Universidad de Oviedo

Estudio con aprendizaje automático de la certificación de datos físicos del detector CMS de LHC (CERN) utilizando Auto Encoders (AEs)

Autor:
Hugo Calvo Castro

Tutores:
Javier Fernández Menéndez
Andrea Tapote Fernández

Universidad de Oviedo

10 de Julio, 2022

Índice general

Capítulo 1

Introducción y motivación

El experimento CMS (Compact Muon Solenoid) del LHC (Large Hadron Colider) del CERN toma datos de colisiones de protones para el estudio de la física de partículas de altas energías. La cantidad de colisiones registradas por el detector es muy grande, y estas han de ser filtradas para evitar que datos de mala calidad entren en el análisis físico.

Uno de los procesos de filtrado se llama DC (Data Certification), en el cual expertos analizan la distribución estadística de los observables físicos registrados por CMS para obtener una muestra en la que todos los subdetectores del experimento funcionen correctamente.

El objetivo de este trabajo es estudiar un método de automatización de este proceso, en concreto la arquitectura de redes neuronales llamada Auto Encoder (AE). La automatización de DC de CMS es clave para el futuro, ya que podría reducir la cantidad de mano de obra experta necesaria y sería posible aumentar

la cantidad de datos procesados, en previsión del futuro del colisionador LHC.

En 2029 se pondrá en marcha el HL-LHC (High Luminosity LHC), una modificación del experimento LHC que brindará un órden de magnitud más colisiones que el actual. La colaboración CMS está investigando varias maneras de adaptar los métodos de DC a esta mejora del LHC, y entre las propuestas más prometedoras se encuentran los AE. Algunos algoritmos de Machine Learning como SVM (Support Vector Machine), kNN (k Nearest Neighbours) o DF (Decision Forest) también se han estudiado, consiguiendo buenos resultados [1].

Comenzaremos describiendo brevemente el Modelo Estándar (ME), la teoría de física de partículas que motiva la existencia de experimentos como LHC y CMS. Se hará un resumen de estos experimentos y los observables físicos que se usan en DC para el análisis. Más adelante, se abundará en el proceso de DC y el paradigma más amplio de Data Quality Monitoring (DQM). Finalmente, se definirá el modelo Auto Enconder y se aplicará a los datos para comprobar si es eficiente en la tarea propuesta.

Capítulo 2

Física de Partículas y Altas Energías

El LHC es un acelerador de partículas circular, dentro del cual se haya el experimento de CMS, un detector multifunción de partículas con el que se mide el resultado de las colisiones. El objetivo del experimento es comprobar la teoría del ME a distintos niveles de energía y a su vez buscar evidencias a favor o en contra de otras teorías física.

2.1. Modelo Estándar

El ME es una teoría cuántica de campos que describe de manera muy precisa la materia y sus interacciones salvo el efecto de la gravedad. Es decir, es una teoría unificada de la fuerza electromagnética, débil y fuerte [2].

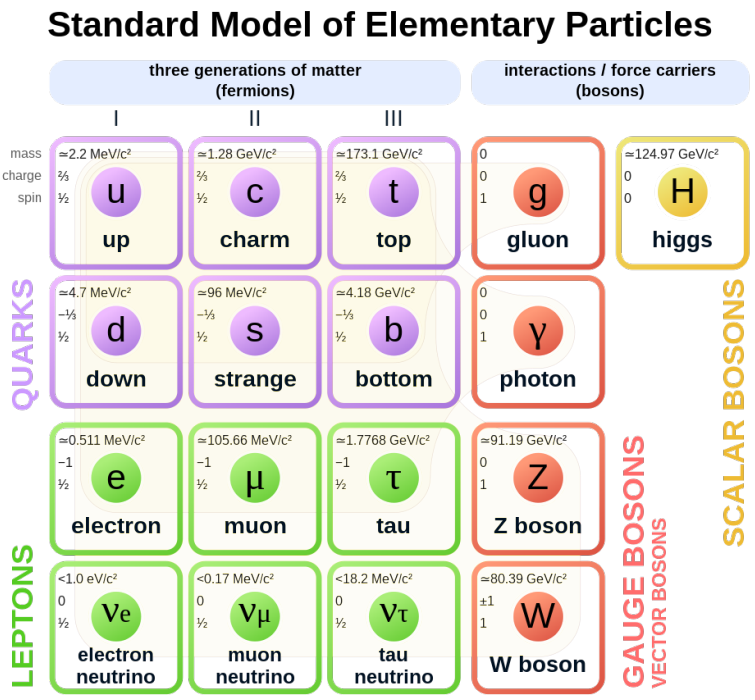


Figura 2.1: Partículas del ME. Fuente: https://en.wikipedia.org/wiki/Standard_Model

Según esta teoría, la materia está compuesta de fermiones, y estos interaccionan entre si mediante el intercabio de bosones vectoriales. El bosón de Higgs es un bosón escalar que no participa como mediador de ninguna interacción, sino que proporciona masa mediante el fenómeno de ruptura espontanea de la simetría (SSB) [3]. Comencemos por describir los elementos de la figura 2.1.

- **Leptones**: Son partículas sin color, con carga débil, y el electrón, mu y tau tienen carga eléctrica (los neutrinos son neutros). Por lo tanto solo interaccionan mediante fuerza débil y eléctromagnética. Son de especial interés para este trabajo los leptones mu, que se encuentran en la segunda generación con una masa de $105.6583745(24) MeV$ [4]. Son la principal fuente de

información sobre las colisiones en el detector CMS, como su nombre indica.

- **Quarks**: Son partículas con color, carga débil y carga eléctrica. Por lo tanto interaccionan mediante todas las fuerzas del ME. Los quarks no se ven nunca en libertad debido a la interacción fuerte que los junta en hadrones como el protón o el neutrón. Si se consigue aislar un quark, se creará un par quark-antiquark por interacción fuerte y se hadronizará en algún estado estable [2].

- **Bosones Gauge**: Son los mediadores de todas las interacciones del ME. Cada uno se acopla a los fermiones que le corresponda (ej: fotón se acopla al campo del electrón) y mediante el intercambio de estos bosones surgen las fuerzas que conocemos. Todos los conocidos tienen spin 1, lo que les da la estructura vectorial ya mencionada. En la teoría libre (i.e. una teoría que solo concibe uno de los bosones), estos tienen que tener masa nula para no violar el principio de invariancia gauge [5]. Pero experimentalmente se encuentra que los bosones débiles sí tienen masa. Este problema se solucionó con el bosón de Higgs.

- **Bosón de Higgs**: Es un bosón escalar, es decir, de spin 0. Al acoplarse a otros campos, mediante el mecanismo de Higgs les otorga masa. De esta manera, es posible tener un bosón vectorial como los $W^{\pm}$ o Z masivos a la vez que repetan el principio gauge. Aunque la teoría de Peter Higgs era muy prometedora, no se pudo comprobar experimentalmente la existencia del bosón de Higgs hasta el año 2012 [6].

Aunque el ME describe de manera precisa 3 fuerzas fundamentales, aún no existe una teoría unificada de las 4. Esto se debe a que el bosón gauge que describe el gravitón debe tener spin 2 (i.e. un campo tensorial), y el esquema para construir teoría cuánticas que sigue el ME no permite bosones con spin 2. Así que queda

un paso para tener una teoría del todo que describa todas las interacciones de la materia.

Las demás fuerzas están adecuadamente descritas por el ME. Se describirán brevemente las características principales de las mismas.

El tiempo de vida medio de un bosón es inversamente proporcional a la anchura de su distribución de masa. Como ya se vio, los fotones no son masivos, por lo que su distancia de interacción es infinita. Por otro lado, los bosones débiles son todos masivos, y a partir de su distribución de masa su rango de interacción resulta ser pequeño.

Los gluones, en cambio, no son masivos, pero lógicamente su distancia de interacción tiene que ser finita. De no ser así, no podrían existir las moléculas ya que la fuerza fuerte uniría los núcleos de los átomos que las componen. La solución al problema es la propia intensidad de la fuerza fuerte. No disminuye con la distancia como la fuerza electromagnética, sino que aumenta, y a una distancia suficientemente grande, es capaz de producir pares quark-antiquark y dejar de ser efectiva.

2.2. Más allá del modelo estándar

Ya se ha visto que el ME describe de manera precisa tres de las cuatro interacciones fundamentales de la materia. Esto es un claro indicio de que la teoría no es completa y debe ser alterada de alguna manera si se busca que describa todos los fenómenos de la naturaleza.

En esta linea de pensamiento, existen más problemas, algunos obvios y otros más sutiles, que nos han hecho poner en duda la validez del ME y proponer generalizaciones que explíquen un mayor número de fenómenos. Una lista de los principales fallos del ME podrían ser:

- **Gravedad**: Lo que se podría llamar modelo estándar de la gravedad es la Relatividad General. Por ahora no se ha conseguido unificar las dos teorías de la misma manera que se unificó la teoría electromagnética y la débil en su momento.
- **Materia y energía oscura**: Las observaciones cosmológicas indican que el universo contiene aproximadamente un 25 % de materia oscura y un 70 % de energía oscura. Se espera que exista una partícula asociada a la materia oscura, pero el problema aún sigue abierto y el ME no incluye nada relativo al tema.
- **Asimetría de materia/antimateria**: El universo está compuesto en su mayor parte de materia, ya que aún no se ha encontrado ninguna sección del mismo compuesto de antimateria. Pero el ME no predice una predilección hacia la materia, por lo que el universo debería estar esencialmente vacío. Una teoría más completa podría explicar la asimetría y producir un modelo del Big Bang más predictivo.
- **Jerarquía de masas**: Es un problema más sutil. Mediante cálculos del ME se ve que la masa del bosón de Higgs debería estar entorno a la masa de Plank [7] ($10^{19} GeV$), pero en cambio su valor experimental es del orden de la masa de los bosones débiles ($100 GeV$), por lo tanto hay una diferencia relativa de 10^{17} que el ME no es capaz de predecir.

La física teórica lleva investigando cómo resolver estas discrepancias desde la

década de los 90.

En el área de la gravedad, existen teorías como Loop Quantum Gravity o la Toería de Cuerdas, pero ninguna es aparentemente una solución a la gravedad cuántica [8]. Aún así, si existiese un supuesto gravitón detectable en el CERN, este estaría muy por encima de la energía de las colisiones que se podrán alcanzar en el futuro más cercano.

El origen de la materia oscura, en cambio, sí tiene propuestas medibles en experimentos del CERN. Por ejemplo, las WIMPS (Weakly Interacting Massive Particles), serían unas partículas fundamentales que interaccionan únicamente mediante fuerza débil y tendrían una masa en el rango de $10GeV - TeV$ [9]. Estos son unos de los candidatos a materia oscura más estudiados gracias a la facilidad en su detección.

De hecho, algunos de los modelos que introducen WIMPS pueden ayudar a solventar el problema de la jerarquía de masas. Lo habitual es el estudio de modelos SUSY (Super Symmetry), que introducen una correspondencia en fermiones y bosones. Si se quiere extender el ME a SUSY, surgen copias de cada una de las partículas (por ejemplo *stop* sería la partícula supersimétrica asociada al quark top). En este tipo de modelos, un *Higgsino*, la pareja supersimétrica del bosón de Higgs, sería un WIMP candidato a matería oscura que permite solucionar el problema de la jerarquía.

En definitiva, la física de parítulcas no acaba con el ME. Las teorías más populares suelen incluir nueva física que podría ser observada en experimentos como los del CERN. A continuación, se describirán las magnitudes relacionadas con aceleradores de partículas que, en el caso de mejorarlas adecuadamente, podrían desembocar en el descubrimiento de nuevas partículas que apoyen estas teorías.

2.3. Física de aceleradores

El principal evento que se observa en los aceleradores es la colisión de partículas (scattering), normalmente de protones o leptones, aunque en algunos casos se han estudiado colisiones de núcleos pesados. Existen dos tipos de colisión que explica la teoría cuántica de campos: elástica e inelástica.

En una colisión elástica se tiene un proceso $a + b \rightarrow a + b$. Es decir, las partículas iniciales y finales son las mismas, sólo cambia su trayectoria en la colisión. Como ya se sabe desde la época de la mecánica clásica, este tipo de proceso debe conservar ciertas magnitudes físicas como la energía y el momento.

En cambio, en una colisión inelástica se tiene $a + b \rightarrow c + d + ...$ En la colisión se producen partículas nuevas, pero no cualquiera, existen reglas de conservación derivadas del ME que explican qué productos se pueden obtener de un proceso como este. Ejemplos de estas cantidades son la energía y el momento como en el caso elástico, pero además el isospin (un número cuántico del ME) y el número bariónico (una ley de conservación empírica).

El objetivo de los aceleradores es el estudio y descrubrimiento de nuevas partículas. Por lo tanto, es deseable que se produzca un gran número de colisiones inelásticas. Las principales magnitudes que indican el número de dichas colisiones son la sección eficaz y la luminosidad [2].

- **Sección eficaz**: Es una medida de la probabilidad de que una reacción ocurra. Se puede definir como $\sigma_{tot} = \frac{\dot{N}}{\Phi N_b}$ donde $\dot{N}$ es el número de reacciones por unidad de tiempo, Φ es el flujo de partículas y N_b es el número de partículas del objetivo por unidad de área.

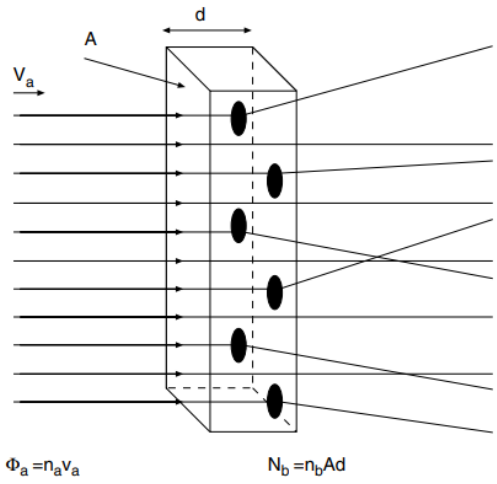


Figura 2.2: Esquema geométrico de un proceso de scattering en el que se identifican las componentes de la sección eficaz. Fuente: [2]

Como existen dos tipos de colisiones, esto se puede reflejar en la sección eficaz como $\sigma_{tot} = \sigma_{el} + \sigma_{inel}$ separandola en una parte elástica y una inelástica. A su vez, dentro de σ_{inel} habrá contribuciones de cada un de las vias por las que puede pasar la colisión (e.g. $p+p \rightarrow p+p+\pi^0$). La sección eficaz depende de la energía de las partículas a colisionar, cuanta más energía tengan (más momento), más inelásticos serán los choques.

- **Luminosidad**: Es una medida del número de colisiones posibles por unidad de tiempo. Tiene unidades de $(area \times tiempo)^{-1}$ y es la medida principal de la potencia de un colisionador de partículas. Se puede estimar para un experimento como el del LHC como,

$$\mathcal{L} = \frac{N_a N_b}{S_{eff} t} F$$

donde N_a y N_b es el número de partículas en cada haz, S_{eff} es el area de colisión de los haces, t es el tiempo que transcurre entre la emisión de los haces y F es un factor geométrico que depende del ángulo (para colisiones

frontales $F = 1$). Normalmente no se considera la luminosidad instantanea, si no la luminosidad integrada $L = \int \mathcal{L} dt$. De esta manera, se puede calcular la luminosidad total del colisionador en períodos extensos de tiempo.

- **Sección eficaz diferencial**: En la práctica, es imposible fabricar un detector que mida todas las colisiones de las partículas del acelerador, porque tendría que rodear completamente el centro de colisiones (es decir, tendría que ser una esfera cerrada). Realmente, el detector tendrá un área A_D, y asumiendo que se geometría está embebida en una esfera, cubrirá un ángulo sólido $\Delta\Omega = A_D/r$ donde r es el radio de dicha esfera. De esta forma, el número de colisiones detectadas por segundo sería,

$$\dot{N} = \mathcal{L}\frac{d\sigma}{d\Omega}\Delta\Omega$$

Para ejemplificar todas estás magnitudes, uno se puede preguntar cuantos bosones de Higgs se producen en el LHC, una cantidad muy importante si fuesemos físicos en 2011 a punto de descubrir esta nueva partícula.

Para empezar, la sección eficaz se suele medir en barn ($1b = 10^{-24}cm^2$). Como ya se mencionó, la sección eficaz depende de la energía, y para una colisión de protones a $7TeV$, se tiene una sección eficaz total de aproximadamente $110mb$ de los que $60mb$ son inelásticos. A partir del ME y cálculos de teoría cuántica de campos se puede estimar que la sección eficaz para producir un bosón de Higgs de $125GeV$ mediante $q + q \to Z + H$ es de $50fb$. Como se puede apreciar, la sección eficaz de la producción de bosones de Higgs por esta vía es aproximadamente 10^{-11} veces la sección eficaz total. Por lo tanto, en los datos recogidos será una pequeña mancha en el fondo de un número enorme de eventos.

La luminosidad del LHC se puede estimar a partir de los valores nominales de los

bunches como $\mathcal{L} = 10^{34} cm^{-2} s^{-1}$.

Finalmente, el número de eventos observables para un canal concreto de colisión se calcularía como $N_{evento/s} = \mathcal{L}\sigma_{evento}$. En este ejemplo, $N_{Higgs/s} = 10^{34} \times 5\ 10^{-38} s^{-1} = 5\ 10^{-4} s^{-1}$. O lo que es lo mismo, **1 bosón de Higgs cada 33 minutos**. He aquí la necesidad de incrementar lo máximo posible la luminosidad. Si se busca una estadística de 5σ para publicar un descubrimiento, se necesita una gran cantidad de detecciones de la partícula, así que o se aumenta la luminosidad o podemos necesitar años hasta acumular la cantidad de datos necesaria.

Para una visión más amplia de la sección eficaz de ciertos eventos, se puede consultar la figura 2.3.

Es interesante notar como la producción de bosones Higgs por las vías ggH y WH aumenta casi un órden de magnitud desde las primeras colisiones del LHC y las más actuales a $13.6TeV$. Se puede ver también que el fondo σ_{tot} es 10^9 mayor que σ_{ggH}, lo que significa que se produce un bosón de Higgs por cada mil millones de otros eventos, cantidad que resalta la necesidad de tener una estadística muy alta para poder observar la distribución de masa de este bosón sobre el fondo.

