715. Presented on Jul 2009.

[5] Fagot, Alexis, *Consolidation and extension of the CMS Resistive Plate Chamber system in view of the high-luminosity LHC upgrade*. PhD thesis, Ghent University, 2020. http://hdl.handle.net/1854/LU-8664361.

[6] W. Riegler, "Induced signals in resistive plate chambers," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **491** no. 1, (2002) 258–271. https://www.sciencedirect.com/science/article/pii/S0168900202011695.

[7] I. Crotty, J. Lamas-Valverde, G. Laurenti, M. C. S. Williams, and A. Zichichi, "The non spark mode and high rate operation of resistive parallel plate chambers," *Nucl. Instrum. Methods Phys. Res., A* **337** (1994) 370–381. https://cds.cern.ch/record/254633.

[8] Reyes-Almanza and et al., "High voltage calibration method for the cms rpc detector," *Journal of Instrumentation* **14** no. 09, (Sep, 2019) C09046. https://dx.doi.org/10.1088/1748-0221/14/09/C09046.

[9] F. Thyssen, "Performance of the resistive plate chambers in the cms experiment," *Journal of Instrumentation* **7** no. 01, (Jan, 2012) C01104. https://dx.doi.org/10.1088/1748-0221/7/01/C01104.

[10] E. Cerron Zeballos, I. Crotty, D. Hatzifotiadou, J. Lamas Valverde, S. Neupane, V. Peskov, S. Singh, M. Williams, and A. Zichichi, "A comparison of the wide gap and narrow gap resistive plate chamber," *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* **373** no. 1, (1996) 35–42. https://www.sciencedirect.com/science/article/pii/0168900295014837.

[11] "Autoencoders." *Available at* https://introml.mit.edu/_static/fall23/LectureNotes/chapter_Autoencoders.pdf.

[12] J. Zhou, "On discrete cosine transform," 2011. https://arxiv.org/abs/1109.0337.

[13] y. Martín Abadi et al., "TensorFlow: Large-scale machine learning on heterogeneous systems." https://www.tensorflow.org/. Software available from tensorflow.org.

[14] F. Chollet, "keras." https://github.com/fchollet/keras, 2015.

[15] **CMS** Collaboration, "Machine Learning approach to CMS RPC HV scan data analysis,". "https://cds.cern.ch/record/2916751".

[16] Z. Li, W. T. Nash, S. P. O. Brien, Y. Qiu, R. K. Gupta, and N. Birbilis, "cardigan: A generative adversarial network model for design and discovery of multi principal element alloys," 2022. https://arxiv.org/abs/2202.00966.

[17] O. Hospodarskyy, V. Martsenyuk, N. Kukharska, A. Hospodarskyy, and S. Sverstiuk, "Understanding the adam optimization algorithm in machine learning," in *Congreso Internacional de Tecnologías e Innovación*. 2024. https://api.semanticscholar.org/CorpusID:271908550.