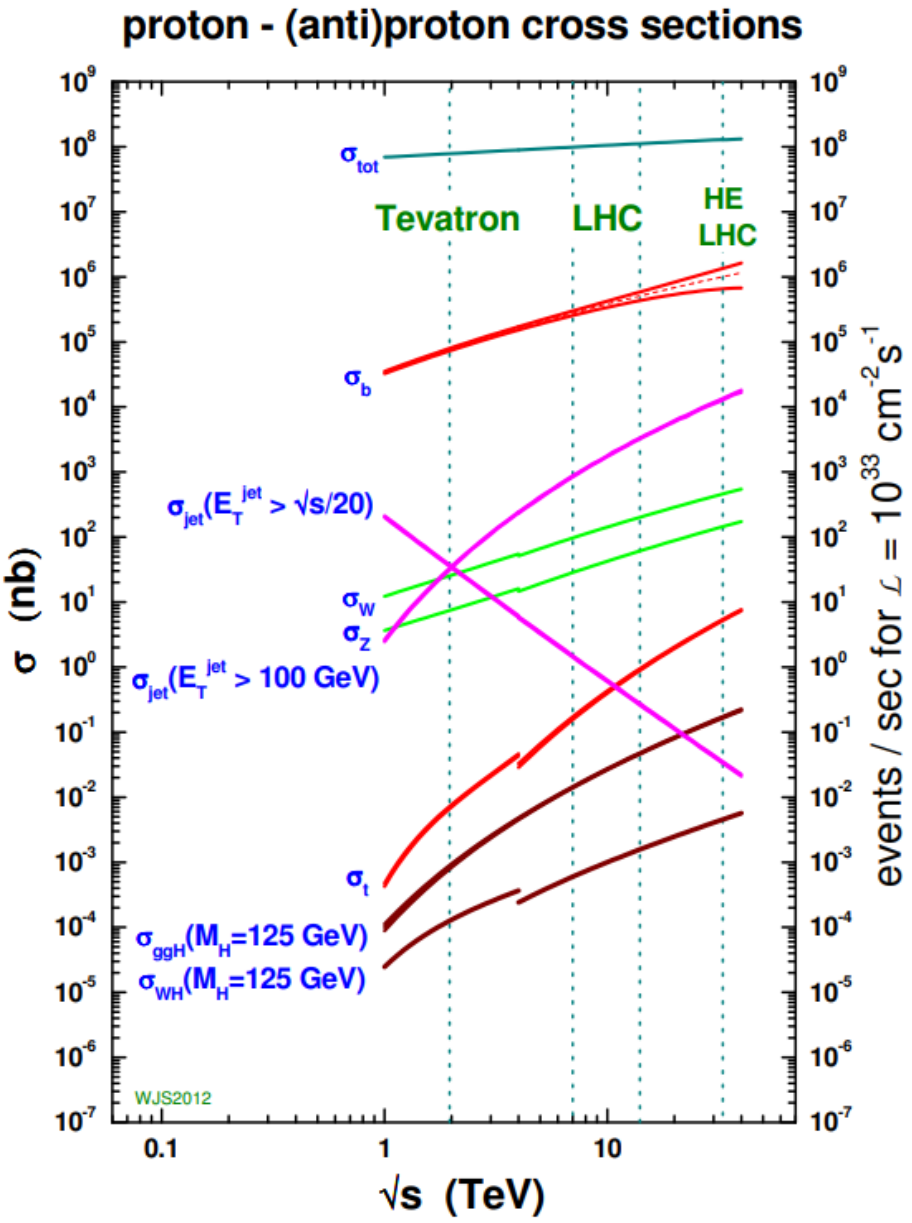


Figura 2.3: Sección eficaz (a la izquierda) y tasa de sucesos esperada (a la derecha) de varios eventos en función de la energía en centro de masas de los haces, junto con las energías usadas en los aceleradores Tevatron, LHC o el posible HE-LHC.

Capítulo 3

LHC y CMS

3.1. LHC y el complejo de aceleradores del CERN

El LHC (Large Hadron Collider) es el colisionador de hadrones circular más potente y grande del mundo. Consta de $27km$ de circunferencia, en los cuales se hallan varios experimentos que miden el resultado de las colisiones con el fin de probar teorías físicas como el ME.

Antes de que los haces (beams) de protones estén a $6.5TeV$ circulando en el LHC, pasan por una serie de aceleradores menores que aceleran gradualmente partículas que son inyectadas secuencialmente en el siguiente trozo del complejo. La construcción de este comenzó en la década de 1950 con el LINAC1, un acelerador lineal de iones ligeros como deuterio y partículas α. A traves de los años, las demás partes de la figura 3.1 fueron construidas. Para el objetivo de este trabajo es suficiente centrarse en la estructura final que inyecta protones en el LHC.

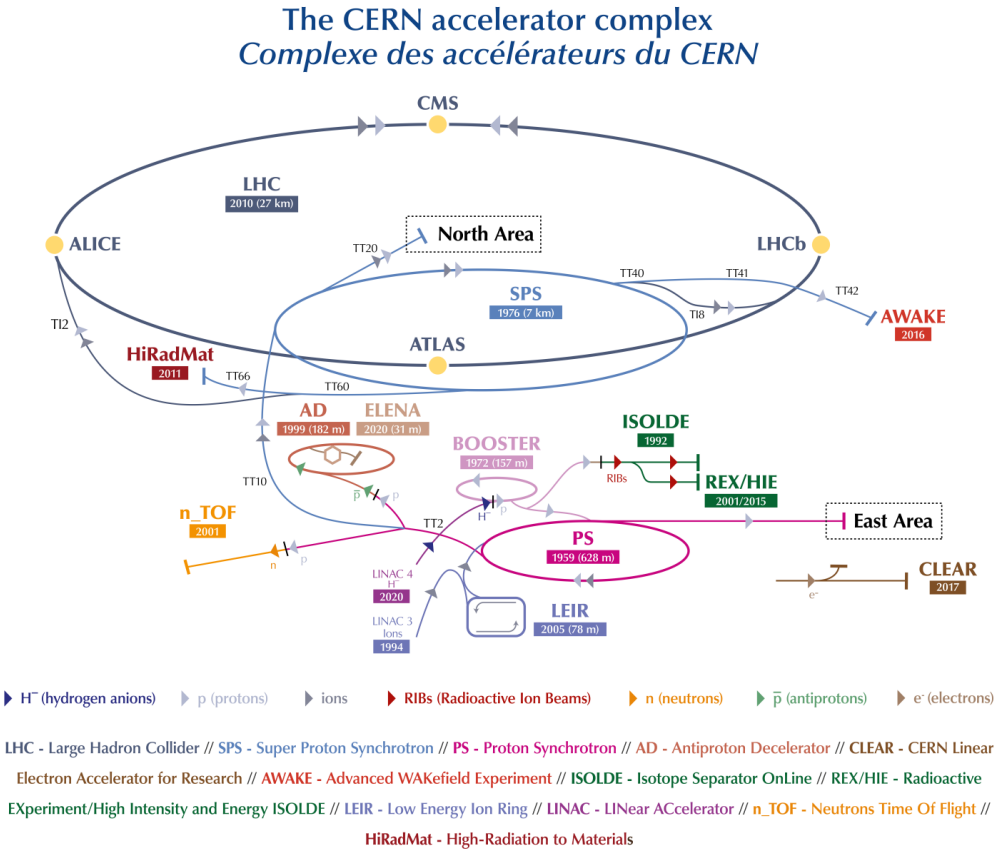


Figura 3.1: Complejo de aceleradores del CERN. Fuente: CERN

- **LINAC4**: Es el sucesor del LINAC2, que aceleraba protones a $50MeV$. Este acelerador lineal es capaz de acelerar iónes de hidrógeno H^- a $160MeV$ en pulsos de $400\mu s$. En el proceso de inyección en el PSB (Proton Synchrotron Booster) se le quitan los electrones a los H^- para dejar solamente protones. Es una de las nuevas componentes que ayudarán a aumentar la luminosidad en esta década para el proyecto de $HL-LHC$.

- **PSB**: Está compuesto por 4 anillos que toman los protones a $160MeV$ y los aceleran a $2GeV$ en su recorrido para inyectarlos en el PS (Proton

Synchrotron). Lleva en funcionamiento desde el año 1972.

- **PS**: Es el primer gran acelerador, de $628m$ de circunferencia, que funciona desde el año 1959. Desde su primer uso ha acelerado gran variedad de partículas, como partículas alpha, oxígeno, azufre y hasta antiprotones y positrones. Es capaz de acelerar protones hasta $26GeV$ antes de inyectarlos en el SPS (Super Proton Synchrotron).

- **SPS**: Es el segundo acelerador más grande del CERN, de $7km$ de circunferencia, que funciona desde el año 1976. Fue el primer acelerador del CERN en hacer grandes descubrimientos de materia exótica, cuando en 1983 se descubrieron los bosones débiles W y Z. Es capaz de acelerar protones a $450GeV$ antes de inyectarlos en el LHC (Large Hadron Colider).

- **LHC**: Es la última pieza del puzle, el acelerador de partículas más grande del mundo, con $27km$ de circunferencia, que funciona desde el año 2008. Comenzó funcionando a $1.18TeV$ de energía en el centro de masa, y a día de hoy alcanza los $14TeV$ (esto es $7TeV$ por haz). Fue esencial para el descubrimiento del bosón de Higgs en 2013. Alimenta varios experimentos, como CMS, ALICE, LHCb, ATLAS y AWAKE. Desde el descubrimiento del bosón de Higgs, no se ha encontrado ninguna partícula elemental nueva, aunque hay teoría como supersimetría que las predicen. En el futuro, hay varias vías de acción para mejorar el acelerador con la esperanza de encontrar nueva física a energías aún mayores.

- **HL-LHC**: Es una propuesta para aumentar la luminosidad del LHC (en un orden de 10). Como ya se explicó antes, la luminosidad es proporcional al número de colisiónes posibles. Por lo tanto un aumento en esta implica más datos para hacer análisis estadístico. En principio se pondrá en marcha en 2029 y funcionará hasta 2040. El aumento en la cantidad de datos tomados

conlleva una necesidad de cambiar la manera en la que estos se almacenan y se analizan. Esta es la clave del trabajo, la automatización de una parte del procesamiento de los datos con vistas al HL-LHC.

- **FCC**: Las siglas significan Future Circular Colider. Es una propuesta de ingeniería civil para construír un nuevo anillo similar al LHC, pero con una circunferencia de $100km$. Sería posible acelerar las partículas hasta $100TeV$ y explorar posibles partículas 10 veces más pesadas que las que seríamos capaces de ver con la tecnología actual. El proyecto está programado para empezar en 2040 cuando acabe el HL-LHC.

El LHC no funciona constantemente, ya que cuando se quiere mejorar su funcionamiento, no se pueden realizar colisiones.

3.2. Períodos de funcionamiento del LHC

Los períodos de funcionamiento del LHC se denominan Run. Entre estos, pueden pasar años en los que el LHC no acelera partículas, estos se denominan Long Shutdown, y se aprovecha para hacer mejoras al acelerador mientras está apagado.

Hasta la fecha, se han realizado dos Run completos. El Run 1 se extendió desde 2009 hasta 2013, con una energía inicial en centro de masas $\sqrt{s} = 7TeV$ que acabó aumentandose a $8TeV$ el último año. El Run 2 duró desde 2015 hasta 2018, a una energía de $13TeV$.

En el primer Long Shutdown (2013-2015), se mejoró la luminosidad del acelerador y la energía en centro de masas de los colisionadores. En la figura 3.2 se puede ver

la luminosidad integrada y la energía en centro de masas durante los diferentes años de funcionamiento del LHC y en concreto del experimento CMS.

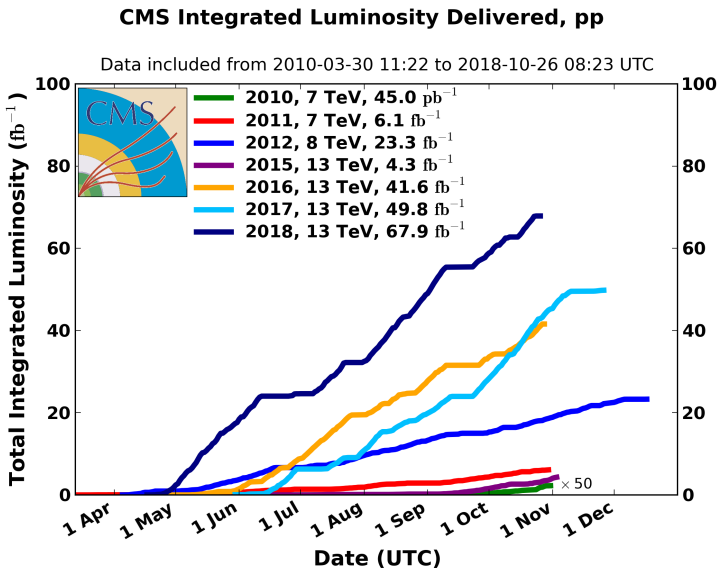


Figura 3.2: Luminosidad integrada de CMS durante diferentes años del funcionamiento del LHC. Aunque los datos vengan sólo del experimento CMS, los valores son representativos del total del LHC, ya que gran parte de las colisiones van dirigidas al CMS.

En el 2018 comezó el segundo Long Shutdown. Durante este, el LHC se preparó con dos mejoras para el Run 3. Primero, un aumento de la energía en centro de masas de $13TeV$ a $13.6TeV$, que brindará colisiones más ricas para el análisis físico. Lo segundo y más importante, un aumento del 50 % en la luminosidad, con el que se obtendrán más datos para realizar mediciones más precisas.

El Run 3 se puso en marcha oficialmente el 5 de julio de 2022 con colisiones a 13.6TeV y se mantendrá activo hasta el final de 2025, momento en el que se iniciará el tercer Long Shutdown para continuar mejorando el acelerador.

En el Long Shutdown 3 se realizarán cambios importantes al LHC para iniciar el

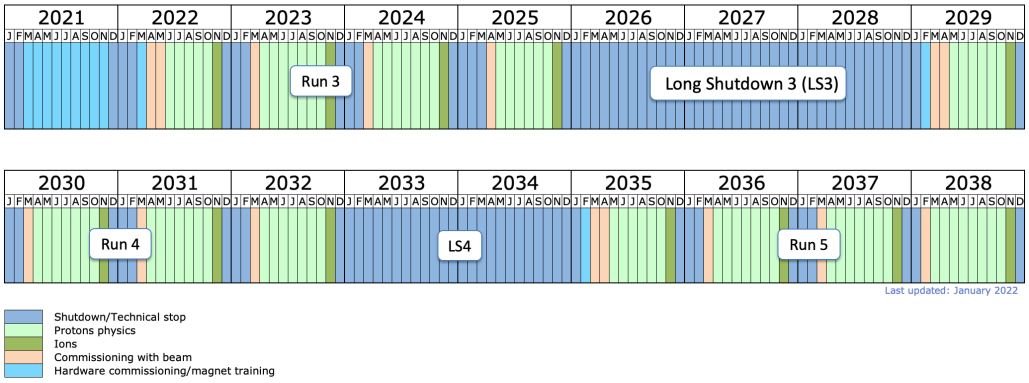


Figura 3.3: Programación del funcionamiento del LHC. Fuente: CERN

nuevo período HL-LHC, que comenzará en 2029 para terminar en 2040. En ese momento, se espera que el LHC haya alcanzado el máximo de sus capacidades, y será más eficiente construir un nuevo experimento para estudiar la física de partículas.

3.3. CMS, el detector

Uno de los experimentos dentro del LHC es CMS, que es un detector de propósito general especializado en la detección de muones.

La colaboración CMS cuenta con más de 4000 integrantes de más de 50 países, como físicos de partículas, ingenieros, informáticos, etc. que trabajan tanto para la construcción y mantenimiento del detector, como para el análisis físico de los datos recogidos por el mismo.

A diferencia de otros detectores del CERN, CMS fue construido fuera del lugar en el que se encuentra actualmente. Se fabricó en el exterior, dividido en 15 piezas, que posteriormente fueron descendidas y ensambladas cerca de Cessy, Francia, en

el lugar en el que se realizan las colisiones. El hecho de trasladar y ensamblar con muy alta precisión un detector de 13.000 toneladas lo convierte en la mayor obra de ingeniería civil después de la construcción del propio LHC.

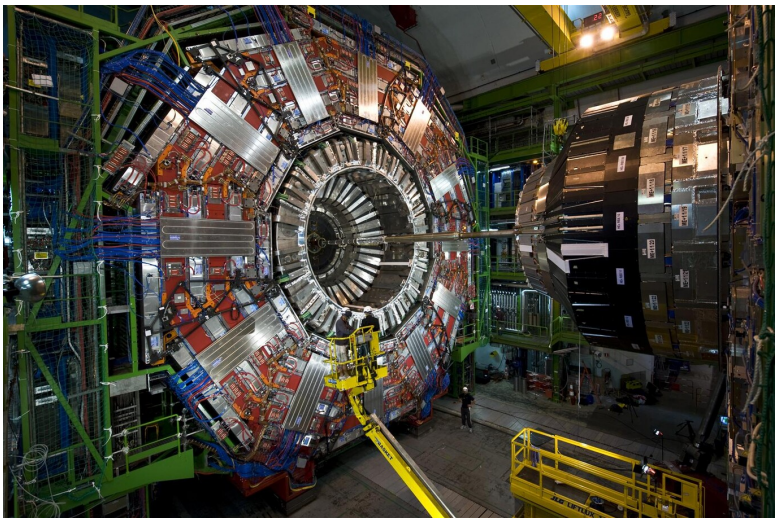


Figura 3.4: Imagen del montaje de CMS. Fuente: [10]

La estructura completa del detector mide $22m$ de largo y $15m$ de diámetro. La pieza principal es el solenoide que rodea las primeras partes del detecor, se puede apreciar en la figura 3.5. Este mide $13m$ de largo y $5.9m$ de diámetro, alcanza los $4T$ (100.000 veces más que el campo magnético terrestre) y está hecho de un material superconductor que hay que mantener refrigerado siempre que se quieran tomar medidas.

Los principales subdetectores de CMS son el tracker, los calorímetros (hadrónicos y electromagnéticos) y los detectores de muones. Se detallará su funcionamiento más adelante en este capítulo.

Comenzaremos por definir el sistema de coordenadas con el que se toman las medidas de CMS.

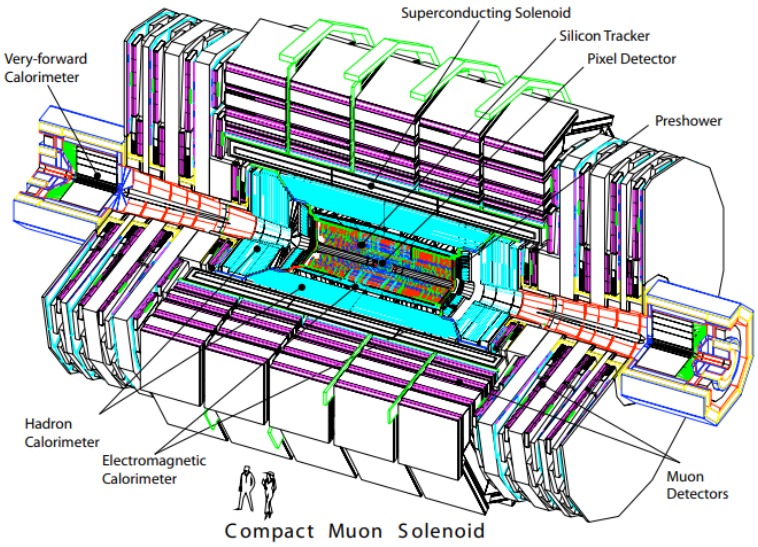


Figura 3.5: Esquema global de la estructura de CMS. Fuente: [11]

3.3.1. Sistema de coordenadas de CMS

El sistema de coordenadas adoptado por CMS tiene el origen en la posición nominal donde ocurren las colisiones. El eje y está orientado hacia arriba y el eje x hacia el centro del LHC. De esta forma, por ortogonalidad se escoge el eje z perpendicular al plano xy en la dirección por la que circulan las partículas.

Se toma también un ángulo polar θ medido a partir del eje z y se define la pseudorapidez $\eta = -\ln\tan\theta/2$, que se prefiere a θ ya que η es un invariante Lorentz. Para completar la descripción cilíndrica del sistema de coordenadas se toma el ángulo ϕ como el formado por la proyección en el plano xy con el eje x.

En la figura 3.6 se puede ver un esquema del sistema de coordenadas descrito.

Este sistema de coordenadas está diseñado en especial para tratar con las colisiones frontales de partículas. En estas, los productos esperan encontrarse principalmente

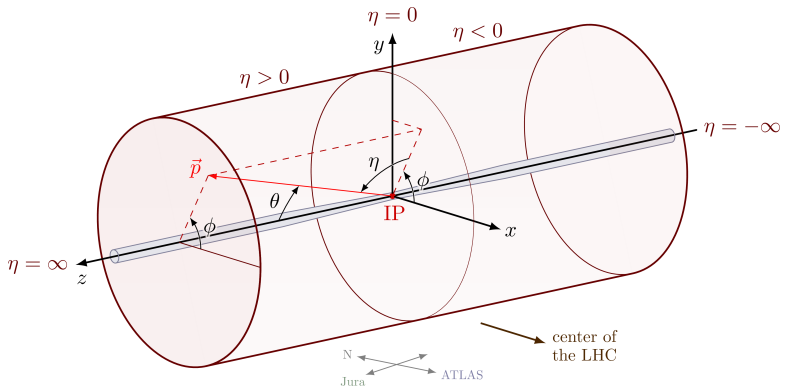


Figura 3.6: Sistema de coordenadas de CMS.

en el plano transverso a la colisión, es decir, el plano xy.

El 4-momento de una partícula es:

$$\mathbf{p} = (E, p_x, p_y, p_z)$$

Que trasladado al sistema de coordenadas particular de CMS se convierte en:

$$\mathbf{p} = (E, p_t, \theta, \phi)$$

Donde se define el momento transverso p_t como $p_t = \sqrt{p_x^2 + p_y^2}$. Las medidas de energía de CMS suelen ir desglosadas también en energía transversa E_t (a partir de p_t) y energía faltante E_t^{miss}, que es la energía que se pierde en la dirección z porque las colisiones no son totalmente frontales.

Para entender la importancia del sistema de coordenadas debemos describir las componentes del detector CMS.

3.3.2. Subdetectores de CMS

- **Tracker**

El sistema Tracker es un conjunto de más de 75 millones de sensores hechos de silicio que tienen el fin de detectar el paso de partículas cargadas. Se ubica en la zona más próxima al punto de colisión y de esta forma es la primera información que se obtiene de la misma. A partir de la activación sucesiva de los pequeños sensores de silicio se puede reconstruir la trayectoria de las partículas en los primeros instantes tras la colisión.

En la figura 3.7 se puede ver la disposición de los sensores del Tracker.

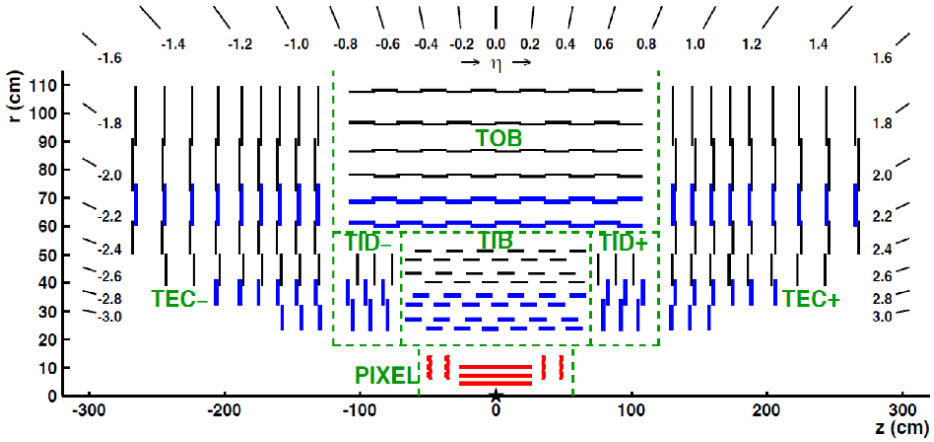


Figura 3.7: Componentes del sistema Tracker de CMS. La representación es una sección del sistema completo que tendría simetría cilíndrica alrededor del eje z. El origen marcado con una estrella es el centro de colisión. Fuente: [12]

Los sensores más próximos a las colisiones (rojo en la figura 3.7) se llaman detectores pixel. Consiste de 3 capas en forma de barril con 2 capas a cada lado como tapas. El tamaño de cada sensor es de $100 \times 150 \mu m$ y se extienden sobre $|z| \leq 46.5 cm$ y r de $4.6 cm$, $7.3 cm$ y $10.2 cm$ para cada capa. El resultado final es

una resolución en z de $20\mu m$ y $10\mu m$ en $r-\theta$.

Los demás sensores se denominan detectores de tiras (Strip Tracker) ya que son mucho más largos que anchos. En la figura 3.8 se pueden ver las diferentes geometrías de los sensores.

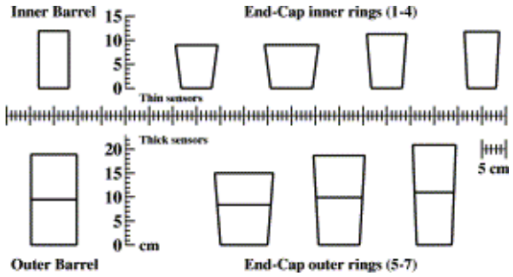


Figura 3.8: Diferentes formas y dimensiones de los sensores del Strip Tracker. Fuente: [13]

En el TIB (Tracker Inner Barrel) y TOB (Tracker Outer Barrel) los sensores se disponen por capas con simetría cilíndrica y su precisión es de $35-52\mu m$ en $r-\theta$ y de $52\mu m$ en z.

En el TID (Tracker Inner Disk) y TEC (Tracker End Cap) se disponen como discos que tapan los cilindros formados por el TIB y el TOB. Su objetivo es detectar partículas con $|\eta|$ elevado.

- **Calorímetro electromagnético**

El calorímetro electromagnético (ECAL) es un conjunto de cristales de tungstenato de plomo ($PbWO_4$) y fotodiodos de avalancha (APD) que tiene como objetivo medir la energía de electrones y fotones.

En la zona del barril se encuentran 61200 cristales y en la zona de las tapas

7324. Estos actuan como centelladores ante las excitaciones producidas por las partículas que interactúan con ellos, es decir, producen una lluvia de fotones muy focalizada. Los APDs detectan estos fotones producidos por los cristales para reconstruir la trayectoria de las partículas.

La elección del material de los cristales es muy importante ya que el experimento CMS requiere un comportamiento muy concreto. El tungstenato de plomo se adapta bien a estos requisitos porque produce unas lluvias de fotones muy concentradas y libera la energía rápido (80 % de la energía liberada en $25ns$). Pero a su vez hay dos problemas principales que fueron resueltos para la implementación en el experimento.

Primero, la emisión de fotones del cristal depende fuertemente de la temperatura a la que se encuentran. Para que el experimento esté bien calibrado se diseñó un sistema de enfriamiento que mantiene el sistema completo a una temperatura que varía solamente en $\pm 0.1^oC$.

Segundo, el tungstenato de plomo produce muy pocos fotones, del orden de $30\gamma/MeV$, y siempre se van a tomar medidas en un campo magnético elevado ($4T$). Por lo que los fotodetectores deben tener una alta ganancia intrínseca y ser resistentes a camposmagnéticos. La solución es el uso de ADPs de silicio en los calorímetros del barril y fototriodos de vacío (VPTs) en los de las tapas.

- **Calorímetro de hadrones**

El calorímetro de hadrones (HCAL), como dice su nombre, tiene como objetivo medir la energía de los hadrones que pasan por estos sensores (protones, neutrones, piones, etc). Gran parte de su trabajo es precisar en la medida de E_T^{miss} y su

diseño está fuertemente influenciado por su localización, mayormente dentro del solenoide.

El material absorbente es latón, ya que tiene una distancia de interacción relativamente corta y no es magnético, por lo que puede estar dentro del solenoide. Además, entre cada capa de latón hay un material centellador acoplado a unas fibras ópticas que recogen la luz emitida por este.

La idea es que cuando un hadrón interaccióne con el latón, este creará una lluvia de partículas que activará el centellador y en la siguiente capa de latón se repetirá el proceso. Este proceso en definitiva se medirá como un *jet* de partículas de las que conoceremos la trayectoria y la energía.

En la figura 3.9 se puede ver un diagrama que señala las diferentes partes del HCAL y la división de las celdas de latón-centellador (llamadas *torres*) presentes en este.

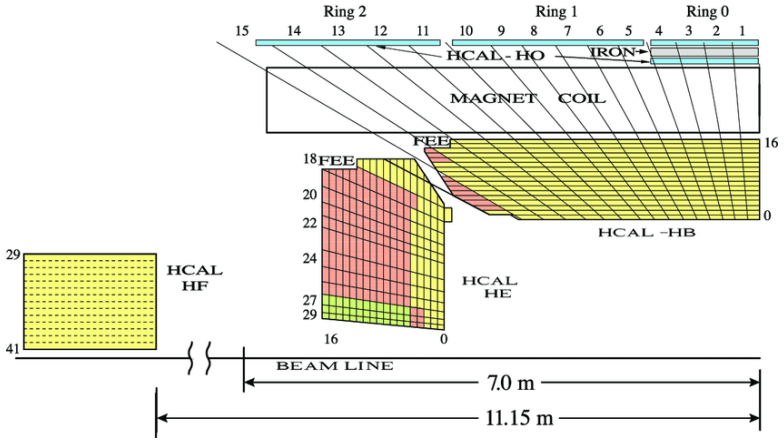


Figura 3.9: Un cuadrante del HCAL. Fuente: CERN

El **Hadron Barrel (HB)**, es la primera parte del HCAL con la que interaccionan partículas. Está compuesto de 2304 torres de dimensiones $\Delta\eta \times \Delta\phi = 0.087 \times 0.087$

y cubre la región $|\eta| \leq 1.4$. Lo forman 15 capas de latón de un grosor de $5cm$.

El **Hadron Outer (HO)** está situado después del solenoide y es una capa de centelladores que se dedica a recoger las partículas que se escapan del HB. Esta pieza del HCAL existe para mejorar la precisión de la medida de E_T^{miss} y cubre la región $|\eta| \leq 1.26$.

El **Hadron Endcap (HE)** cumple la misma función que el HB, pero cubre la región $1.3 \leq |\eta| \leq 3$.

Finalmente, el **Hadron Forward (HF)**, es la pieza más alejada del centro de colisión y cubre la región $3 \leq |\eta| \leq 5$. Está diseñado específicamente para hadrones neutros, que son los que tienen más probabilidad de hallarse en $|\eta|$ elevados. Está compuesto de acero y cuarzo, que detecta las partículas por la luz Cherenkov que generan al interaccionar con las fibras de cuarzo.

Como se puede comprobar, para $|\eta| \leq 5$, el HCAL es completamente hermético, es decir, las partículas no tienen ningún hueco por el que pasar sin ser detectadas. De esta manera, se minimiza la posibilidad de que una partícula generada en la colisión no sea detectada y se piense, por ejemplo, que hay nueva física detrás de esa medida.

- **Sistema de muones**

Como indica el nombre CMS, es imporante detectar muones de manera precisa. De hecho, en este experimento, los muones producidos en el centro de colisión se miden 3 vecen: en el tracker, después del solenoide y en el flujo de retorno.

En la figura 3.10 se analiza la precisión en la medida del momento de los muones

para muones con diferentes valores del mismo.

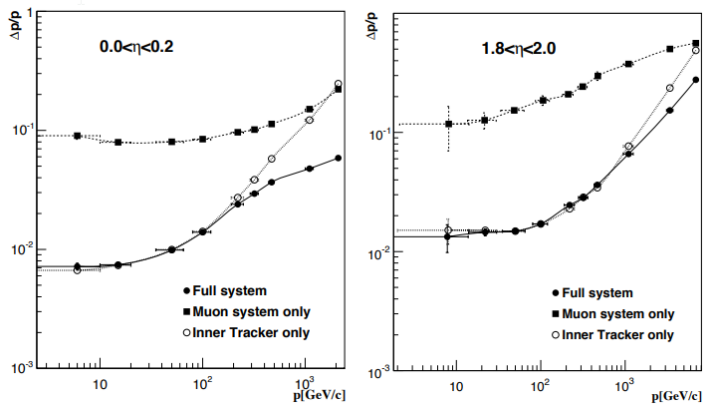


Figura 3.10: Precisión relativa de la medida del momento de los muones usando solamente el sistema tracker, solamente el sistema de muones y ambos. En la primera gráfica se ven datos de medidas en el barril y en el segundo en el endcap. Fuente: [11]

Como se puede ver, los dos sistemas se complementan. Para valores del momento bajos, el tracker domina por un orden de magnitud y es suficiente para conseguir una medida precisa, pero para momentos altos el sistema de muones aporta gran parte de la precisión de las medidas.

En la figura 3.11 se puede ver de manera esquemática la sección de un cuadrante del sistema de muones.

Los diferentes sensores que conforman este sistema son los **Drift Tubes (DT)**, los **Cathode Strip Chambers (CSC)** y los **Resistive Plate Chambers (RPC)**. Como se puede observar, los RPC coinciden con los DT y los CSC para $|\eta| \leq 1.6$ (en la actualidad han sido ampliados a $|\eta| \leq 2.1$). Esto da lugar a una medida relativamente redundante que ayuda a comprobar que el sistema funciona adecuadamente.

Entre los grupos de sensores, se colocan las Return Yoke (rectánculos blancos en

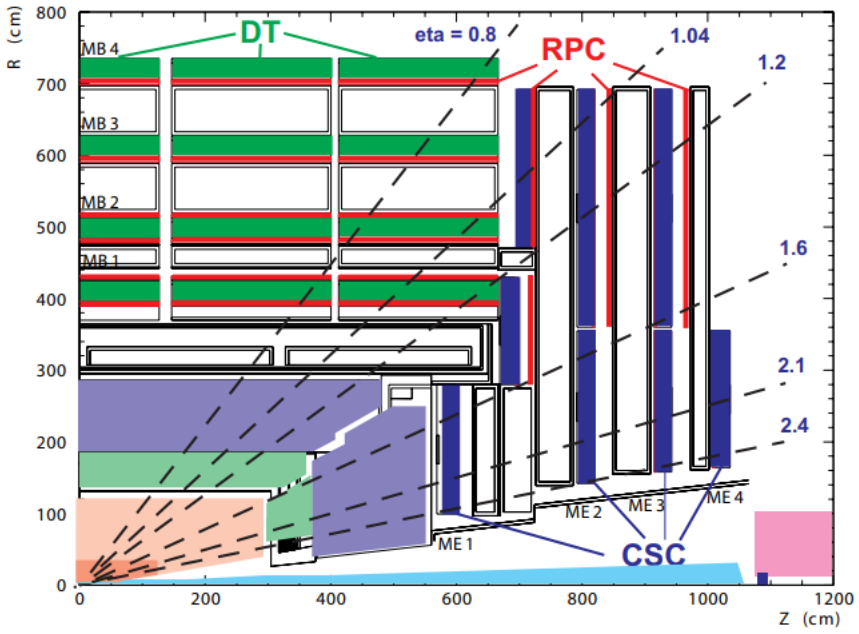


Figura 3.11: Sección del sistema de muones. Fuente: [11]

3.11). Son grandes trozos de acero que sirven como material absorbente para filtrar partículas que puedan alterar la medida de muones (como electrones, hadrones, etc).

Los **DT** son tubos plásticos llenos de gas con un hilo cargado positivamente en el centro situados únicamente en la zona del barril. Cuando un muón pasa por el gas, desplaza electrones de los átomos del mismo que se ven atraídos hacia el hilo, al contactar con este se realiza la medida de la posición del muón. La resolución de la posición es $\approx 200\mu m$. En CMS hay un total de 250 DT y el 30 % fueron construidos en un laboratorio del CIEMAT (Centro de Investigaciones Energéticas, Medioambientales y Tecnológicas) en España.

Los **CSC** están formados por un conjunto de tiras (Strips) de cobre cargadas negativamente (Cathode) y cables cargados positivamente perpendiculares a las

tiras. Están sumergidos en un gas, que por un efecto similar al de los DT generan una avalancha de electrones que generan la medida en la matriz de tiras y cables. La medida de la posición es muy rápida y tiene una precisión de $\approx 200\mu m$. En CMS hay 468 CSC, la mitad en cada *endcap*.

Los **RPC** consisten de dos placas con cargas opuestas que contienen un gas. De nuevo, el proceso es muy similar alos DT y CSC: cuando pasa un muón se produce una avalancha de electrones del gas, que son medidos por una placa metálica que está adherida al ánodo. La medida producida en los RCP es muy rápida, aunque menos precisa que la de los otros sensores del sistema de muones. Se usa principalmente como *trigger* para decidir si los datos de la colisión son suficientemente buenos como para almacenarlos.

En la figura 3.12 se pueden ver ejemplos de las trayectorias de diferentes partículas al pasar por los subdetectores de CMS.

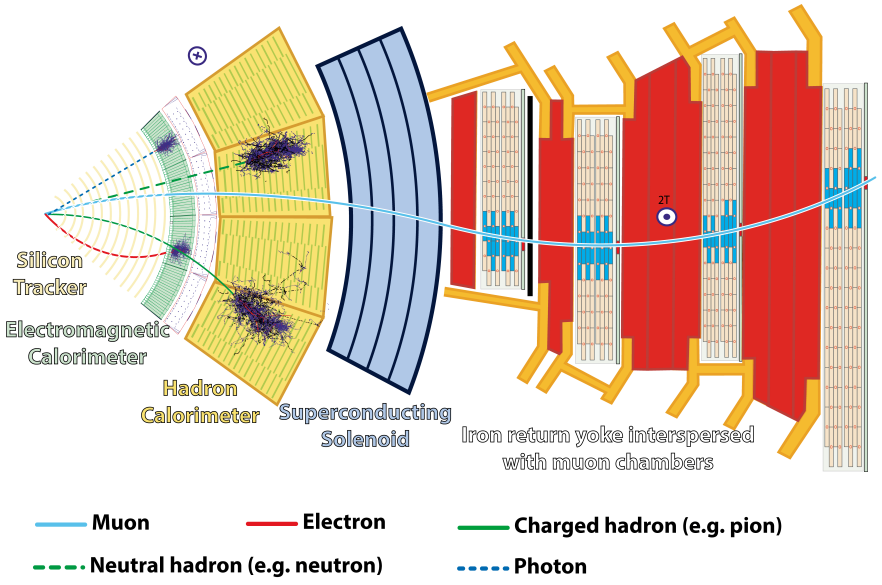


Figura 3.12: Ejemplo de detección de diferentes partículas dentro de CMS. Fuente: CERN

La trayectoria de todas las partículas cargadas se ve curvada por el campo magnético generado por el solenoide, mientras que las neutras, como el neutrón y fotón, siguen lineas rectas. Los electrones y fotones serán detectados por el ECAL y ahí finalizará su trayectoria. Los hadrones sufrirán el mismo proceso en el HCAL. Cualquier partícula que consiga escapar el ECAl y HCAL será interceptada por las Return Yokes del sistema de muones, y en este se espera medir únicamente la trayectoria de los mismos.

Con este esquema, queda determinado el funcionamiento del experimento CMS, pasaremos a describir aspectos de los datos que analizaremos posteriormente.

3.4. Granularidad y observables físicos

La granularidad se define como el detalle de la serie de datos que se esté analizando. En el caso de CMS, los datos se dividen en secciones temporales más o menos largas con las que posteriormente se puede hacer un análisis estadístico general del intervalo en cuestión.

Los datos de CMS se recogen en **Lumisections (LS)**, que duran aproximadamente $23s$. Un conjunto de apoximadamente 500 LS compone un **run**, que no tiene relación con los Run1, Run2 o Run3 definidas previamente.

Los datos recogidos de las colisiones se almacenan como observables físicos (histogramas) que comprenden el momento transverso p_t, la pseudorapidez η, el ángulo azimutal ϕ, y χ^2.

El objetivo de esta clasificación es discriminar las muestras de datos *malos* a partir

de la distribución de estos observables. De esta forma, si se mira colectivamente la distribución de un run completo, se perderá resolución, es decir, posibles LS buenas serán descartadas indebidamente.

3.5. Muestras de datos usadas

En los siguientes capítulos del trabajo se analizarán datos provenientes de 4 épocas (conjunto de runs) que datan del año 2018. En el cuadro 3.1 se puede ver una pequeña descripción de estas épocas.

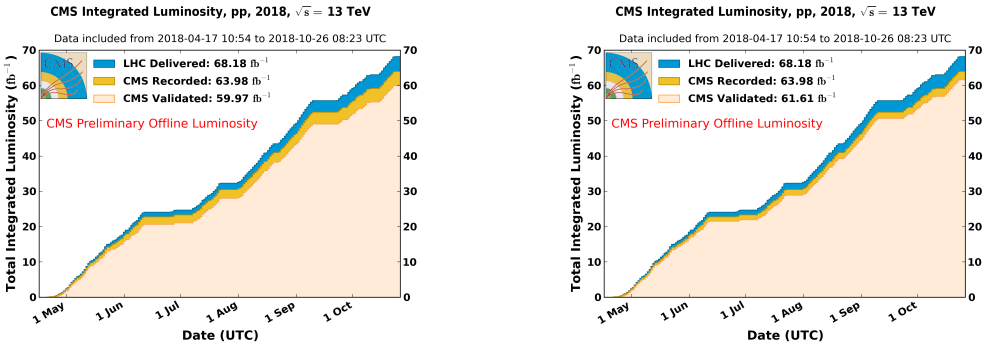
	Run inicial	Run final	E_{CM}(TeV)	Luminosidad integrada (fb^{-1})	Fecha inicial	Fecha final
2018A	315252	316995	13	14.00	26/04/2018	28/05/2018
2018B	317080	319310	13	7.10	28/05/2018	06/07/2018
2018C	319337	320065	13	6.94	08/07/2018	23/07/2018
2018D	320673	325175	13	31.93	01/08/2018	24/10/2018

Cuadro 3.1: Resumen de las épocas que se analizarán en los próximos capítulos.

Cada conjunto de datos está compuesto de histogramas asociados a cada uno de los 4 observables p_t, ϕ, η y χ^2, con una granularidad de LS (ejemplos de todos estos histogramas se pueden ver en la figura 5.1). Estas LS vienen acompañadas de una etiqueta de *bueno* o *malo* que puso un experto al run al que pertenecen.

Para entender el trabajo que se hace en CMS para mantener la calidad de los datos, se presenta en la figura 3.13 la luminosidad integrada *delivered*, *recorded* y *validated* en CMS en las épocas que estudiremos.

En la figura 3.13, *delivered* hace referencia a todas las colisiones que se producen en el punto de colisión de CMS. *Recorded* son las colsiones que se guardan en el disco, ya sean *buenas* o *malas*. *Validated* son las que pasan el proceso de certificación y



(a) (b)

Figura 3.13: Luminosidad integrada de los datos recogidos por CMS durante las 4 épocas que se estudiarán en este trabajo. A la izquierda (a) Validated = 59.97 fb^{-1} incluye certificación de todo el detector, mientras que a la derecha (b) Validated = 61.61 fb^{-1} incluye únicamente los sistemas de detección de muones (cámara de muones y detector de trazas). Fuente: CERN

son útiles para análisis físico. La luminosidad *validated* de 3.13(b) es ligeramente mayor que la de 3.13(a) porque para el estudio de muones no es necesario certificar con el detector completo.

En definitiva, la eficiecia proceso de certificación actual basado en expertos es de 93.7 %. Este número podría variar al implementar técnicas de automatización y aumentar la granularidad del análisis, pero se espera una eficiencia alrededor de 95 % en general.

A continuación se explicarán los procesos y técnicas mediante los cuales se filtran los datos de CMS en función de estos observables y la granularidad.

Capítulo 4

DQM

El grupo de Data Quality Monitoring (DQM) del experimento CMS tiene tres objetivos principales [14] [15]:

- **Monitorización** de los componentes del detector. Avisa cuando uno de estos funciona incorrectamente para que se pueda tomar acción rápidamente.
- **Data Certification (DC)**, un proceso en el que se detecta qué fracción de los datos se tomó mientras el detector no funcionaba correctamente.
- Apoyar a la **depuración** del detector. Devuelve información detallada sobre las posibles causas del comportamiento anómalo del detector, para que los expertos puedan solventar estos problemas más facilmente.

Todas estar tareas son similares y están relacionadas entre ellas, pero difieren en su aplicación. El valor esperado de la **monitorización** y **DC** es en esencia un dato binario que clasifica los datos como *buenos* o *malos*. La principal diferencia

es que el primero se realiza en tiempo real (*online*) durante el funcionamiento del detector, así que debido a la alta frecuencia de datos regristrados se tolera que algunos falsos positivos y negativos se guarden. El DC en cambio tiene como objetivo detectar los datos *malos* con la mayor precisión posible, por lo que los datos pueden tardar hasta semanas en ser procesado por expertos (*offline*).

Como herramienta de *debugging*, el grupo de DQM cuenta con una gran cantidad de gráficos, en concreto histogramas, sobre cantidades medidas por CMS durante un intervalo de tiempo catalogadas según el subdetector que realizó la medida. Los expertos pueden analizar estos gráficos para comprender el origen de los problemas del detector.

En definitiva, el grupo de DQM existe para mantener un buen funcionamiento del detector y asegurarse de que solamente se utilicen datos *buenos* para su aplicación en análisis físicos.

4.1. Flujo de trabajo de DQM

En la figura 4.1 se pueden ver las diferentes componentes del sistema de DQM. La mayor parte del código usado en el proceso de DQM forma parte de CMSSW (CMS software), un repositorio de código abierto que contiene un gran número de herramientas usadas en CMS para adquirir, analizar y validar los datos entre muchas otras funciones. El código se puede encontrar en [16], está escrito principalmente en C++ pero se aplica a través de una interfaz con Python.

El flujo de los datos se organiza en 3 niveles. Primero, los datos son procesados con alguna parte del código orientado a DQM de CMSSW, como en el *High Level*

Trigger (DQM for HLT), reconstrucción de datos offline y validación Monte Carlo. Luego, estos son organizados y representados en DQMGUIs (DQM Graphical User Interface). Entonces, los expertos pueden acceder a las herramientas de DQMGUI para dedicarse al análisis sin tener que lidiar con datos en bruto. Una herramienta adicional notable es el Historic DQM (HDQM) que ayuda a visualizar los datos como una serie temporal para ayudar a detectar patrones en períodos más extendidos de tiempo. Finalmente, los expertos deciden si los datos son *buenos* o *malos* y estos son añadidos al *Run Registry*, que es una base de datos donde se almacenan los runs y LS junto con sus etiquetas.

A partir de los datos almacenados en el *Run Registry*, se genera el *JSON de oro*, un documento que contiene todas las LS asociadas a runs que han sido certificadas como *buenas* para el detector completo. También existe un documento llamado *MUON JSON*, en el que solo se emplea certificación del tracker y cámaras de muones, los únicos subdetectores que intervienen en la medida de muones.

A continuación se describirán brevemente las diferentes componentes de DQM.

El **sistema online de DQM** es un conjunto especial de dispositivos que toma una fracción del HLT para generar elementos de monitorización en tiempo real. Los datos se recogen a una frecuencia de $50 - 100Hz$ y están compuestos de una mezcla de cantidades físicas como p_t, η, ϕ, etc. Los datos se representan durante el funcionamiento de CMS en la sala de control para que los expertos los juzguen y tomen decisiones.

El **sistema offline de DQM** puede considerarse como el siguiente paso después del tramo *online* que siguen los datos de los eventos registrados por CMS. En esta etapa los expertos disponen de un tiempo más prolongado para analizar los runs almacenados en DC. Cuando un run es clasificado como *bueno*, se entiende que

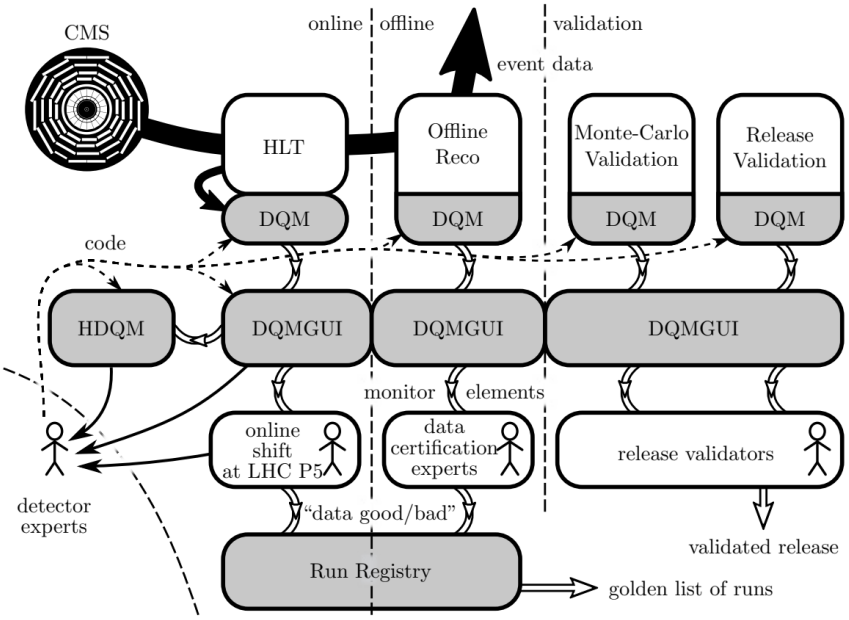


Figura 4.1: Esquema de los sistemas de DQM y el orden en el que se procesan los datos hasta ser almacenados a largo plazo. Fuente: [14].

todas las LS que lo componen son buenas y estas se añaden al *JSON de oro*.

El proceso de **Release Validation** existe para probar nuevos desarrollos en CMSSW para DQM. Si se quiere probar una mejora o un cambio del código usado en DQM, se realiza con datos antiguos que ya han sido analizados o con datos generados por simulacion Monte Carlo de eventos. Una vez comprobado que el código nuevo funciona adecuadamente, este será añadido al repositorio de CMSSW.

Los datos usados en los 3 anteriores sistemas se suben a **DQMGUI** para el análisis de los expertos. Cada una de las componentes de DQM se sube a una instancia independiente de DQMGUI, siendo la *offline* la más extensa. Al hablar de DQMGUI nos referimos al conjunto de servidores que contienen las bases de datos de

las LS, las páginas web asociadas a estos, y las herramientas de visualización que permiten que los expertos los analicen. En la figura 4.2 se puede ver la apariencia de DQMGUI a la hora de analizar datos.

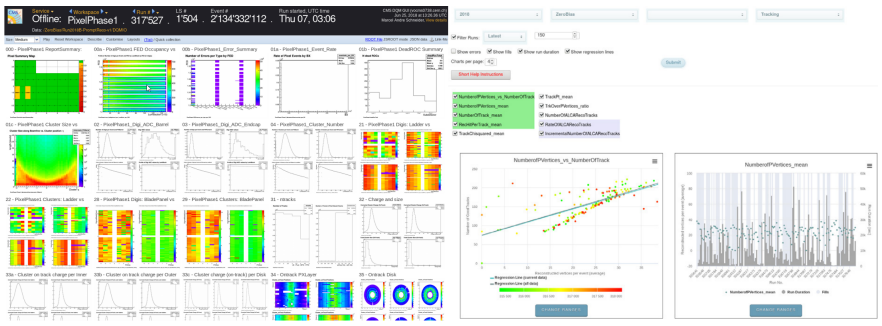


Figura 4.2: A la derecha DQMGUI y a la izquierda HDQM. Fuente: [14].

4.2. Retos actuales y futuros de DQM

En general, el sistema de DQM ha funcionado bien hasta el Run2. Uno de los principales problemas es la **acumulación masiva** de elementos de monitorización. Como es muy sencillo desarrollar nuevos elementos, a veces con el proposito de tener información extra sobre el detector por si acaso es necesaria en el futuro, la cantidad de elementos de monitorización ha crecido mucho en los últimos años. En principio, esto no debería ser un problema porque DQM está diseñado con este propósito.

En un momento de el Run2, este hecho de generar una enorme cantidad de elementos de monitorización causó problemas de memoria en la reconstrucción, debidos a sutilezas informáticas sobre paralelización. Al final este problema de memoria fue solucionado dando marcha atrás y restringiendo la paralelización de los cálculos.

Estos cambios en el código fueron bastante repentinos e invasivos. Lo que nos lleva

al segundo reto planteado para el Run3 y el futuro, mejorar el mantenimiento y organización de CMSSW para DQM. EN esencia, CMSSW funciona de manera secuencial, añadiendo bloques de código sobre los anteriores para incrementar la funcionalidad del repositorio. A lo largo de los años, se ha incrementado mucho el número de bloques en la cadena y se han extendido los anteriores, sin tener en cuenta las dependencias cruzadas entre estos y posibles redundancias. Esta mala práctica en el proceso de mantenimiento puede afectar a areas relacionadas con DQM como la calibración del detector, por lo que gran parte del esfuerzo actual se dedica a documentar y organizar el repositorio de CMSSW.

Gran parte del mencionado esfuerzo de mantenimiento de CMSSW va a ir dedicado en los próximos años a rediseñar el *Run Registry* y DC. En previsión del aumentado flujo de datos que vendrá con HL-LHC y un interés creciente por realizar DC con una granularidad más fina de LS, se están investigando nuevas técnicas que apoyen y agilicen el proceso de DC en el futuro.

4.3. Preparaciones para HL-LHC: automatización de DC

Como ya se ha comentado, actualmente el proceso de DC (y DQM *online*) es realizado por expertos que analizan los histogramas en DQMGUI manualmente en busca de anomalías. Como la cantidad de datos para analizar es muy extensa, DC se realiza *run-by-run* en vez de *LS-by-LS*. Este contexto implica varios problemas.

Primero, **la granularidad es muy baja**. Dentro de un run puede haber LS anómalas que no se detectan y esto afectará al análisis físico. De manera contraria, en un run clasificado como *malo* puede haber LS *buenas*, que podrían aumentar

la cantidad de datos enviados al *JSON de oro*.

Segundo, el proceso se basa en **trabajo humano**. Esto tiene dos inconvenientes: los humanos son relativamente lentos en analizar los datos, y tras una larga jornada inspeccionando histogramas, la fatiga puede causar errores a la hora de juzgar si estos son *buenos* o *malos*.

Además, en el futuro, el Run3 y el HL-LHC presentarán más problemas porque la luminosidad se verá aumentada hasta un factor 10. Por lo que el uso exclusivo de humanos jamás será capaz de analizar estas cantidades de datos, y menos si se tiene en cuenta que se quiere aumentar la granularidad.

Por lo tanto, la propuesta más estudiada para solucionar todos estos problemas a la vez es la aplicación de *Machine Learning* (ML) para automatizar parte del proceso. Claramente, el uso de técnicas de ML tiene la ventaja de ser capaz de procesar datos a una velocidad órdenes de magnitud superior a la humana. Pero un modelo de ML puede cometer errores, por lo que la idea es mantener un equipo de expertos para que analicen los histogramas que el ordenador no sepa clasificar.

A continuación, se abundará en la implementación de ML para DC, y los posibles retos que se presentan a la hora de escoger una estructura de red neuronal y problemas intrínsecos de la detección de anomalías.

Capítulo 5

Redes neuronales aplicadas a DC

El concepto de ML y red neuronal se ha estudiado extensivamente en el pasado para apoyar en la taréa de DC de CMS. Desde la más sencilla SVM (Support Vector Machine) hasta los más complejos CVAE (Convolutional Variational Autoencoder). El problema de decidir que arquitectura debería ser usada en la práctica aún está abierto, pero la tendencia hacia redes neuronales profundas (DNN) está clara.

A la hora de escoger la arquitectura de una red neuronal para resolver una tarea concreta, es muy importante tener en cuenta la estructura de los datos y la naturaleza del problema. Por ejemplo, si se quiere distinguir entre dos tipos de imágenes, es lógico pensar que una salida binaria es óptima, pero si el objetivo es reconstruir trayectorias de partículas, esto ya no será el caso.

En lo que nos concierne, el apartado de DC de CMS requiere en esencia una salida binaria: determinar si el run (o LS) es *bueno* o *malo*. Antes de describir las características que clasifican los datos en estas categorías, deberíamos conocer que tipo de datos se analizará. En nuestro caso, serán histogramas de los diferentes observables que mide CMS para los muones detectados en una LS.

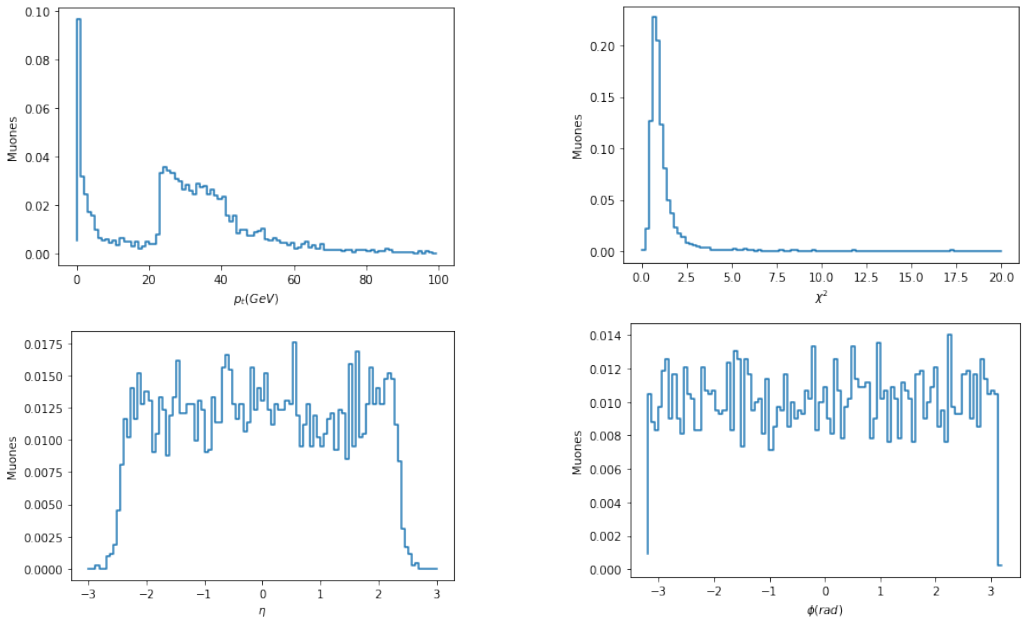


Figura 5.1: Ejemplos de histogramas clasificados como buenos de la serie 2018A para todos los observables que se van a estudiar.

El problema a la hora de automatizar DC es que las anomalías pueden venir de una gran cantidad de fuentes: fallo completo en la toma de datos, fallo de una componente, error en la medida por causas desconocidas, etc. Es decir, intentar discernir la bondad de los datos de forma algorítmica es un proceso extremadamente complejo, ya que habría que tener en cuenta todas las posibles fuentes de error.

Gracias a los avances en computación de las últimas dos décadas, las redes neu-

ronales dedicadas a detección de anomalías pueden ser de ayuda para la taréa en cuestión. En la fase de aprendizaje de la red, esta encontrará las características más relevantes que hacen que un histograma sea *bueno* y lo clasificará como tal. Aún así en la literatura [17] [18] se han encontrado varios problemas con este tipo de datos concreto, que se expondrán a continuación.

5.1. El problema del class imbalance

Una de las características de la muestra de datos es la gran cantidad de datos buenos, ya que en el proceso de DQM *online* se descarta la mayoría de datos anómalos mientras el detector está funcionando. Pero como ya se comentó en la sección anterior, este sistema deja pasar algunos falsos positivos y negativos, por eso es necesario el proceso de DC.

De esta manera, la cantidad de datos anómala es menor del 5 % en general. Este hecho dificulta en gran medida la fase de entrenamiento, ya que la red estará mucho más expuesta a datos *buenos* que *malos*. Por lo tanto, aprenderá de manera precisa las características de los datos *buenos*, pero es posible que no tenga tan en cuenta los datos malos y los clasifique erroneamente.

Existen varias propuestas que pretender solucionar este problema, y una de las más prometedoras es la que se estudiará en este trabajo, los *Auto Encoders* (AE). El principal beneficio que tienen en la tarea de detección de anomalías es que solo se entrenan con datos *buenos*, así que no hay ningún problema con no exponer a la red a datos malos en la fase de entrenamiento.

5.2. El problema del sobreentrenamiento

El sobreentrenamiento es un problema general de las DNN. Ocurre cuando la red 'memoriza' los datos de entrenamiento y no es capaz de generalizar a otras muestras de datos. La capacidad de una red se puede definir como la cantidad de variables que puede aprender, entre los pesos y sesgos de las neuronas y sus conexiones. Si la capacidad es suficientemente grande, es muy probable que la red memorice como clasificar los datos con los que la entrenamos.

Por ejemplo, si se tiene una muestra con 60.000 datos, los cuales pueden tomar 10 valores cada uno, una red con más de 600.000 parámetros para entrenar es capaz de memorizar las características de cada uno de los datos.

El inconveniente del sobreentrenamiento es que, aunque el error en la clasificación de los datos de entrenamiento sea muy bajo, al exponer la red a una nueva muestra de datos que no haya visto nunca, no estará preparada para obtener información relevante y clasificarlo.

Existen muchas propuestas para minimizar el sobreentrenamiento. Lo más básico es reducir el número de neuronas y la profundidad, aunque sería un intercambio por menos potencia a la hora de detectar características en los datos. Otras soluciones más modernas involucran redes variacionales, que muestrean de una distribución estadística en una o varias de sus capas; así la red no puede aprender de los datos, ya que dos instancias de clasificación del mismo dato darán resultados diferentes por el muestreo probabilístico. Una última propuesta interesante para reducir el sobreentrenamiento es la modificación de la función de pérdida: en este caso podríamos penalizar a la red por usar muchas neuronas en una capa dada. Este proceso se llama *sparsity*, y beneficia a la red por usar solo la información

relevante de los datos.

5.3. El problema de las etiquetas

Este es un problema específico de esta muestra. Cada uno de los datos corresponde a una lumisection, y tiene una etiqueta de *bueno* o *malo*. Estas etiquetas realmente corresponden al run al que pertenece la LS, así que no existe una manera certera y directa de saber cuando la LS es *buena* (en un run clasificado como *malo* se consideran los histogramas de todas las LS en conjunto, no individualmente. Puede haber LS buenas en Runs malos y viceversa). Aún así, en el caso real, todas las LS de un *run bueno* se consideran buenas, por lo que es una buena aproximación incluir todas las LS de las *run buenas* en el conjunto de entrenamiento.

5.4. Tipos de entrenamiento de redes mejor adaptados a DC

Al hablar del tipo de entrenamiento de una red, uno se refiere a si esta dispone de las respuestas a la tarea que se le proponga. Por ejemplo, si la tarea es de clasificación, podríamos proporcionarle a la red las categorías de cada uno de los datos, o podríamos no hacerlo y dejas que esta descubriese las diferentes categorías únicamente a partir de los datos.

Se plantean ahora diferentes estrategias de entrenamiento de redes:

- **Supervisado**: Se proporciona la clase o etiqueta que debe aprender la red

para todos los datos de entrenamiento.

- **No supervisado**: No se usa ninguna etiqueta para el entrenamiento. Principalmente se usa este método como una manera más natural de aprendizaje, en el que la red aprenderá las categorías por si misma.
- **Semi-supervisado**: Es un punto medio entre los dos métodos anteriores. Normalmente se usa una gran cantidad de datos sin etiquetar y una pequeña cantidad de datos etiquetados. De nuevo es útil para las mismas tareas que el aprendizaje no supervisado, pero la inclusión de datos etiquetados puede mejorar considerablemente la precisión de las predicciones.

En nuestro caso, ya se explicó la problemática del *class imbalance* y el etiquetado. Por lo tanto es lógico proponer entrenamiento no supervisado para la tarea de DC, donde se proporcionen solo datos *buenos* sin avisar a la red de este hecho. De esta forma podremos por un lado olvidar el problema del *class imbalance* porque solo se usará una de las clases para entrenar, y por otro podremos comprobar cuantas LS de los runs *malos* son realmente *buenas*.

Una de las arquitecturas de red más usadas en aprendizaje no supervisado para detección ded anomalías son los AE. Ya han sido estudiados para DQM con datos de los DT y dan muy buenos resultados cuando se emplean correctamente [17] [19].

5.5. Autoencoders para detección de anomalías

[20]Los AE son una clase de redes muy simple, que se divide en dos partes. Primero, un *encoder* que reduce la dimensión de los datos. El espacio sobre el que se encuentran los datos depués de pasar por el *encoder* se llama espacio latente.

La segunda parte es un *decoder*, que toma valores del espacio latente y aumenta su dimensión hasta la de los datos originales.

A continuación introducimos la notación básica necesaria para definir un AE [20]:

Símbolo	Descripción
$\mathbf{x}$, $\mathbf{h}$, $\mathbf{z}$	Valor de entrada, latente y salida
d_v, d_h	Dimensión de la capa de entrada y latente
$\mathcal{W}_v$, $\mathcal{W}_h$	Pesos de la capa de entrada y latente
$\mathbf{b}_v$, $\mathbf{b}_h$	Sesgos de la capa de entrada y latente
$\mathcal{J}(\cdot)$, $\mathcal{L}(\cdot)$	Función de coste y pérdida
$\sigma(\cdot)$, $\delta(\cdot)$	Funciónes de activación

Un AE, entonces podría expresarse como un conjunto de operaciones lineales, dadas por los pesos $\mathcal{W}$ y los sesgos $\mathbf{b}$ y operaciones no lineales dadas por las funciones de activación $\sigma(\cdot)$ y $\delta(\cdot)$.

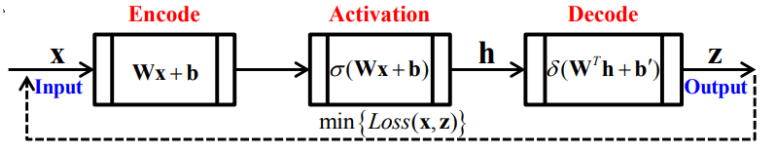


Figura 5.2: Diagrama de flujo general de un AE. Fuente: [20]

Entonces, dado un dato inicial $\mathbf{x} \in \mathbb{R}^{d_v}$, se calcula su representación en el espacio latente como $\mathbf{h} = \sigma(\mathcal{W}_v\mathbf{x} + \mathbf{b}_v)$. Habitualmente se toma $\sigma : \mathbb{R} \to [0, 1]$ y se aplica componente a componente para contener el espacio latente en la caja $[0, 1]^{d_h}$. A continuación se toma el dato $\mathbf{h} \in \mathbb{R}^{d_h}$ y obtiene la salida del AE como $\mathbf{z} = \delta(\mathcal{W}_h\mathbf{h} + \mathbf{b}_h)$.

Como queremos que la entrada $\mathbf{x}$ y la salida $\mathbf{z}$ sean similares, es habitual normalizar $\mathbf{x}$ para que tome valores en $[0, 1]^{d_v}$ y luego tomar $\delta : \mathbb{R} \to [0, 1]$ la segunda

función de activación para forzar $\mathbf{z} \in [0,1]^{d_v}$. Esto es un truco que ayuda a que los parámetros de la red no diverjan, y en casos concretos pueden usarse diferentes funcionen de activación.

Ahora, lo que se necesita para tener un modelo funcional de AE es una manera de determinar los pesos y sesgos $\{\mathcal{W}_v, \mathcal{W}_h, \mathbf{b}_v, \mathbf{b}_h\}$. De forma paralela a cualquier otra estructura de ML esto se hace proponiendo una función de coste $\mathcal{J}$ que se pretende minimizar con los parámetros de la red. La forma habitual de la función de coste para un AE clásico es:

$$\mathcal{J}(\mathcal{W}_v, \mathcal{W}_h, \mathbf{b}_v, \mathbf{b}_h) = \mathcal{L}(\mathbf{x}, \mathbf{z}) - \lambda g(\mathcal{W}_v, \mathcal{W}_h)$$

La primera parte es el error de reconstrucción, y se puede tomar como $\mathcal{L}(\mathbf{x}, \mathbf{z}) = ||\mathbf{x}-\mathbf{z}||_2$, por ejemplo. Cualquier función que mida la distancia entre dos puntos en $\mathbb{R}^{d_v}$ sería una elección válida para $\mathcal{L}$, ya que el objetivo de esta parte de la función de coste es hacer que el AE sea aproximadamente una operación identidad.

El segundo es un término regularizador, que suele definirse como $g(\mathcal{W}) = 1/2(||\mathcal{W}_v||_F^2 + ||\mathcal{W}_h||_F^2)$, donde $||\cdot||_F$ es la norma de Frobenius para matrices. Minimizar este factor ayuda a mantener la magnitud de los parámetros del AE lo más pequeños posibles, y de esta forma se previene en cierta medida el sobreentrenamiento.

Para minimizar $\mathcal{J}$, se suele usar un algoritmo de descenso del gradiente estocástico (SGD), y nosotros usaremos de forma general una modificación de SDG llamado Adams [21].

Como se puede comprobar, hasta ahora solo se ha explicado el desarrollo de un AE de 3 capas, pero si el problema es complejo, lo normal es generalizar el concepto

a una red profunda, que tendría N capas de *encoder* y *decoder* y 1 capa para la representación latente. Entonces, un AE profundo cuenta con $2N + 1$ capas, y es capaz de captar patrones mucho más complejos en los datos que un AE simple de 3 capas. El problema de aumentar tanto el número de parámetros es el sobreentrenamiento, y habrá que tener presentes técnicas que ayuden a minimizarlo.

La idea detrás de la aplicación de AEs a detección de anomalías es entrenarlos únicamente con los datos *buenos*. De esta forma, la red aprenderá las características más relevantes de estos datos y será capaz de reconstruirlos fielmente. Cuando se aplique el AE a datos anómalos, en cambio, la red no estará preparada para reconstruirlos y el valor de la función de error será muy grande. Por lo tanto se define un valor de corte en el error, a partir del cual se considerará que los datos son anómalos.

5.6. El problema del espacio latente

Los datos que queremos analizar con el AE pueden pertenecer a dos clases, *buenos* o *malos*. Esta clasificación define una partición en el espacio de origen $\mathbb{R}^{d_v}$, y por lo tanto, también define una partición en el espacio latente, digamos $\Omega \in \mathbb{R}^{d_h}$.

Cuando entrenamos un AE clásico, no es seguro que esta partición de Ω sea conexa y continua. Es decir, existe la posibilidad de que dos datos muy próximos pertenezcan a clases diferentes, hecho que normalmente se atribuye a sobreentrenamiento. Este hecho se ejemplifica en la figura 5.3.

En nuestro caso, los datos *buenos* serán histogramas muy similares entre si. Por lo tanto, estarán próximos en $\mathbb{R}^{d_v}$ y formarán un elipsoide en este espacio. Entonces,

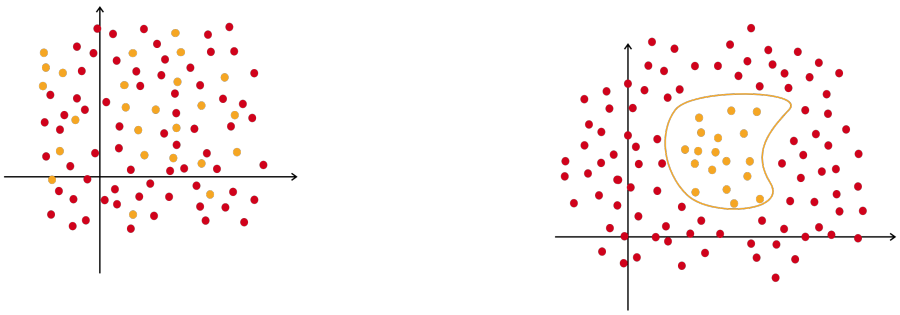


Figura 5.3: Ejemplos de espacios latentes en 2 dimensiones, en rojo la clase *mala* y en naranja, *buena*. A la izquierda se tiene un espacio latente aleatorio. A la derecha, uno determinado y bien clasificado.

el AE debería definir una partición de Ω similar. Lo que buscamos en una división similar a la de la izquierda en la figura 5.3.

Para asegurar que la red producirá un espacio latente con este buen comportamiento, se han desarrollado diversas técnicas que modifican la estructura del AE en mayor o menor medida. En este trabajo se considerará únicamente el caso del Variational Autoencoder (VAE) y una variante simple llamada β-VAE.

5.7. Autoencoder variacional para mejorar el espacio latente

La idea fundamental detras del VAE, es cambiar la naturaleza puntual de la representación en el espacio latente de los datos por una probabilística. De esta forma, un único dato $\mathbf{x}$ tiene asociado en el espacio latente una distribución de la que se tomará una muestra. La red deberá estar preparada para poder reconstruir la imágen original a partir de un muestreo de esta distribución, por lo que los pun-

tos cercanos en el espacio latente necesariamente devuelven resultados cercanos en el espacio original. Esta regularización impide que ocurra lo que se enseñó en la figura 5.3.

La teoría detrás de los VAE se basa en la Inferencia Bayesiana y la Inferencia Variacional [22]. Primero, suponemos que existe una variable h que genera x, como si fuese una codificación de x. Nosotros conocemos x, pero queremos inferir las características que tendría h a partir de los datos conocidos. Es decir, buscamos $p_\theta(h|x)$, la distribución de h condicionada por x con parámetros θ. Según el teorema de Bayes, se puede escribir:

$$p_\theta(h|x) = \frac{p_\theta(x|h)p_\theta(h)}{p_\theta(x)} \tag{5.1}$$

Pero resulta que el cáculo de $p_\theta(x)$ no se puede realizar analíticamente en general, por lo que la solución natural es aproximar $p_\theta(h|x)$ por otra distribución $q_\phi(h|x)$ conocida y optimizar la aproximación usando métodos de Inferencia Variacional.

La distancia entre dos distribuciones estadísticas se suele definir como la divergencia de Kullback-Leiber D_{KL}, y en nuestro caso se buscará el conjunto de parámetros ϕ que minimizan:

$$\min_\phi D_{KL}(q_\phi(h|x)||p_\theta(h|x)) = \min_\phi \mathbb{E}_{q_\phi(h|x)} \log \frac{q_\phi(h|x)}{p_\theta(h|x)} \tag{5.2}$$

Este problema se puede reescribir en términos más simples como un problema de maximización del *evidence of lower bound* (ELBO),

$$ELBO = \mathbb{E}_{q_\phi(h|x)} \log p(x|h) - D_{KL}(q_\phi(h|x)||p_\theta(h)) \tag{5.3}$$

El primer término se entiende como la probabilidad de reconstrucción, que queremos maximizar para que el VAE sea aproximadamente una transformación identidad. El segundo término ayuda a asegurarse que la distribución q_ϕ se aproxime a la distribución a priori p_θ.

En la práctica, el *encoder* aprenderá a encontrar h a partir de x, es decir $q_\phi(h|x)$, mientras que el *decoder* aprenderá lo contrario, $p_\theta(x|h)$. Mientras z sigue una distribución estadística a priori que se puede escoger, normalmente, como una Gaussiana $h_j \sim \mathcal{N}(\mu_j, \sigma_j)$ para cada componente j de z. Entonces, la función de coste a minimizar se puede definir como:

$$\mathcal{J}(\mathbf{x}, \mathbf{z}, \phi) = \mathcal{L}(\mathbf{x}, \mathbf{z}) + \sum_j D_{KL}(q_{j,\phi(h|x)}||\mathcal{N}(\mu_j, \sigma_j)) \tag{5.4}$$

Donde $\mathcal{L}$ es una métrica en $\mathbb{R}^{d_v}$ (por ejemplo $||\cdot||_2$) y j son las componentes del espacio latente. En la figura 5.4 se puede ver un esquema de la estructura del VAE.

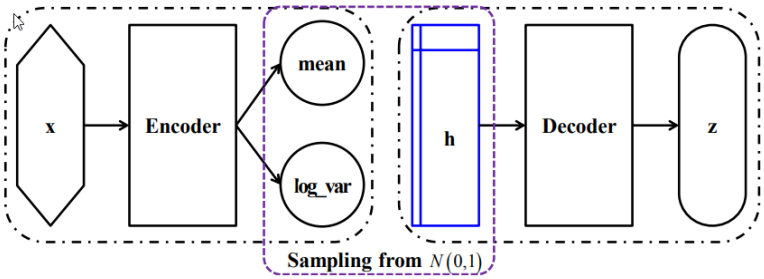


Figura 5.4: Diagrama de flujo de un VAE con distribución a priori $\mathcal{N}(0, 1)$. Fuente: [20]

En la capa central donde ocurre el muestreo probabilístico, hay un detalle sutil pero clave para poder entrenar esta red con un método de descenso del gradiente y *backpropagation*. Los parámetros que se quieren aprender son $\phi = \{\mu_j, \sigma_j\}_j$, por lo tanto hay que encontrar una expresión analítica de la distribución de la que se muestrea en función de estos parámetros. Esto es posible gracias a la reparametrización de la variable aleatoria. Es decir, h_j se expresa como $h_j = g_j(\epsilon)$, con $\epsilon \sim \mathcal{N}(0,1)$ y $g_j(\epsilon) = \mu_j + \sigma_j\epsilon$. Entonces, $h_j \sim \mathcal{N}(\mu_j, \sigma_j)$ pero se puede calcular su derivada con respecto de μ_j y σ_j facilmente.

Como ya se ha comentado, el VAE soluciona el problema del sobreentrenamiento del espacio latente, convirtiendolo en una partición conexa y continua de $\mathbb{R}^{d_h}$. Pero aún así, puede ocurrir que este espacio latente esté *entangled* (entrelazado), es decir, que las clases se superpongan demasiado y sus fronteras estén difusas. Esto puede causar confusión en la detección de anomalías porque habrá una región de confusión en la que la red decidirá de manera prácticamente aleatoria si los datos son *buenos* o *malos*.

La solución más simple surge al considerar los β-**VAE**, un tipo de VAE que añade un hiperparámetro $\beta > 1$ que da más peso a la divergencia KL. Diversos estudios han comprobado que dar más importancia a este término de la funcion de coste genera un VAE más *disentangled* (desentrelazado, separado) que la red original. En este caso se tendría,

$$\mathcal{J}(\mathbf{x}, \mathbf{z}, \phi) = \mathcal{L}(\mathbf{x}, \mathbf{z}) + \beta \sum_j D_{KL}(q_{j,\phi(h|x)} || \mathcal{N}(\mu_j, \sigma_j)) \tag{5.5}$$

El algoritmo de entrenamiento de la red sería exactamente el mismo que el VAE clásico, pero una elección adecuada de β suele dar mejores resultados y una cla-

sificación de anomalías más robusta.

Los últimos avances presentados por [19], sugieren un modelo de Conditional VAE (CVAE), que define una red con las propiedades del β-VAE sin tener que modificar manualmente ningún hiperparámetro. Aunque en su investigación se concluye que el CVAE es un modelo suficientemente robusto como para enviarlo a producción e incorporarlo al proceso de CMS (en su caso tratan con problemática de HLT, no DC), los retos que se encuentran a la hora de entrenar la red y proponer un algoritmo de clasificación de anomalías nos llevan al siguiente apartado.

5.8. Cómo entrenar un VAE y cómo aplicarlo a detección de anomalías

Parte de la discusión sobre la introducción de VAE en DC trata sobre cómo entrenar la red y qué métricas usar para la clasificación. Aunque, como ya se comentó, el problema del *sobreentrenamiento* está casi resuelto al introducir un muestreo probabilístico, hay estudios que sugieren que entrenar un VAE para detección de anomalías solo con datos *buenos*, como haríamos con un AE clásico, causa un *sobreentrenamiento* a la distribución normal. Este comportamiento hace que pierda buenas propiedades el VAE porque no se está realizando correctamente la aproximación variacional $q_\phi(h|x) \approx p_\theta(h|x)$.

Una solución propuesta a este problema es introducir una pequeña cantidad de datos *malos* en la fase de entrenamiento y resulta que así se soluciona parcialmente este problema. En cuanto a las métricas que se pueden usar para clasificar los datos, es similar a los AE clásicos. Una opción es usar la función de pérdida $\mathcal{L}(\mathbf{x}, \mathbf{z})$, y en algunos casos se plantea usar D_{KL}. [19] Propone usar ambos criterios

unidos mediante un OR lógico, es decir, si uno o ambos superan el valor de corte, el dato se considera *malo*.

Otra idea para la fase de entrenamiento es exponer a la red a un número elevado de datos *malos* y *buenos*, y ver si un valor alto de β para un β-VAE genera un modelo en el que las dos clases están bien separadas en el espacio latente. Esto se explorará en varios experimentos posteriormente y es más una técnica exploratoria que una solución de producción para DC.

Capítulo 6

Métricas para la evaluación de redes neuronales

Ya se ha comentado la estructura de los modelos que se van a estudiar en este trabajo. Ahora, definiremos las técnicas utilizadas para comparar estos modelos.

Habitualmente, el conjunto de datos con los que se puede entrenar la red se divide en tres partes, el **conjunto de entrenamiento**, el **conjunto de prueba (test)**, y el **conjunto de validación**. En nuestro caso, tomaremos solo uno de entrenamiento y uno de validación, ya que nuestro objetivo no es un modelo de producción y no necesitamos realizar pruebas extensivas.

El conjunto de validación es una parte de los datos con la que no se entrena la red, y por lo tanto será una buena manera de comprobar cómo generaliza la red a datos que nunca ha visto. Ya que la red puede aprender carácterísticas particulares de la muestra de entrenamiento, se usarán varias métricas que una vez

calculadas sobre el conjunto de entrenamiento y validación se pueden comparar. Si la red presenta sobreentrenamiento, estas métricas tendrán valores dispares para los dos conjuntos, mientras que un modelo que gneraliza bien presentaría valores similares.

A continuación se definirán las métricas mencionadas que serán útiles para evaluar el funcionamiento de los modelos.

6.1. Matriz de confusión

La matriz de confusión, también conocida como tabla de contingencia en un contexto más general, es una métrica usual para modelos de clásificación. Será una tabla $N \times N$, en la que las filas representas las clases reales y las columnas, las predichas. En nuestro caso particular, la matriz de confusión será 2×2 y sus elementos serán los *verdaderos positivos* (TP), *verdaderos negativos* (TN), *falsos positivos* (FP), y *falsos negativos* (FN). Los últimos dos valores se conocen como errores de tipo I (FP) y tipo II (FN). Un buen modelo de clasficación tendrá errores muy bajos, por lo que la matriz de confusión será principalmente diagonal.

Pero no es suficiente inspeccionar la diagonal principal, ya que en modelos no balanceados, como el nuestro, una clase será mucho más común que la otra. Por lo tanto, un modelo que clasifica todo como *bueno*, no cometería ningún error de tipo II y un porcentaje muy bajo de tipo I, pero estaría clasificando mal todos los datos *malos*. De esta forma, definen varias métricas a partir de la matriz de confusión que apoyan a la evaluación de modelos de clasificación:

- **Accuracy**: Mide la cantidad de aciertos frente al total, lo que es muy similar

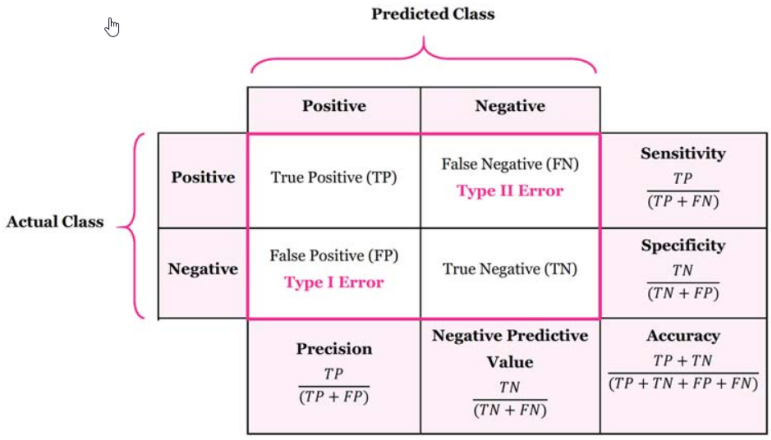


Figura 6.1: Matriz de confusión para una variable dicotómica y métricas más importantes. Fuente: [23]

a ver como de diagonal es la matriz de confusión. Por este motivo, no es una buena métrica para determinar si modelos no balanceados son buenos.

- **Precisión**: Mide cuantos de los datos clasificados como positivos por el modelo son realmente positivos. Es muy importante en nuestro caso ya que queremos minimizar todo lo posible el número de falsos positivos.

- **Sensibilidad**: Mide la cantidad de datos *buenos* bien clasificados por el modelo. $TP + FN$ es la cantidad total de datos *buenos* en la muestra de datos mientras que TP es la cantidad de datos *buenos* bien clasificados por el modelo.

- **Especificidad**: Es equivalente a la sensibilidad pero para los datos *malos*.

- **Valor F1**: Es una métrica importante que une la precisión y la sensibilidad del modelo mediante la media armónica. Es decir,

$$F_1 = \frac{2}{\frac{1}{precision} + \frac{1}{sensibilidad}}$$

La métrica más importante para nuestro problema es el valor F1. Por un lado, queremos que la precisión sea lo más alta posible para asegurarnos de que el mínimo número posible de datos malos se escapan en el proceso de DC. A la vez, es conveniente tener un valor elevado de sensibilidad, ya que esto indicará que no descartamos muchos datos *buenos*.

En la realidad, no es suficiente tener un valor elevado de F1, ya que se le da más importancia a la precisión que a la sensibilidad. Una opción es introducir pesos en la media harmónica, pero no existe un convenio para elegirlos y por lo tanto tampoco sería de gran ayuda. Lo habitual es juntar los valores de precisión y F1 con la curva ROC, que se explicará a continuación.

6.2. Curva ROC

[24]La curva ROC (*Receptor Operator Characteristic*) representa la tasa de falsos positivos (FPR, $1 - especificidad$) y verdaderos positivos (FNR, *sensibilidad*), en función de un valor de corte. En nuestro caso, por ejemplo, el valor de corte será un valor de la función de pérdida (error cuadrático medio) a partir del cual clasificaremos los datos como *malos*.

Normalmente la curva ROC se representa junto a una recta de pendiente 1, que representaría la curva ROC asociada a un modelo completamente aleatorio.

Por último, el área bajo la curva (AUC) de la curva ROC, es un buen parámetro que indica el buen comportamiento de un modelo. Cualquier valor superior a 0.5 significa que el modelo es mejor que asignar clases aleatoriamente, y para que un modelo sea realmente útil en un entorno de producción se espera un valor de AUC

al menos de 0.9.

Otra ventaja de usar la curva ROC es que da una idea visual de que valor de corte tomar para el problema dado. En nuestro caso debemos elegir un punto que cumpla que FPR sea muy bajo y TPR sea muy alto. Si el modelo no permite esto no será adecuado para usarse en un entorno de producción.

Un ejemplo de curva ROC se puede ver en la figura 6.2

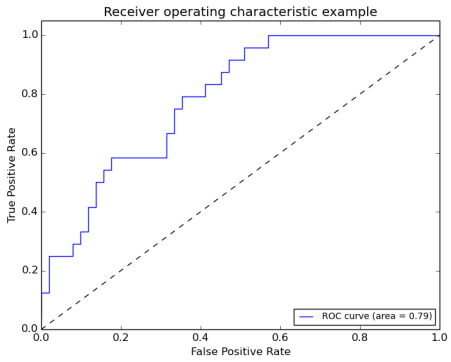


Figura 6.2: Ejemplo de una curva ROC y su AUC. Fuente: [25]

Capítulo 7

Aplicación de técnicas de aprendizaje automático para detección de anomalías en DC

En este capítulo finalmente utilizaremos todas las técnicas expuestas en el capítulo anterior para intentar resolver el problema de automatización de DC. Primero se describirán en detalle las muestras de datos que analizaremos, para luego aplicar los AE y VAE para detección de anomalías.

Las muestras de datos pertenecen a varios runs que ocurrieron durante el año 2018. Contamos con 4 muestras correspondientes a las épocas 2018A, 2018B, 2018C, 2018D que cuentan con histogramas de aproximadamente $50,000$ LS para los observables p_t, χ^2, η y ϕ.

7.1. Descripción y pre-procesamiento de los datos

Las muestras de datos con las que trataremos contienen los siguientes datos sobre cada LS:

- **Fromrun**: Es un número entero que identifica la run a la que pertenece la LS.
- **Fromlumi**: Un número entero que identifica la LS dentro de la run correspondiente.
- **Labels**: Etiqueta que se le ha dado a la run que la clasifica como *buena* (True) o *mala* (False). Más adelante se abundará en la fiabilidad de esta eqtiqueta al analizar los datos con granularidad de LS.
- **Hname**: Representa la variable correspondiente al histograma. Por ejemplo $GlbMuon\_Glb\_pt$ significaría momento transverso p_t de muones globales.
- **Histo**: Contiene los valores de las cuentas del histograma asociado a la variable en cuestión para la LS señalada.
- **Entries**: Número total de cuentas registradas en la LS.
- **Xbins**: Número de divisiones en el histograma.
- **Xmin, Xmax**: Valor mínimo y máximo para las divisiones del histograma.

La estrategia de pre-procesado de los datos depende mucho del problema en cuestión. Normalmente se intentaría eliminar el máximo número de datos obviamente malos (entradas vacías por ejemplo) de la muestra de datos para no confundir al modelo. Pero en nuestro caso se ha estudiado que un número pequeño de *outliers*,

es decir, datos que no siguen la norma, puede reducir considerablemente el sobre-entrenamiento.

De esta forma, se opta por eliminar únicamente los datos obviamente mal etiquetados. En la figura 7.1 se puede ver la distribución de datos *buenos* y *malos* en función del número de entries.

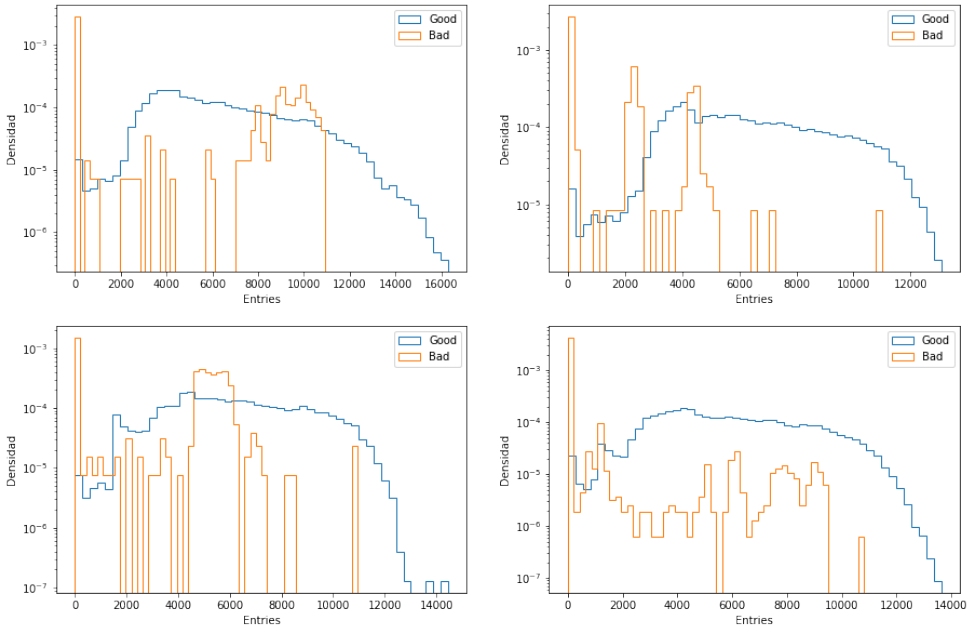


Figura 7.1: Densidad de datos *buenos* y *malos* en función del número de entries para las 4 series de datos que tenemos.

Como se puede observar, la mayoría de datos *malos* tienen menos de 2000 entries por LS, mientras que los buenos tienen la mayoría más de este número. Por lo tanto, al eliminar los histogramas con menos de 2000 entries no estamos perdiendo una cantidad sustancial de histogramas y estamos evitando que la red aprenda características poco importantes de los mismos. En la figura 7.2

Los histogramas restantes que corresponden a la categoría *bad* serán en su mayoría

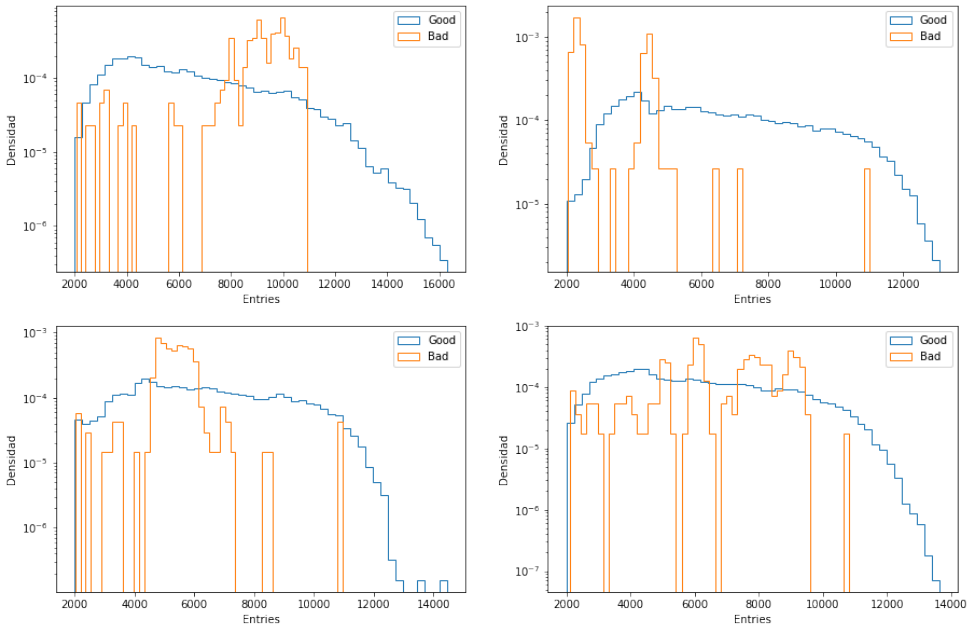


Figura 7.2: Densidad de datos *buenos* y *malos* en función del número de entries para las 4 series de datos que tenemos una vez se aplica el filtro $entries \geq 2000$.

interesantes para validar la red, ya que su distribución no será obviamente mala. Además se puede ver que las distribuciones de la figura 7.2 son distintas entre sí, por lo que podemos asegurar que continen anomalías de origenes distintos. De esta forma, a la hora de comprobar el funcionamiento del AE que entrenemos, podremos ver si actúa mejor detectando anomalías de algúna muestra de datos en particular.

Otro beneficio de aplicar un filtro tan simple es que dejamos que ciertos datos mal etiquetados penetren en el período de entrenamiento de la red. Como se mencionó antes, esto ayudará a reducir el sobreentrenamiento.

Aplicando el el filtro $entries \geq 2000$ se obtienen las siguientes muestras de datos:

Época	Bueno	Malo	Proporción de malos	Media de entries
2018A	51050	248	0.48 %	6284
2018B	28166	233	0.82 %	6394
2018C	26293	396	1.51 %	6238
2018D	123205	589	0.47 %	5894

Cuadro 7.1: Descripción estadística del número de entries por LS para cada época.

Otra manipulación que suele aplicarse antes de entrenar una red es normalizar los datos en alguna norma. En nuestro caso concreto usaremos una normalización en la norma l^1, lo que significa normalizar por la suma del valor absoluto de cada componente. Es decir, si $x = (x_i)_{1 \leq i \leq N}$, se puede definir la norma l^1 como,

$$||x||_{l^1} = \sum_{i=1}^{N} |x_i|$$

Por lo tanto, como los datos pertenecientes a histogramas tienen componentes positivas, normalizar mediante la norma l^1 convierte los histogramas en densidades de probabilidad. Normalizar los datos suele ayudar a que el modelo converja adecuadamente y es un procedimiento estándar en ML.

Una vez pre-procesados los datos, podemos pasar a entrenar las redes.

7.2. Construcción del AE

En lo relativo a la construcción de modelos de ML se usará el paquete de python **TensorFlow** [26] y en especial el módulo **Keras** [27] del mismo. Para el tratamiento de las muestras de datos se usará **Pandas** [28] y para el tratamiento

numérico **NumPy** [29]. Para el preprocesado y cálculo de métricas como la matriz de confusión y curvas ROC se usará **Scikit-learn** [25]. Para las representaciones gráficas se usará el paquete **Matplotlib** [30].

En la sección 5.5 ya se dieron las guías generales sobre como entrenar un AE para detección de anomalías. En el caso concreto que nos incumbe, se tienen 4 observables diferentes en los que se pueden encontrar anomalías, y dependiendo del error, pueden encontrarse en una o más variables para que se considere una LS anómala. Por lo tanto, se ha decidido que la mejor opción es entrenar 4 AEs diferentes para cada una de las variables que aprenderán a reconstruir las mismas.

Luego, para cada LS se tendrá un error a partir de la función de pérdida entre el dato original y el reconstruido para cada una de las variables, y este se espera del mismo órden para todas ellas ya que están normalizadas. El error de la LS entonces se define como el máximo de estos 4 errores y la LS se clasificará como anómala si este valor es superior a un corte que se definirá más adelante.

En la figura 7.3 se puede ver un esquema del proceso descrito.

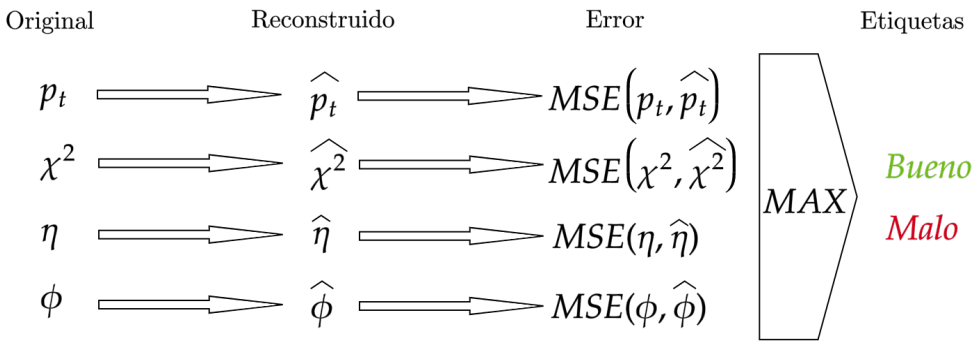


Figura 7.3: Diagrama de flujo para obtener etiquetas a partir de los datos originales.

En este trabajo se quiso explorar el rendimiento de los AE dependiendo de la dimensión latente escogida. Por lo tanto, en un paso inicial se entrenaron 4 AE con 128 y 64 nodos conectados a una capa latente de dimensión 2, 4, 8 y 16. Estos AE se entrenaron exclusivamente con el 80 % de los datos *buenos* de la muestra 2018A[1] y se validaron sobre el restante de datos *buenos* y la totalidad de datos *malos*. De esta forma, se obtienen las siguientes curvas ROC:

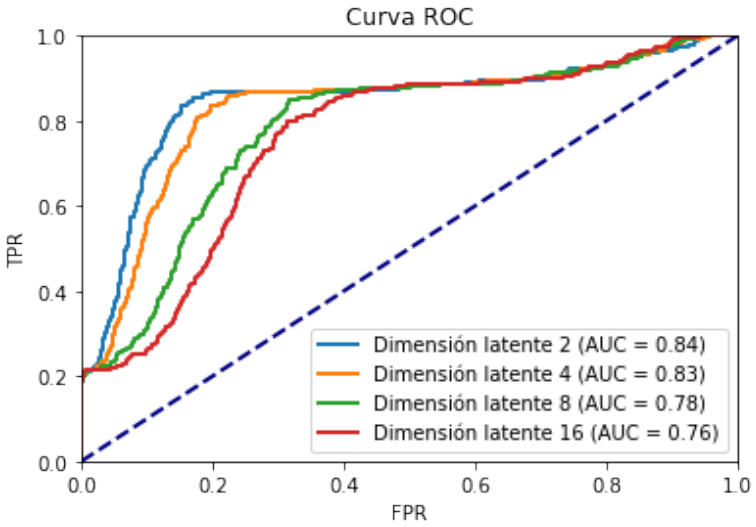
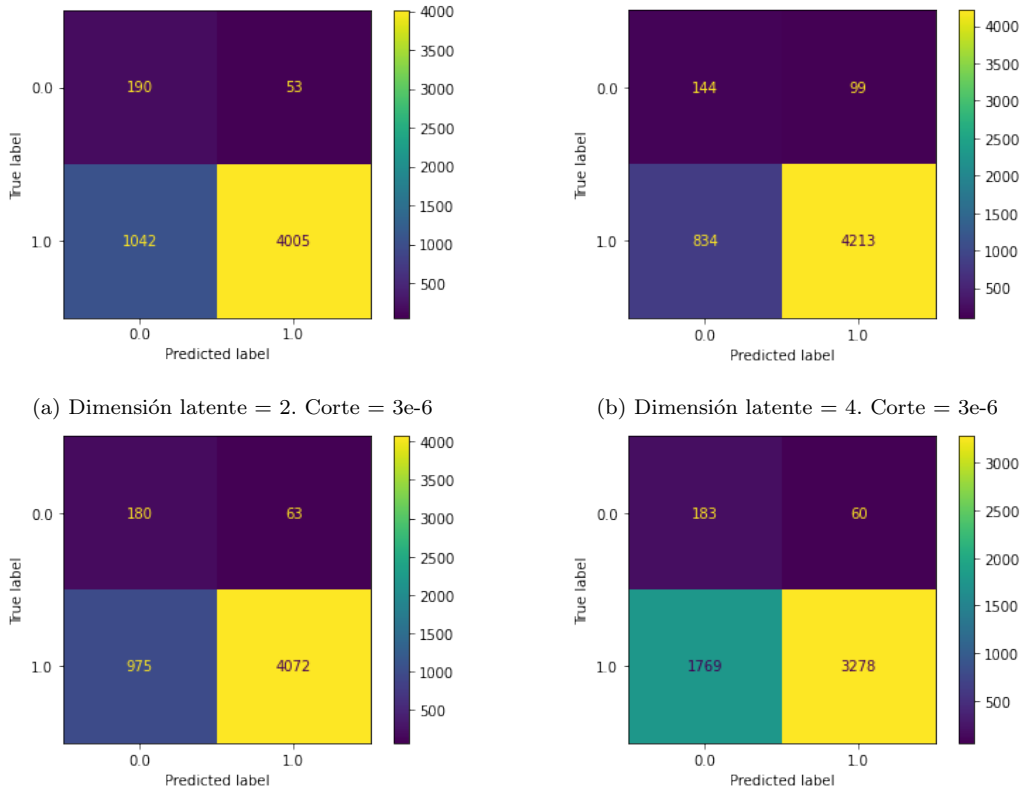


Figura 7.4: Comparación de las diferentes curvas ROC y AUC para los diferentes valores de dimensión latente explorados.

Para obtener las respectivas matrices de confusión se tienen que escoger valores concretos para el corte del MSE. Para este experimento se elegirán cortes que devuelvan una tasa de falsos positivos menor que 0.25.

Como se puede ver en la figura 7.5, los AE con dimensión latente 2, 4 y 8 son muy

[1] La elección concreta de la época sobre la que entrenar el AE no debería tener efectos notables sobre su rendimiento, ya que los datos *buenos* son de la misma naturaleza en todas las épocas. Otra buena decisión sería entrenar con la muestra 2018D ya que es la más amplia, pero se decidió usar esta para analizar posteriormente el funcionamiento del AE entrenado.



(a) Dimensión latente = 2. Corte = 3e-6

(b) Dimensión latente = 4. Corte = 3e-6

(c) Dimensión latente = 8. Corte = 2.8e-6

(d) Dimensión latente = 16. Corte = 2e-6

Figura 7.5: Comparativa de la matriz de confusión para cada AE entrenado con diferentes valores de corte para el MSE.

similares en su rendimiento, mientras que el de dimensión 16 funciona claramente peor por su gran tasa de falsos negativos. Podemos ahora analizar las métricas propuestan en la sección 6.1 para analizar en detalle que estructura de AE es mejor.

Dimensión latente	Precision	Especificidad	Sensibilidad	Exactitud	F1
2	0.99	0.78	0.79	0.79	0.88
4	0.98	0.59	0.83	0.82	0.90
8	0.98	0.74	0.81	0.80	0.89
16	0.98	0.75	0.65	0.65	0.78

Cuadro 7.2: Métricas obtenidas a partir de las matrices de confusión de la figura 7.5 para las distintas dimensiones latentes.

A partir de las métricas del cuadro 7.2 queda claro que los AE entrenados con dimensión latente 4 y 16 son los que peor han puntuado, el de 4 por su baja especificidad y el de 16 por su baja sensibilidad y exactitud. En cambio, las redes entrenadas con dimensión latente 2 y 8 presentan un rendimiento muy bueno. En lo siguiente se estudiará en concreto cómo generaliza la red de dimensión latente 8 a los demás muestras de datos para ver si podría ser utilizada en el caso real de DC de CMS.

7.3. Estudio del AE con dimensión latente 8

Para estudiar el rendimiento del AE con dimensión latente 8 que se entrenó en la sección anterior se calculará la curva ROC y matriz de confusión de esta sobre los demás muestras de datos: 2018B, 2018C, 2018D. En este caso tendremos un conjunto de histogramas de LS más realista, ya que cuentan con más datos clasificados como *buenos* (en la anterior sección el 80 % se utilizaron en el entrenamiento y no en la validación).

Primero, se calcula la curva ROC las muestras usando el mismo método descrito en la sección 7.2, como se puede ver en la figura 7.6.

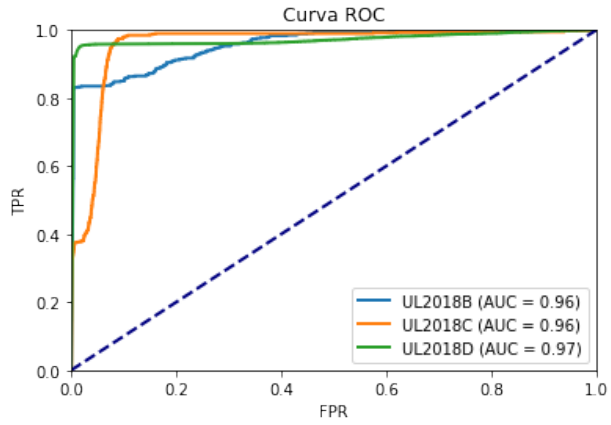


Figura 7.6: Curvas ROC y AUC asociadas a cada uno de los muestras de datos restantes. Para su cálculo ya se ha tenido en cuenta el máximo del MSE entre el dato original y el reconstruido para los 4 observables como valor de corte.

El AE tiene un rendimiento mejor que el observado en la figura 7.4 para todos los muestras de datos. Esto es debido en gran medida a la mayor cantidad de datos *buenos* que contribuyen a aumentar la AUC porque el AE está preparado para clasificarlos correctamente.

Para encontrar un buen valor de corte para el MSE, podemos calcular la sensibilidad frente al parámetro de corte del modelo sobre cada muestra de datos. Esta gráfica se puede ver en la figura 7.7.

Como se puede ver en la figura 7.7, se puede dividir cada una de las curvas en 4 escalones. El primero, para un corte del orden de 10^{-6}, representa un modelo que clasifica casi la totalidad de los datos buenos correctamente. Los demás escalones se pueden asociar a los distintos tipos de error que se puede encontrar la red. Sería interesante en un estudio más profundo analizar cómo son los histogramas

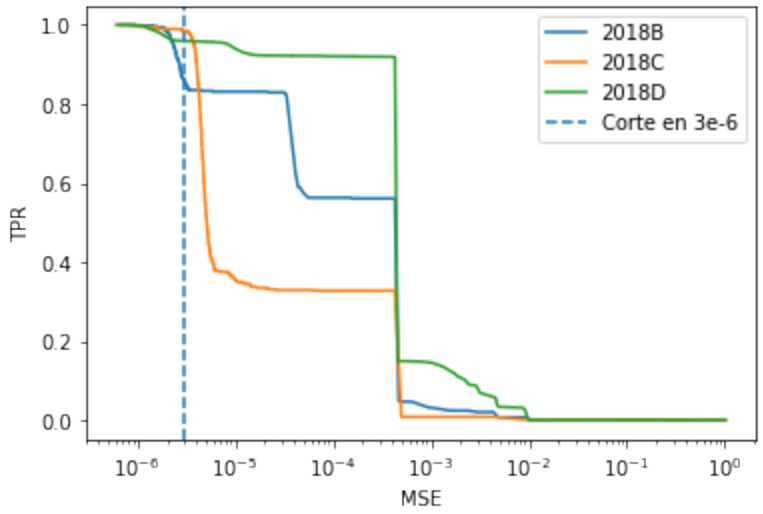


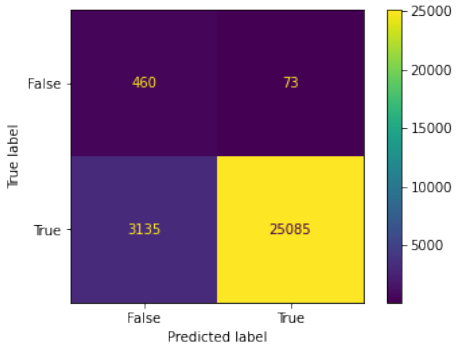
Figura 7.7: TPR vs corte del MSE. Además, como linea discontinua se representa el corte elegido para los próximos experimentos.

asociados a estos escalones.

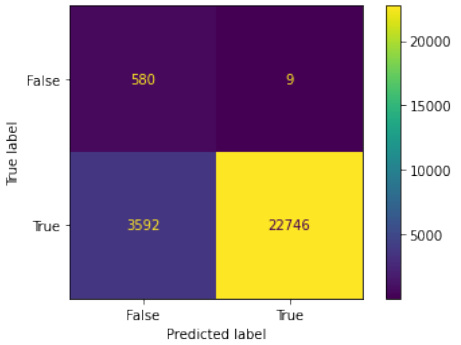
Para el propósito de estudiar el funcionamiento de las red en mayor profundidad se tomó un valor de corte del MSE de $3e-6$. La decisión surge porque con este valor la red es capaz de distinguir todos los tipos de error, mientras se mantiene un valor alto de precisión.

A continuación, debemos calcular las matrices de confusión asociadas a estas muestras de datos para evaluar la potencia real del AE. En la figura 7.8 se pueden ver las matrices de confusión calculadas a partir del corte mencionado del MSE de $3e-6$.

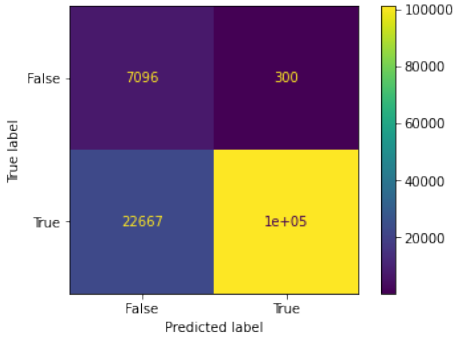
A primera vista, el modelo funciona bien para todas las muestras. Se puede apreciar un valor bajo de falsos positivos, aunque el valor de falsos negativos es considerable.



(a) 2018B



(b) 2018C



(c) 2018D

Figura 7.8: Matrices de confusión para los muestras de datos restantes con valor de corte del MSE de $5e-6$.

Para analizar en detalle el funcionamiento del AE debemos analizar las métricas expuestas en 6.1 de nuevo. Los resultados se pueden ver en la tabla 7.3.

Como se puede observar, el AE tendría un rendimiento excelente en un uso real de DC. Una precisión tan alta significa que más del 99.7 % de los datos clasificados como positivos son realmente positivos. Es decir, sólamente el 0.3 % de los datos clasificados por la red como *buenos*, pueden ser en realidad *malos*.

Por otro lado, como se puede ver en la figura 7.8, una porción considerable de los datos clasificados como *buenos* por expertos acaban siendo clasificados como *malos*

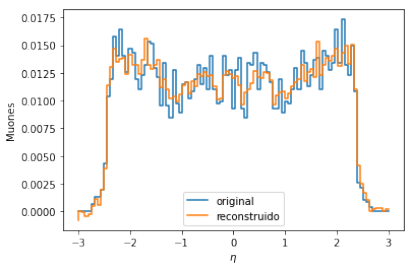
Muestra de datos	Precision	Especificidad	Sensibilidad	Exactitud	F1
2018B	0.997	0.863	0.889	0.888	0.940
2018C	0.999	0.985	0.864	0.866	0.927
2018D	0.997	0.959	0.815	0.823	0.897

Cuadro 7.3: Métricas obtenidas a partir de las matrices de confusión de la figura 7.8 para los distintos muestras de datos aplicando el AE entrenado con dimensión latente 8.

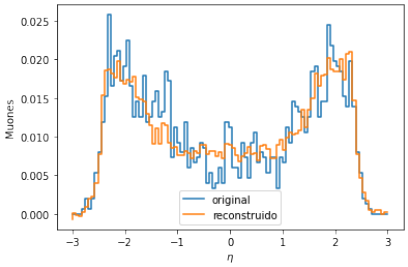
por la red, como indica el valor inferior a 0.9 de la sensibilidad para estas muestras. Esto puede deberse a sobreentrenamiento con el conjunto de entrenamiento, en el que se encuentra una sensibilidad al considerar la muestra completa 2018A de 0.9975, que es considerablemente mayor que en el resto de muestras. Aun así, esto no es suficiente evidencia como para concluir que el sobreentrenamiento es relevante.

De esta forma, si se usase esta red concreta en DC de CMS, alrededor del 15 % de los datos se perderían debido a falsos negativos. Dependiendo de la importancia de ganar granularidad y automatizar DC, esto puede valer la pena, pero lo ideal sería conseguir un modelo con una sensibilidad más elevada.

Para explorar la razón de esta sensibilidad más baja proponemos observar los histogramas originales y reconstruidos de un ejemplo de una LS *buena* frente a una *mala* para el observable η, como se puede ver en 7.9. Notar que para el cálculo del MSE del AE se utiliza el máximo MSE de los 4 observables y en este razonamiento solo se está considerando uno de los observables por simplificar.



(a) LS *buena*



(b) LS *mala*

Figura 7.9: Comparación de histogramas originales y reconstruidos para uns LS *buena* y una *mala* de la muestra 2018D. El MSE resulta ser $1.47e-6$ y $7.51e-6$ respectivamente.

Resulta que la red es capaz de reconstruir ambos histogramas *buenos* y *malos* correctamente, aunque en los *malos* se encuentre un MSE más elevado. Entonces, el MSE asociado a LS *buenas* y *malas* puede ser del mismo orden, lo que lleva a confusión a la hora de que la red tome decisiones. Por esto, al querer un valor de precisión muy cercano a 1, estaremos sacrificando la sensibilidad. Para solucionar esto con un AE clásico, se propone reducir su capacidad, es decir, reducir el número de capas o el número de neuronas que hay en ellas. Esto podría resultar en un AE que reconstruya peor las LS *malas* y por lo tanto un modelo en el que no se sacrifique tanta sensibilidad a favor de una precisión elevada. También podría ser útil explorar técnicas de regularización de redes como las explicadas en 5.5, que no se implementaron en nuestro modelo.

También es interesante ver la dependencia del MSE en función de la LS. Como estas están ordenadas cronológicamente, podríamos encontrar períodos prolongados con un MSE elevado debido a un fallo técnico en CMS o alguna correlación entre los picos del MSE. Para apoyar las gráficas se representará el valor de corte de $3e-6$ del MSE, por lo que los puntos que aparezcan bajo el corte serán clasificados como *buenos*, mientras que los superiores, como *malos*.

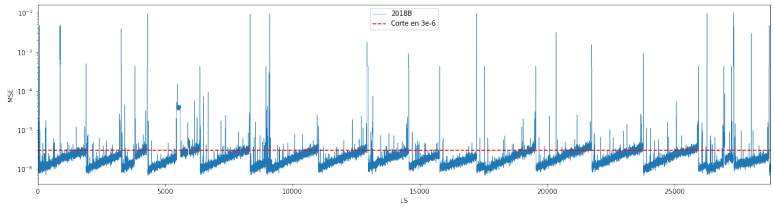


Figura 7.10: MSE vs LS para la muestra 2018B.

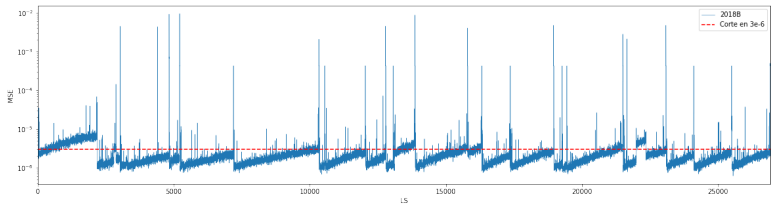


Figura 7.11: MSE vs LS para la muestra 2018C.

En las figuras 7.10 y 7.11 se puede ver el MSE para cada LS ordenadas cronológicamente. En principio no se encuentra ningún período anómalo con un MSE elevado, y de hecho se encuentra un patrón periódico en las dos gráficas. Podría decirse que el MSE toma una forma similar a una serie de dientes de sierra de longitud similar. Esto se puede relacionar directamente con la luminosidad instantanea que varía durante cada fill del LHC, como se puede ver en la figura **??**.

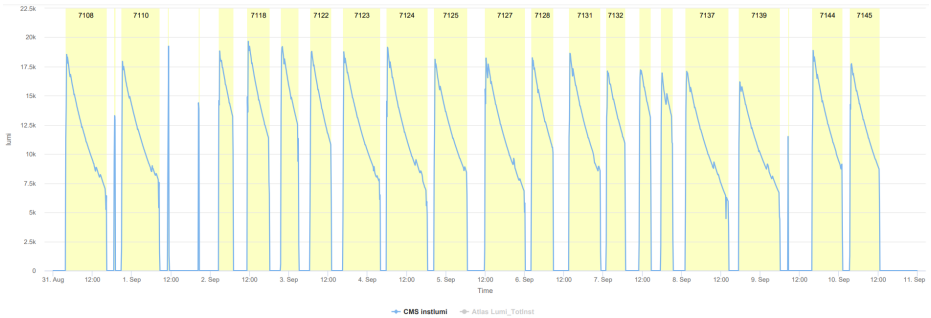


Figura 7.12: Luminosidad instantanea en función del tiempo. cada región coloreada de amarillo representa un run.

La luminosidad instantanea del LHC disminuye a lo largo del run porque cada

vez hay menos protones en los haces debido a las colisiones que se producen entre ellos. Entonces, al haber menor luminosidad, habrá menos colisiones y por lo tanto menos muones detectados por cada LS a medida que el fill se acaba. Esto causa un aumento en el MSE de reconstrucción porque las distribuciones empíricas de los 4 observables serán más irregulares por contener menos información estadística.

Se puede ver claramente la correlación entre los picos de sierra observados en la figura 7.12 y el MSE calculado en las figuras 7.10 y 7.11. Esto es una buena señal de que el AE funciona bien ya que es capaz de reconstruir un patron real sin tener información del mismo.

Aún así, hay un número importante de picos del MSE más elevados dentro de cada fill observado en 7.10 y 7.11, que no se puede atribuir al descenso de la luminosidad instantanea. Seremos incapaces en este trabajo de explorar cada uno de los picos, pero varios de la época 2018B fueron documentados como un agujero en un sector de las cámaras de muones que se puede ver en la figura 7.13

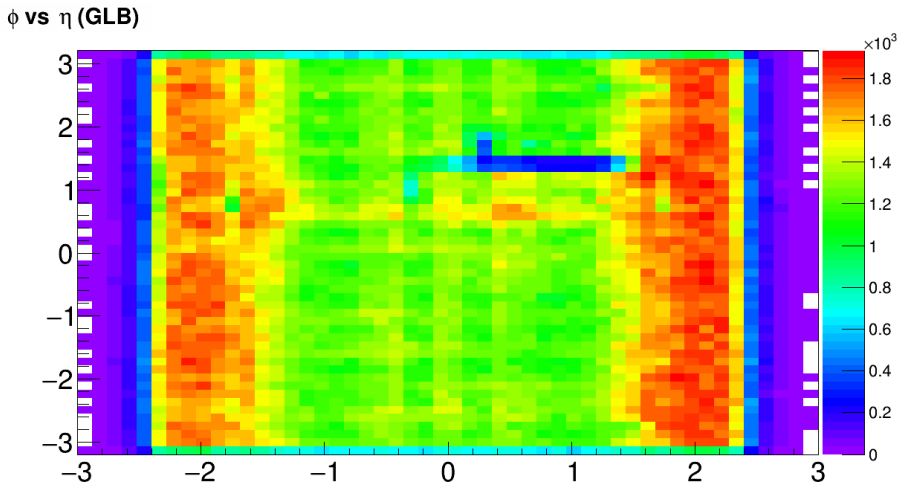


Figura 7.13: Agujero en un sector de las cámaras de muones en $\phi \approx 1.5rad$, $\eta \approx 0-1.5$ durante la época 2018B.

Este agujero en la cámara de muones implicó una pérdida del 40 % en la eficiencia en la región afectada cuando surgía el problema, y apareció en diversos runs que fueron clasificados como buenos. Un ejemplo de dichos runs es el 317292 de la época 2018B, y en la figura 7.14 se puede ver el MSE de esta run en función de cada una de sus LS ordenadas cronológicamente.

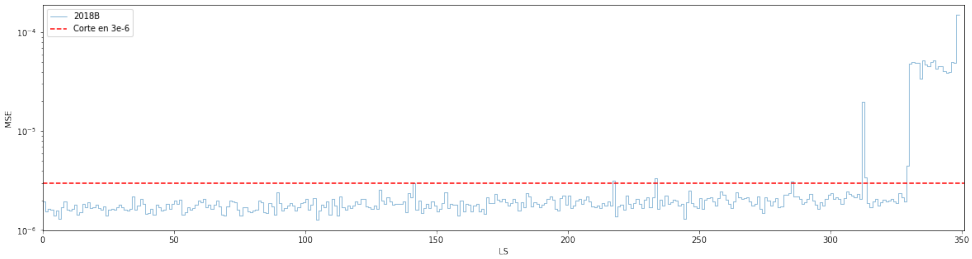


Figura 7.14: MSE vs LS para la muestra 2018B en un run que presenta el agujero descrito en la figura 7.13.

A partir de la LS 330 del run, aparece el problema del agujero en las cámaras de muones, y se puede ver perfectamente como aumenta el MSE en este período. Esto es un claro ejemplo de las ventajas de automatizar DC, ya que esta run fue clasificada como *buena* por los expertos, mientras que nuestro AE podría haber distinguido individualmente qué LS fueron afectadas para clasificarlas como *malas*.

Solo queda analizar el MSE vs LS para la época 2018D, que se puede ver en la figura 7.15. Tiene una única peculiaridad que la separa de las épocas 2018B y 2018C.

En la figura 7.15, sí que se ven dos período extenso de tiempo con MSE elevado, acompañado de los dientes de sierras esperados que representan los fills y los picos esporádicos que se ven en las otras 2 épocas. Estos períodos de tiempo con un MSE tan elevado se corresponden con un gran cantidad de LS casi vacías con menos de 100 muones. Claramente, los runs correspondientes a estas épocas fueron

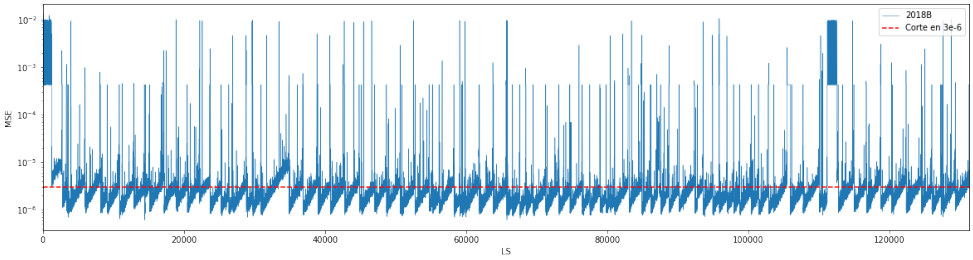


Figura 7.15: MSE vs LS para la muestra 2018D.

clasificados como *malos*, así que no es un escenario en el que la automatización sea crucial, pero es un ejemplo de buen funcionamiento del AE.

En conclusión, el AE clásico puede ser una buena primera herramienta para automatizar DC. La tasa de falsos positivos se encuentra muy baja, pero la de falsos negativos es elevada. Existen ciertos problemas a resolver que no se tratarán en este trabajo, como un método fiable para analizar posibles falsos positivos en el uso real de la red, y más pruebas sobre datos que contengan anomalías de diferente naturaleza. Además, el AE es capaz de reconstruir los diferentes fills de cada época como dientes de sierra, y se comprueba que es capaz de distinguir al menos 2 tipos de anomalías concretas ya conocidas. En un futuro, sería útil diseñar herramientas que acopladas a un AE o un modelo similar sean capaces de detectar el tipo de anomalía en tiempo real para reportarlo.

7.4. VAE

Por desgracia, hemos sido incapaces de entrenar de manera útil un VAE para cualquiera de los muestras de datos dados y el método descrito en la sección 5.8. El problema que encontramos es que la función de coste diverge en aproximadamente 3 iteraciones independientemente de hiperparámetros como el *learning rate*, *batch*

size, β, etc.

La única forma con la que hemos conseguido un modelo convergente es aplicando la función de activación sigmoide a la capa del espacio latente asociada a la media y varianza de la distribución codificada. El problema en este caso es que no hay suficiente poder predictivo como para que la red distinga los datos *buenos* de los *malos*. Como se puede ver en la figura 7.16, la representación en el espacio latente de los datos *buenos* y *malos* no tiene ninguna diferencia.

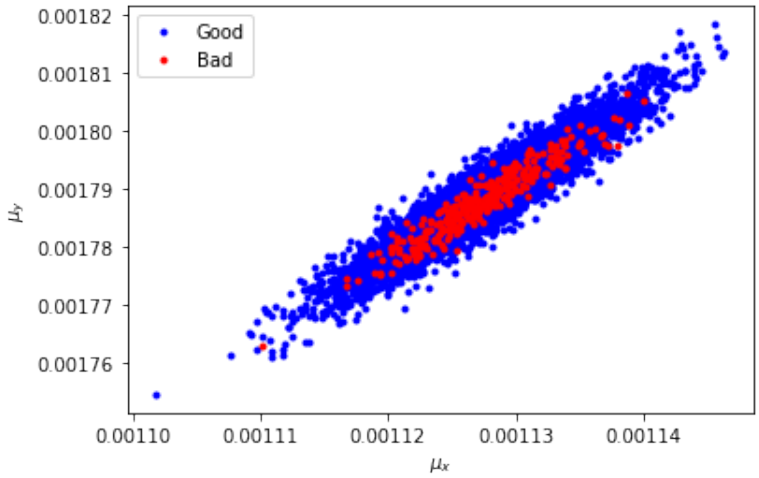


Figura 7.16: Representación codificada en el espacio latente de dos dimensiones de la media para histogramas de ϕ de la muestra de datos 2018D.

Como cabe esperar a partir de esta representaciónb latente, podemos ver en su curva ROC en la figura 7.17, que no es mejor que una decisión aleatoria a la hora de clasificar los datos.

Está claro que el VAE debería ser un modelo mejor en todos sus aspectos que el AE clásico, pero hace falta más investigación sobre como implementarlo en el caso de DC de CMS. Aún así, sí que se ha comprobado que el AE clásico es un buen modelo con alto poder predictivo y podría ser de utilidad en el caso real hasta

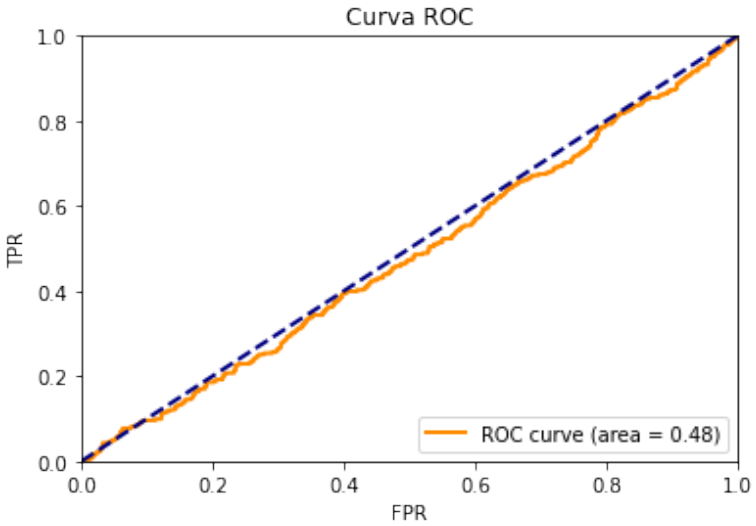


Figura 7.17: Curva ROC y AUC asociado al VAE entrenado sobre los datos 2018D obtenida a partir de un conjunto de validación compuesto del 10 % de datos *buenos* y todos los *malos* de la misma muestra.

tener un modelo funcional de VAE.

Capítulo 8

Conclusiones

Durante este trabajo se ha estudiado un método concreto de ML para automatizar la taréa de DC de CMS, el AE. Esta técnica de aprendizaje no supervisado es útil en problemas de detección de anomalías porque no se ve afectada por el *class imbalance* ya que se entrena únicamente con una clase de datos, en nuestro caso los clasificados como *buenos*.

Uno de los problemas encontrados durante el estudio son los fallos en la clasificación por expertos de los datos. Las etiquetas que ponen estos expertos van asociadas a los runs y no a las LS, por lo que puede haber una cantidad de datos clasificados erroneamente. A pesar de esto, se vio que no es un fallo alarmante y de hecho puede ayudar a reducir el sobreentrenamiento el hecho de tener una pequeña cantidad de anomalías en el conjunto de entrenaminto.

A la hora de decidir una arquitectura concreta de AE para la taréa de DC de CMS, se estudiaron varias posibilidades con distinta dimensión latente (2, 4, 8 y

16), y al final nos decantamos por 8 aunque todas podrían haber sido propuestas válidas. Esta red se entrenó con el 80 % de los datos *buenos* de la muestra 2018A, y proporcionó un AUC de 0.78, una especificidad de 0.74, sensibilidad de 0.81, exactitud de 0.80 y función F1 de 0.89 sobre los datos de validación. Sobre las demás muestras de datos 2018B, 2018C y 2018D aumentó el AUC a 0.96, 0.96 y 0.97 respectivamente, lo cual significa un gran poder predictivo en un caso más realista de DC que el conjunto de validación. Como se ve en el cuadro 7.3 el AE generaliza adecuadamente sobre estas muestras, y puede ser utilizado en la práctica debido a que tiene una precisión superior a 99.7 % en todos los casos. Este elevado valor de precisión permite un uso mínimo de expertos para descartar falsos positivos, pero a coste de sacrificar el 15 % de los verdaderos positivos que la red no es capaz de clasificar adecuadamente.

Una clara evidencia del buen funcionamiento del AE se ve en las figuras 7.10, 7.11, 7.14 y 7.15. En estas, se comprobó que el MSE aumenta al disminuir la cantidad de muones detectados por CMS, y se ve una correlación entre los picos de estas gráficas con la luminosidad instantanea del LHC para cada fill. Además, se estudiaron 2 tipos diferentes de anomalías que el AE es capaz de detectar. El primero, un agujero en la cámara de muones de CMS que se corresponde con períodos de MSE elevado y por lo tanto LS clasificadas como *malas*. El segundo, períodos extensos en los que CMS no detecta muones aunque los datos sean grabados, que el AE también clasifica como *malos*.

En el estudio del VAE, no se encontraron buenos resultados debido a problemas con la convergencia de la red y un bajo poder predictivo. Esto podría deberse a un problema en el esquema de entrenamiento propuesto. En principio, en un estudio más intensivo del VAE para DC deberían encontrarse resultados aún mejores que el AE clásico, pero esto supera los objetivos de este trabajo.

En conclusión, el AE clásico resulta ser una herramienta muy potente, con un gran poder predictivo y fácil de implementar, que podría ser usada en el futuro para apoyar el proceso de DC para CMS con la revisión de expertos de algunos histogramas que el AE no sepa clasificar de manera certera, aunque se debería estudiar un método para generar un AE con sensibilidad mayor.

Bibliografía

[1] M. Stankevicius et al. Comparison of supervised machine learning techniques for cern cms offline data certification. In *Doctoral Consortium/Forum@ DB&IS*, pp. 170–176, 2018.

[2] B. Povh et al. *Particles and Nuclei*. Springer Berlin Heidelberg, 2015.

[3] V. A. Bednyakov et al. On the higgs mass generation mechanism in the standard model. *Physics of Particles and Nuclei*, 39(1):13–36, jan 2008.

[4] P.A. Zyla et al. (Particle Data Group). Review of particle physics. *Progress of Theoretical and Experimental Physics*, 2020(8), August 2020.

[5] S. F. Novaes. Standard model: An introduction. *10th Jorge Andre Swieca Summer School: Particle and Fields*, pp. 5–102, January 1999.

[6] ATLAS Colaboration. Observation of a new particle in the search for the standard model higgs boson with the ATLAS detector at the LHC. *Physics Letters B*, 716(1):1–29, sep 2012.

[7] C. Csáki and P. Tanedo. Beyond the standard model. *2013 European School of High-Energy Physics, Paradfurdo, Hungary, 5 - 18 Jun 2013, pp.169-268 (CERN-2015-004)*, 2016.

[8] S. Carlip et al. Quantum gravity: A brief history of ideas and some prospects. *International Journal of Modern Physics D*, 24(11):1530028, sep 2015.

[9] J. L. Feng. Dark matter candidates from particle physics and methods of detection. *Ann. Rev. Astron. Astrophys. 48: 495, 2010*, March 2010.

[10] CERN. Cms images gallery: https://home.cern/resources/image/experiments/cms-images-gallery, 2017.

[11] CMS Collaboration. *CMS Physics : Technical Design Report Volume 1: Detector Performance and Software*. Technical design report. CMS. CERN, Geneva, 2006.

[12] B. Vormwald. The CMS inner tracker – transition from LHC run i to run II and first experience of run II. In *Proceedings of The European Physical Society Conference on High Energy Physics — PoS(EPS-HEP2015)*. Sissa Medialab, mar 2016.

[13] P. Azzurri. The CMS silicon strip tracker. *Journal of Physics: Conference Series*, 41:127–134, may 2006.

[14] V. Azzolini et al. The data quality monitoring software for the CMS experiment at the LHC: past, present and future. *EPJ Web of Conferences*, 214:02003, 2019.

[15] L. Tuura et al. CMS data quality monitoring: Systems and experiences. *Journal of Physics: Conference Series*, 219(7):072020, apr 2010.

[16] CMS Collaboration. CMSSW (CMS Software). https://github.com/cms-sw/cmssw.

[17] A. A. Pol et al. Anomaly detection using deep autoencoders for the assessment of the quality of the data acquired by the CMS experiment. *EPJ Web of Conferences*, 214:06008, 2019.

[18] P. Calafiura et al. *Artificial Intelligence for High Energy Physics*. WORLD SCIENTIFIC, mar 2022.

[19] A. A. Pol et al. Anomaly detection with conditional variational autoencoders. In *ICMLA 2019 - 18th IEEE International Conference on Machine Learning and Applications*, 18th International Conference on Machine Learning Applications, Boca Raton, United States, December 2019. arXiv.

[20] G. Dong et al. A review of the autoencoder and its variants: A comparative perspective from target recognition in synthetic-aperture radar images. *IEEE Geoscience and Remote Sensing Magazine*, 6(3):44–68, sep 2018.

[21] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. *International Conference on Learning Representations*, December 2014.

[22] D. P. Kingma and M. Welling. An introduction to variational autoencoders. *Foundations and Trends® in Machine Learning*, 12(4):307–392, November 2019.

[23] Rapidminer. Confusion matrix. *https://rapidminer.com/glossary/confusion-matrix/*.

[24] T. Gneiting and P. Vogel. Receiver operating characteristic (roc) curves. https://doi.org/10.48550/arXiv.1809.04808, 2018.

[25] F. Pedregosa et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[26] TensorFlow Developers. TensorFlow: Large-scale machine learning on heterogeneous systems, 2022. Software available from tensorflow.org.

[27] F. Chollet et al. Keras. https://keras.io, 2015.

[28] J. Reback et al. pandas-dev/pandas: Pandas 1.4.3, 2022.

[29] C. R. Harris et al. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[30] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.